INTERNATIONAL
HELLENIC
UNIVERSITY

Design and development of the website of a conference and the accompanying mobile applications utilizing cross-platform tools

George Georgiadis

SID: 3306170002

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Mobile and Web Computing

OCTOBER 2019

THESSALONIKI – GREECE

Design and development of the website of a conference and the accompanying mobile applications utilizing cross-platform tools

George Georgiadis

SID: 3306170002

Supervisor:	Prof. Stavros Stravrinides
Supervising Committee Members:	Assoc. Prof. Name Surname
	Assist. Prof. Name Surname

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Mobile and Web Computing

OCTOBER 2019

THESSALONIKI – GREECE

Abstract

This dissertation was written as a part of the MSc in Mobile and Web Computing at the International Hellenic University. I would like to thank Dr. Stavrinides for supervising this dissertation.

Events as well as scientific conferences require online presence for a number of reasons. This dissertation studies essential concepts of web technologies and software development in the context of designing and developing the website of a conference and the accompanying mobile applications utilizing cross – platform development tools. In particular, the 5th International Interdisciplinary Chaos Symposium on Chaos and Complex Systems is used as a case study. A fully functional website having all the necessary features is developed with WordPress CMS and a mobile application that is compatible with Android and iOS is also developed with Ionic Framework 4 for the user to be able to register for the conference via mobile devices.

Contents

Table of Figures

1 Introduction

78% of people interested in events search for them on the internet [53]. Hence, the online presence of an event is necessary for a number of reasons [53]. First and foremost, the website increases potential attendees and can be used as a marketing tool for the promotion of the event [53]. Secondly, the website provides the capability of analyzing the audience of the event by means of the website analytics [53]. Thirdly, the website can provide answers to frequent questions and avert responses to individual inquiries [53]. Lastly, the website enhances post – event communication, since attendees are able to receive follow – up news and stay in touch via social media [53].

A scientific conference is an event that needs a website for informing potential attendees about the event and for guiding them through registration [54]. It is essential that the website of a scientific conference promote the logo, the name and the dates of the event, describe the location the event takes place, inform the visitors about the keynote speakers, provide instructions about abstract and paper submission, introduce the committees, provide guidance through registration, offer the capability of contacting the organizers, have links to the social media profiles of the event, make the detailed schedule of the event easily accessible, mention the sponsors of the event, present the history of the event and the reasons for attending it, include material from past events, provide answers to frequently asked questions and have a blog containing insights into the event and allowing the interaction with the visitors [54].

One of the web development trends in 2019 is the cross – platform app development [55]. Users load websites as well as native applications on smartphones and tablets [55]. Therefore, the website should be responsive to such devices and has its own native application being compatible with smartphones and tablets.

The chapter “Background Theory” makes a short introduction to Content Management Systems and to cross – platform application development. It, subsequently, discusses essential concepts of web security, web design principles, classification of websites, website design evaluation metrics, Search Engine Optimization and web hosting. The necessary theoretical background for software development is discussed in the next subchapter. In addition, RESTful APIs, the JSON data-interchange language and the Cross – Origin Resource Sharing mechanism applied to the development of the mobile application are sufficiently analyzed. Last but not least, the structure of Ionic Framework 4, the involved frameworks and programming languages also used for the development of the mobile application are discussed in the last subchapter.

The chapter "Website and Mobile Application Development" includes the analysis and design phases of the website and of the mobile application. The functional and nonfunctional requirements as well as the use cases are defined for each software application. The website's structure is described with a sitemap diagram and the purchase process on the website is illustrated with a robustness diagram. On the other hand, the user navigation through the mobile application is illustrated with a navigation flow diagram.

Finally, the conclusions drawn from the dissertation's work and proposals for future work are mentioned in the chapter "Conclusions".

Appendix A contains screenshots of the website's pages, Appendix B contains screenshots of the mobile application's pages and Appendix C contains the source code of the mobile application.

2 Background Theory

2.1 Content Management Systems

Content Management Systems offer the opportunity to deliver high quality websites without requiring advanced coding skills. The most popular among them are the open source CMS WordPress, Joomla! and Drupal as they occupy the largest share of the market (~70%) [1].

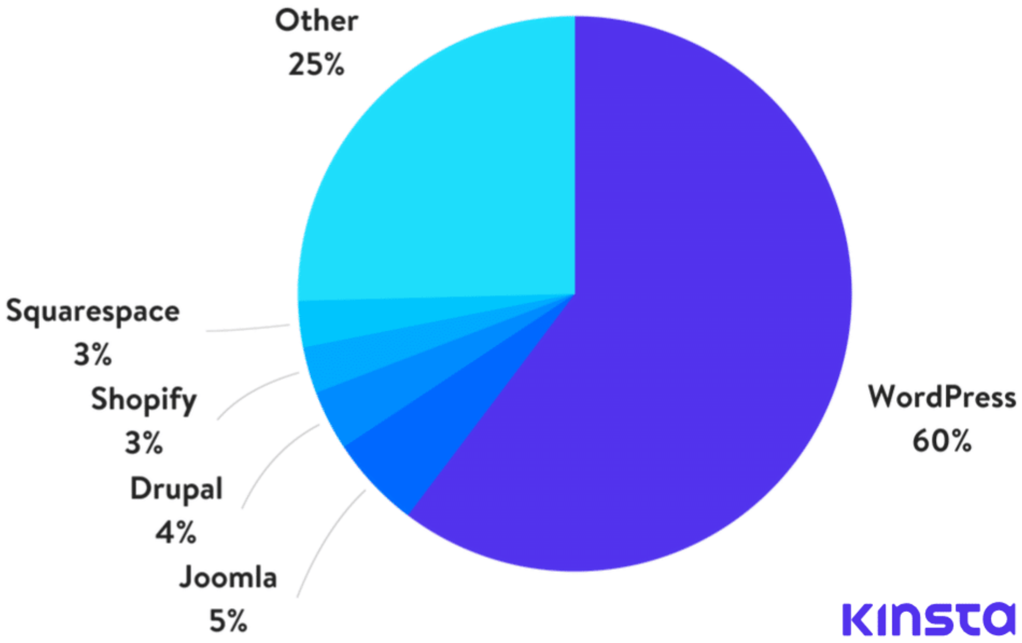


Figure 1 CMS Market share [2]

WordPress has been developed since 2003 and relies on PHP and SQL database technologies, while JavaScript has been integrated into WordPress environment during the last years. WordPress maintains the 60% of the CMS market share, that is, approximately 600 million websites on the World Wide Web [1].

Joomla! Started as an open source project and has been a CMS alternative since 2005. The presence of Joomla! on the CMS market is significant given that almost 60 million websites are based on Joomla! which remains the second most popular CMS. Its principle technologies and its architecture are similar to those of Wordpress. Specifically, Joomla! websites are developed in PHP and SQL databases and they are built on the model consisting of the front-end (the accessible to the public part of the website) and of the back-end administration area [1].

On the other hand, Drupal is older than WordPress and Joomla! as it has been developed as an open source community project since 2001. However, it is ranked third in the preferences of website developers [1]. Drupal is considered the most technologically advanced Content Management System and is particularly preferred among big enterprise websites [1].

The selection of the proper CMS should take into consideration the following factors [1][2]:

- **Cost**: Apart from the hosting service and domain registration costs, the expenses of the website's construction may include premium plugins, themes and extensions. The more professional and complex a website, the greater the chance of incorporating paid add-ons. Given that Joomla! and Drupal are more complex than Wordpress and due to the law of supply and demand, the likelihood of website construction with Wordpress being less expensive is higher.
- **Ease of use**: Although both Joomla! and Drupal have user-friendly interfaces, Wordpress is considered to be the most simple platform for the construction of a website.
- **Customizability**: Wordpress is the most compact CMS for customizing the website due to the wide variety of tools it contains (templates, themes, components, plugins and languages), while both Joomla! and Drupal lack such resources, which are needed to be built or searched independently.
- **Performance**: Drupal can create the fastest websites supporting large number of visitors since it requires less resources from the server. Joomla! is also regarded as faster than Wordpress. However, the website's performance heavily depends on the speed of the hosting service, on the caching and on the development techniques, which means that the Wordpress website can still be competitive.
- **Security:** Vulnerabilities and security flaws are common issues among websites regardless of the CMS with which they are built.

- **Search Engine Optimization:** WordPress enhances SEO with features such as custom URLs, SEO-friendly markup, content optimization tools, mobile responsive design. Furthermore, Drupal utilizes meta tags, plugins and caching to improve the impact of the website on the search engines. On the other hand, Joomla! is deprived of URL rewrites, metadata and tag optimization, which contribute to SEO.
- **Team collaboration:** All CMS offer the capability of collaborative creation by determining roles, granting permissions and access levels and keeping track of the website's history.
- **Scalability:** Drupal is the most scalable among the three CMS and it is often used for the construction of enterprise websites. Joomla! is considered the most appropriate CMS for building social networks. Although Wordpress is frequently used for blogging and e-commerce websites, it can successfully be utilized for the construction of any type of website.

2.2 Cross-platform application development

Building software being capable of running on several platforms has certain advantages over writing native applications which function on a unique operating system. Maintenance and development costs are significantly reduced since the same technical knowledge and resources are implemented for a product, which, otherwise, needs more than one Integrated Development Environments for developing, testing and implementation [3].

Mobile application development supports the cross-platform software development with the aid of the open source tools listed below [3][4]:

- **Apache Cordova** can be used for the creation of applications for Android, iOS, Windows Phone 8, Blackberry, Ubuntu, Firefox OS, LG WebOS and Amazon FireOS. Mobile applications are built with HTML, CSS and Javascript. A disadvantage of Apache Cordova is that the design and the architecture of the application are decisions the user has to make.
- **Apache PhoneGap** is more effective than Cordova, since it includes a Graphical User Interface in contrast with the command line environment of Cordova, and its applications are compatible with Android, iOS and Windows Phone 8.

Although it simplifies the development process, it does not constitute a reliable solution to creating cross-platform mobile applications.

- **Ionic Framework** incorporates Angular with its own User Interface and inherits many characteristics from the two previous tools. It is considered the prevailing cross-platform tool and its applications are compatible with Android, iOS, Windows Phone 8, Chrome and Desktop. However, the open source version of Ionic Framework confines the application development due to significant restrictions.
- **Framework7** is an effective variant of Cordova thanks to the addition of two critical layers regarding the structure and the interface elements. On the other hand, it does not have strong community support, which puts the development of Android and iOS applications at risk.
- **React Native** inherits its core application framework from React. Reacts interprets the JavaScript source code and is able to immediately convert it to Android and iOS native elements. It can be considered a reliable framework on the grounds that Facebook and Instagram applications are built with React Native.
- **NativeScript** provides a cross-platform development experience, which is similar to React Native, whereas the development approach resembles that of Ionic Framework. NativeScript is based on JavaScript and TypeScript and its applications are compatible with Android and iOS.
- **Flutter** is an application development framework created by Google. It employs its own programming language named Dart, which aspires to replace JavaScript. Flutter is at an early development stage and its dependence on Dart and on Google Material Design renders it a controversial cross-platform mobile application development tool.
- **Jasonette** relies on server-side rendering, which means JSON layouts are retrieved from the server and parsed for each mobile screen with which the user interacts. However, in the case of non-static applications extra functionality on the server side is required to deal with the JSON layouts, which makes the application development too complex.

2.3 Website security

Security is a critical component of every web application and deals with issues arising from various scales and types of attacks. Since a website is exposed on the internet, security definitely needs to be incorporated into the website's architecture [6].

2.3.1 Secure Sockets Layer

TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) are the dominant protocols being responsible for the transmission of packets over the internet. Both protocols do not implement any encryption, which may result in the sniffing of the data delivered from the sender to the receiver [5]. The Secure Sockets Layer is an enhanced version of TCP providing encryption, data integrity and end-point authentication. SSL is implemented in the application layer. The SSL code is embedded in both the client and the server. The sender forwards the data to the SSL socket, the data is encrypted and the encrypted data is sent through the TCP socket of the sender, which is received from the TCP socket and then decrypted at the SSL socket at the receiver's side [5].

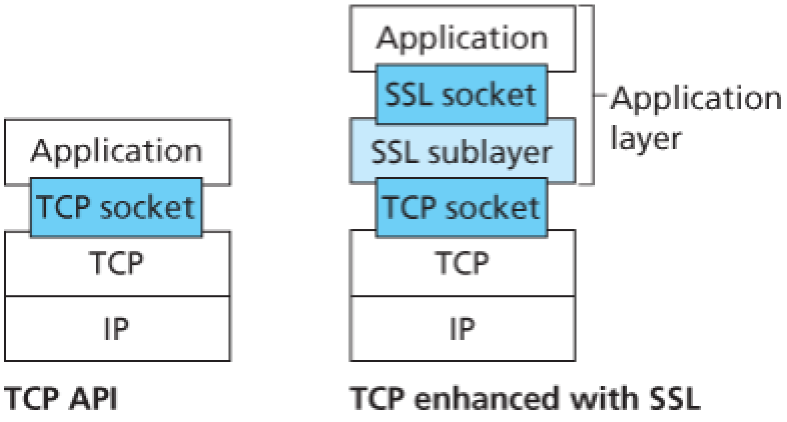


Figure 2 SSL architecture [5]

SSL communications are comprised of three stages [5]:

I. Handshake: The server possesses a private-public key pair and a certificate that associates its identity with the public key. The client aims at establishing a TCP connection with the server, at verifying the identity of the server and at sending a master secret key to the server in order for both the client and the server to generate the symmetric keys needed for the SSL communication session. In particular, the client sends a SYN (synchronize) message and the server responds with a SYNACK (synchronize-acknowledge) message. This means that the TCP connection is established. Then the client sends a “hello” message to the server and the server responds with a certificate containing its public key. Since the certificate has been issued by a Certificate Authority, the client trusts the response from the server. Finally, the client generates a master secret and encrypts it with the server’s public key, which is subsequently sent to the server and decrypted with the server’s private key.

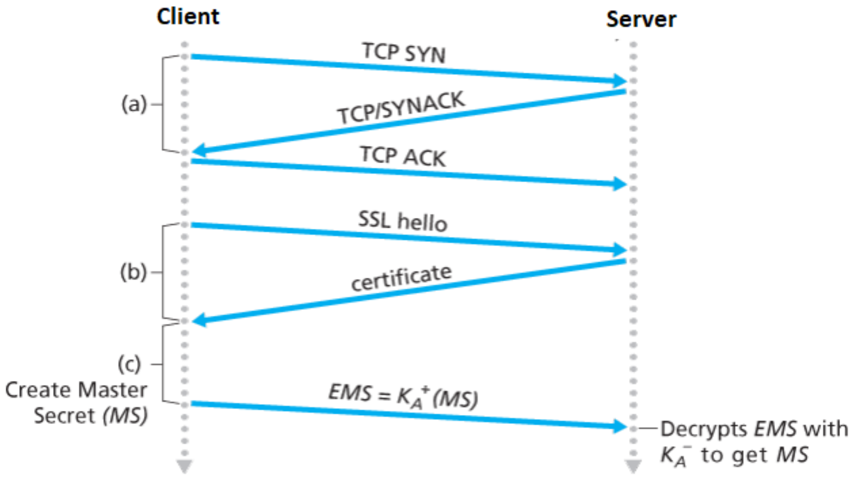


Figure 3 SSL handshake process [5]

II. Key derivation: The master secret generated during handshake can be the symmetric session key for any further encryption and integrity tests. However, it is considered safer to utilize different encryption and integrity test each time. For this reason, both the client and the server generate four keys being segments of the master secret: two session keys for data encryption and two MAC keys (Media Access Control) keys for data integrity verification. The two keys for each purpose refer to the data transfer from the client to the server and to the data transfer from the server to the client.

III. Data transfer: SSL divides the data byte stream into SSL records. Each SSL record includes a MAC key in a hash function and is encrypted with the proper session key. The packet containing both the MAC key and the encryption key is then passed to the TCP socket and sent to the destination. When the client or the server is to terminate the SSL connection, the last SSL record sent to the receiver contains proper indication in the type field.

SSL was initially developed by Netscape in 1996 and has been widely used by mail servers, web browsers, web servers, commerce sites and online payments [5].

2.3.2 Transport Layer Security

SSL was the first encryption protocol for securing internet communications. Nowadays is replaced by the Transport Layer Security (TLS) protocol, the function of which resembles that of SSL. Initially, the client and the server acknowledge each other with SYN and SYNACK messages. The client begins the handshake by sending a “Hello” message, which includes the TLS version and the encryption algorithms (cipher suite) supported by the client as well as a random string named the “client random” [6]. The server responds with a SSL certificate, a selected cipher suite and a random string named the “server random” [6]. The server’s SSL certificate is verified by the client with recognizing the Certificate Authority having issued it. After the confirmation of the server’s identity, the client generates the premaster secret being encrypted with the server’s public key, which then is decrypted with the private key at the server side [6]. The session keys at the client side and at the server side are produced from the master secret, the “client random” and the “server random”. At the end of the handshake process, both the client and the server send each other encrypted with the session key messages indicating that the handshake finished [6]. Although TLS handshakes in principle implement asymmetric encryption on the grounds that they employ a public-private key pair, there are handshake variations according to which the server’s private key is not utilized during the handshake process. Instead, the server and the client attempt separately to generate a common master secret [6].

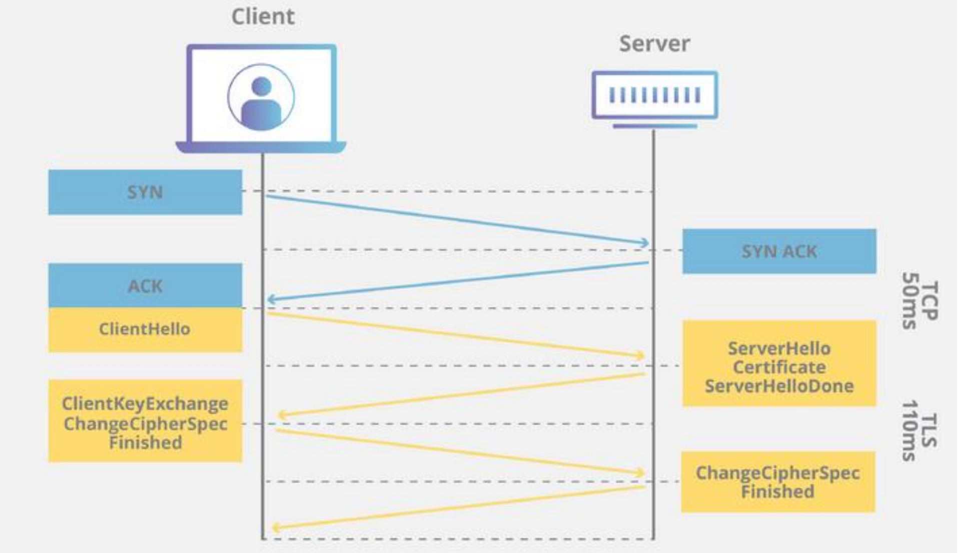


Figure 4 TLS handshake process [6]

TLS prevents sensitive data from being revealed over the network and intruders from tampering with the communication data between the client and the server [7]. However, the handshake adds latency to the server's response and consumes resources due to the slow asymmetric encryption [7]. The estimated latency is approximately 5ms and the estimated increase in CPU usage is 2% [7].

2.3.3 HyperText Transfer Protocol Secure

HTTP (HyperText Transfer Protocol) is the primary protocol for the communication between web browsers and websites [6]. HTTPS being the secure version of HTTP is critical to applications, which involve sensitive data, such as logging into bank accounts, healthcare services and email services [6]. HTTPS encrypts data with the TLS protocol. As a result, even if data is sniffed due to interception, it will be encrypted and not readable. In addition, HTTPS avert unauthorized third parties injecting advertisements into web content [6].

More and more servers require HTTPS for security reasons and more and more browsers require HTTPS for certain features such as the geolocation. Google and Firefox also endorse encrypted HTTP by gradually abolishing unencrypted HTTP [7]. Not to mention that users expect safety URL indicators such as the green padlock when they visit e-commerce websites as they are concerned about their privacy [7].

2.3.4 Threats and countermeasures

Since a website is an information system, the information it manages is regarded as an important asset. Every information system needs to guarantee the confidentiality, the integrity and the availability of the information [8]. Confidentiality being an essential component of privacy refers to the prevention of unauthorized disclosure of data. Similarly, integrity ensures that data is not modified in an unauthorized or undesirable manner and availability ensures that data is accessible as required [8].

The threats to information systems can be classified into four groups [8]:

I. **Interception** attacks affect the confidentiality of information by granting unauthorized users access to data. An example of interception is the unauthorized viewing or copying of a file.

II. **Interruption** attacks render information assets unusable temporarily or on a constant basis. They can be attacks against both availability and integrity. Examples of interruption are a Denial of Services attack on a mail server and the manipulation of the database services to block access to the database.

III. **Modification** attacks aim at the loss of availability and integrity and include tampering with the data. An example of modification is the alteration of the configuration file of a web server resulting in the change of the server's function.

IV. **Fabrication** attacks are considered attacks against availability and integrity. They include the generation of data and processes being similar to those of the information system. Examples of fabrication are the generation of fake data in a database and the generation of enough data, process and network traffic to consume the system resources.

Specifically, a website may encounter the following threats [9]:

I. **Defacements** refers to the modification of the website's appearance.

II. **SEO spam** affects the search engine results by injecting code into the website. That code redirects to other websites, thus, search engines rank the website as risky.

III. **Malicious redirect attacks** link the website with infected websites. When the website's domain is requested, browser is redirected to infected websites.

IV. **iFrame injections** install a hidden iFrame on the website. That iFrame appearing like an ad is actually an infected website that loads along with the website.

V. **Phishing scams** disguise themselves as trustworthy websites and services to collect sensitive information such as usernames, passwords and credit card details.

VI. **Backdoor shells** are pieces of PHP code installed on the website that grant administration privileges to the attacker.

The countermeasures dealing with the threats to a website vary and be reactive or proactive. First and foremost, the website's security policy determines the security requirements and what should be done in order to avert a security issue or in case a security issue arises. In particular, it briefly and at a high level involves security practices that the user of the information system should follow [8]. The security policy is written on a document covering the topics of acceptable use of information technology, training and awareness, physical security, password policy and emergency planning [8].

Many websites encounter broken authentication vulnerabilities or, in other words, the administration login page is exposed to brute-force attacks. Thus, the administration login page should not be publicly accessible [8]. Another common website vulnerability that is exploited by malicious attacks, such as SQL injection, Cross-Site Scripting (XSS) and Remote File Inclusion (RFI), derives from the lack of input sanitization [8]. That is, the attacker sends data exploiting security holes in order to perform unauthorized actions [8]. Input sanitization should not only be applied to the website content such as form inputs, but URLs should also be sanitized [8].

Website vulnerability problems are often solved by updating the CMS version and by installing patches [9][10]. Moreover, anti-malware software scans for malicious content in the website and cleanses it. Web Application Firewalls can block suspicious incoming and outgoing traffic once it reaches the server, while DNS Level Website Firewalls block malicious traffic before it reaches the server by routing it through cloud proxy servers [10]. Setting strong passwords, determining user permissions and user's accountability can also be effective proactive actions [9][10]. An extra level of website security measures may include: multifactor authentication, limitation of login attempts, disabling theme and plugin editors, filtering access to the administration panel by IP address and disabling PHP execution [9][10].

2.4 Website design principles

According to Human-Computer Interaction a User Interface design should comply with Nielsen's ten usability heuristics. Nielsen's heuristics apply to website design [11][12]:

I. **There must be visibility of the system status**. The users should recognize whether the site is loaded or loading and receive appropriate feedback.

II. **There must be connection between the real world and the system**. The users should be able to recognize the words and concepts that are familiar to them. They should be able to navigate by following the website's logical structure.

III. **There must be user control and freedom**. The website should provide redo and undo action options.

IV. **The User Interface design must be consistent and follow standards**. This means that the responsive design of the website should support different devices and facilitate the navigation flow.

V. **There must be error prevention**. The website should warn the user in case they attempt to submit invalid inputs or perform risky actions.

VI. **The User Interface design must enhance recognition rather than recall**. The website should provide clear instructions and help the users easily retrieve or remember information.

VII. **There must be flexibility and efficiency of use**. The website design should meet the needs of both novice and expert users. It should contain shortcuts to frequent actions.

VIII. **The User Interface design must be aesthetic and minimalistic**. That is, the website information should be represented visually when possible and redundant information should be avoided.

IX. **The User Interface design must help users recognize, diagnose and recover from errors.** The website should inform the users about errors occurred (e.g. 404 page not found) and recommend solutions.

X. **The User Interface design must include help and documentation**. Such information should be easily accessible to the users of the website and precisely describe the steps the users have to take in order to carry out a specific task.

More design principles improving user's experience and satisfaction are recommended in [13]:

XI. The website design should implement **visual hierarchy**. This does not necessarily mean that the essential elements of the website should be bigger than the less important elements, but the essential elements should be such highlighted that they attract the attention of the user.

XII. The website design may be dimensioned in accordance with the **golden ratio** as it is aesthetically appealing. One dimension of the layout should be submultiple or multiple of the other dimension of the layout and the multiplier should equal 1.618.

XIII. **Hick's law** indicates that the more the available choices the longer it takes to make a decision. In other words, distracting and redundant options should be omitted.

XIV. **Fitt's law** indicates that the time needed to move to a target area is logarithmically proportional to the distance to the target and logarithmically inversely proportional to the size of the target. Fitt's law suggests that it be easier to use an object being big and close to the user. This resembles the rule of target size, which dictates that the size of an object should be proportional to the expected frequency of its utilization.

XV. **The rule of thirds** is applied to images. If an image is equally divided into nine equal parts by two vertical lines and two horizontal lines and the essential visual components are placed along the line or at the intersections of the lines, the image becomes aesthetically more pleasing.

XVI. **Gestalt's law** dictates that the human eye perceives the individual parts of a picture after it perceives the picture as a whole. Concerning the website design, elements forming a distinct group should be placed near each other and should have common shape, color or other common features, or form as a group unfinished known shapes. Moreover, clustering of website elements can be carried out according to the law of symmetry.

XVII. **White space** is essential to clean design. White space is utilized for the hierarchy of information. That is, it contributes to making the user focus on the important information of the website.

XVIII. **Occam’s razor** recommends that hypotheses being more than the necessary ones be avoided. Occam’s razor endorses website design conveying straightforward messages and not letting the users get confused.

2.5 Classification of websites

The classification of websites is not an easy task due to the different emerging types of websites and due to the mixed purposes they serve [15]. Selcuk Cebi based on user’s expectations and on the purposes of the websites proposes a classification of websites, which, as he claims, includes all types of websites [15]:

- **Commercial website**: The purpose of the website is the profit and users expect spending their money to purchase products or services. Depending on the parties involved in the transaction, such websites are further into B2B, B2C and C2C, where B stands for business and C stands for consumer.
- **Service website**: The purpose of the website is to provide either information or entertainment or communication or “self-service” operations (e.g. internet banking, e-government website) at no cost.
- **Mixed type website**: The purpose of the website is a combination of the purpose of a commercial website and that of a service website. For instance, a gambling websites aims at making money and providing entertainment simultaneously.

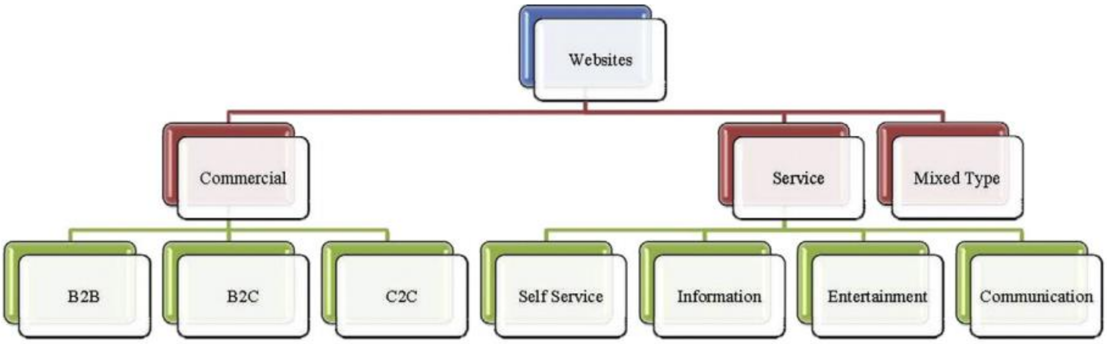


Figure 5 Classification of websites proposed by Selcuk Cebi [15]

Another classification of websites relies on the information content and on the website services and divides websites into four types [14]:

- **Presence websites**: The website consists of static web pages and a few online services. Such websites may belong to small and medium-sized enterprises or refer to conferences. As their name indicate, their purpose is to inform the public about the existence of their owners or of an event.

- **Catalogue websites**: These websites contain complex informational content and a few online services. They administer large amounts of data. For instance, an academic website administers programmes, courses, schedules, e-learning platforms, projects etc.
- **Service-oriented websites**: Despite their informational content being basic, they are able to deliver large amounts of data. They provide deliver complex services being delivered via a user-friendly interface. Research engines and online messaging services are examples of service-oriented websites.
- **Web information systems**: Their informational content and services are sophisticated. Commercial websites, intranets and extranets of companies can be characterized as web information systems.

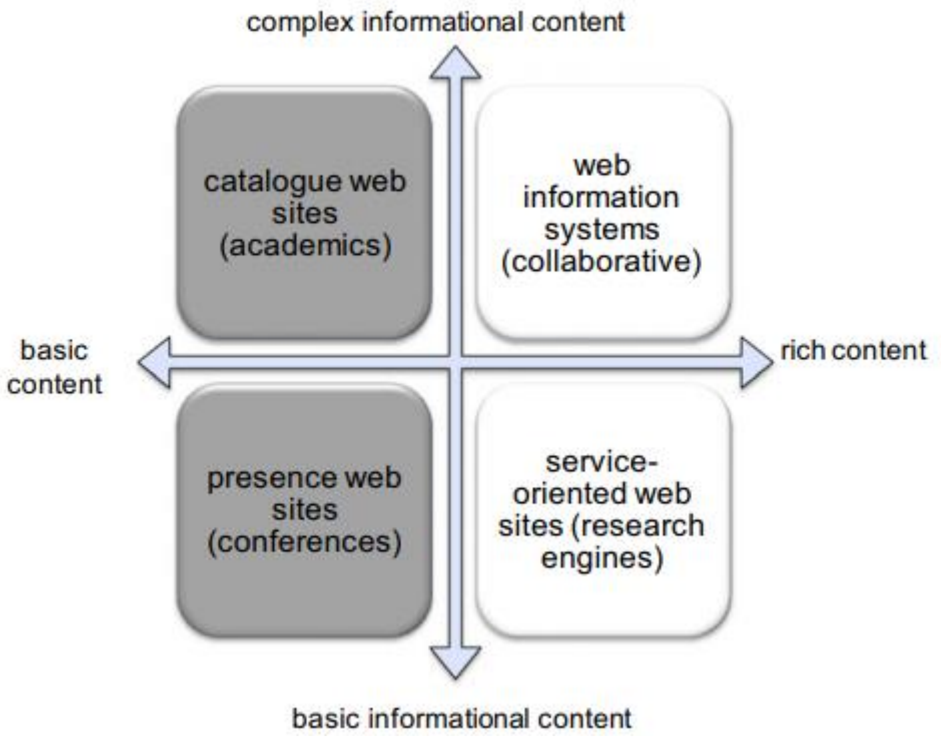

complex informational content
catalogue web sites (academics)
web information systems (collaborative)
basic content
rich content
presence web sites (conferences)
service-oriented web sites (research engines)
basic informational content

Figure 6 Classification of websites according to their informational content and services [14]

Moreover, a recent classification of websites identifies ten popular types of websites [16]:

- **E-commerce**: Online shops that sell products, which consumers can order and purchase from home.
- **Blogs**: Personal websites where people write about their personal lives, their viewpoints and about other topics they wish to discuss.

- **Informational**: These websites provide educational material, tutorials and tips for a variety of topics. They are usually found in the search engine results when the answer to a particular problem is searched.
- **Online community**: An online forum is an example of an online community. Their purpose is the communication of people with common interests and working together.
- **Photo sharing**: Users can exchange photos through such websites or even earn money by selling photos (e.g. pixabay.com).
- **Resume**: Personal websites whose owner present their resume in order to showcase their skills to potential employers.
- **Portfolio**: The purpose of such a website is similar to that of a Resume website. However, the owners showcase their work as well.
- **Catalogue and brochure**: Such websites promote products and services, which are not purchased online, to attract potential customers.
- **Business directory**: They provide list of resources such as phone directories or service directories.
- **Bio**: Authors or artists own such websites. Bio websites contain the biography and catalogues of the work of their owners, as well as blogs and social media links.

2.6 Website design evaluation metrics

The evaluation of user’s experience is of utmost importance, as flaws that need to be fixed may be detected in the website design. Evaluation metrics help detect such flaws by examining the user’s experience.

The lostness index is calculated for assessing user’s navigation on the website. Specifically, it compares the effort to identify a particular web page to the optimal path. The lostness index is given by the formula [11][17]:

$$L=\sqrt{\left(\frac{N}{S}-1\right)^{2}+\left(\frac{R}{N}-1\right)^{2}}$$

Where N equals the number of unique web pages the user visits to perform the task, S equals the total number of web pages visited including revisits to the same web page and R equals the number of the web pages of the optimal path that should be visited to perform the task. Lostness index being lower than 0.4 indicates that there are no vavigation issues on the website, while lostness index values being higher than 0.42 indicate that the user is expected to encounter nagivation problems [11][17].

The Unified Theory of Acceptance and Use of Technology (UTAUT) assesses the user's intention to use a specific system and helps identify the key factors to acceptance in any context [18]. Performance expectancy, effort expectancy, social influence and facilitating conditions are the parameters of UTAUT that influence user's behavior and they are affected by gender, age, experience and voluntariness of use [18].

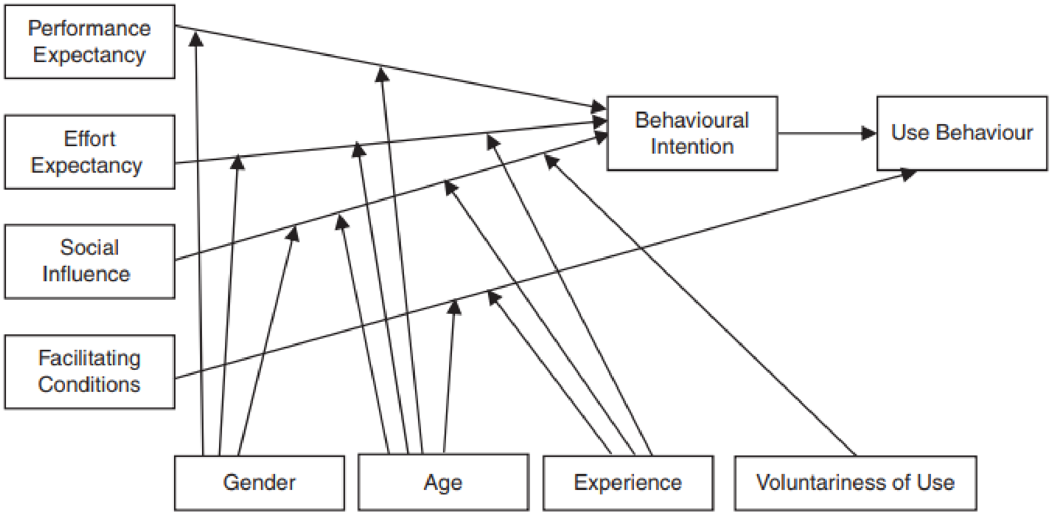


Figure 7 Unified Theory of Acceptance and Use of Technology model [18]

Performance expectancy is defined as "the degree to which the use of technology helps consumers perform certain activities", effort expectancy as "the ease with which consumers use technology", social influence as "the extent to which consumers believe that important others (e.g., family and friends) think they should use a particular technology" and facilitating conditions as "the consumers' perceptions of the resources and support available to perform a behavior" [19].

The Technology Acceptance model is determined by three parameters: perceived usefulness, perceived ease of use and perceived playfulness [11]. In particular, perceived usefulness refers to the degree to which the users believes the use of a particular system will increase their job performance and perceived ease of use refers to the degree to which the users believe that the system will be free of effort, while perceived playfulness examines whether the user focuses on an activity, whether the activity triggers user's cognitive curiosity and whether the user enjoys the interaction with the system [11]. What the users perceive is the satisfaction they receive by interacting with the system. However, the confirmation they receive is calculated if expected usefulness, expected ease of use and expected playfulness are subtracted from the values of the perceived variables [11].

A well-known framework for the evaluation of software quality is the ISO 9126 standard [20]. ISO 9126 defines six quality characteristics, each of which is composed of a number of sub-characteristics that can be in turn divided into attributes. The values of sub-characteristics and attributes are computed with the aid of metrics [21]. Quality factors that can be measured during the development process are referred to as internal and those being measured during the testing process are referred to as external factors [21]. In addition, the quality-in-use factors describe the user's perspective of quality [21]. The characteristics and the sub-characteristics of the ISO 9126 standard with their explanations are summarized in the table below.

Table 1 ISO 9126 software quality model [20]

Characteristic	**Sub-characteristic**	**Explanation**
Functionality	Suitability	Can the software perform the tasks required?
	Accurateness	Is the result as expected?
	Interoperability	Can the system interact with another system?
	Security	Does the software prevent unauthorized access?
Reliability	Maturity	Have most of the faults in the software been eliminated over time?
	Fault tolerance	Is the software capable of handling errors?
	Recoverability	Can the software resume working and restore lost data after failure?
Usability	Understandability	Does the user comprehend how to use the system easily?
	Learnability	Can the user learn to use the system easily?
	Operability	Can the user use the system without much effort?
	Attractiveness	Does the interface look good?
Efficiency	Time behavior	How quickly does the system respond?
	Resource utilization	Does the system utilize resource efficiently?
Maintainability	Analyzability	Can faults be easily diagnosed?
	Changeability	Can the software be easily modified?

	Stability	Can the software continue functioning if changes are made?
	Testability	Can the software be tested easily?
Portability	Adaptability	Can the software be moved to other environments?
	Installability	Can the software be installed easily?
	Conformance	Does the software comply with portability standards?
	Replaceability	Can the software easily replace other software?
(Refers to all characteristics)	Compliance	Does the software comply with laws or regulations?

2.7 Search Engine Optimization

Search Engine Optimization is the process of adaption and refinement of the website’s content in order for the search engines to be capable of discovering it effectively [22]. Keyword relevance plays an important role in SEO as web pages are usually indexed by keywords which briefly describe the content of the web page and facilitate searching [23]. Search engines apply algorithms that search for relevance to a keyword and rank the destination websites in order of quality [22]. Keyword relevance is measured with keyword distance or semantic proximity, which is defined as follows [22]: Assuming that K_1 and K_2 that are two sets of documents indexed by keywords k_1 and k_2 correspondingly, the formula for computing the semantic proximity of keywords k_1 and k_2 is:

$$S_p(k_1, k_2) = \frac{|K_1 \cap K_2|}{|K_1 \cup K_2|}$$

Search engines use bots called crawlers to continuously scan the web and index web pages. In order for a website to be displayed in the search results, the search engine should have previously indexed it [24]. Crawlers scan the website’s code for specific tags and descriptions and for links. The number and the quality (depending on the source) of the inbound links increase the credibility value of the website’s content [24]. At this point best practices increasing the discoverability of the website are listed [9][24]:

- The front page should describe the value proposition of the website with generic keywords.

- Unique content on each web page is more likely to be indexed by search engines. The content of web pages can be enriched with links to other websites and resources.
- The effect of blogging on the website is significant since every new post can be indexed as new web page.
- The title tag of the web page is listed when the web page appears in the search results. Therefore, every web page or post should have a representative title and contain keywords and phrases covering its content.
- Assigning names to the URLs of the web pages (permalinks) organizes the URL structure of the website and help search engines understand the content of the web page.
- The images of the website should be tagged with informative text (alt tags) and should have names being indicative of their content.
- If the website receives inbound traffic due to referrals from other websites being highly ranked by search engines, the search engines conceive the website as important. Thus, having authoritative websites post links to the website improves its ranking in the search results.
- Providing links to other web pages within the website in the form of "anchor text" makes those web pages more interesting to crawlers, hence, they rank higher in the search results.
- Headline tags assist both the visitors and the crawlers in identifying the important content of the webpage.
- Meta description is a text snippet in the HTML code of the web page that precisely describes what the particular web page is about. Crawlers search for meta descriptions in order to place the description of the web page under the "blue link" title tag in the search results. In addition, meta keywords is another text snippet in the HTML code which consists of a few keywords being relevant to the page's content.
- Submitting the website to a search engine enables the search engine to review the website. Once the website is submitted, the search engine checks it for crawling errors and begins crawling and indexing it.

- A crawler first examines the robots.txt file of the website so as to know what and where to look for. Therefore, the proper configuration of the robots.txt file can point out the information of the website that the crawlers should give priority.
- The registration of the website in directories adds links to the website from trusted sources.
- Social media integration assists in sharing the website’s content easily. Search engines rank higher content being shared on social media as the impact of social media is considered influential.
- Breadcrumbs are displayed on the top of a web page and allow the visitors to navigate the site easily. They are also used by crawlers for figuring out the website’s structure. Similarly, pagination solves navigation issues and improves crawling results.
- The performance and the valid code of the website affect how search engines crawl it.
- Long-tail keywords are keywords that are not often searched. The long-tail keywords should be related to the website’s content.
- Web analytics shows the list of search keywords that produce traffic to the website. This way, important keywords can be tracked and SEO results can be reviewed.

2.8 Web Hosting

A website needs a domain name to replace the Internet Protocol (IP) address. The IP address of the website is corresponded to the domain name of the website via the Domain Name System (DNS). The domain name of the website is its ID on the World Wide Web and it should be characteristic so that the users are able to remember and locate it effortlessly.

Aside from the domain name registration, the website data must be stored or hosted on a web server usually offered by a company and accompanied with software helping build the website. There are four main types of web hosting [32] [33]:

I. **Shared hosting:** The website shares the server resources with other websites hosted on the same server. Such web hosting services can be cheap and technically easy to use. However, the website performance may be low and affected by neighboring websites.

II. **VPS hosting:** Unlike shared hosting, Virtual Private Server hosting splits the server into virtual servers. As a result, each website has its own resources as if it was hosted on a dedicated server. In addition, the owner of the website has root access and is able to customize his server's environment.

III. **Dedicated hosting:** The owner of the website totally controls the dedicated web server. The website performance increases and so does its cost.

IV. **Cloud hosting:** Although the function of cloud hosting resembles that of VPS hosting, it is ranked as a better solution than the latter, since it leverages the resources of many servers, whereas VPS hosting relies on standard server specifications. Cloud hosting can handle traffic spikes and mitigate DDoS attacks.

2.9 Software Development

The Software Development Life Cycle (SDLC) bridges the gap between the real world problems and the software that should be developed to deliver solutions [26]. Planning, analysis, design and implementation are the main phases of SDLC. In particular, during the planning phase the reasons for which the software should be developed and the work plan are determined. The analysis determines the users of the system, what the system does, when and where it will be used [26]. During the design phase, answers to the question "how the system shall operate" are given and decisions about the necessary software, hardware and network components are made. Finally, the software is built during the implementation phase, during which the software is also tested, reviewed and all the specifications defined in the previous phases are reflected on the product [26].

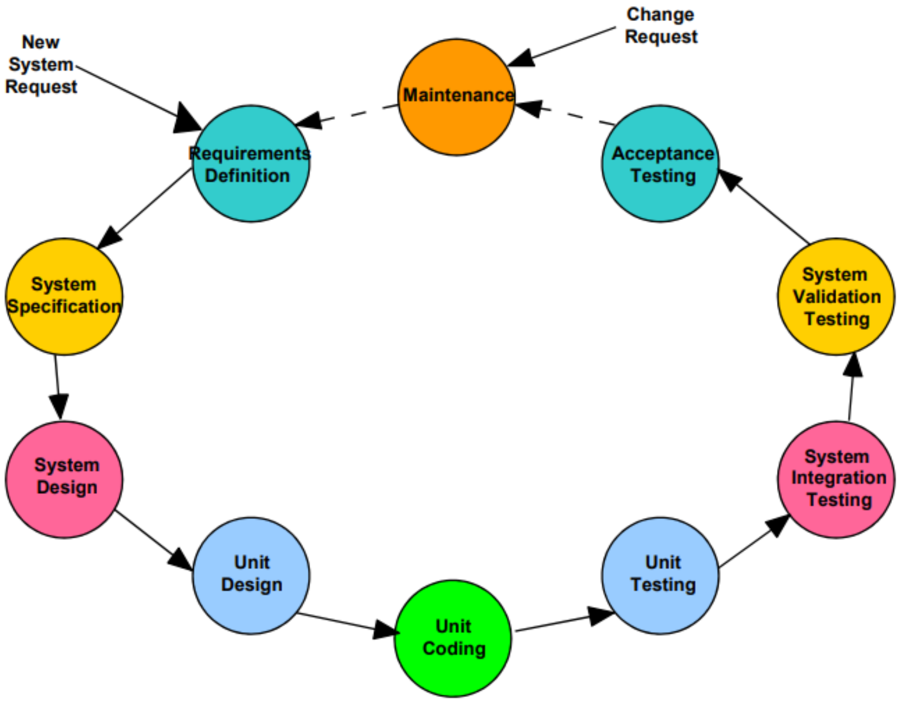


Figure 8 Software Development Life Cycle [25]

2.9.1 Methodologies

Even though development methodologies are alternative versions of the SDLC model, the sequences of their intermediate steps may differ from each other. The most prevalent development methodologies are the waterfall and the agile methodologies.

According to the waterfall development methodology, the work in each phase needs to be accomplished in order to move to the next phase. This approach has the key advantage of the software requirement being specified long before the implementation commences [26]. However, reconsidering the requirements in case an issue arises during testing or in case a new proposal shows up is both expensive due to the need for going through all the phases again, and difficult to manage due to the time having elapsed since the first phase and due to the time needed for making changes to the whole project [26].

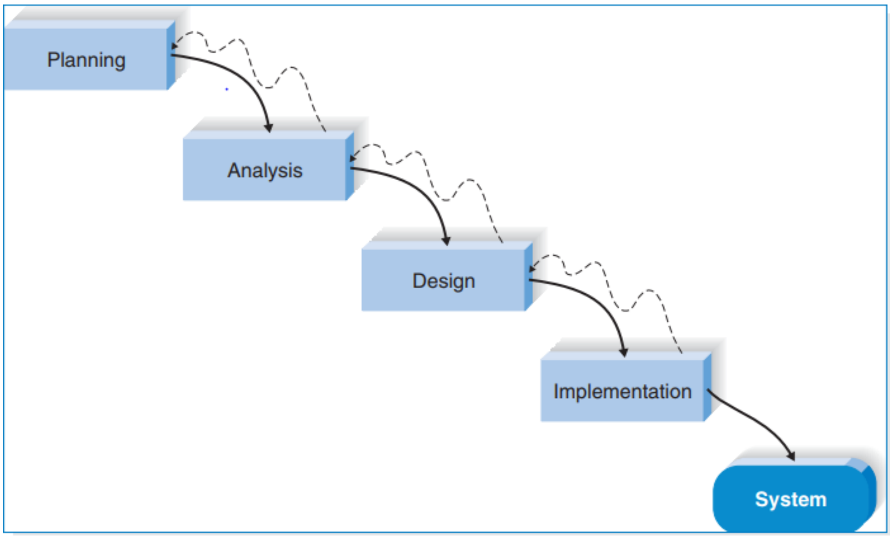


Figure 9 Waterfall development methodology [26]

Contrarily, the agile development methodologies, such as extreme programming (XP), Scrum and Dynamic Systems Development Method (DSDM), break down the project into short iterative applications, each of them requires a complete cycle of planning, analysis, design and implementation. Agile development methodologies can be significantly useful for developing projects with unclear requirements and short time schedules [26].

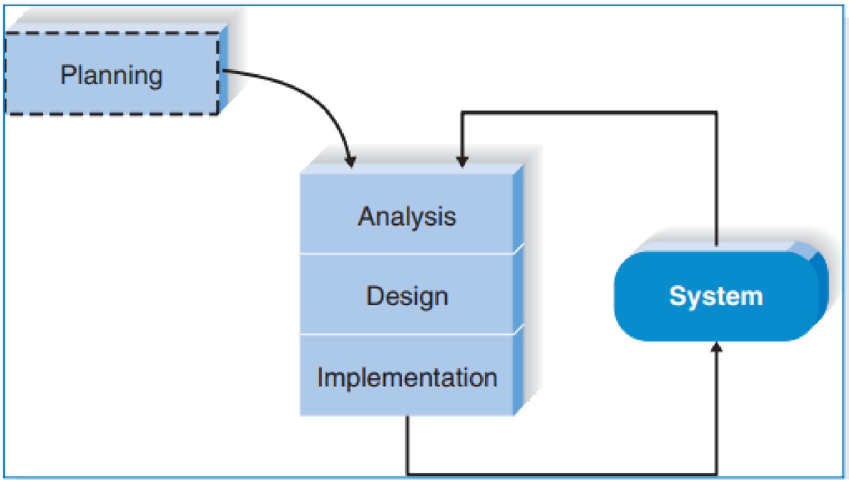


Figure 10 Agile development methodology [26]

2.9.2 Functional and Nonfunctional Requirements

Functional requirements are statements of what services the software should provide to meet the user’s needs. Nonfunctional requirements are statements of what characteristics and constraints the system should have. Functional requirements are directly related to the functions of the system, while nonfunctional requirements are not (e.g. system's response time) [25] [26].

Nonfunctional requirements may affect the overall system and if they are not met, the system may not function at all. They are classified into [25]:

- ✓ **Product requirements:** They specify the system's behavior in a particular manner (e.g. execution speed) and they are related to concepts such as usability, security, dependability, performance and space.
- ✓ **Organizational requirements:** They derive from organizational policies and procedures (e.g. standards) and they are related to the environment and to the system's operation and development.
- ✓ **External requirements:** They derive from factors being external to the system such as the legislation or the system's interoperability and they are related to regulations, legislation, accounting, safety and ethics.

Before proceeding to the design phase, during which the developer figures out how the system performs specific tasks, these tasks must be specified. The analysis of requirements is a strenuous process, since stakeholders (end-users, managers, maintenance engineers and trade unions) are unable to state requirements or express them in their own terms or they may have conflicting interests. In addition, organizational and political factors affect the definition of requirements. Thus, the developer should debate with the stakeholders to discover every possible requirement, cluster, prioritize and negotiate the requirements in order to resolve any conflicts. Lastly, the developer should document the result of the analysis [25]. Provided that requirements may change due to new stakeholders or due to other factors, the analysis of requirements must be an iterative process including requirements elicitation, specification and validation [25].

2.9.3 Sitemaps

The sitemap diagram of a website depicts the hierarchical structure of the pages and visualizes the routes to specific content [35]. The sitemap diagram may be an outline listing the names of the pages and their hierarchical level or contain visual elements for representing the content of each page [35].

A website should also have an XML sitemap that specifies the website structure and the links to important content [36]. Thus, the XML sitemap helps search engines crawl the website and index the pages effectively.

2.9.4 Use Cases

Use cases are utilized as an effective means for expressing user requirements by illustrating the required user's interactions with the system in order to carry out certain tasks. They depict in detail the functional requirements of the system and contribute to the understanding of special cases, exceptions and error handling [26].

The preconditions in a use case define the system's state before the use case commences. The use case commences with a triggering event which may be external or temporal. The normal course of the use case lists the major steps to be performed in order to produce a certain system response [26]. Each step has an input and produces an output. The normal course is not the only path to finish successfully the use case, though. The user or the actor may follow alternative courses based on the logic of the use case. Each path of the use case the actor takes is called scenario. A use case also involves post-conditions and exceptions [26]. Specifically, post-conditions often influence the preconditions of the next use case in the series, and exceptions are error conditions not leading to the successful conclusion of the use case [26].

The Use Case diagram is a common UML (Unified Modeling Language) technique for presenting the system's functionality at a high level [26]. A use case diagram illustrates the use cases with [26]:

- ✓ **The actor:** The entity interacting with and receiving responses from the system.
- ✓ **Association relationships:** Lines connect the actors to the use cases with which they interact. Asterisks drawn at the ends of associations show multiplicity.
- ✓ **Use cases:** Each use case is represented by an oval and a descriptive verb phrase.
- ✓ **System boundary:** The system boundary is a box enclosing the use cases and separating external from internal system entities.
- ✓ **«Include» and «exclude» stereotypes:** The «include» stereotype denotes that a use case includes another use case in the flow of events, whereas the «exclude» stereotype denotes special cases that may occur in the flow of events.

2.9.5 Robustness Diagram

The robustness diagram is a method bridging the gap between the analysis and the design of software [37]. Although it is not included in the UML specifications, it employs stereotyped communication objects [37]:

- **Boundary object:** The interface with which the user interacts. Concerning a website, boundary objects can represent web pages or screens.
- **Entity object:** They are abstractions that represent physical or virtual entities.
- **Control objects:** They connect boundary objects to entity objects.

The robustness diagram combines the features of the class diagram and the activity diagram, since it both displays the involved classes and the software's functionality [37]. However, it does not directly connect classes to specific functions of the software. In a robustness diagram the objects are nouns and the communication messages are verbs. In particular, verbs and nouns can communicate with each other, verbs can also talk to each other, while nouns are not allowed to communicate with each other [37].

2.10 RESTful API

REST invented by Roy Fielding in 2000 stands for Representational State Transfer and is an architecture for distributed hypermedia systems and for the design of interconnected applications and web services over HTTP (Hypertext Transfer Protocol) [1].

2.10.1 RESTful Requirements

In order for an Application Programming Interface (API) to be characterized as RESTful certain requirements must be met [1]:

- **Uniform interface:** The web page resources should have unique logical URIs (Uniform Resource Identifiers) that are utilized for fetching the data. Every resource representation should be concise and include URIs linking to related data. All resource representations must follow certain standards regarding the name conventions, the data format (XML or JSON) and the link format, and must be accessible via a common method such as HTTP GET.
- **Client-server separation:** The client and the server application interfaces must be distinct from each other. This means that the client application interface

should only contain the resource URIs and should not have access to data storage.

- **Stateless:** The client – server interaction should be stateless. The server does not save anything about the HTTP requests the client makes and about the sessions of the client. The client requests should contain the required information, such as information about authorization and authentication, so as the server to understand them.
- **Cacheable:** The resources must be labeled as cacheable or non-cacheable. The caching of the server responses can be carried out on the server side or on the client side.
- **Layered system:** A REST system may be deployed on several layers, which means that the client may send the requests to an intermediary server rather than the back-end server.
- **Code on demand:** Despite the fact the client requests and the server responses are often static representations of resources in XML or JSON format, they can also be scripts of executable code.

2.10.2 HTTP Methods

RESTful APIs allow the implementation of every CRUD (Create, Retrieve, Update, and Delete) operation on the resources of a web application [1].

The HTTP GET method is used for the retrieval of the resource representations and does not change the values of the resources [1]. In case the resource is found on the server, the server response to the HTTP GET request is the code 200 (OK). Otherwise, the returned response code is 404 (NOT FOUND) [1]. In case the HTTP GET request is not syntactically correct, then the returned response code is 400 (BAD REQUEST) [1]. The HTTP GET method is an idempotent method or, in other words, it produces the same results no matter how many times it is executed [1].

The HTTP POST method is used for the creation of new subordinate resources within a collection of resources [1]. The server response to a successful HTTP POST request is the code 201 (Created) and contains information about the status of the request, information about the new resource and a location header [1]. In case the resource created by the HTTP POST request cannot be identified by a URI, the response code returned is 200 (OK) or 204 (No Content). In contrast with HTTP GET requests, HTTP POST requests are not cacheable, unless Cache-Control or Expires headers are included in the server response [1]. Moreover, making two identical HTTP requests results in two different resources with the same content [1].

2.10.3 HATEOAS

HATEOAS (Hypermedia as the Engine of Application State) is integral part of the REST architecture and differentiates it from other web application architectures [1]. The term “hypermedia” describes the content consisting of links to media types such as images, videos and text [1]. By using hypermedia links the client can navigate to resources in the same way a website visitor navigates to web pages by clicking on hyperlinks [1]. This way the hypermedia links contained in the server response determine the state of the application and the server is able to modify the URIs without affecting the clients [1].

2.10.4 Richardson Maturity Model

Leonard Richardson examined a hundred different web service designs and classified them into four categories based on the extent to which they comply with the REST architecture [40]. The determinant factors of the maturity of a web service are the URIs, the HTTP methods and the hypermedia the web service employs [40].

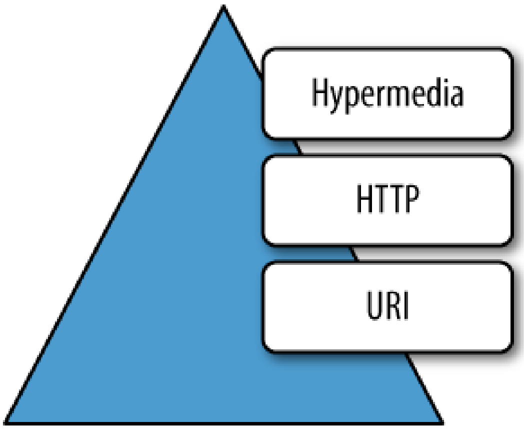


Figure 11 Factors determining the maturity of a web service according to the Richardson maturity model [40]

The table below summarizes the four categories of web services according to the Richardson maturity model.

Table 2 Categories of web services according to the Richardson maturity model [40].

Maturity level	**Specifications**
Zero	The web service does not utilize any URIs, HTTP methods and hypermedia. It has a single URI pointing to a single endpoint and uses the HTTP POST method to transfer Simple Object Access Protocol payloads.
One	The web service uses many URIs and assigns a URI to each individual resource. However, it employs only the HTTP POST method.
Two	The web service uses numerous URIs and assigns a URI to each individual resource. It also employs all HTTP verbs and is able to perform the CRUD operations on each resource.
Three	The web service uses numerous URI-addressable resources, employs all HTTP verbs and implements HATEOAS, which makes the discoverability of resources and the navigation to them easy.

2.10.5 WordPress Rest API

The WordPress REST API provides the capability of receiving data in JSON format from and sending data in JSON format to endpoints associated with WordPress data types [42]. Developers are able to remotely modify the content of the WordPress website from external applications, such as mobile applications, by utilizing any programming language [42].

The WordPress REST API can be described by [42]:

- ➢ **Routes – Endpoints:** They are URIs associated with data types and different HTTP methods.
- ➢ **Requests:** The WP_REST_Request class is the fundamental class handling internal PHP and remote requests submitted to the API. A WP_REST_Request object is generated upon the submission of a request.
- ➢ **Responses:** The WP_REST_Response class is the fundamental class handling the responses containing the data returned from endpoints or the errors produced by improper requests.
- ➢ **Schema:** It defines the structure of the API data and the API properties, such as the input parameters the API accepts and authentication methods for requests.

- **Controller Classes:** They administer the API Schema, handle requests to the API and determine the responses of the API.

2.10.6 WooCommerce Rest API

The WooCommerce REST API is fully compatible with the WordPress REST API and it is actually a subset of the WordPress REST API. This means that the WooCommerce data is processed in the same way the WordPress data is processed via the WordPress REST API [43].

The control of access to certain endpoints of the WooCommerce REST API (e.g. customers, orders) is achieved by implementing basic authentication. Specifically, WooCommerce uses two keys, which are the consumer key and the consumer secret, to grant access and execution privileges to HTTP requests [43]. Aside from the basic authentication method, the HTTP requests to the WordPress REST API can be authenticated by means of OAuth authentication or cookie authentication [44].

2.11 JSON

JSON (JavaScript Object Notation) based on a subcategory of the JavaScript programming language (Standard EMCA-262 ECMAScript) is a structured data-interchange language that supports data interchange among all programming languages [41].

The two fundamental structures in JSON are the collections of name-value pairs and the ordered lists of values [41]. The first structure is met as object or record or struct or dictionary or hash table or keyed list, and the latter is met as array or vector or list or sequence in other programming languages [41]. The facts that JSON facilitates the interchange of complex data among incompatible programming languages and that JSON does not require specific behaviors render JSON an ideal data – interchange language [41].

A JSON value may be an object or an array or a string or a number or a string or a Boolean or null [41]. JSON syntax includes braces, brackets, commas and colons for the determination of the different JSON data types [41]. The following figures illustrate the JSON syntax of the JSON data types [41].

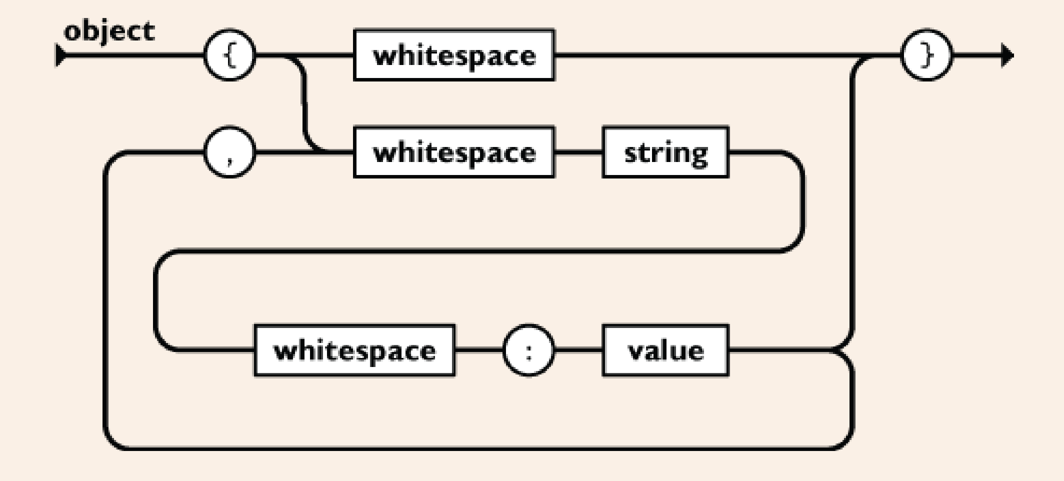


Figure 12 JSON object syntax [41]

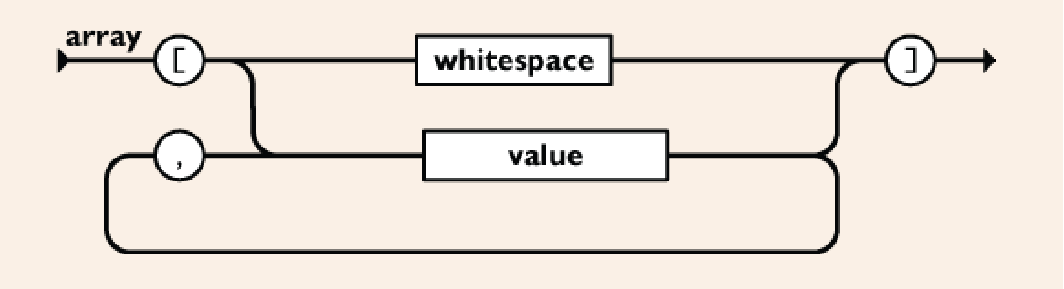


Figure 13 JSON array syntax [41]

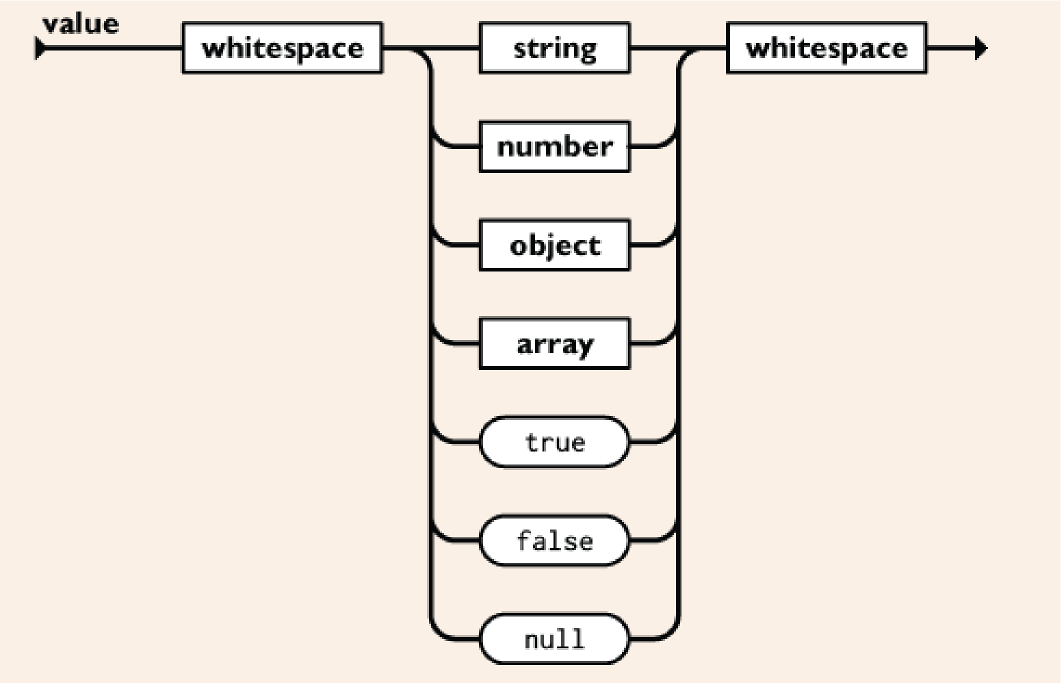


Figure 14 JSON variable or constant initialization and assignment statement [41]

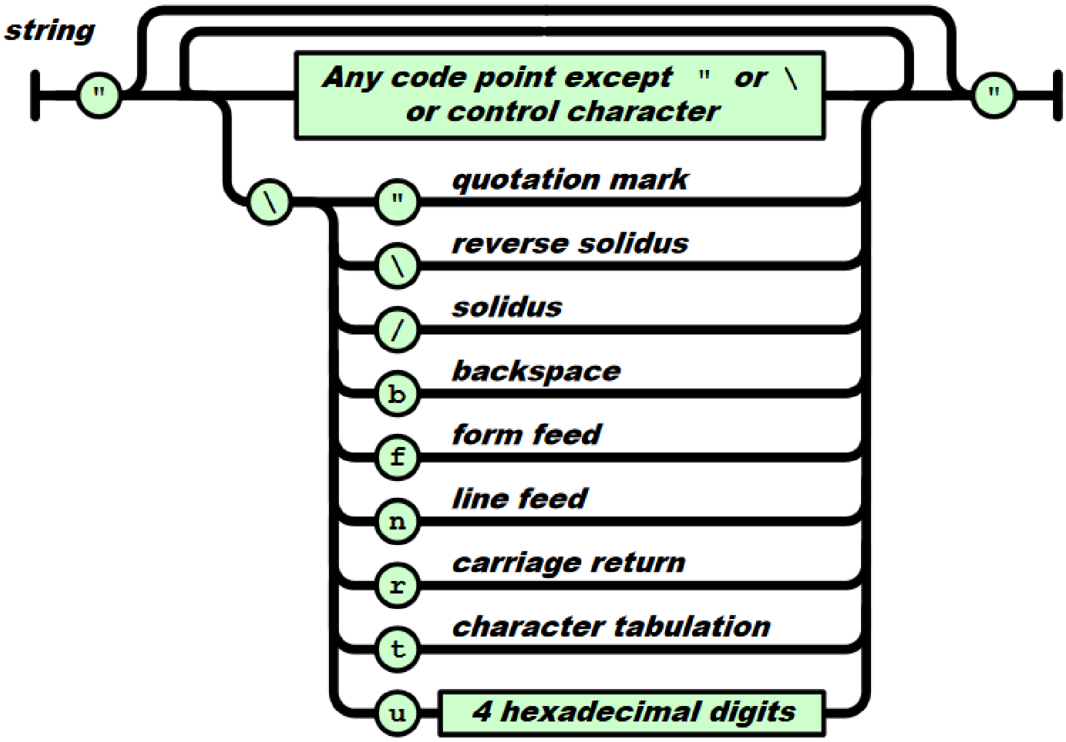


Figure 15 JSON string formation [41]

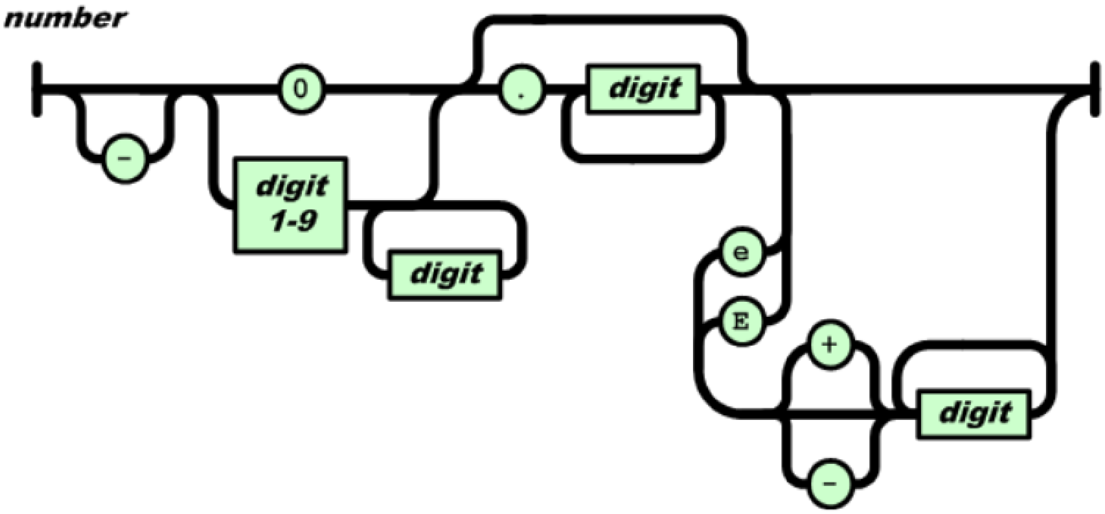


Figure 16 JSON number formation [41]

2.12 Cross – Origin Resource Sharing (CORS)

Due to security reasons, it is not feasible to call domain X from domain Y. If this was feasible, some JavaScript or AJAX malicious code could retrieve information from another website or perform advanced requests (POST, DELETE, PUT) without the user being able to notice [45][46]. However, cross – domain communication from the browser is necessary as resources not being hosted on a website may be needed [46]. Cross – Origin Resource Sharing (CORS) is a standard mechanism that enables resources belonging to domains outside the origin parent domain to be requested and loaded on the parent domain [45][46].

The addition of proper headers to the HTTP request allows the browser to allow or deny the cross – domain communication in accordance with the security settings [45]. The CORS mechanism works as follows [46]: The browser adds the domain to the header of the request along with key "Origin". The server checks whether the domain belongs to the allowed domains, which the server usually specifies in the .htaccess file. If the domain is an allowed domain, the server responds with a header named "Access – Control – Allow – Origin" and with the requested data. Otherwise, the response is an error.

2.13 Ionic Framework

Ionic Framework as an open source toolkit released under the MIT license is utilized for the development of high – quality cross - platform mobile applications [48]. It relies on web technologies such as HTML, CSS and JavaScript and its applications are able to run on iOS, Android, Windows Phone, desktop and the web as Progressive Web Apps [48].

2.13.1 Ionic Stack

Ionic mobile applications are developed in accordance with the three-layered Ionic Stack. In other words, they are built with the use of three primary technologies: Ionic, Angular and Cordova [47].

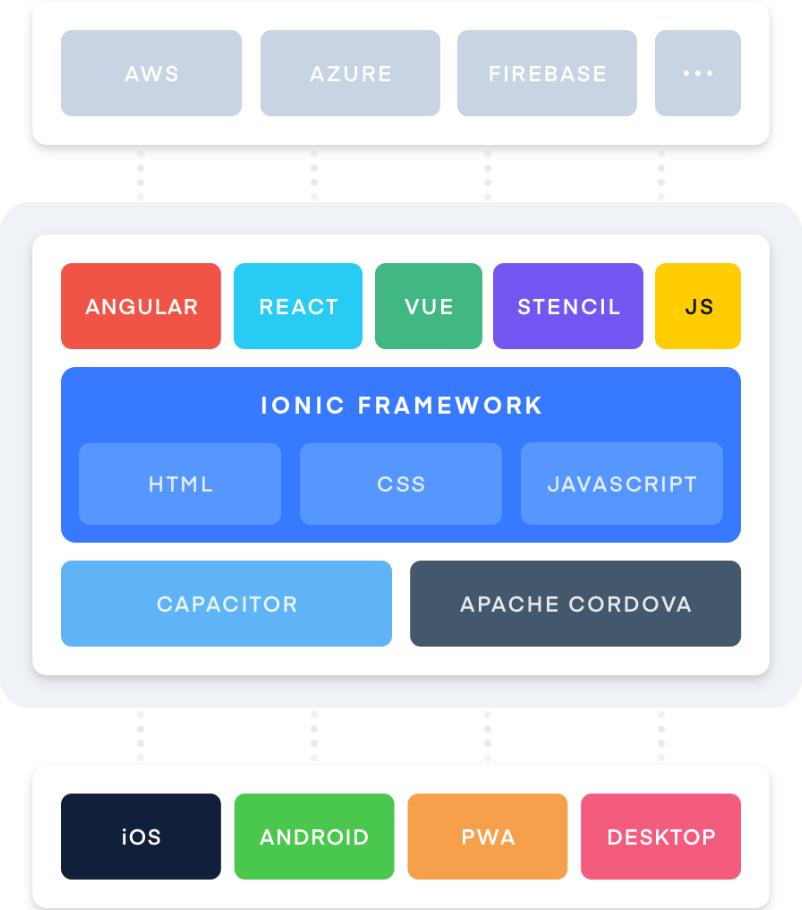


Figure 17 Ionic Stack [49]

The primary characteristic of the Ionic Framework is building the mobile application UI with interface elements which are not included in the web – based application development [47]. For instance, if a UI component cannot be provided as a native HTML element, Ionic Framework extends the HTML library and builds the component, which looks and interacts with the user just like the native element, by combining HTML, CSS and JavaScript code [47].

Angular or AngularJS is a web application framework supported by Google for developing complex, single-page web applications [47]. Ionic Framework leverages the characteristics of Angular for building custom UI components [47].

Cordova enables the development of native applications with web technologies via an integrated WebView [47]. In fact, it is the interface linking the WebView with the native layer of the device [47].

2.13.2 Ionic CLI

The Ionic Command – Line Interface (CLI) built with TypeScript and Node.js is a key tool for developing Ionic applications [48]. Ionic CLI is responsible for [47]:

- Creating a new application.
- Providing information about the application.
- Installing external packages on the application.
- Managing the available platforms.
- Managing Cordova plugins.
- Generating pages, components, directives, pipes, services and tabs for the application.
- Previewing and testing the application in a browser by running the local development server.
- Emulating the application on an Android or iOS or Windows emulator.
- Capturing the logs of the local development server.

2.13.3 Node.js

Node.js is an open source cross – platform environment for building server – side network applications written in JavaScript and being compatible with OS X, Windows and Linux [50]. It also provides an extensive library that facilitates the development of web applications using Node.js [50].

All Node.js APIs are asynchronous and event – driven, which means that the Node.js server does not wait for an API to return data. Instead, the Node.js server calls the next API and utilizes a notification mechanism for receiving the response from the previous API [50]. For this reason, Node.js is based on a single – threaded model with event looping being able to serve a larger number of requests than the Apache HTTP server [50]. In addition, the Node.js is built on Google Chrome’s V8 JavaScript Engine. As a result, the Node.js library executes code very fast and Node.js applications produce data in chunks instead of buffering data [50].

2.13.4 TypeScript

There are two versions of the next – generation JavaScript programming language: ES6 and TypeScript [47]. ES6 was formally presented in June 2015 and it is natively supported by browsers [47]. TypeScript supported by Microsoft leverages the features of ES6 and is the main programming language for Angular and Ionic applications [47].

Variables

Variables in TypeScript are declared with the keyword *let.* In ES6 variables were declared with the keyword *var* and were accessible outside the function in which they were defined. On the contrary, *let* variables are confined in the block in which they are declared [47].

Classes

The class syntax in TypeScript is similar to the object – oriented inheritance model of JavaScript [47]. The class attribute declaration, the constructor function and the class attribute and methods invocation resemble those of programming languages such as C# or Java [47].

Promises

The Promise object in TypeScript is used for performing asynchronous computations in a simpler way than traditional callbacks. In particular, a Promise describes an operation being not yet carried out but being expected to be carried out in the future [47][48]. This type of functionality is needed when submitting requests to remote servers or loading local data [47].

A Promise can be [47]:

- ✓ *Pending:* The Promise's result is not available since the asynchronous operation producing the result has not been completed yet.
- ✓ *Fulfilled:* The asynchronous operation is completed and a value result is assigned to the Promise's result.
- ✓ *Rejected:* The asynchronous operation fails and the Promise's result contains the reason of the operation's failure.

The Promise object registers callbacks to receive the result of the asynchronous operation with its *then* method. Promises provide the advantage of the rest of the code being able to execute while the response from the server is being expected [47].

Observables

Observables are objects implemented using the RxJS library. An Observable can resolve to multiple values asynchronously, whereas a Promise can resolve to a single value asynchronously [47]. Moreover, an Observable can be cancelled and re-executed with the use of an operator, whereas a Promise need to have access to the function returned from the Promise so that it is re – executed [47].

Arrow functions

Arrow functions in TypeScript render the code more concise and simplify the operations of functions. Arrow functions with the use of the => symbol replace the keywords *function* and *return* [47]. Arrow functions are often used in array manipulation.

Data types

TypeScript supports the JavaScript primitive data types: String, Number and Boolean. In addition, it supports arrays associated with any valid data type. *Any*, *null*, *undefined* and *void* are special data types in TypeScript and represent entities behaving just as their type annotation indicates [47].

2.13.5 Angular

Angular is based on the popular MV* (Model View Whatever) model of Google [47]. In Ionic 4+ the official stack of Angular for building the routing of an app and a series of Angular features are employed [48].

Components

“Component” is a decorator that characterizes a class as an Angular component and contains metadata which determines the processing, the instantiation and the usage of the component at runtime [51]. Ionic Frameworks allows the customization of components with the implementation of templates [47].

Inputs

Angular passes information into the Components by means of the Input module. For instance, in order to pass the variable “currentUser” into the <current–user> Component, the HTML code should look like: <current-user [user] = “currentUser”> </current-user>. Within the class an @Input binds to the user variable with the following syntax: @Component ({ selector : ‘current-user’, template: ‘ <div>{{user.name}}</div> ’ }) export class UserProfile {@Input user; constructor() {}

} [47]. Hence, the Component's template is enabled to render out the value of the user.name variable [47].

Templates

Templates are snippets of HTML code that Angular executes in conjunction with specific elements and attributes to generate the dynamic page content [47].

In the previous example the expression {{user.name}} performs the one-way data binding or rendering [47]. The binding properties of the Component are defined in the same way as currentUser property of the <current-user> Component is defined in the previous example [47]. Moreover, the two-way data binding in Angular can be achieved with the use of the ngModel notation (e.g. <input [(ngModel)] = “username”> </input>). This syntax synchronizes the this.username Component attribute with the input value [47]. Last but not least, a template may function in particular way when directives with the asterisk symbol are declared inside it. For instance, the directive ngFor in the template <componentX *ngFor = “let item of items”> </componentX> executes the componentX Component for each item of the collection named items [47]. However, the directive never renders componentX Component since it is executed in a template [47].

Events

Events in Angular are declared with the parentheses notation and activate methods of the Component's class [47]. For instance, the (click) event activate the methodToExecute() method when the button “Click” is clicked: <button (click) =”methodToExecute()”> Click </button>.

In order to broadcast a custom event from Component X to Component Y, the Output and EventEmitter modules need to be imported in the Component X class [47]. The new event is declared with the Output decorator in the Component X class and an EventEmitter object is assigned to it. The new event is triggered by itself and the new event object executes its emit(parameter) method to broadcast the desired parameter to Component Y. Component X is imported to the Component Y class, which has a method for handling the event [47].

Pipes

Pipes in Angular convert the value of a variable to a new value [47]. For instance, the value of the birthday variable within the interpolation is converted to a readable date value by means of the Date pipe function: <p> {{birthday | date}} </p> .

2.13.6 Apache Cordova

Apache Cordova is an open source framework that facilitates the development of native mobile applications with web technologies such as HTML, CSS and JavaScript. Apache Cordova renders the web application within a WebView. A WebView is a considered a native application component displaying web content within a native application [47]. The web application running in the WebView behaves just like any other web application running on a browser, since it displays HTML pages, executes JavaScript code, interacts with remote servers and runs media files [47]. Such an application is called a hybrid application.

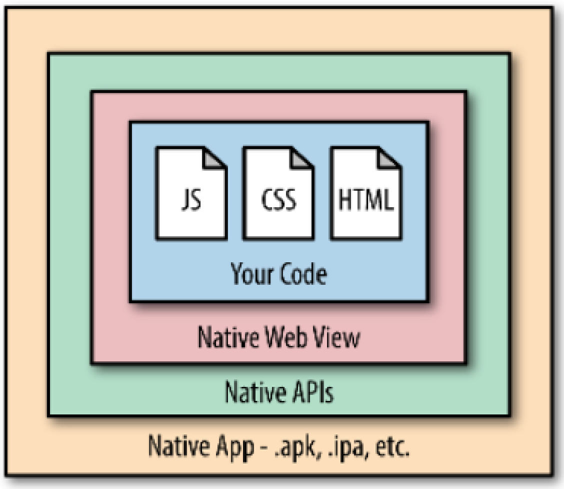
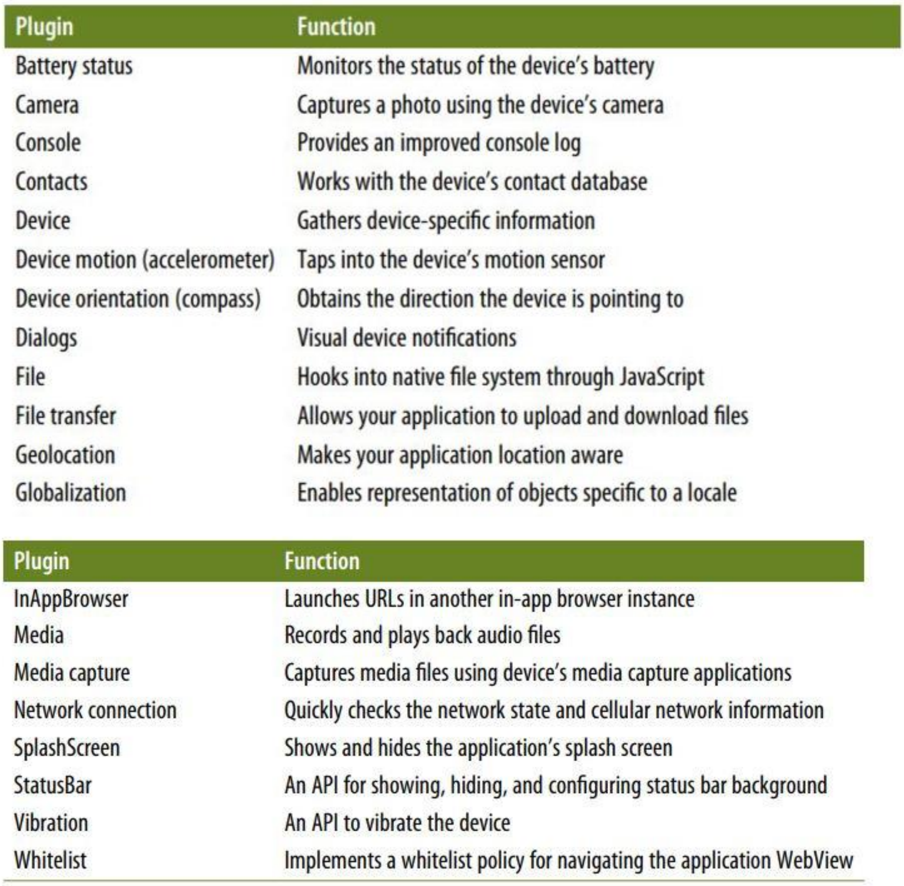


Figure 18 Composition of Apache Cordova applications [47]

Although web-based applications do not have direct access to hardware and software of the device on which they run, Apache Cordova provides access to the device features with certain JavaScript APIs exposed as a set of plugins [47].



Plugin	Function
Battery status	Monitors the status of the device's battery
Camera	Captures a photo using the device's camera
Console	Provides an improved console log
Contacts	Works with the device's contact database
Device	Gathers device-specific information
Device motion (accelerometer)	Taps into the device's motion sensor
Device orientation (compass)	Obtains the direction the device is pointing to
Dialogs	Visual device notifications
File	Hooks into native file system through JavaScript
File transfer	Allows your application to upload and download files
Geolocation	Makes your application location aware
Globalization	Enables representation of objects specific to a locale
InAppBrowser	Launches URLs in another in-app browser instance
Media	Records and plays back audio files
Media capture	Captures media files using device's media capture applications
Network connection	Quickly checks the network state and cellular network information
SplashScreen	Shows and hides the application's splash screen
StatusBar	An API for showing, hiding, and configuring status bar background
Vibration	An API to vibrate the device
Whitelist	Implements a whitelist policy for navigating the application WebView

Figure 19 Apache Cordova plugins [47]

2.13.7 Sassy CSS

Sassy (Syntactically Awesome Style Sheets) CSS (Cascading Style Sheets) is a stylesheet programming language compiling to CSS and allowing variables, mixins, nested rules and functions to be used in a CSS – compatible syntax [52]. Since SCSS is a superset of CSS, any CSS code can be executed as SCSS code [52].

SCSS uses the dollar symbol $ to declare its variables and provides the capability of nesting the CSS selectors in correspondence with the visual hierarchy of the HTML code [52]. Mixins create groups of CSS declarations that are often used, hence, enhance the reusability of the CCS code [52]. Moreover, the notation @extend is used for sharing CSS properties from one selector to another, hence, multiple class names on HTML elements are avoided [52]. Math operators are also supported in SCSS [52]. All the previous features render SCSS more effective than CSS.

3 Website and Mobile Application Development

As part of this dissertation, a website for the conference event is developed with WordPress. A mobile application facilitating the registration to the conference event and running on Android and iOS is also developed with Ionic Framework 4.

3.1 Conference Website

Initially, the functional and the nonfunctional requirements of the Conference website are defined. The design phase of the website begins with its sitemap in order for the hierarchical structure of the web pages to be visualized. As a second step of the design, the use cases are specified and analyzed. Additively, the process of purchasing products on the website is described with a robustness diagram. The implementation of the website is carried out with Wordpress using sample data from the 5th International Interdisciplinary Chaos Symposium on Chaos and Complex Systems [27]. Several Wordpress plugins such as Elementor, All-in-One WP Security, WooCommerce, Clicky Analytics, GDPR Cookie Consent, Mailchimp, WooCommerce Checkout Manager, WooCommerce Menu Cart, WooCommerce PayPal Checkout Gateway, Yoast SEO, Font Awesome, Elementor Addon Elements, Contact Form 7, Checkout Field Editor and Tidio Chat are also installed to add the appropriate functionality and meet the website’s requirements.

3.1.1 Functional and Nonfunctional Requirements

The website’s behavior, features and functions in terms of the visitor and of the administrator are listed below:

- ✓ The website shall inform the visitor about the conference committees.
- ✓ The website shall ask the visitor to submit relevant research papers.
- ✓ The website shall provide photographs from previous events.
- ✓ The website shall provide answers to frequently asked questions about the conference.
- ✓ The visitor shall be able to submit questions via an appropriate form.

- ✓ The website shall inform the visitor about the disciplines and the topics of the conference.
- ✓ The website shall present the conference venue to the visitor and provide relevant instructions.
- ✓ The website shall grant the visitor the capability of easily contacting the conference organizers.
- ✓ The website shall present the keynote speakers of the conference to the visitor.
- ✓ The visitor shall be able to read and download the conference schedule.
- ✓ The website shall instruct the visitor in submitting abstracts for the conference.
- ✓ The website shall inform the visitor about the registration details.
- ✓ The visitor shall be able to access and download archives from previous events.
- ✓ The website shall provide links to the social media accounts of the conference.
- ✓ The website shall clearly introduce the conference milestones to the visitor.
- ✓ The visitor shall be able to post comments in the blog section.
- ✓ The visitor shall be able to subscribe to the conference newsletter.
- ✓ The visitor shall be able to browse the available products (tickets and services).
- ✓ The visitor shall be able to place an order and easily complete the transaction.
- ✓ The visitor shall be able to review the order and update the shopping cart by adding or removing products.
- ✓ The website shall produce a receipt detailing customer’s order information including the customer’s personal details, the items purchased, the cost of each item, the VAT and the total cost.
- ✓ The website shall embed a Chatbot guiding the visitor.
- ✓ The administrator shall be able to log in to the WordPress panel.
- ✓ The administrator shall be able to change the website’s appearance.
- ✓ The administrator shall be able to customize the content and the appearance of each individual page.
- ✓ The administrator shall be able to receive messages from the contact form and from the FAQ contact form.
- ✓ The administrator shall be able to post new articles in the blog.
- ✓ The administrator shall be able to moderate comments submitted to the blog.

- ✓ The administrator shall be able to inspect the security status of the website and harden the security measures.
- ✓ The administrator shall be able to manage the logs and analytics of the website.
- ✓ The administrator shall be able to install any necessary plugins.
- ✓ The administrator shall perform search engine optimization on the website.
- ✓ The administrator shall be able to modify the source code of the website.
- ✓ The administrator shall be able to edit the products offered on the website.
- ✓ The administrator shall be able to modify the status of the orders made.
- ✓ The administrator shall be able to send newsletter to subscribers.
- ✓ The administrator shall be able to modify the purchase processes.

The website should also meet certain nonfunctional requirements:

- ✓ The website should be hosted on a web server providing at least a dual 2GHz+ CPU capacity, 6 GB RAM capacity and 10 GB hard disk space. The web server should also provide unlimited bandwidth, 1 Gbps connection, cPanel account, SSH access, FTP access,, email accounts, access to the latest MySQL database, web file manager, UPS power backup and should support SSL encryption, HTTPS, subdomain creation, built-in security, 24/7 network monitoring, site backups, the latest version of Apache software or nginx, the latest version of PHP, Linux OS, JavaScript and access to log files.
- ✓ The domain name of the website should be indicative of its content.
- ✓ The website should be able to deliver its content to a considerable number of concurrent users.
- ✓ The website should verify and validate all user inputs and alert the user with notifications in case of errors.
- ✓ The website should be compatible with any browser and machine provided that the minimum hardware and software requirements are met.
- ✓ The website should be scalable and expandable.
- ✓ The website should be backed up and restored if needed.
- ✓ The website should retain log files of the inbound and outbound traffic.
- ✓ The website should comply with international regulations and legal frameworks such as GDPR.

- ✓ The website should be kept up to date. The CMS used should notify the administrator about its available updates and about extension updates.
- ✓ The user should navigate the website and complete any action without much effort.
- ✓ The website should notify customers about the progress of their orders with e-mails.
- ✓ The website should have a blog section containing articles and news about the conference.

3.1.2 Sitemap

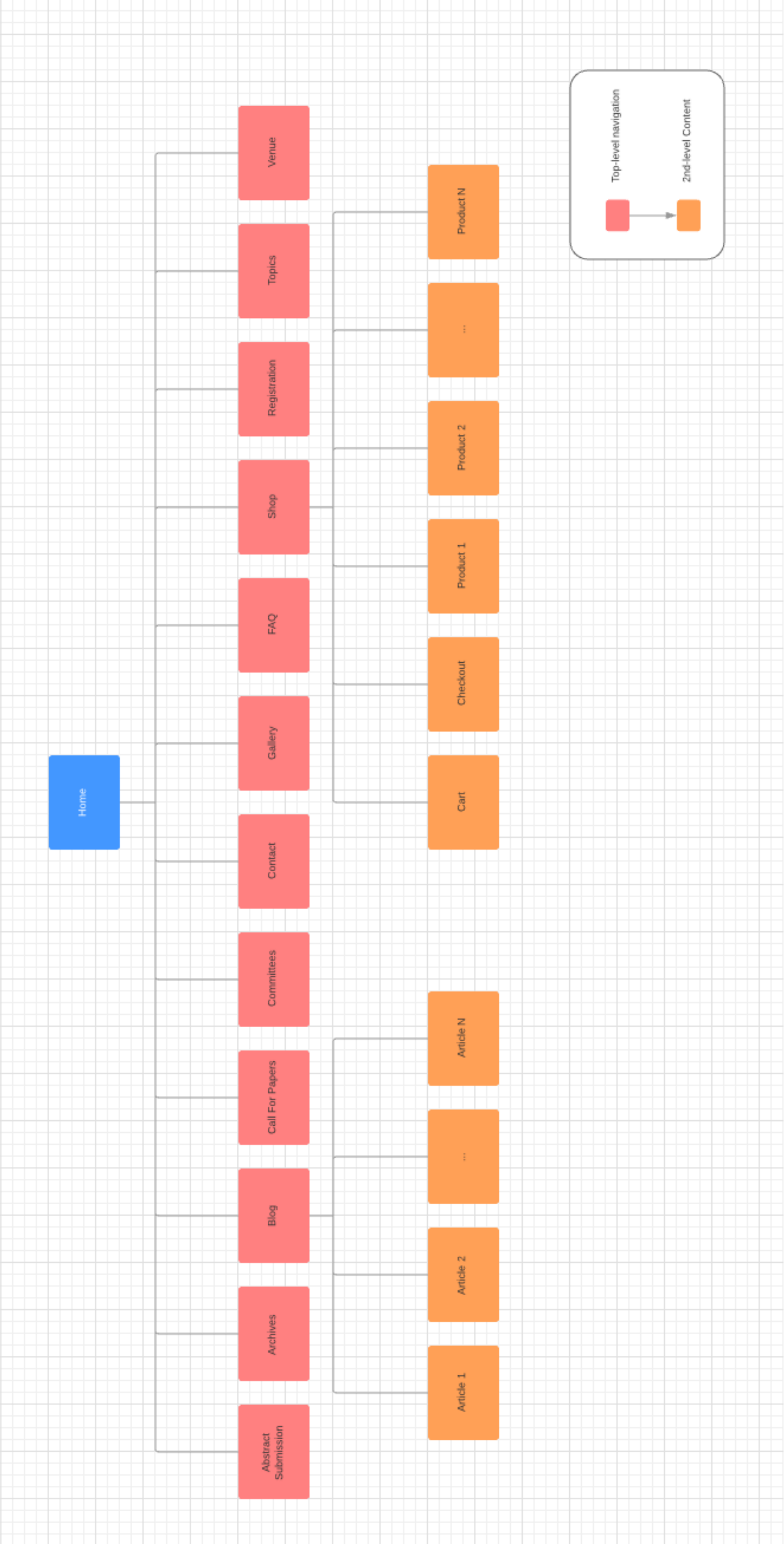


Figure 20 Conference website sitemap diagram

3.1.3 Use Cases

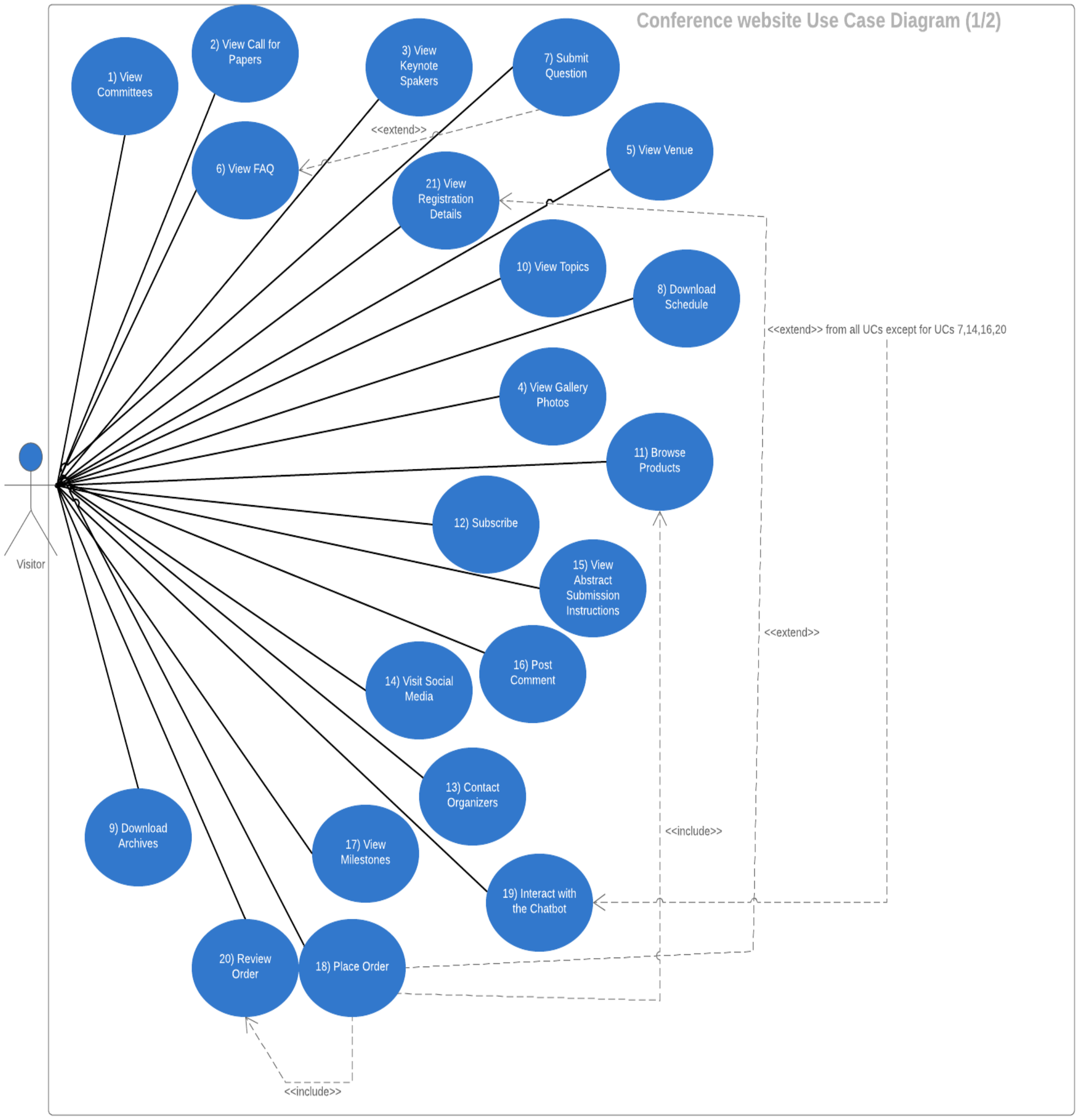


Figure 21 Conference website Use Case diagram (1/2)

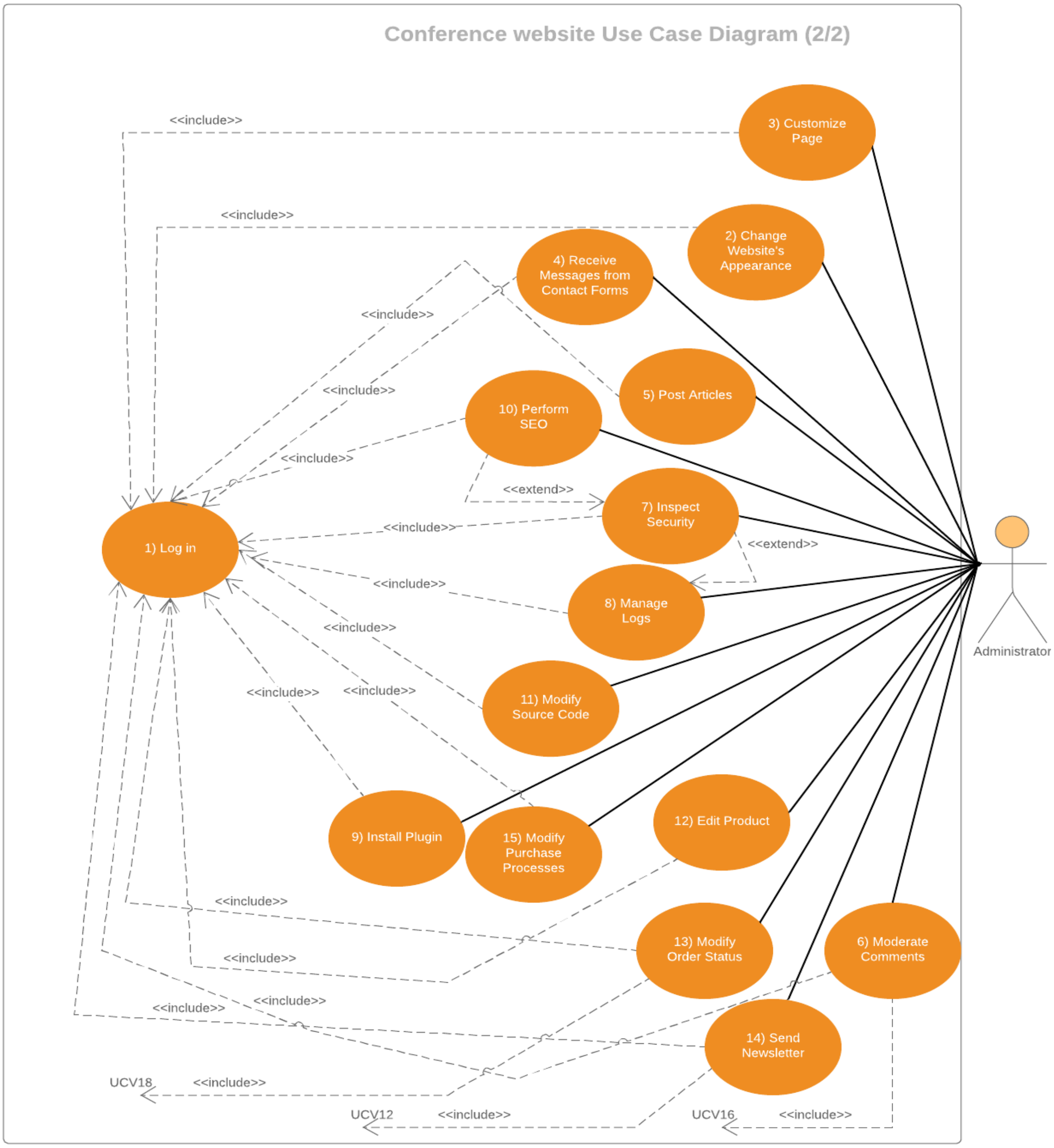


Figure 22 Conference website Use Case diagram (2/2)

ID	UCV1
Title: Description	View Committees: The visitor reads the topics of the committees of the conference.
Actor	Visitor
Preconditions	Enters the website's URL in the web browser.
Normal course	1a) Clicks on the "About" item of the main menu. 2a) Selects "Committees" from the drop-down menu to load the Committees page.
Alternative course	1b) Scrolls down the Home page. 2b) Clicks on the "View the Committees" button to load the Committees page. 1c) Scrolls down the Home page. 1c) Clicks the "Committees" hyperlink in the footer area to load the Committees page.

ID	UCV2
Title: Description	View Call for Papers: The visitor reads the appeal for papers being relevant to the conference.
Actor	Visitor
Preconditions	Enters the website's URL in the web browser.
Normal course	1a) Clicks on the "About" item of the main menu. 2a) Selects "Call for Papers" from the drop-down menu to load the Call for Papers page.
Alternative course	1b) Scrolls down the Home page. 2b) Clicks on the "Call for Papers" button to load the Committees page. 1c) Scrolls down the Home page. 2c) Clicks the "Call for Papers" hyperlink in the footer area to load the Call for Papers page.

ID	UCV3
Title: Description	View Keynote Speakers: The visitor learns about the keynote speakers of the conference.
Actor	Visitor
Preconditions	Enters the website's URL in the web browser.
Normal course	1) Scrolls down the Home page. 2) Clicks on the photograph of each keynote speaker.

ID	UCV4
Title: Description	View Gallery Photos: The visitor views photographs from previous conferences.
Actor	Visitor
Preconditions	Enters the website's URL in the web browser.
Normal course	1a) Clicks on the "About" item of the main menu. 2a) Selects "Gallery" from the drop-down menu to load the Gallery page.
Alternative course	1b) Scrolls down the Home page. 2b) Clicks on the "View Gallery" button to load the Gallery page. 1c) Scrolls down the Home page. 2c) Clicks the "Gallery" hyperlink in the footer area to load the Venue page.

ID	UCV5
Title: Description	View Venue: The visitor views details about the conference venue, such as the hotel facilities and location.
Actor	Visitor
Preconditions	Enters the website's URL in the web browser.
Normal course	1a) Clicks on the "About" item of the main menu. 2a) Selects "Venue" from the drop-down menu to load the Venue page.
Alternative course	1b) Scrolls down the Home page. 2b) Clicks on the "Learn about the Venue" button to load the Committees page. 1c) Scrolls down the Home page. 2c) Clicks the "Venue" hyperlink in the footer area to load the Venue page.

ID	UCV6
Title: Description	View FAQ: The visitor reads frequently asked questions about the conference.
Actor	Visitor
Preconditions	Enters the website's URL in the web browser.
Normal course	1a) Clicks on the "About" item of the main menu. 2a) Selects "FAQ" from the drop-down menu to load the FAQ page.
Post-conditions	The visitor submits a question to the conference organizers in case his question is not found on the FAQ page. (UCV7)
Alternative course	1b) Scrolls down the Home page. 2b) Clicks on the "Read FAQ" button to load the FAQ page. 1c) Scrolls down the Home page. 2c) Clicks the "FAQ" hyperlink in the footer area to load the FAQ page.

ID	UCV7
Title: Description	Submit Question: The visitor submits a question to the conference organizers.
Actor	Visitor
Preconditions	Enters the website’s URL in the web browser.
Normal course	1) Fills in the fields of the contact form (name, e-mail and question) on the FAQ page. 2) Clicks the “Send Question” button. 3) The website confirms that the question is received.

ID	UCV8
Title: Description	Download Schedule: The visitor downloads the conference schedule.
Actor	Visitor
Preconditions	Enters the website’s URL in the web browser.
Normal course	1) Scrolls down the Home page. 2a) Clicks the “Download Schedule” button.
Alternative course	2b) Interacts with the graphical elements in the “Schedule” section of the Home page.

ID	UCV9
Title: Description	Download Archives: The visitor downloads the proceedings of previous conferences.
Actor	Visitor
Preconditions	Enters the website’s URL in the web browser.
Normal course	1a) Clicks on the “About” item of the main menu. 2a) Selects “Archives” from the drop-down menu to load the Archives page. 3) Depending on the proceedings he wants to download from the list, he clicks the corresponding “Download” button.
Alternative course	1b) Scrolls down the Home page. 2b) Clicks on the “Access CCS Archives” button to load the Archives page. 1c) Scrolls down the Home page. 2c) Clicks the “Archives” hyperlink in the footer area to load the Archives page.

ID	UCV10
Title: Description	View Topics: The visitor reads the disciplines and the topics of the conference.
Actor	Visitor
Preconditions	Enters the website's URL in the web browser.
Normal course	1a) Clicks on the "About" item of the main menu. 2a) Selects "Topics" from the drop-down menu to load the Topics page.
Alternative course	1b) Scrolls down the Home page. 2b) Clicks on the "Read the Topics" button to load the Topics page. 1c) Scrolls down the Home page. 1c) Clicks the "Topics" hyperlink in the footer area to load the Topics page.

ID	UCV11
Title: Description	Browse Products: The visitor browses the products offered (tickets, taxi services, number of abstracts).
Actor	Visitor
Preconditions	Enters the website's URL in the web browser.
Normal course	1a) Clicks on the "Register" button in the banner area to load the Shop page. 2a) Clicks the title of a product to load the details of the particular product.
Alternative course	1b) Scrolls down the Home page. 2b) Clicks on the "Buy Ticket" button in the "Book your Seat" section to load the Shop page. 3b) Clicks the title of a product to load the details of the particular product.

ID	UCV12
Title: Description	Subscribe: The visitor subscribes to the conference newsletter.
Actor	Visitor
Preconditions	Enters the website's URL in the web browser.
Normal course	1) Scrolls down the Home page. 2) Inserts his e-mail address into the "e-mail" field in the "Receive newsletter" section. 3) Clicks the "Subscribe" button.

ID	UCV13
Title: Description	Contact Organizers: The visitor sends a message to the conference organizers.
Actor	Visitor
Preconditions	Enters the website’s URL in the web browser.
Normal course	1a) Clicks on the “About” item of the main menu. 2a) Selects “Contact” from the drop-down menu to load the Contact page. 3) Fills in the fields of the contact form (name, phone, e-mail, subject and message). 4) Clicks the “Contact Us” button. 5) The website confirms that the message is received.
Exceptions	3c) The visitor does not fill in the contact form properly. 3d) The website asks him to fill in the contact form properly.
Alternative course	1b) Scrolls down the Home page. 2b) Clicks the “Contact” hyperlink in the footer area to load the Contact page.

ID	UCV14
Title: Description	Visit Social Media: The visitor connects to the social media accounts of the conference.
Actor	Visitor
Preconditions	Enters the website’s URL in the web browser.
Normal course	1) Scrolls down the Home page. 2) Clicks on the respective social media account icon in the footer area to connect to the social media account.

ID	UCV15
Title: Description	View Abstract Submission Instructions: The visitor receives instructions about submitting abstracts for the conference.
Actor	Visitor
Preconditions	Enters the website’s URL in the web browser.
Normal course	1a) Clicks on the “About” item of the main menu. 2a) Selects “Abstract Submission” from the drop-down menu to load the Abstract Submission page.
Post-conditions	4) The visitor downloads the abstract template by clicking the corresponding button. 5) Submits online the abstract by clicking the corresponding button.
Alternative course	2a) Scrolls down the Home page. 2b) Clicks the “Abstract Submission” hyperlink to load the Abstract Submission page. 3a) Clicks on the “Submit Abstract” button in the banner area to load the Abstract Submission page.

ID	UCV16
Title: Description	Post Comment: The visitor posts a comment in the blog section.
Actor	Visitor
Preconditions	Enters the website’s URL in the web browser.
Normal course	1a) Clicks on the “Blog” item of the main menu to load the Blog page. 2a) Clicks on the title of an article to open the article page. 3) Fills in the fields of the comment form (name, e-mail, website and message) 4) Clicks the” Post Comment” button.
Exceptions	3a) The visitor does not fill in the contact form properly. 3d) The website asks him to fill in the contact form properly.
Alternative course	2b) Searches for the article by using the search bar or by clicking the respective tag or the respective category or the respective archive, and selects the title of an article to open the article page.

ID	UCV17
Title: Description	View Milestones: The visitor views the milestones of the conference.
Actor	Visitor
Preconditions	Enters the website's URL in the web browser.
Normal course	1) Scrolls down the Home page. 2) Taps on the flip boxes in the "Important Deadlines" section.

ID	UCV18
Title: Description	Place Order: The visitor places an order and completes the transaction.
Actor	Visitor
Preconditions	UCV11,UCV20
Normal course	1) Selects products and adds them to the shopping cart by clicking on the "Add to Cart" button of each product. 2) Clicks the "Proceed to checkout" button on the Cart page. 3a) Fills in his billing details on the Checkout page. 4) Selects the payment method (direct bank transfer or via the PayPal payment gateway) and completes the transaction. 5) The website sends an e-mail to the visitor which confirms that the order is received and includes the order details. The website also informs the visitor about the bank account details in case he selects the direct bank transfer method and redirects him to the PayPal payment gateway to complete the online transaction in case he selects the second payment method.
Exceptions	3b) The visitor does not fill in the billing details properly and the website asks him to fill in correctly the fields.

ID	UCV19
Title: Description	Interact with the Chatbot: The visitor interacts with the ChatBot and receives guidance concerning the website.
Actor	Visitor
Preconditions	Enters the website’s URL in the web browser.
Normal course	1) The Chatbot asks the visitor if he needs help. 2a) The visitors gives an affirmative response. 3a) The Chatbot lists the potential actions the visitor can take. 4a) The visitor selects an action.
Post-conditions	After the interaction with the Chatbot, the visitor may accomplish all the use cases having a UCV prefix except for UCV7, UCV14, UCV16 and UCV20.
Alternative course	2b) The visitor gives a negative response. 3b) The Chatbot does not list the potential actions.

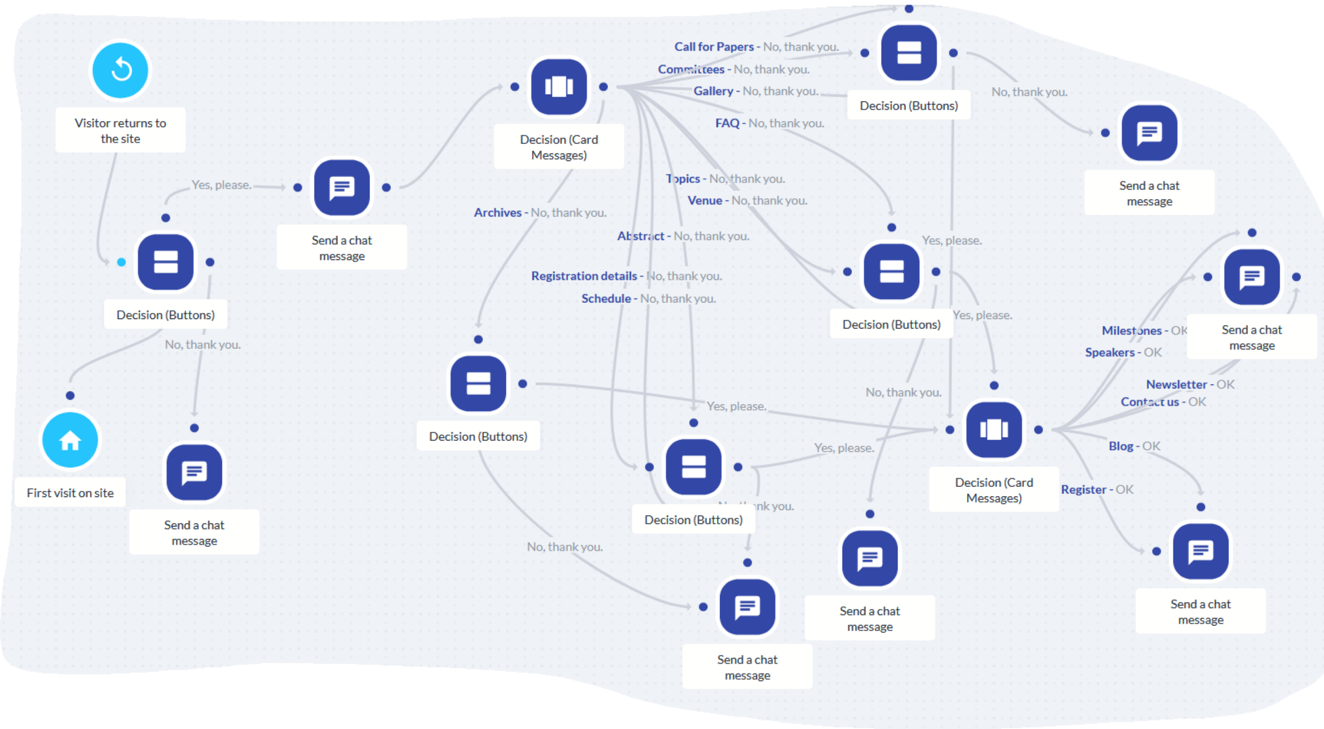


Figure 23 Functionality flow diagram of Tidio Chatbot

ID	UCV20
Title: Description	Review Order: The visitor reviews the order and adds product to or removes products from the cart any time before completing the transaction.
Actor	Visitor
Preconditions	Enters the website's URL in the web browser.
Normal course	1) Clicks the "Continue shopping" button on the Cart page to add products to the shopping cart. 2) Modifies the quantity of the products or removes products from shopping cart on the Cart page. 3) Clicks the "Update cart" on the Cart page to apply the changes he made.
Alternative course	Clicks on the shopping cart icon of the main menu to open the Cart page.

ID	UCV21
Title: Description	View Registration Details: The visitor reads the registration details about the conference.
Actor	Visitor
Preconditions	Enters the website's URL in the web browser.
Normal course	1a) Clicks on the "About" item of the main menu. 2a) Selects "Registration" from the drop-down menu to load the Registration Details page.
Post-conditions	The visitor clicks the "Buy Ticket" hyperlink to load the Shop page.
Alternative course	1b) Scrolls down the Home page. 2b) Clicks on one of the "More Details" buttons in the "Pricing Plans" section to load the Registration Details page. 1c) Scrolls down the Home page. 2c) Clicks the "Registration" hyperlink in the footer area to load the Registration Details page.

ID	UCA1
Title: Description	Log in: The administrator logs in to the Wordpress admin panel or to the cPanel of the web hosting service.
Actor	Administrator
Normal course	1) Enters the URL to load the authentication page. 2) Inserts his credentials. 3) Clicks the "Log In" button.

ID	UCA2
Title: Description	Change Website’s Appearance: The administrator changes the main appearance of the website.
Actor	Administrator
Preconditions	UCA1
Normal course	1) Clicks the “Appearance” tab in the admin panel. 2) Selects “Theme” to change the website’s theme. 3) Selects “Customize” to change general elements of the website such as the header, the banner and the footer. 4) Selects “Widgets” to add or remove widgets. 5) Selects “Menus” to change the main menus appearing on the website. 6) Clicks “Save” or “Publish” to apply changes.

ID	UCA3
Title: Description	Customize Page: The administrator customizes an individual page.
Actor	Administrator
Preconditions	UCA1
Normal course	1) Clicks the “Pages” tab in the admin panel. 2a) Clicks the “Edit” hyperlink to edit general settings and the content of the page or clicks the “Add New” button to add a new page. 3) Clicks the “Update” or the “Publish” button to apply changes.
Alternative course	2b) Clicks the “Edit with Elementor” hyperlink to edit the page with the Elementor plugin.

ID	UCA4
Title: Description	Receive Messages from Contact Forms: The administrator receives and reads the messages from the website’s contact forms.
Actor	Administrator
Preconditions	UCA1
Normal course	1) Opens the e-mail account he determined in the “Contact” section of the admin panel and to which the messages from the contact forms are forwarded. 2) Receives two different e-mail types, one from the “Contact Us” form and one from the “Question” form.

ID	UCA5
Title: Description	Post Articles: The administrator posts a new article in the Blog page.
Actor	Administrator
Preconditions	UCA1
Normal course	1) Clicks the “Blog Posts” tab in the admin panel. 2a) Clicks on the “Add New” button to write a new article. 3) Clicks the “Publish” button.
Alternative course	2b) Clicks the “Edit” hyperlink below an existing article to edit it.

ID	UCA6
Title: Description	Moderate Comments: The administrator moderates a comment posted in the blog section.
Actor	Administrator
Preconditions	UCA1, UCV16
Normal course	1) Clicks the “Comments” tab in the admin panel. 2) Selects a comment to moderate. 3) Approves or edits or reply to the comment.

ID	UCA7
Title: Description	Inspect Security: The administrator inspects the security status of the website.
Actor	Administrator
Preconditions	UCA1
Normal course	1) Clicks the “WP Security” tab in the admin panel to launch the dashboard of the All-in-One WP Security plugin. 2) Examines the current security risks of the website.
Post-conditions	The administrator may harden the security settings by activating the Login Lockdown, setting firewall rules, scheduling frequent file scans and setting file permissions.

ID	UCA8
Title: Description	Manage Logs: The administrator inspects the security status of the website.
Actor	Administrator
Preconditions	The administrator logs in to the Wordpress admin panel or to the cPanel of the web hosting service.
Normal course	1a) Logs in to his personal account on https://www.clicky.com, to which he linked the Clicky Analytics plugin. 2) Audits the inbound traffic and outbound traffic to the website. 3) Extracts conclusions about the website's effectiveness and security.
Post-conditions	UCA7, UCA10
Alternative course	1b) Logs in to the cPanel of the web hosting service and navigates to the "Metrics" section.

ID	UCA9
Title: Description	Install Plugin: The administrator installs any necessary plugins for the website's function.
Actor	Administrator
Preconditions	UCA1
Normal course	1) Clicks the "Plugin" tab in the admin panel. 2a) Clicks on the "Add New" button to add a new plugin. 3) Clicks on the "Install" button to install a specific plugin. 4) Clicks the "Activate" hyperlink to activate the plugin.
Post-conditions	The administrator may uninstall the plugin by clicking the "Deactivate" and "Delete" hyperlinks.

ID	UCA10
Title: Description	Perform SEO: The administrator performs Search Engine Optimization on the website.
Actor	Administrator
Preconditions	UCA1
Normal course	1) Clicks the "SEO" tab in the admin panel to launch the Yoast SEO plugin. 2) Inspects the SEO status of the website. 3) Clicks the "Configuration Wizard" hyperlink to customize the SEO settings of the website.

ID	UCA11
Title: Description	Modify Source Code: The administrator modifies the source code of the website.
Actor	Administrator
Preconditions	The administrator logs in to the WordPress admin panel or to the cPanel of the web hosting service.
Normal course	1a) Clicks the “Appearance” tab in the admin panel. 2a) Selects “Theme Editor” to access the website’s files. 3a) Modifies the source code of a file. 4a) Clicks the “Update File” button to apply changes.
Alternative course	1b) Logs in to the cPanel of the web hosting service and selects the “File Manager” option. 2b) Locates the file. 3b) Uploads the file with the modified source code and replaces the existing file.

ID	UCA12
Title: Description	Edit Product: The administrator edits a product appearing on the Shop page.
Actor	Administrator
Preconditions	UCA1
Normal course	1) Clicks the “Pages” tab in the admin panel. 2) Clicks the “Edit” hyperlink to edit properties of an existing product or clicks the “Add New” button to add a new product. 3) Clicks the “Update” or the “Publish” button to apply changes.

ID	UCA13
Title: Description	Modify Order Status: The administrator edits a product appearing on the Shop page.
Actor	Administrator
Preconditions	UCA1, UCV18
Normal course	1) Clicks the “WooCommerce” tab in the admin panel and. 2) Selects the “Order” item. 3) Clicks the title of an order. 4) Changes the order status to “Pending payment” or “Completed” or “Cancelled”. 5) Clicks “Update” to update the order status.

ID	UCA14
Title: **Description**	Send Newsletter: The administrator sends newsletter to the subscribers of the website.
Actor	Administrator
Preconditions	UCA1, UCV12
Normal course	1) Logs in to his personal account on https://www.mailchimp.com, to which he linked the Mailchimp plugin. 2) Clicks on the “Create” button. 3) Imports the list of subscribers. 4) Composes the newsletter e-mail. 5) Clicks the “Send” button.

3.1.4 Purchase Process

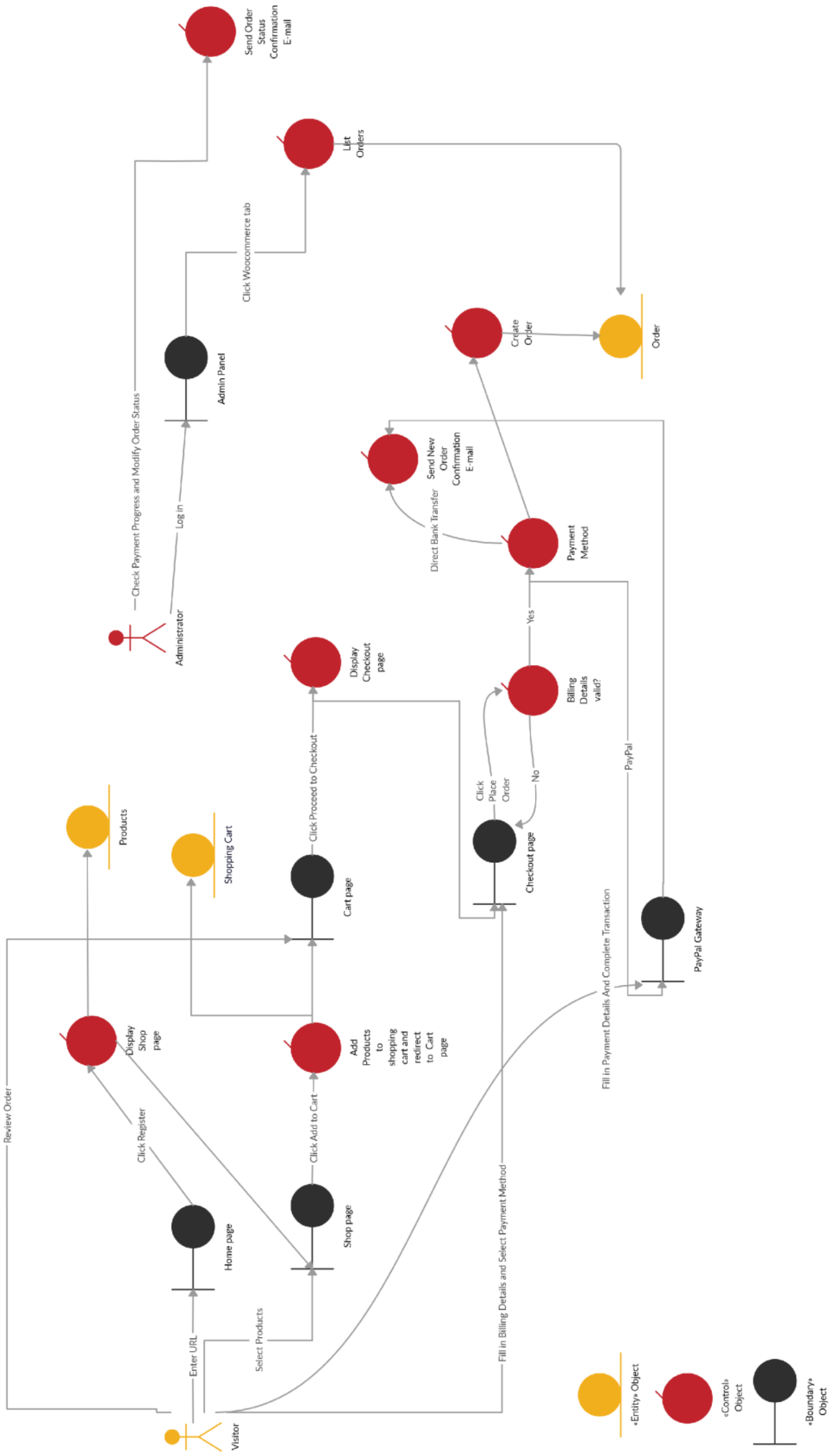


Figure 24 Robustness diagram of the purchase process on the conference website

3.2 Mobile Application

The functional and the nonfunctional requirements of the mobile application are defined as part of the analysis phase. The design phase includes the navigation flow and the use cases specification of the mobile application. The implementation of the application is carried out with Ionic Framework 4. The application is tested on the integrated emulator of Ionic Framework and on the Android Studio emulator.

3.2.1 Functional and Nonfunctional Requirements

The application’s behavior, features and functions in terms of the user are listed below:

- ✓ The user shall be able to view the homepage listing the available products when the app is launched.
- ✓ The user shall be able to navigate through the app with the aid of a side menu.
- ✓ The user shall be able to view general information about the app.
- ✓ The user shall be able to read the registration details of the conference.
- ✓ The user shall be able to list the available products by category.
- ✓ The user shall be able to view the details and the reviews of each available product.
- ✓ The user shall be able to add each available product to the shopping cart.
- ✓ The user shall be able to navigate to the Cart page and modify its content by changing the quantity of each product and by removing items from the shopping cart.
- ✓ The user shall be able to fill in his billing details and select the payment method on the Checkout page.
- ✓ The user shall be able to place an order and finalize it by paying via Direct Bank Transfer or via the PayPal gateway.
- ✓ The app shall be able to display a proper notification message in case the user attempts to proceed to the Checkout page with empty shopping cart.
- ✓ The app shall be able to display a proper notification message in case the user attempts to place an order without having properly filled in his billing details.
- ✓ The app shall be able to display a proper notification message with the order number after the user successfully placing his order.

The mobile application should also meet certain nonfunctional requirements:

- ✓ The app should be developed with the latest versions of Ionic Framework and those of the accompanying frameworks.
- ✓ The app should run on Android and iOS.
- ✓ The app should fetch on demand the necessary information from the REST API of the WordPress website.
- ✓ The app should post on demand the necessary t the WordPress website.
- ✓ The app should make the user aware of the pricing practices.
- ✓ The app should store the items in the cart irrespective of the app's state (active or inactive).
- ✓ The app should have a straightforward navigation flow supported with appropriate graphical elements.
- ✓ The app should integrate the PayPal payment gateway.
- ✓ The app should responsive to the user input or to external applications and return to the same state.
- ✓ The app should be scalable and handle as much data as needed.
- ✓ The app should load its content in less than 4 seconds on a regular device.
- ✓ The app should be reliable in such a way that acknowledges significant actions of the user with confirmation messages and anticipated responses.
- ✓ The app should secure user's information.
- ✓ The app should be available and accessible.
- ✓ The app should be responsive to different mobile devices and screen sizes.
- ✓ The app should utilize the internet connection of the mobile device to function as expected.

3.2.2 Navigation Flow

The navigation flow diagram illustrates the user navigation to the pages of the mobile application and the interaction of the application with data interfaces such as the WordPress REST API and the PayPal API.

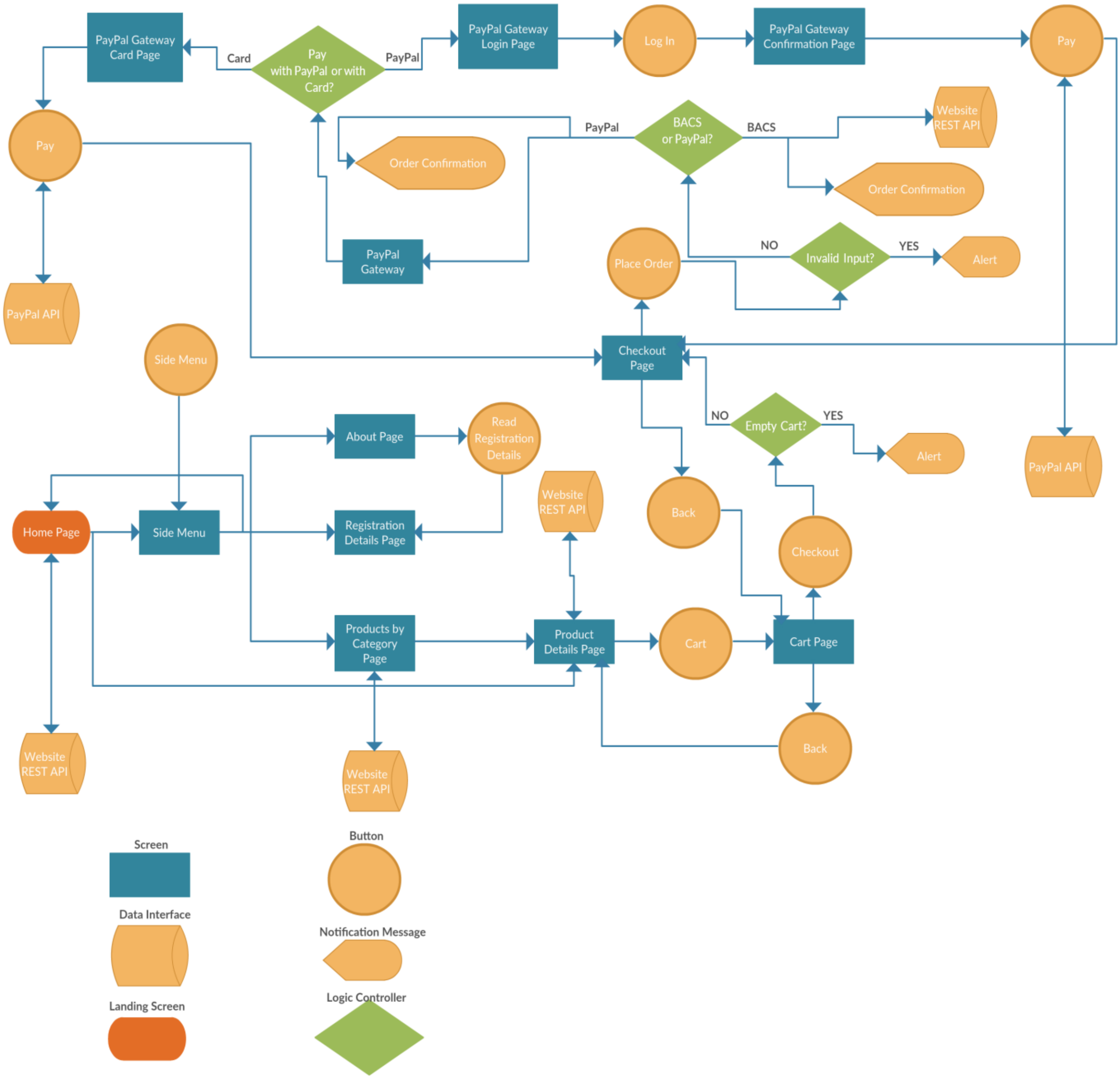


Figure 25 Mobile application Navigation Flow diagram

3.2.3 Use Cases

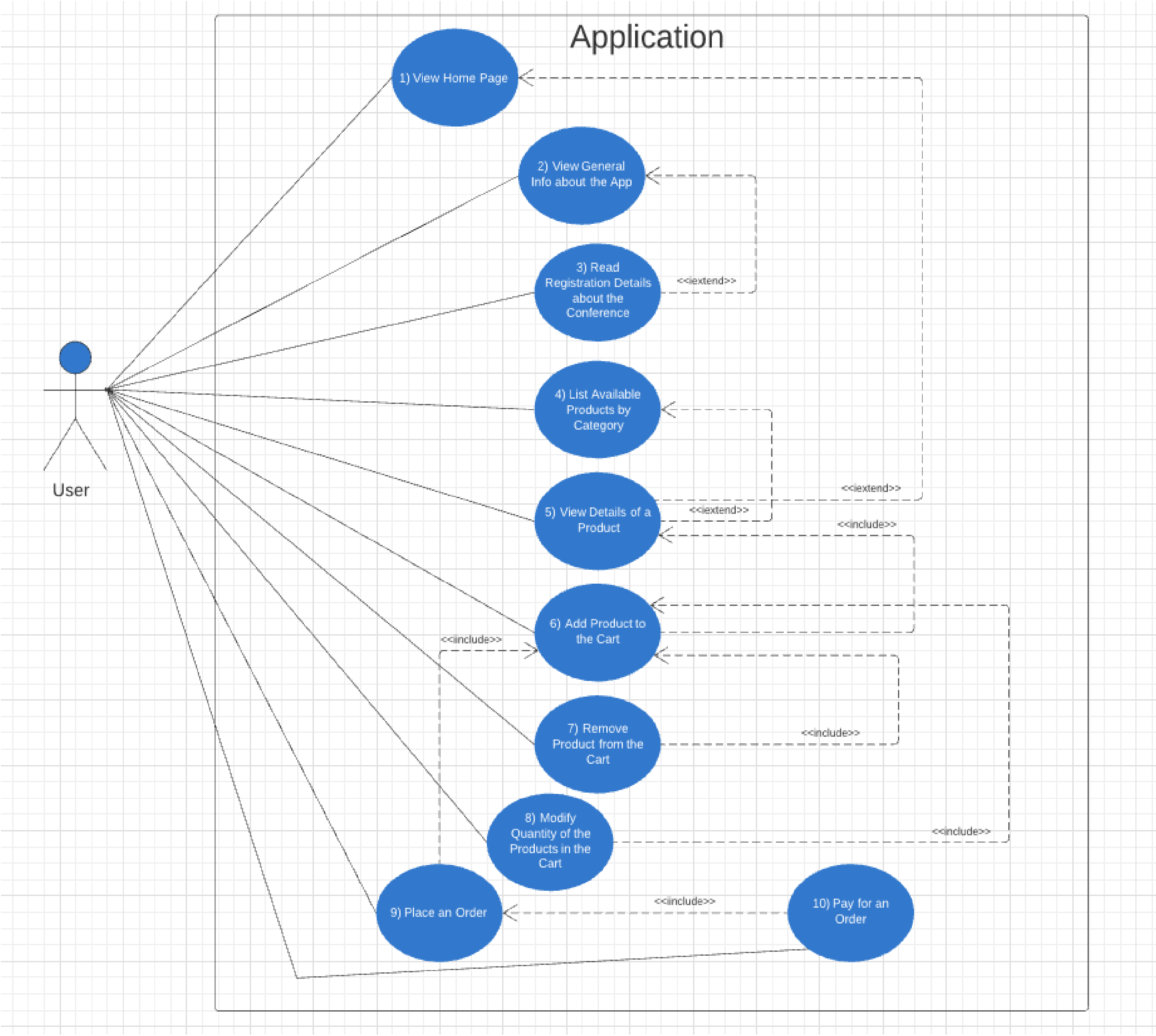

Application
User
1) View Home Page
2) View General Info about the App
3) Read Registration Details about the Conference
4) List Available Products by Category
5) View Details of a Product
6) Add Product to the Cart
7) Remove Product from the Cart
8) Modify Quantity of the Products in the Cart
9) Place an Order
10) Pay for an Order
<<extend>>
<<extend>>
<<extend>>
<<include>>
<<include>>
<<include>>
<<include>>
<<include>>

Figure 26 Mobile application Use Case diagram

ID	UC1
Title: Description	View Home Page: The user views the Home page.
Actor	User
Preconditions	Launches the app.
Normal course	1a) The Home page listing all the available products comes into view.
Post-conditions	The user selects a product of the list of the available products to open the Product Details page.
Alternative course	1b) Opens the side menu. 2) Selects “Home” from the side menu. 3) The Home page listing all the available products comes into view.

ID	UC2
Title: Description	View General Info about the App: The user views general information about the app.
Actor	User
Preconditions	Launches the app.
Normal course	1) Opens the side menu. 2) Selects “About” from the side menu.
Post-conditions	The user taps on the “Read Registration details” button to open the Registration Details page.

ID	UC3
Title: Description	Read Registration Details about the Conference: The user reads the registration details about the conference.
Actor	User
Preconditions	Launches the app.
Normal course	1a) Opens the side menu. 2a) Selects “Registration Details” from the side menu.
Alternative course	b) UC2

ID	UC4
Title: Description	List Available Products by Category: The user views the available products by category.
Actor	User
Preconditions	Launches the app.
Normal course	1) Opens the side menu. 2) Selects “Abstract” or “Taxi” or “Ticket” from the side menu to list the products of the abstract or taxi or ticket category.
Post-conditions	The user selects a particular product to open the Products Details page.

ID	UC5
Title: Description	View Details of a Product: The user views the details and the reviews of a product.
Actor	User
Normal course	a) UC1
Alternative course	b) UC4

ID	UC6
Title: Description	Add Product to the Cart: The user adds a product to the shopping cart.
Actor	User
Preconditions	UC5
Normal course	The user taps on the “Add to Cart” button.

ID	UC7
Title: Description	Remove Product from the Cart: The user removes a product from the shopping cart.
Actor	User
Preconditions	UC6
Normal course	1) Taps on the cart button on the Product Details page to open the Cart page. 2) Taps on the red button on the right side of the product he wishes to remove from the shopping cart.

ID	UC8
Title: **Description**	Modify Quantity of the Products in the Cart: The user increases or decreases the quantity of the products in the shopping cart.
Actor	User
Preconditions	UC6
Normal course	1) Taps on the cart button on the Product Details page to open the Cart page. 2) Taps on the “plus” button to increase or on the “minus” button to decrease the quantity of a product in the shopping cart.

ID	UC9
Title: **Description**	Place an Order: The user proceeds to the Checkout and places an order.
Actor	User
Preconditions	UC6
Normal course	1) Taps on the cart button on the Product Details page to open the Cart page. 2) Taps on the “Checkout” button to open the Checkout page. 3a) Fills in his billing details, selects the payment method and taps on the “Place Order” button. 4a) The order is successfully placed and a confirmation message comes into view.
Exceptions	3b) The app notifies the user that the shopping cart is empty and the Checkout page is not loaded. 4b) The app notifies the user that he entered invalid input and the order is not placed.
Alternative course	2c) The user taps on the “Back” button on the Cart page and navigates back to the Product Details page. 3c) The user taps on “Back” button on the Checkout page and navigates back to the Cart page. The

ID	UC10
Title: Description	Pay for an Order: The user pays for the order placed via Direct Bank Transfer or via the PayPal payment gateway.
Actor	User
Preconditions	UC9
Normal course	1a) The PayPal payment gateway is launched. 2a1) Selects to pay with PayPal. 3a1) Enters his PayPal credentials and taps on the “Log In” button. 4a) Taps on the “Pay” button to complete the transaction.
Exceptions	The user enters invalid PayPal credentials or invalid credit card credentials.
Alternative course	1b) The user logs in to his bank account and completes the transaction by using the bank account details from the Registration Details page. 2a2) Selects to pay with credit card. 3a2) Enters his credit card details.

4 Conclusions

Certain improvements [56] have been made to the original website [27] regarding User Experience:

- ✓ The landing page provides a button that links to the significant task of abstract submission, whereas the landing page of the original website provides only a button linking to the registration page.
- ✓ The landing page highlights the important deadlines with flip boxes, whereas the landing page of the original website presents the important deadlines with text.
- ✓ The landing page, along with the posters of past events, contains buttons linking to important pages of the website, whereas the landing page of the original website contains only the posters of past events in the archives section.
- ✓ The landing page contains a visualization of the event's schedule in the schedule section, whereas the landing page of the original website contains only the link to download the schedule as a pdf file.
- ✓ The landing page presents the pricing plans for registering for the conference and reminds the visitors of booking their seats, whereas the landing page of the original website has no such section.
- ✓ The landing page contains a section in the footer area for the visitor to subscribe to newsletter, whereas the landing page of the original website does not provide such a capability.
- ✓ The landing page has buttons linking to the social media profiles of the conference, whereas the landing page of the original website does not have this feature.
- ✓ The landing page includes the different web pages in the "About" menu in the header area, whereas the landing page of the original website has a tab in the header area for each of the different web pages.
- ✓ The website integrates a Chabot that helps the visitor navigate through the website, whereas the original website does not have such a feature.

- ✓ The website contains hyperlinks to the different web pages in the footer area, whereas the original website does not.
- ✓ The website streamlines the order process by providing an e-commerce solution for listing and selecting products as well as for placing the order, whereas the original website has only a registration form for placing the order.
- ✓ The website offers the capability of online payments with PayPal and credit card, whereas the original website does not.
- ✓ The website asks the visitor to consent for cookie usage, whereas the original website does not.
- ✓ The website contains a web page with photographic material from previous events, whereas the original website does not.
- ✓ The website contains a web page through which answers to frequently asked questions are provided, whereas the original website does not.
- ✓ The website has two contact forms via which the visitor can contact and submit questions to the event organizers, whereas the original website does not.
- ✓ The website has a blog page containing insights into the event and allowing the interaction with the visitors, whereas the original website does not.

The mobile application interconnected with the website via the WordPress REST API and built with Ionic Framework 4 [57] is compatible with Android and iOS and facilitates the registration for the event. This means that the attendee can download the app from the Google Play Store or from the App Store, install it on his mobile device and place the order just as if he carried out the registration process on the website.

Proposals for future work include the conversion of the website into a Progressive Web Application. The website could also have a membership area, which the visitor is able to access after signing up for a personal account. Aside from the account details and the personal information of the member, the membership area could have a forum where the members would debate on topics related to the event. In addition, the mobile application could be expanded on other platforms and its content could grow to such an extent that all the website’s information and functionality are embedded in the application.

Bibliography

[1] N. Schäferhoff, "CMS Comparison: WordPress vs Joomla vs Drupal - WebsiteSetup", *websitesetup.org*, 2019. [Online]. Available: https://websitesetup.org/cms-comparison-wordpress-vs-joomla-drupal/

[2] "Joomla vs WordPress - Which One is Better? (Pros and Cons)", *Kinsta Managed WordPress Hosting*, 2019. [Online]. Available: https://kinsta.com/blog/joomla-vs-wordpress/

[3] "Getting Started With Cross-Platform App Development In 2019 - By", *Hackernoon.com*, 2019. [Online]. Available: https://hackernoon.com/getting-started-with-cross-platform-app-development-in-2019-dd2bf7f6161b

[4] "Free Cross-Platform Mobile App Development Tools Compared - 2018", *Outsystems.com*, 2019. [Online]. Available: https://www.outsystems.com/blog/free-cross-platform-mobile-app-development-tools-compared.html

[5] J. Kurose and K. Ross, *Computer networking*. Boston, Mass.: Pearson, 2017.

[6] 2019. [Online]. Available: https://www.cloudflare.com/learning

[7] "TLS vs. SSL" - 5 Things To Know (Differences, Protocols, & Handshakes) - HostingAdvice.com", *HostingAdvice.com*, 2019. [Online]. Available: https://www.hostingadvice.com/how-to/tls-vs-ssl/

[8] D. Baltatzis, "Information Systems Security Lectures", *International Hellenic University*, 2019.

[9] L. Sabin-Wilson, *WordPress all-in-one for dummies*, 4th ed. John Wiley & Sons, Inc, 2019.

[10] “The Ultimate WordPress Security Guide – Step by Step (2019)”, *Wpbeginner.com*, 2019. [Online]. Available: https://www.wpbeginner.com/wordpress-security/

[11] C. Moridis and A. Ampatzoglou, "Human-Computer Interaction, Design and User Experience", *International Hellenic University Lectures*, 2018.

[12] "10 Usability Heuristics for User Interfaces in Web Design", *Medium*, 2019. [Online]. Available: https://medium.com/@toddohanian/10-usability-heuristics-for-user-interfaces-in-web-design-c179aa39b54e

[13] P. Laja, "8 Effective Web Design Principles to Know in 2019 | CXL", *CXL*, 2019. [Online]. Available: https://conversionxl.com/blog/universal-web-design-principles

[14] E. Cocquebert, D. Trentesaux and C. Tahon, "WISDOM: A website design method based on reusing design and software solutions", *Information and Software Technology*, vol. 52, no. 12, pp. 1272-1285, 2010.

[15] S. Cebi, "Determining importance degrees of website design parameters based on interactions and types of websites", *Decision Support Systems*, vol. 54, no. 2, pp. 1030-1043, 2013.

[16] "10 Most Popular Types of Websites", *Webyurt.com*, 2019. [Online]. Available: http://www.webyurt.com/popular-types-of-websites

[17] A. Darem and Suresha, "The effect of website design and structure on disorientation and lostness in hyperspace", *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 6, 2013.

[18] M. Williams, N. Rana and Y. Dwivedi, "The unified theory of acceptance and use of technology (UTAUT): a literature review", *Journal of Enterprise Information Management*, 2014.

[19] K. Al-Qeisi, C. Dennis, E. Alamanos and C. Jayawardhena, "Website design quality and usage behavior: Unified Theory of Acceptance and Use of Technology", *Journal of Business Research*, vol. 67, no. 11, pp. 2282-2290, 2014. Available: 10.1016/j.jbusres.2014.06.016.

[20] B. Chua and L. Dyson, "Applying the ISO 9126 model to the evaluation of an e-learning system", 2004.

[21] P. Botella et al., "ISO/IEC 9126 in practice: what do we need to know?" .

[22] E. Seufert, *Freemium economics*. Amsterdam: Elsevier/Morgan Kaufmann, 2014.

[23] L. Metcalf and W. Casey, *Cybersecurity and Applied Mathematics*. Saint Louis: Elsevier Science, 2016.

[24] “Introduction to Search Engine Optimization Getting Started With SEO to Achieve Business Goals”, *Hubspot.com*, 2019. [Online]. Available: https://www.hubspot.com/

[25] A. Dennis, B. Wixom and R. Roth, *Systems analysis and design*.

[26] C. Tjortjis, "Software Development Methodologies Lectures", *International Hellenic University*, 2018.

[27] "CCS2019", *Ccs2019.org*, 2019. [Online]. Available: http://www.ccs2019.org/

[28] "Web Hosting Features - Web Hosting Functions - Bluehost", *Bluehost.com*, 2019. [Online]. Available: https://www.bluehost.com/hosting-features

[29] [5]"Server Hardware Requirements Guide - Atlassian Documentation", *Confluence.atlassian.com*, 2019. [Online]. Available: https://confluence.atlassian.com/doc/server-hardware-requirements-guide-30736403.html

[30] “Web server requirements (software) | Web-Site-Scripts.com", *Web-site-scripts.com*, 2019. [Online]. Available: http://www.web-site-scripts.com/knowledge-base/article/AA-00468/0/Web-server-requirements-software.html

[31] "Hosting Requirements for WordPress", *WordPress.org*, 2019. [Online]. Available: https://wordpress.org/about/requirements/

[32] "Different Types of Web Hosting Explained (Shared, VPS, Dedicated)", *HostingFacts.com*, 2019. [Online]. Available: https://hostingfacts.com/different-types-of-web-hosting/

[33] "Web Hosting & Domain 101: How Hosting a Website Works", *Web Hosting Secret Revealed*, 2019. [Online]. Available: https://www.webhostingsecretrevealed.net/web-hosting-beginner-guide/

[34] "Tidio Live Chat | Free Livechat Solution", *Tidio Live Chat*, 2019. [Online]. Available: https://www.tidio.com/

[35] "Website Sitemap | WebMistress.com", *Webmistress.com*, 2019. [Online]. Available: http://www.webmistress.com/web-design-development/web-sites/website-sitemap

[36] "What is an XML sitemap and why should you have one? • Yoast", *Yoast*, 2019. [Online]. Available: https://yoast.com/what-is-an-xml-sitemap-and-why-should-you-have-one/

[37] L. Kollár and N. Sterbinszky, "Case study in system development - Notes", 2014.

[38] *Lucidchart.com*, 2019. [Online]. Available: https://www.lucidchart.com

[39] "Creately - Your Documents", *App.creately.com*, 2019. [Online]. Available: https://app.creately.com

[40] R. Talada, A. Matos, R. Verma, D. D and S. Masne, "What is REST – Learn to create timeless REST APIs", *Restfulapi.net*, 2019. [Online]. Available: https://restfulapi.net

[41] "JSON", *Json.org*, 2019. [Online]. Available: https://www.json.org/

[42] "WordPress Developer Resources", *Developer.wordpress.org*, 2019. [Online]. Available: https://developer.wordpress.org/rest-api/

[43] "WooCommerce REST API Documentation", *Woocommerce.github.io*, 2019. [Online]. Available: https://woocommerce.github.io/woocommerce-rest-api-docs

[44] O. Alam, "How to Setup and Use WordPress REST API: Basic Authentication", *The Official Cloudways Blog*, 2019. [Online]. Available: https://www.cloudways.com/blog/setup-basic-authentication-in-wordpress-rest-api/

[45] "Complete Guide to Cross-Origin Resource Sharing (CORS) - KeyCDN Support", *KeyCDN*, 2019. [Online]. Available: https://www.keycdn.com/support/cors

[46] J. Usmani, "How to enable Cross-site XMLHttpRequests (CORS) using .htaccess", *iPragmatech*, 2019. [Online]. Available: https://www.ipragmatech.com/enable-cors-using-htaccess/

[47] C. Griffith, *Mobile App Development with Ionic, Revised Edition Cross-Platform Apps with Ionic, Angular, and Cordova*. O'Reilly, 2017

[48] "Ionic - Cross-Platform Mobile App Development", *Ionic Framework*, 2019. [Online]. Available: https://ionicframework.com

[49] "My real clip domain - a lot of cliparts", *Myrealdomain.com*, 2019. [Online]. Available: https://myrealdomain.com/

[50] "Node.js tutorial", *Tutorialspoint.com*, 2019. [Online]. Available: https://www.tutorialspoint.com/nodejs/nodejs_tutorial.pdf

[51] "Angular", *Angular.io*, 2019. [Online]. Available: https://angular.io/

[52] "Sass: Syntactically Awesome Style Sheets", *Sass-lang.com*, 2019. [Online]. Available: https://sass-lang.com/

[53] "Why an event website is necessary?” *Medium*, 2019.

[Online]. Available: https://medium.com/hubilo-officil-blog/why-an-event-website-is-necessary-7ba123bf51c0

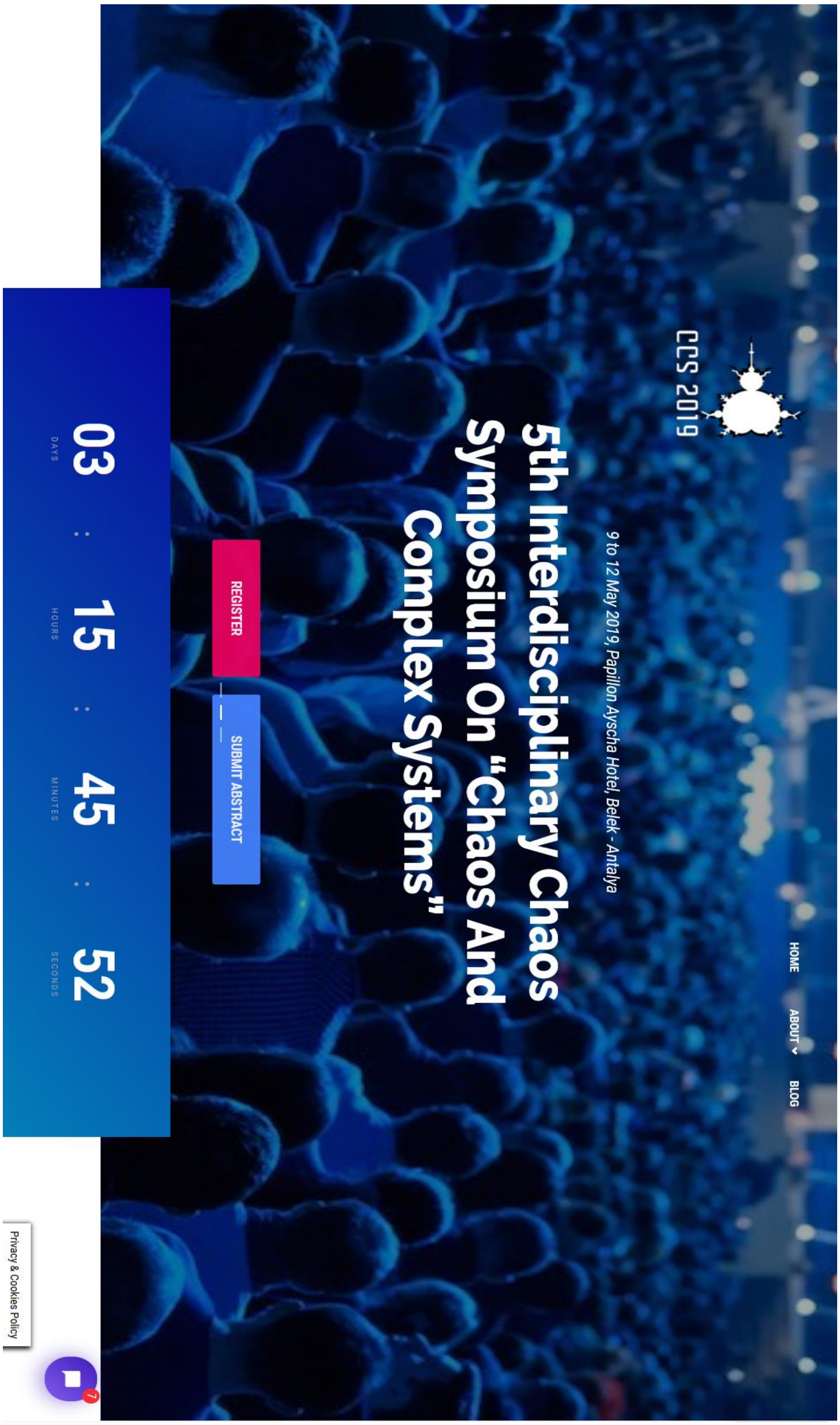
[54] B. Campbell, P. Casserly and N. Pascual, "14 Critical Elements to Include on Your Conference Website", *Blog | Ex Ordo*, 2019. [Online]. Available: https://www.exordo.com/blog/14-elements-include-conference-website/

[55] "Web Development Trends 2019 - KeyCDN", *KeyCDN*, 2019. [Online]. Available: https://www.keycdn.com/blog/web-development-trends-2019

[56] Demonstration of the conference website [Online]. Available: https://conference1.geowebsite.net/

[57] Demonstration of the Ionic Framework application for registering for the conference [Online]. Available: https://www.youtube.com/watch?v=7HATWzcBPQ4

Appendix A

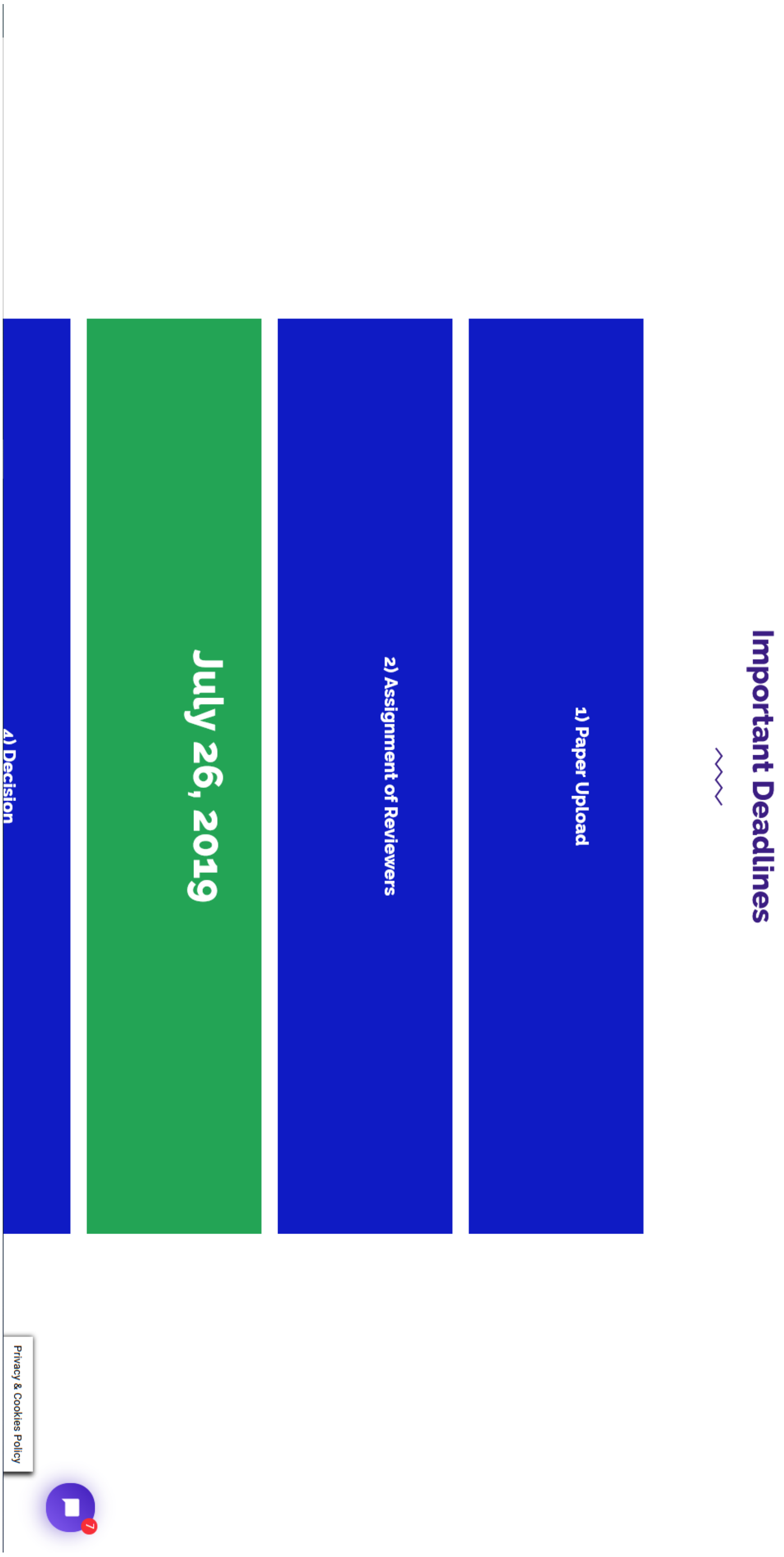


CCS 2019

9 to 12 May 2019, Papillon Ayscha Hotel, Belek - Antalya

5th Interdisciplinary Chaos Symposium On "Chaos And Complex Systems"

Figure 27 Website Home Page 1/8



Important Deadlines

1) Paper Upload

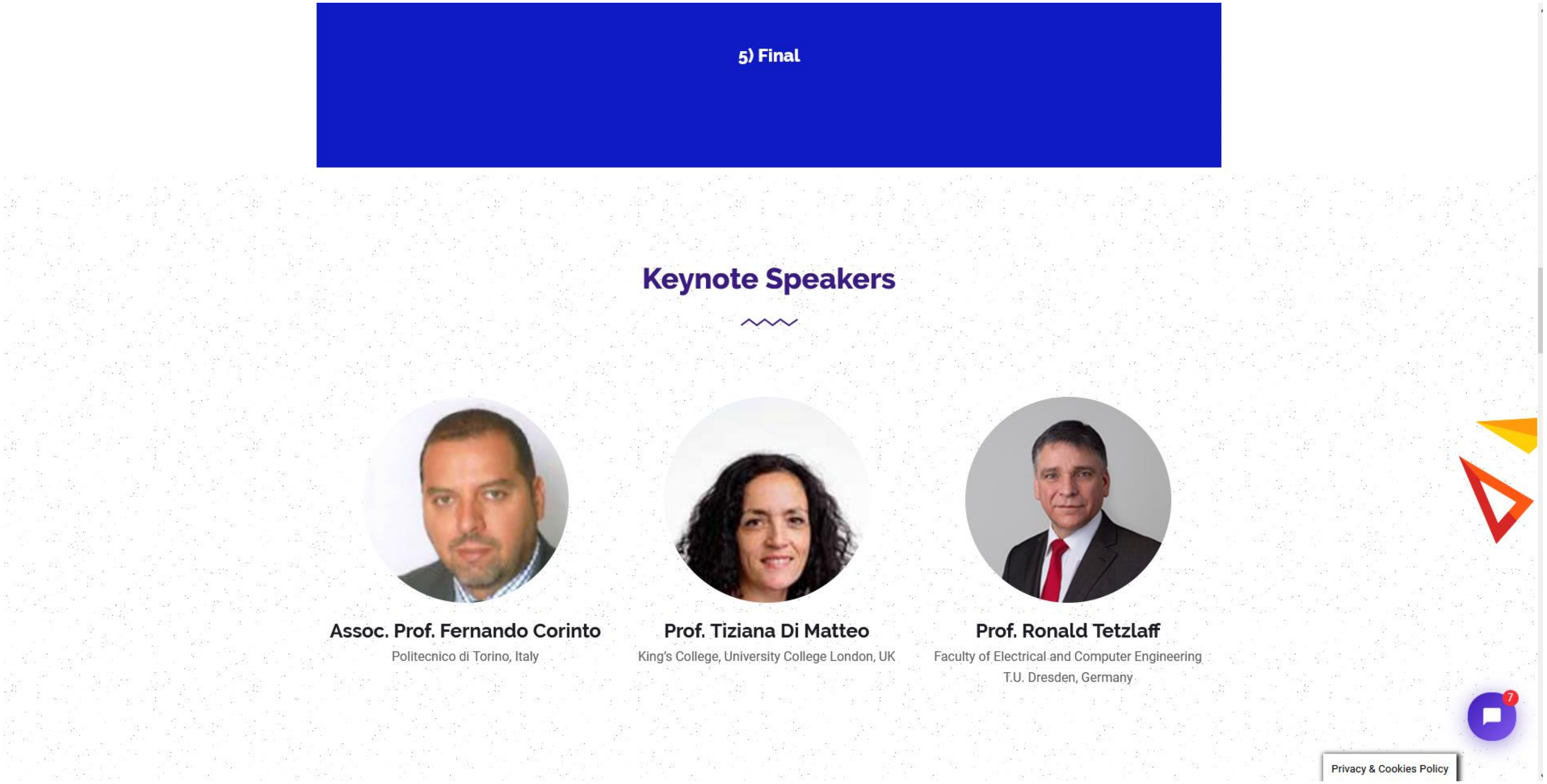
2) Assignment of Reviewers

July 26, 2019

4) Decision

Privacy & Cookies Policy

Figure 28 Website Home Page 2/8



5) Final

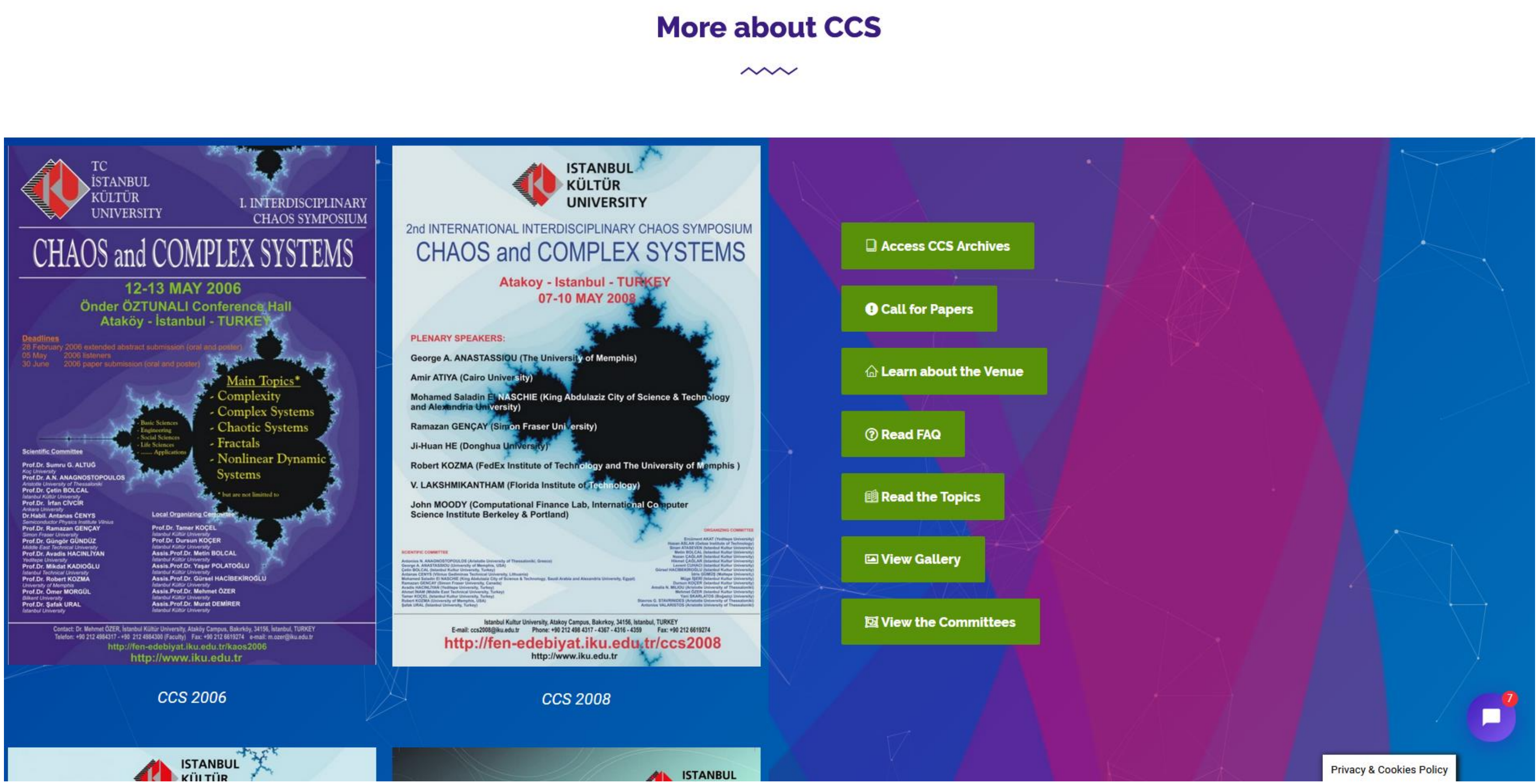
Keynote Speakers

Assoc. Prof. Fernando Corinto
Politecnico di Torino, Italy

Prof. Tiziana Di Matteo
King's College, University College London, UK

Prof. Ronald Tetzlaff
Faculty of Electrical and Computer Engineering
T.U. Dresden, Germany

Figure 29 Website Home Page 3/8



More about CCS

TC
İSTANBUL
KÜLTÜR
UNIVERSITY

I. INTERDISCIPLINARY
CHAOS SYMPOSIUM

CHAOS and COMPLEX SYSTEMS

12-13 MAY 2006
Önder ÖZTUNALI Conference Hall
Ataköy - İstanbul - TURKEY

Deadlines
28 February 2006 extended abstract submission (oral and poster)
05 May 2006 listeners
30 June 2006 paper submission (oral and poster)

Main Topics*
- Complexity
- Complex Systems
- Chaotic Systems
- Fractals
- Nonlinear Dynamic Systems

- Basic Sciences
- Engineering
- Social Sciences
- Life Sciences
- .... Applications

* but are not limitted to

Scientific Committee
Prof.Dr. Sumru G. ALTUĞ
Koç University
Prof.Dr. A.N. ANAGNOSTOPOULOS
Aristotle University of Thessaloniki
Prof.Dr. Çetin BOLCAL
İstanbul Kültür University
Prof.Dr. İrfan CİVCİR
Ankara University
Dr.Habil. Antanas ČENYS
Semiconductor Physics Institute Vilnius
Prof.Dr. Ramazan GENÇAY
Simon Fraser University
Prof.Dr. Güngör GÜNDÜZ
Middle East Technical University
Prof.Dr. Avadis HACINLİYAN
Yeditepe University
Prof.Dr. Mikdat KADIOĞLU
İstanbul Technical University
Prof.Dr. Robert KOZMA
University of Memphis
Prof.Dr. Ömer MORGÜL
Bilkent University
Prof.Dr. Şafak URAL
İstanbul University

Local Organizing Committee
Prof.Dr. Tamer KOÇEL
İstanbul Kültür University
Prof.Dr. Dursun KOÇER
İstanbul Kültür University
Assis.Prof.Dr. Metin BOLCAL
İstanbul Kültür University
Assis.Prof.Dr. Yaşar POLATOĞLU
İstanbul Kültür University
Assis.Prof.Dr. Gürsel HACİBEKİROĞLU
İstanbul Kültür University
Assis.Prof.Dr. Mehmet ÖZER
İstanbul Kültür University
Assis.Prof.Dr. Murat DEMİRER
İstanbul Kültür University

Contact: Dr. Mehmet ÖZER, İstanbul Kültür University, Ataköy Campus, Bakırköy, 34156, İstanbul, TURKEY
Telefon: +90 212 4984317 - +90 212 4984300 (Faculty) Fax: +90 212 6619274 e-mail: m.ozer@iku.edu.tr
http://fen-edebiyat.iku.edu.tr/kaos2006
http://www.iku.edu.tr

CCS 2006

ISTANBUL
KÜLTÜR
UNIVERSITY

2nd INTERNATIONAL INTERDISCIPLINARY CHAOS SYMPOSIUM
CHAOS and COMPLEX SYSTEMS

Atakoy - Istanbul - TURKEY
07-10 MAY 2008

PLENARY SPEAKERS:

George A. ANASTASSIOU (The University of Memphis)

Amir ATIYA (Cairo University)

Mohamed Saladin El NASCHIE (King Abdulaziz City of Science & Technology and Alexandria University)

Ramazan GENÇAY (Simon Fraser University)

Ji-Huan HE (Donghua University)

Robert KOZMA (FedEx Institute of Technology and The University of Memphis)

V. LAKSHMIKANTHAM (Florida Institute of Technology)

John MOODY (Computational Finance Lab, International Computer Science Institute Berkeley & Portland)

SCIENTIFIC COMMITTEE
Antonios N. ANAGNOSTOPOULOS (Aristotle University of Thessaloniki, Greece)
George A. ANASTASSIOU (University of Memphis, USA)
Cetin BOLCAL (Istanbul Kultur University, Turkey)
Antanas CENYS (Vilnius Gediminas Technical University, Lithuania)
Mohamed Saladin El NASCHIE (King Abdulaziz City of Science & Technology, Saudi Arabia and Alexandria University, Egypt)
Ramazan GENCAY (Simon Fraser University, Canada)
Avadis HACINLIYAN (Yeditepe University, Turkey)
Ahmet INAM (Middle East Technical University, Turkey)
Tamer KOÇEL (Istanbul Kultur University, Turkey)
Robert KOZMA (University of Memphis, USA)
Şafak URAL (Istanbul University, Turkey)

ORGANIZING COMMITTEE
Ercüment AKAT (Yeditepe University)
Hasan ASLAN (Gebze Institute of Technology)
Sinan ATASEVEN (Istanbul Kultur University)
Metin BOLCAL (Istanbul Kultur University)
Nazan ÇAĞLAR (Istanbul Kultur University)
Hikmet ÇAĞLAR (Istanbul Kultur University)
Levent CUHACI (Istanbul Kultur University)
Gürsel HACIBEKIROĞLU (Istanbul Kultur University)
Idris GÜMÜŞ (Maltepe University)
Müge IŞERI (Istanbul Kultur University)
Dursun KOÇER (Istanbul Kultur University)
Amalia N. MILIOU (Aristotle University of Thessaloniki)
Mehmet ÖZER (Istanbul Kultur University)
Yani SKARLATOS (Boğaziçi University)
Stavros G. STAVRINIDES (Aristotle University of Thessaloniki)
Antonios VALARISTOS (Aristotle University of Thessaloniki)

Istanbul Kultur University, Atakoy Campus, Bakırkoy, 34156, Istanbul, TURKEY
E-mail: ccs2008@iku.edu.tr Phone: +90 212 498 4317 - 4367 - 4316 - 4359 Fax: +90 212 6619274
http://fen-edebiyat.iku.edu.tr/ccs2008
http://www.iku.edu.tr

CCS 2008

ISTANBUL
KÜLTÜR

ISTANBUL

Figure 30 Website Home Page 4/8

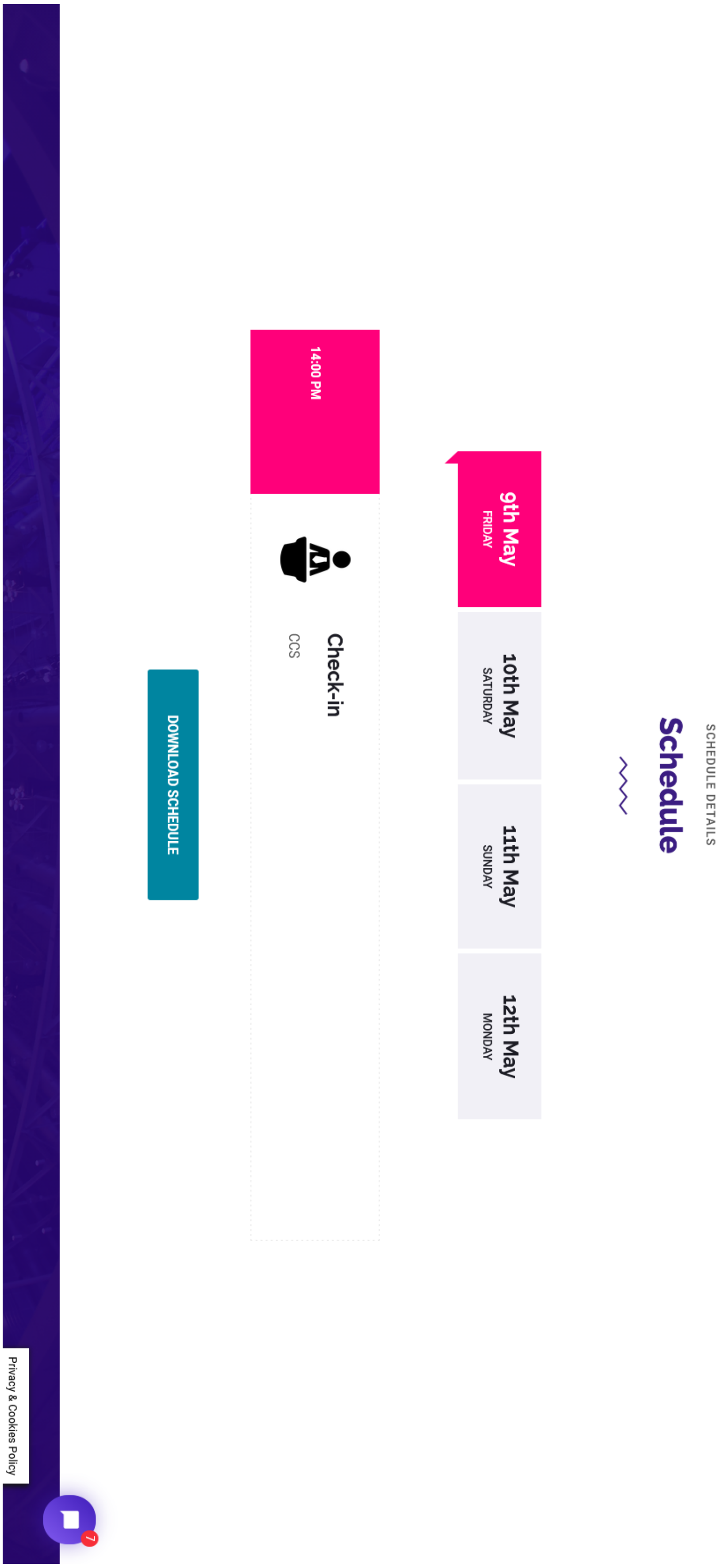
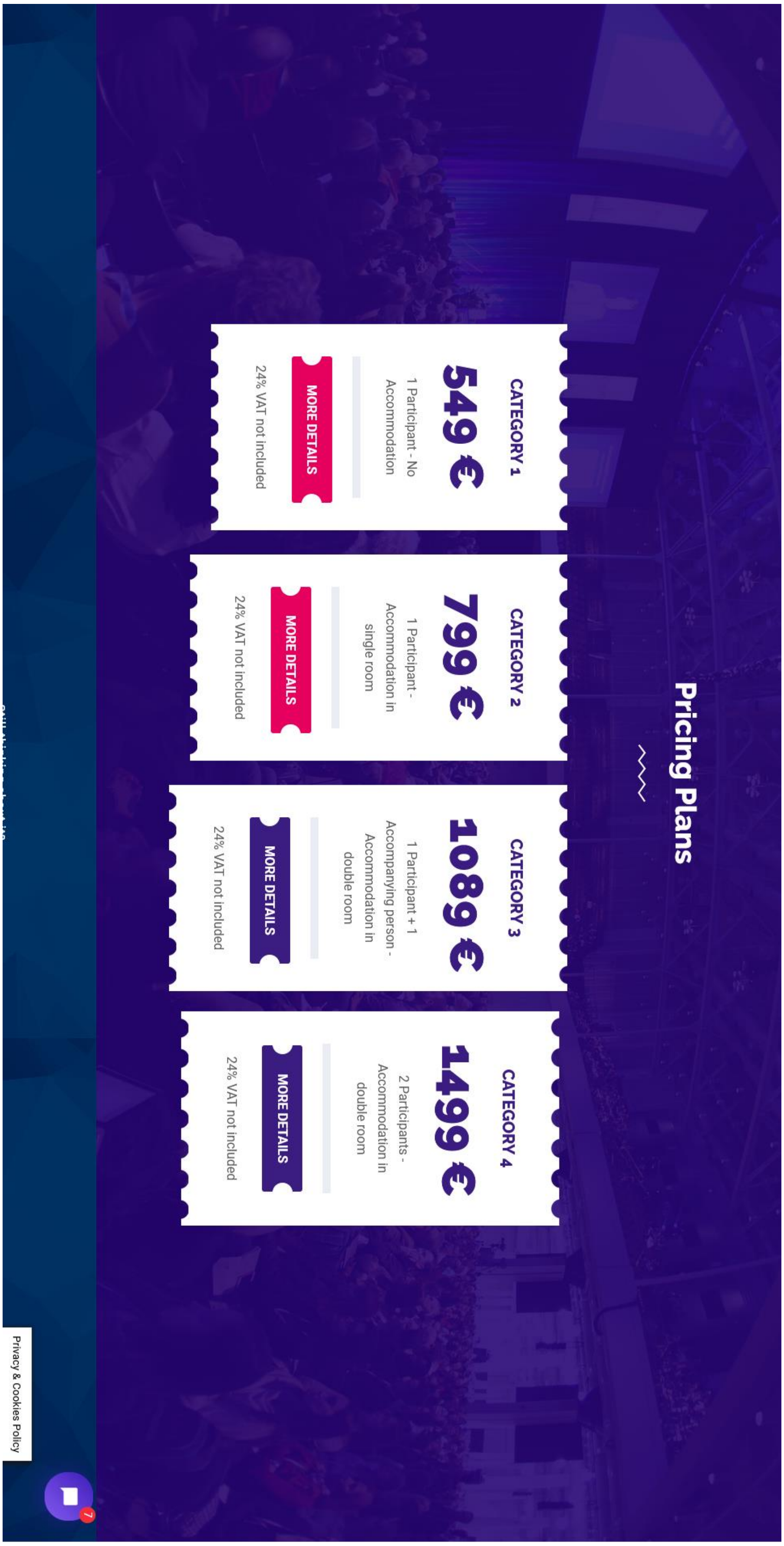


Figure 31 Website Home Page 5/8



Pricing Plans

CATEGORY 1

549 €

1 Participant - No Accommodation

24% VAT not included

CATEGORY 2

799 €

1 Participant - Accommodation in single room

24% VAT not included

CATEGORY 3

1089 €

1 Participant + 1 Accompanying person - Accommodation in double room

24% VAT not included

CATEGORY 4

1499 €

2 Participants - Accommodation in double room

24% VAT not included

Figure 32 Website Home Page 6/8

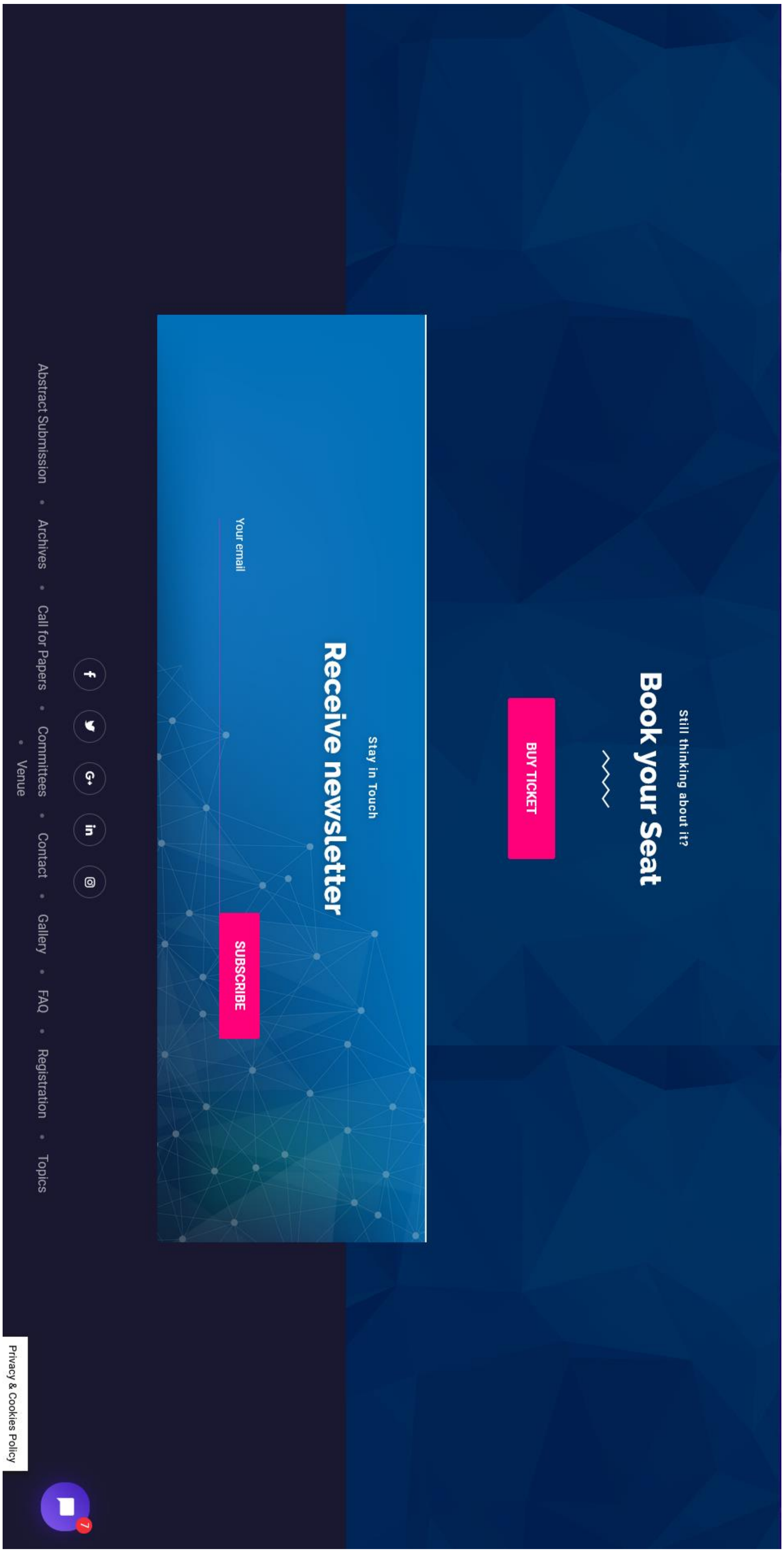
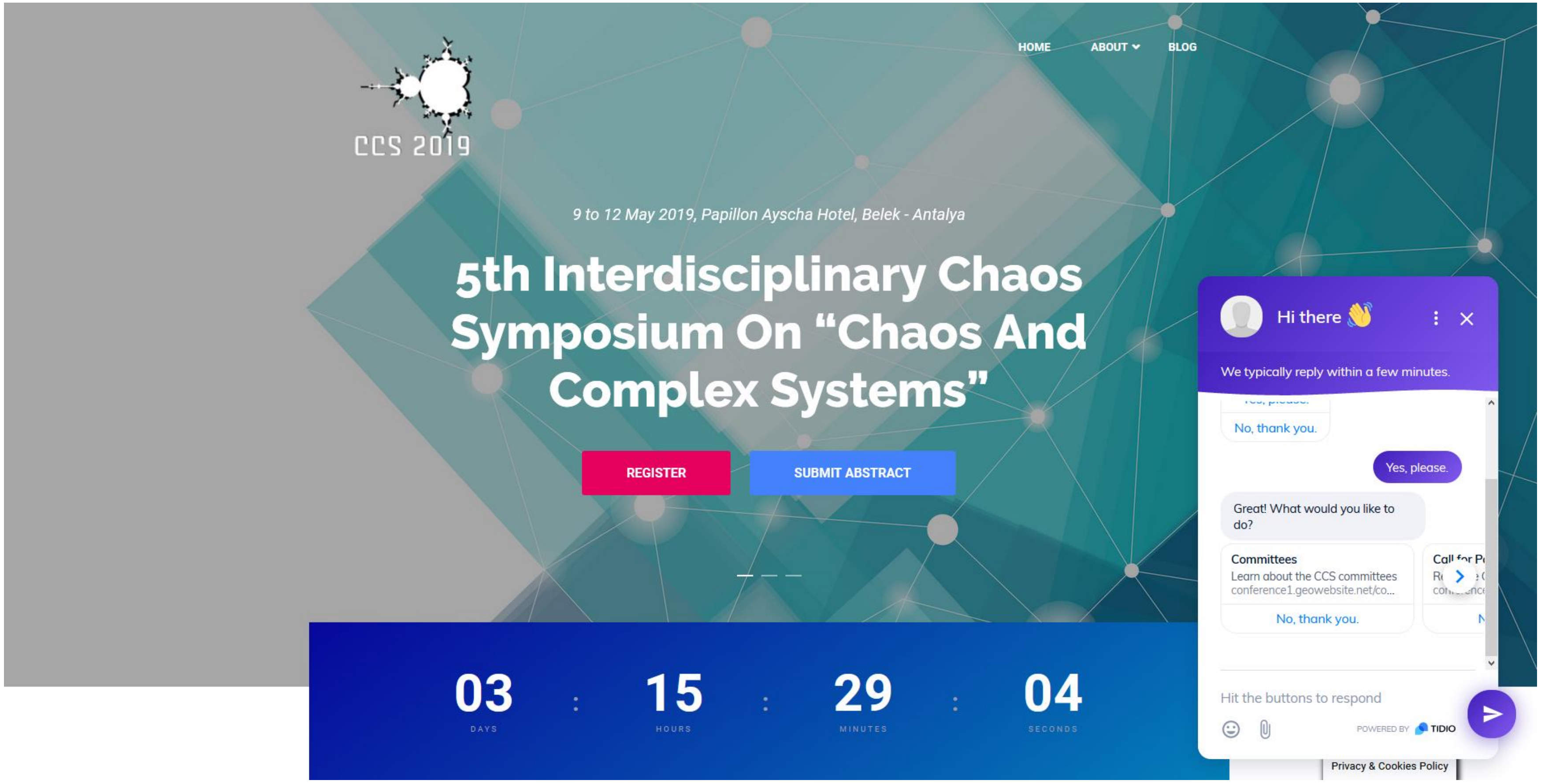


Figure 33 Website Home Page 7/8



CCS 2019

HOME ABOUT BLOG

9 to 12 May 2019, Papillon Ayscha Hotel, Belek - Antalya

5th Interdisciplinary Chaos Symposium On “Chaos And Complex Systems”

REGISTER SUBMIT ABSTRACT

03 DAYS : 15 HOURS : 29 MINUTES : 04 SECONDS

Hi there 👋

We typically reply within a few minutes.

No, thank you.

Yes, please.

Great! What would you like to do?

Committees
Learn about the CCS committees
conference1.geowebsite.net/co...

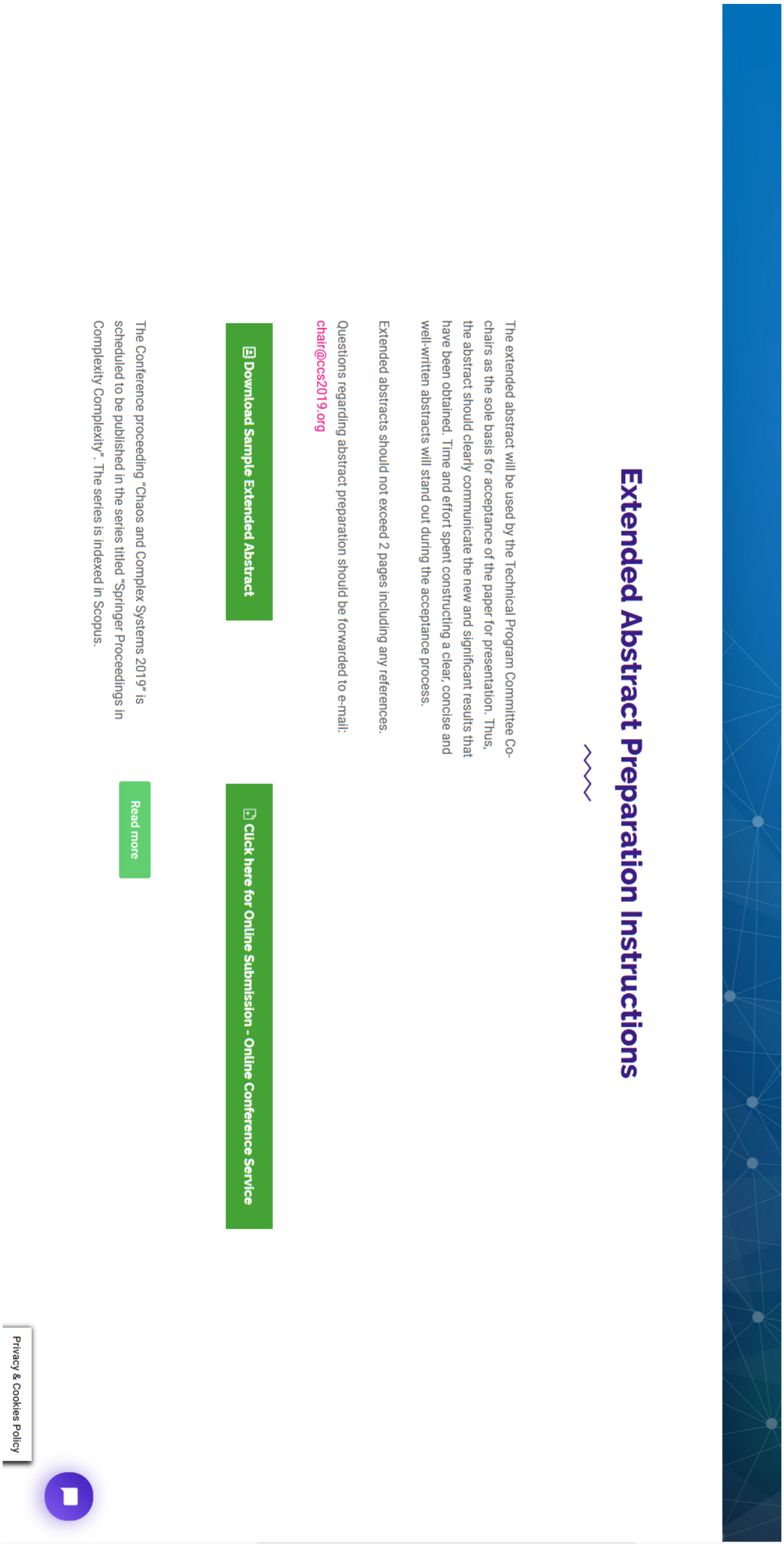
No, thank you.

Hit the buttons to respond

POWERED BY TIDIO

Privacy & Cookies Policy

Figure 34 Website Home Page 8/8



Extended Abstract Preparation Instructions

The extended abstract will be used by the Technical Program Committee Co-chairs as the sole basis for acceptance of the paper for presentation. Thus, the abstract should clearly communicate the new and significant results that have been obtained. Time and effort spent constructing a clear, concise and well-written abstracts will stand out during the acceptance process.

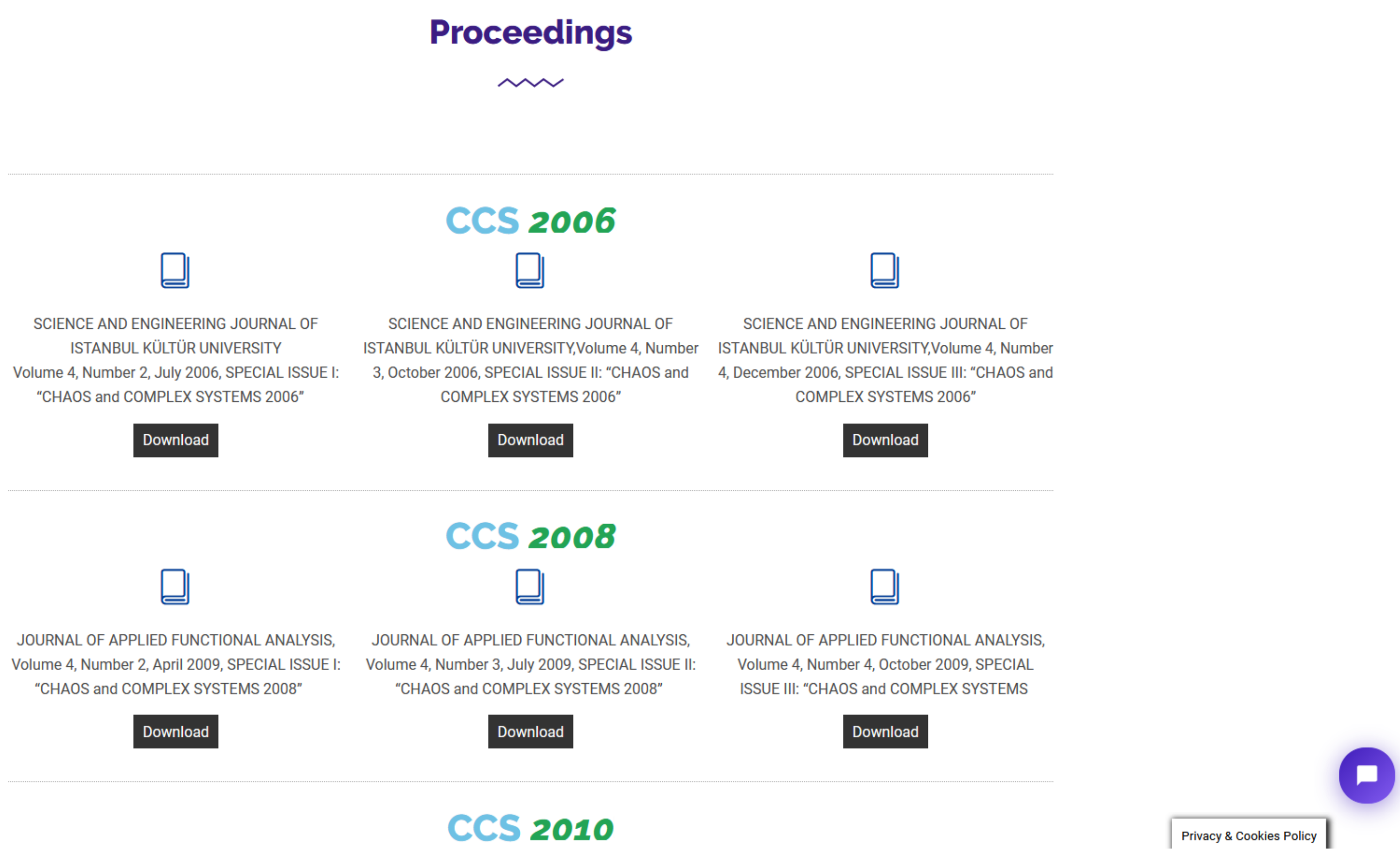
Extended abstracts should not exceed 2 pages including any references.

Questions regarding abstract preparation should be forwarded to e-mail: chair@ccs2019.org

Download Sample Extended Abstract
Click here for Online Submission - Online Conference Service

The Conference proceeding "Chaos and Complex Systems 2019" is scheduled to be published in the series titled "Springer Proceedings in Complexity Complexity". The series is indexed in Scopus.

Figure 35 Website Abstract Submission page



Proceedings

CCS 2006

SCIENCE AND ENGINEERING JOURNAL OF ISTANBUL KÜLTÜR UNIVERSITY
Volume 4, Number 2, July 2006, SPECIAL ISSUE I: "CHAOS and COMPLEX SYSTEMS 2006"

Download

SCIENCE AND ENGINEERING JOURNAL OF ISTANBUL KÜLTÜR UNIVERSITY,Volume 4, Number 3, October 2006, SPECIAL ISSUE II: "CHAOS and COMPLEX SYSTEMS 2006"

Download

SCIENCE AND ENGINEERING JOURNAL OF ISTANBUL KÜLTÜR UNIVERSITY,Volume 4, Number 4, December 2006, SPECIAL ISSUE III: "CHAOS and COMPLEX SYSTEMS 2006"

Download

CCS 2008

JOURNAL OF APPLIED FUNCTIONAL ANALYSIS, Volume 4, Number 2, April 2009, SPECIAL ISSUE I: "CHAOS and COMPLEX SYSTEMS 2008"

Download

JOURNAL OF APPLIED FUNCTIONAL ANALYSIS, Volume 4, Number 3, July 2009, SPECIAL ISSUE II: "CHAOS and COMPLEX SYSTEMS 2008"

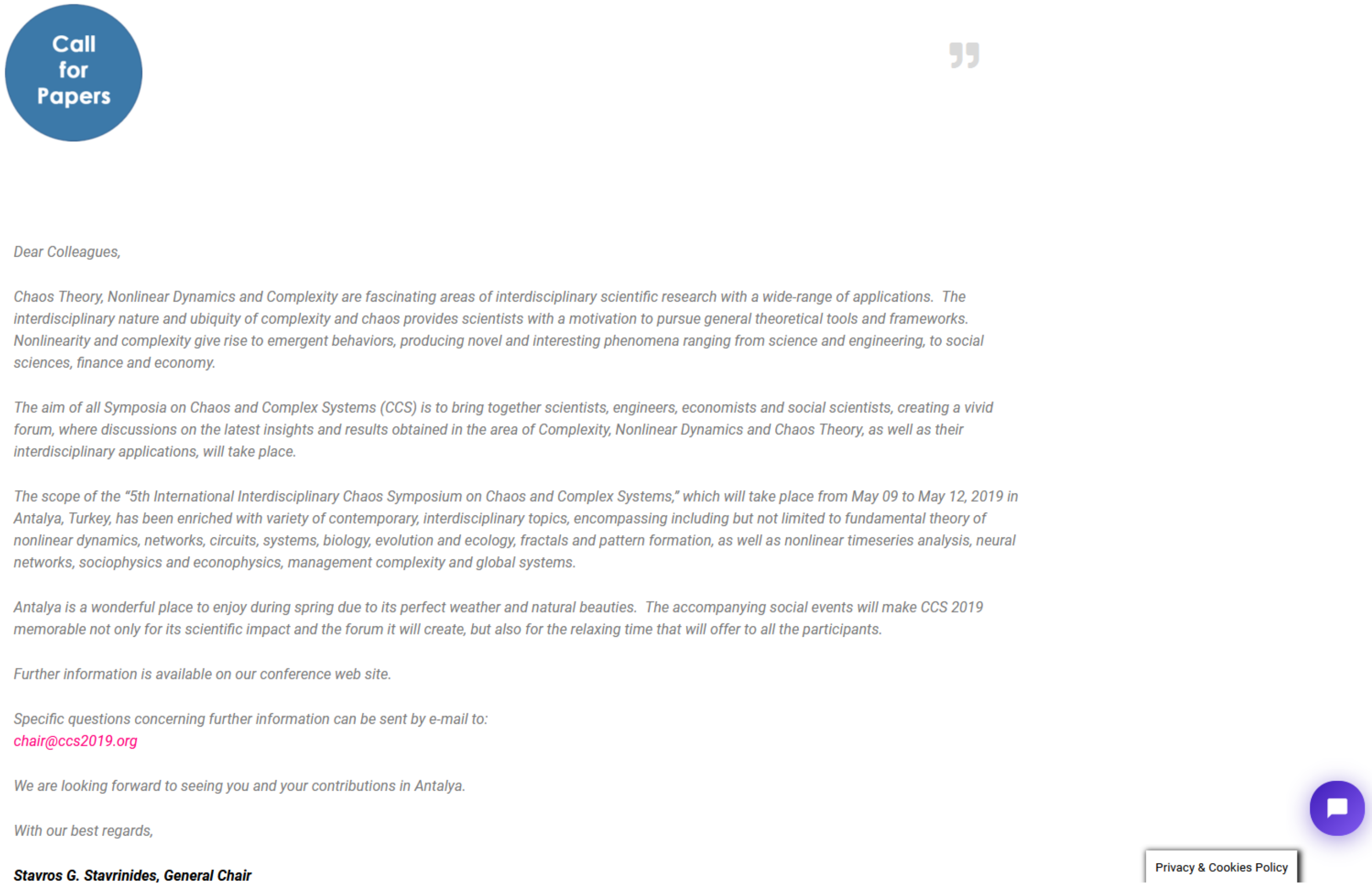
Download

JOURNAL OF APPLIED FUNCTIONAL ANALYSIS, Volume 4, Number 4, October 2009, SPECIAL ISSUE III: "CHAOS and COMPLEX SYSTEMS

Download

CCS 2010

Figure 36 Website Archives page



Dear Colleagues,

Chaos Theory, Nonlinear Dynamics and Complexity are fascinating areas of interdisciplinary scientific research with a wide-range of applications. The interdisciplinary nature and ubiquity of complexity and chaos provides scientists with a motivation to pursue general theoretical tools and frameworks. Nonlinearity and complexity give rise to emergent behaviors, producing novel and interesting phenomena ranging from science and engineering, to social sciences, finance and economy.

The aim of all Symposia on Chaos and Complex Systems (CCS) is to bring together scientists, engineers, economists and social scientists, creating a vivid forum, where discussions on the latest insights and results obtained in the area of Complexity, Nonlinear Dynamics and Chaos Theory, as well as their interdisciplinary applications, will take place.

The scope of the "5th International Interdisciplinary Chaos Symposium on Chaos and Complex Systems," which will take place from May 09 to May 12, 2019 in Antalya, Turkey, has been enriched with variety of contemporary, interdisciplinary topics, encompassing including but not limited to fundamental theory of nonlinear dynamics, networks, circuits, systems, biology, evolution and ecology, fractals and pattern formation, as well as nonlinear timeseries analysis, neural networks, sociophysics and econophysics, management complexity and global systems.

Antalya is a wonderful place to enjoy during spring due to its perfect weather and natural beauties. The accompanying social events will make CCS 2019 memorable not only for its scientific impact and the forum it will create, but also for the relaxing time that will offer to all the participants.

Further information is available on our conference web site.

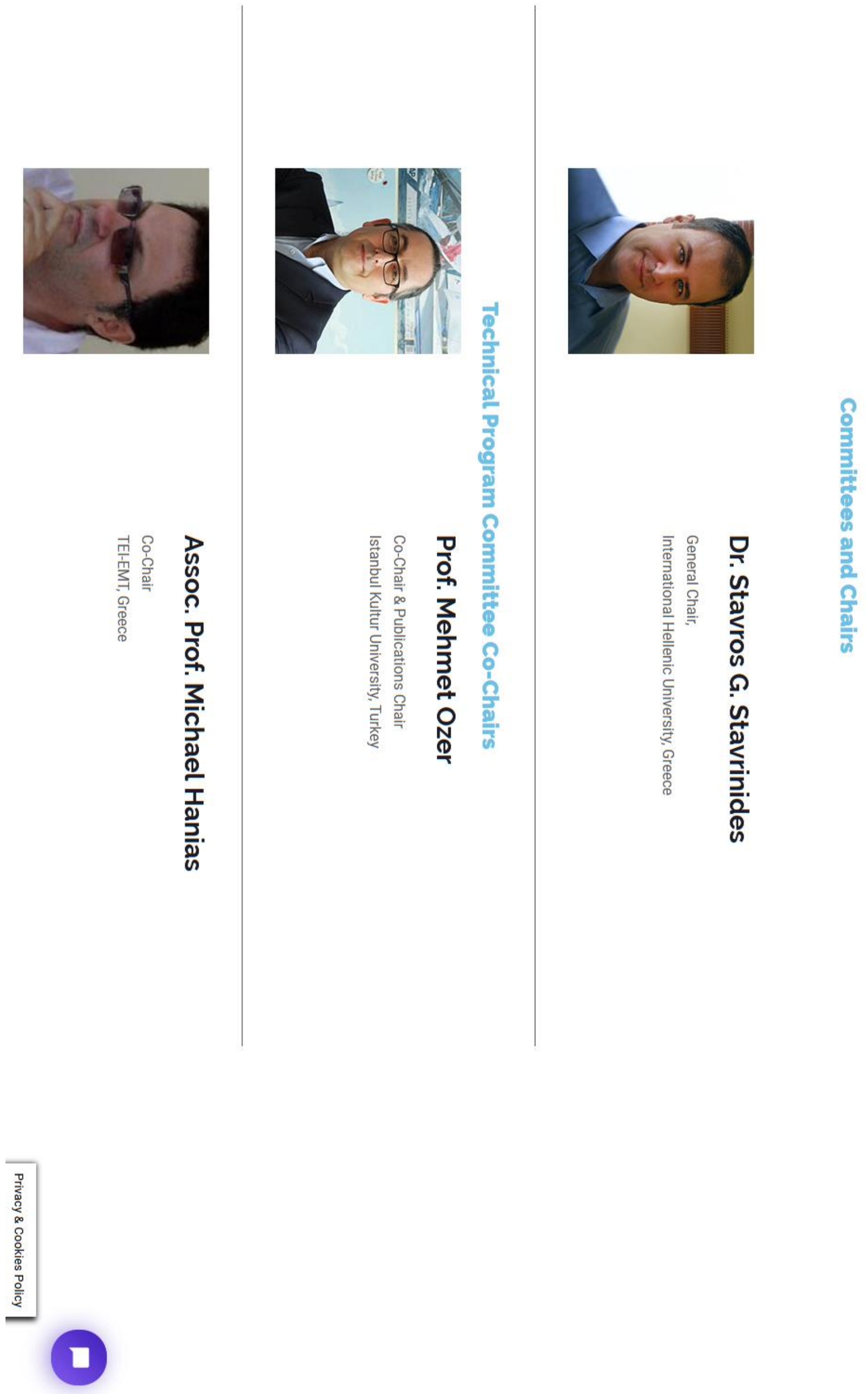
Specific questions concerning further information can be sent by e-mail to:
chair@ccs2019.org

We are looking forward to seeing you and your contributions in Antalya.

With our best regards,

Stavros G. Stavrinides, General Chair

Figure 37 Website Call for Papers page



Committees and Chairs

Dr. Stavros G. Stavrinides

General Chair,

International Hellenic University, Greece

Technical Program Committee Co-Chairs

Prof. Mehmet Ozer

Co-Chair & Publications Chair

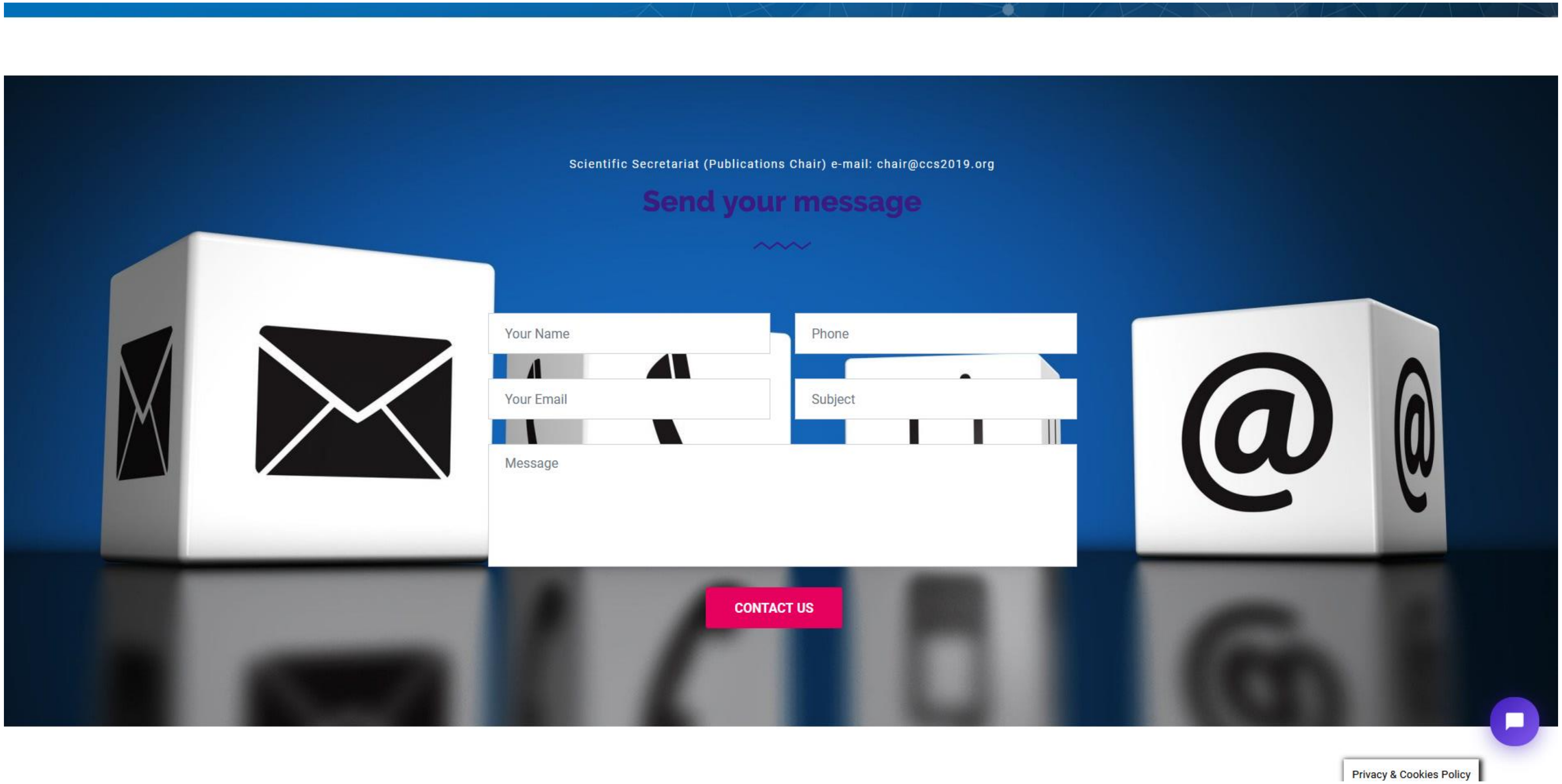
Istanbul Kultur University, Turkey

Assoc. Prof. Michael Hanias

Co-Chair

TEI-EMT, Greece

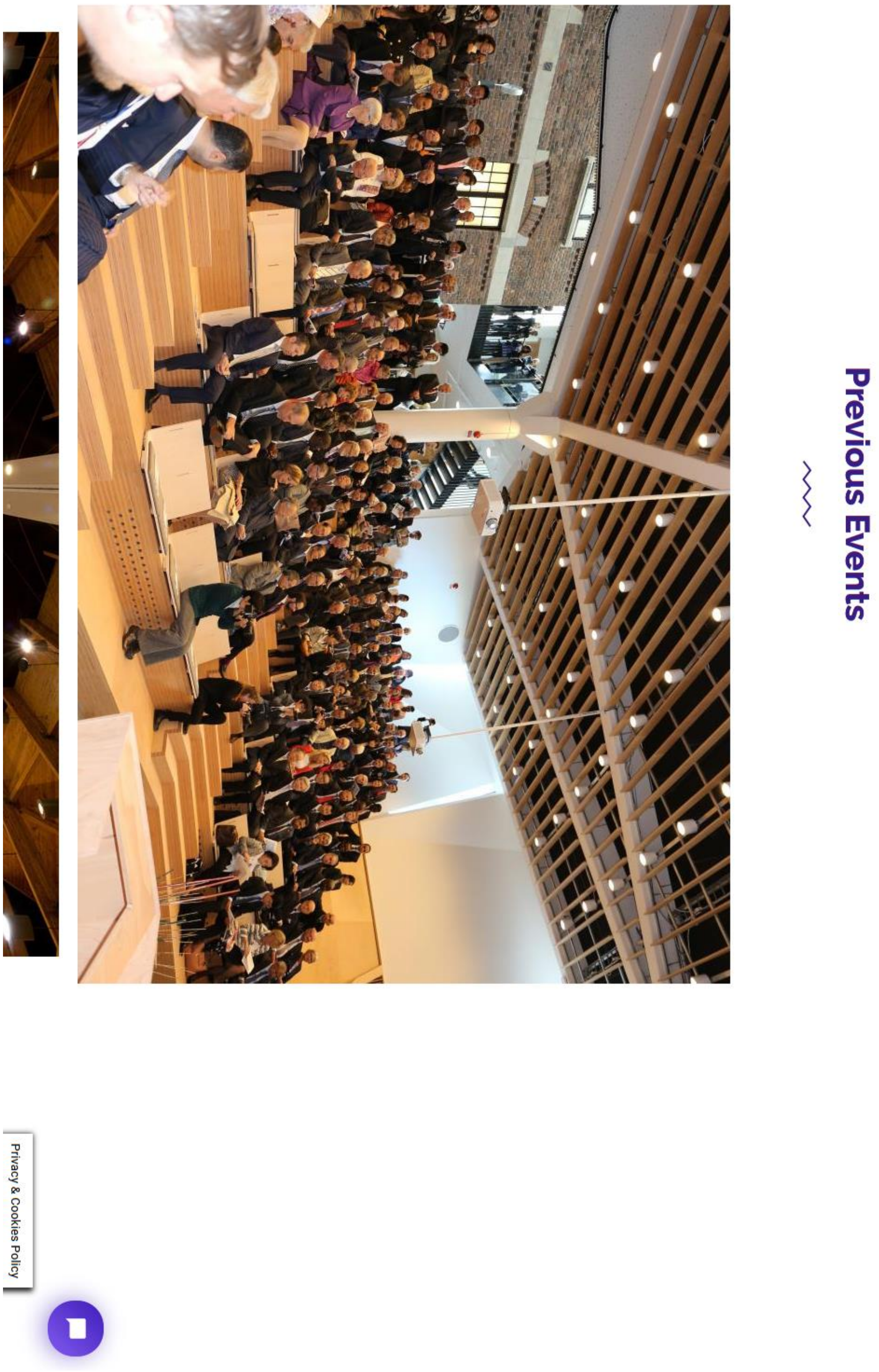
Figure 38 Website Committees page



Scientific Secretariat (Publications Chair) e-mail: chair@ccs2019.org

Send your message

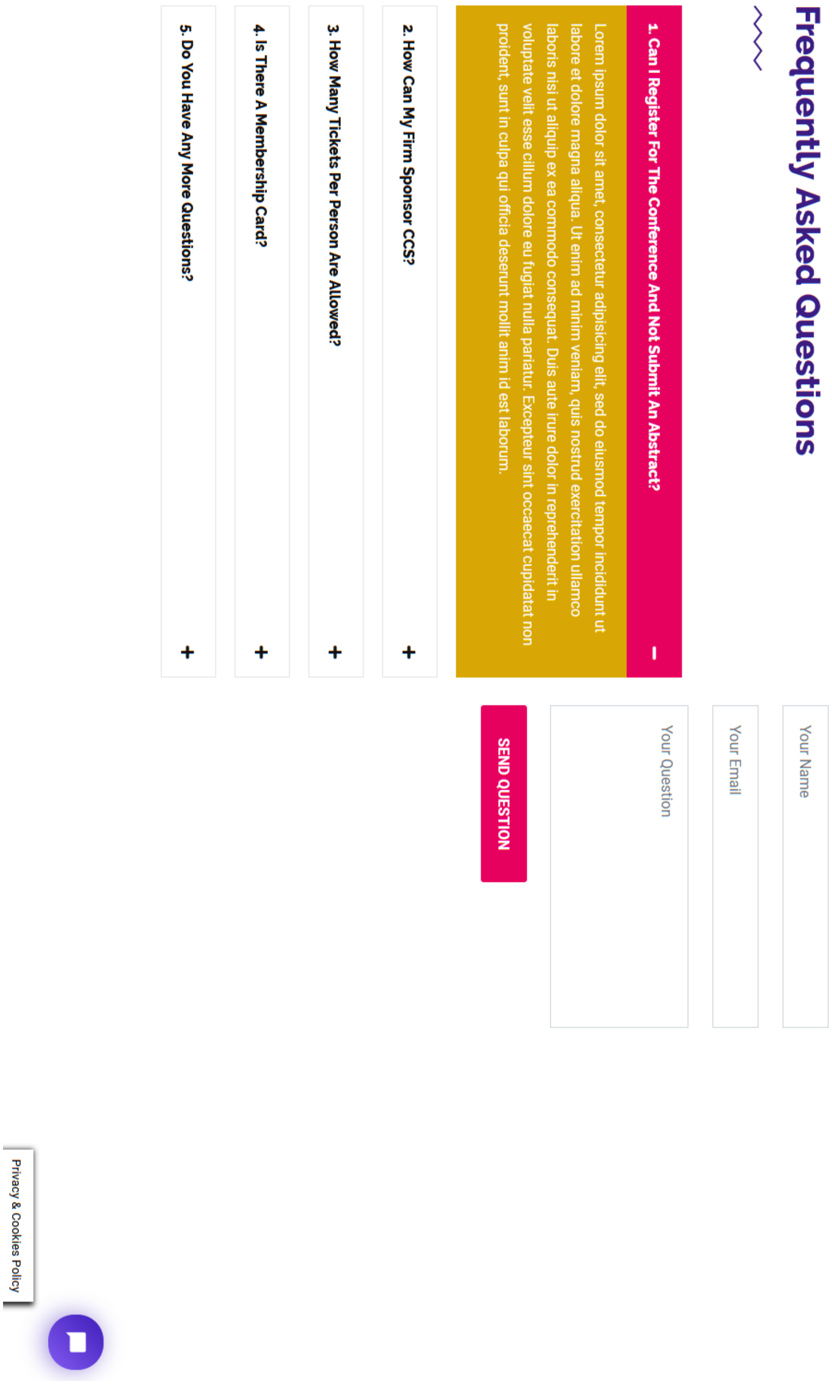
Figure 39 Website Contact page



Previous Events

Privacy & Cookies Policy

Figure 40 Website Gallery page



Frequently Asked Questions

1. Can I Register For The Conference And Not Submit An Abstract?

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

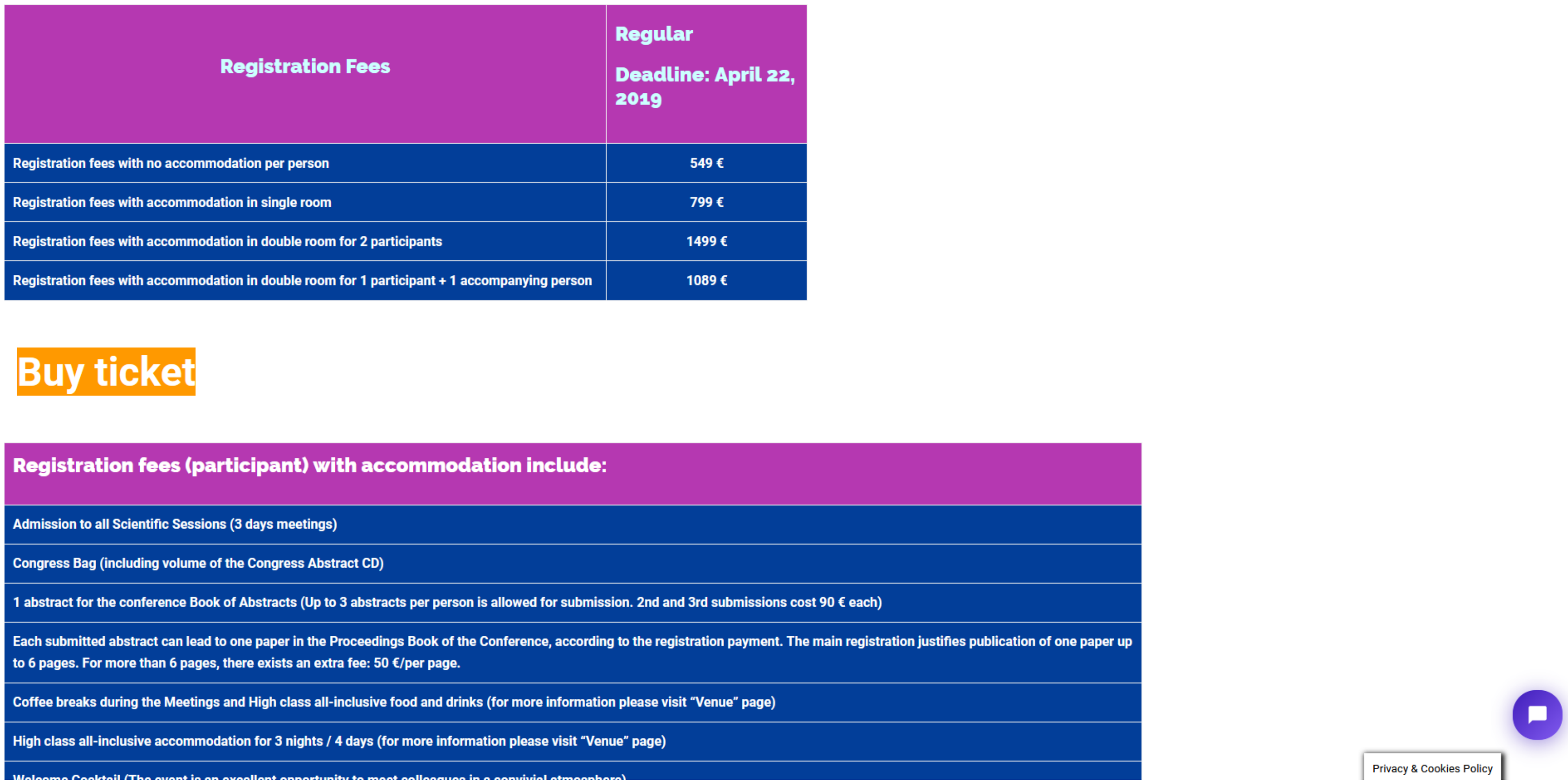
2. How Can My Firm Sponsor CCS?

3. How Many Tickets Per Person Are Allowed?

4. Is There A Membership Card?

5. Do You Have Any More Questions?

Figure 41 Website FAQ page

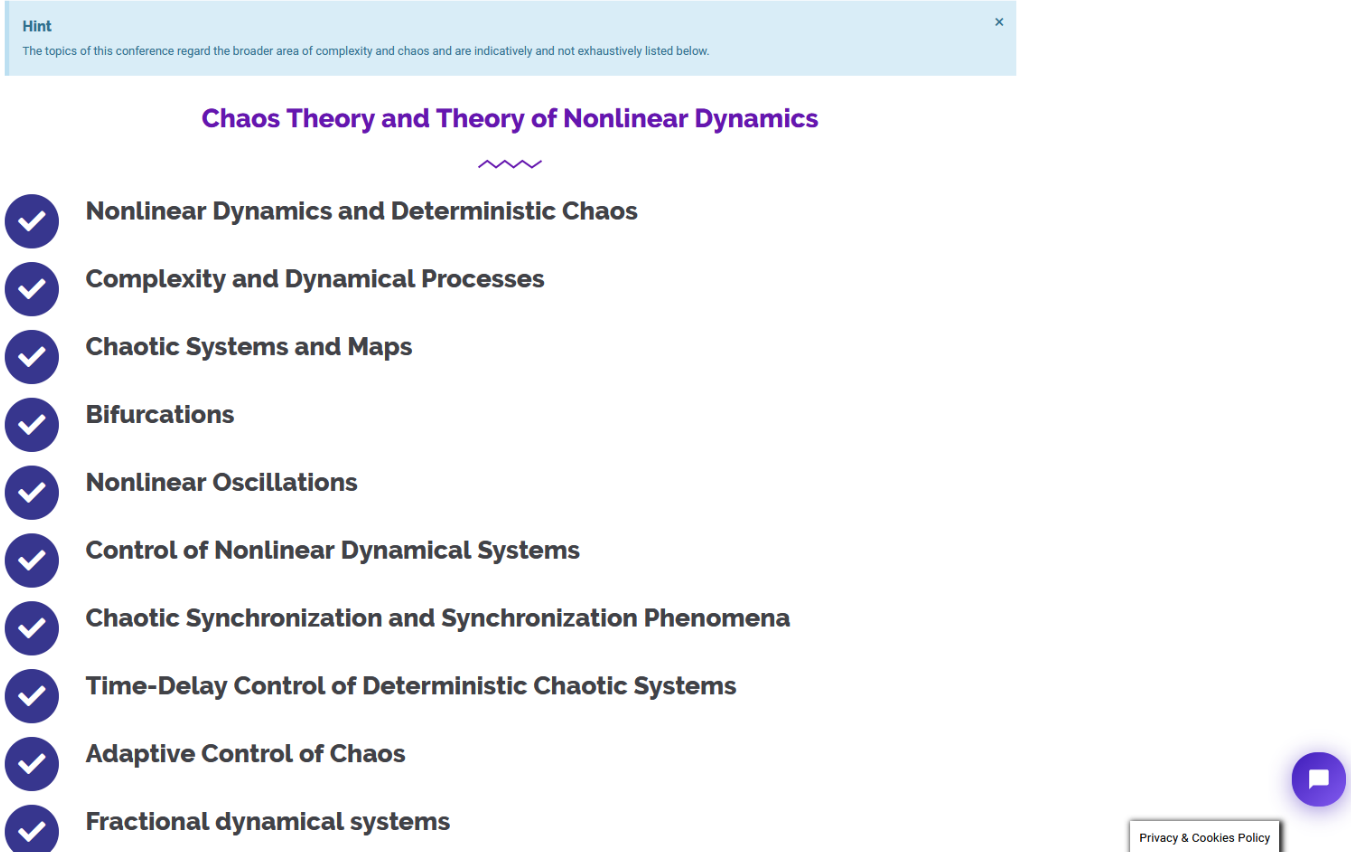


Registration Fees	Regular Deadline: April 22, 2019
Registration fees with no accommodation per person	549 €
Registration fees with accommodation in single room	799 €
Registration fees with accommodation in double room for 2 participants	1499 €
Registration fees with accommodation in double room for 1 participant + 1 accompanying person	1089 €

Buy ticket

Registration fees (participant) with accommodation include:
Admission to all Scientific Sessions (3 days meetings)
Congress Bag (including volume of the Congress Abstract CD)
1 abstract for the conference Book of Abstracts (Up to 3 abstracts per person is allowed for submission. 2nd and 3rd submissions cost 90 € each)
Each submitted abstract can lead to one paper in the Proceedings Book of the Conference, according to the registration payment. The main registration justifies publication of one paper up to 6 pages. For more than 6 pages, there exists an extra fee: 50 €/per page.
Coffee breaks during the Meetings and High class all-inclusive food and drinks (for more information please visit “Venue” page)
High class all-inclusive accommodation for 3 nights / 4 days (for more information please visit “Venue” page)
Welcome Cocktail (The event is an excellent opportunity to meet colleagues in a convivial atmosphere)

Figure 42 Website Registration Details page



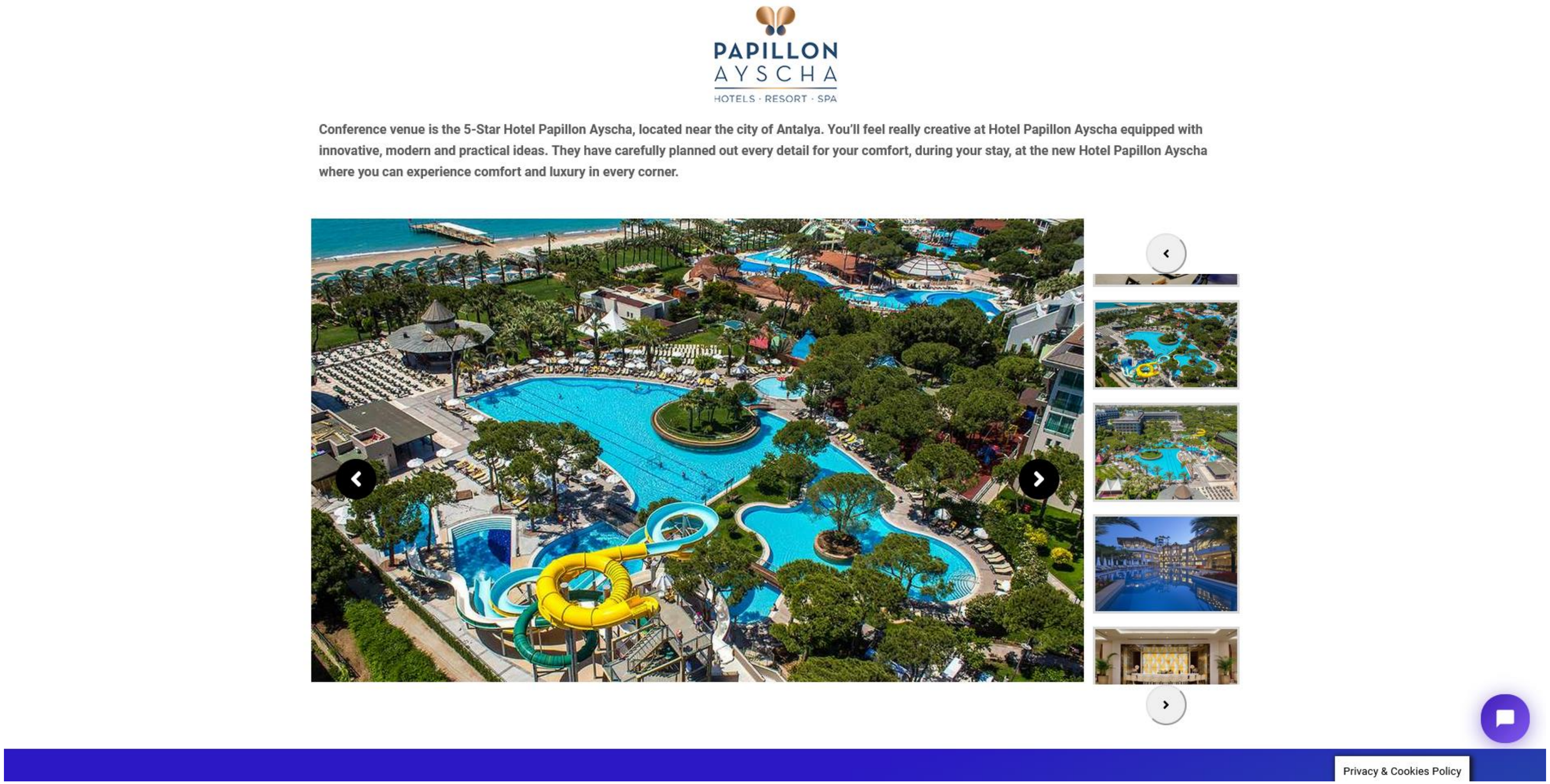
Hint

The topics of this conference regard the broader area of complexity and chaos and are indicatively and not exhaustively listed below.

Chaos Theory and Theory of Nonlinear Dynamics

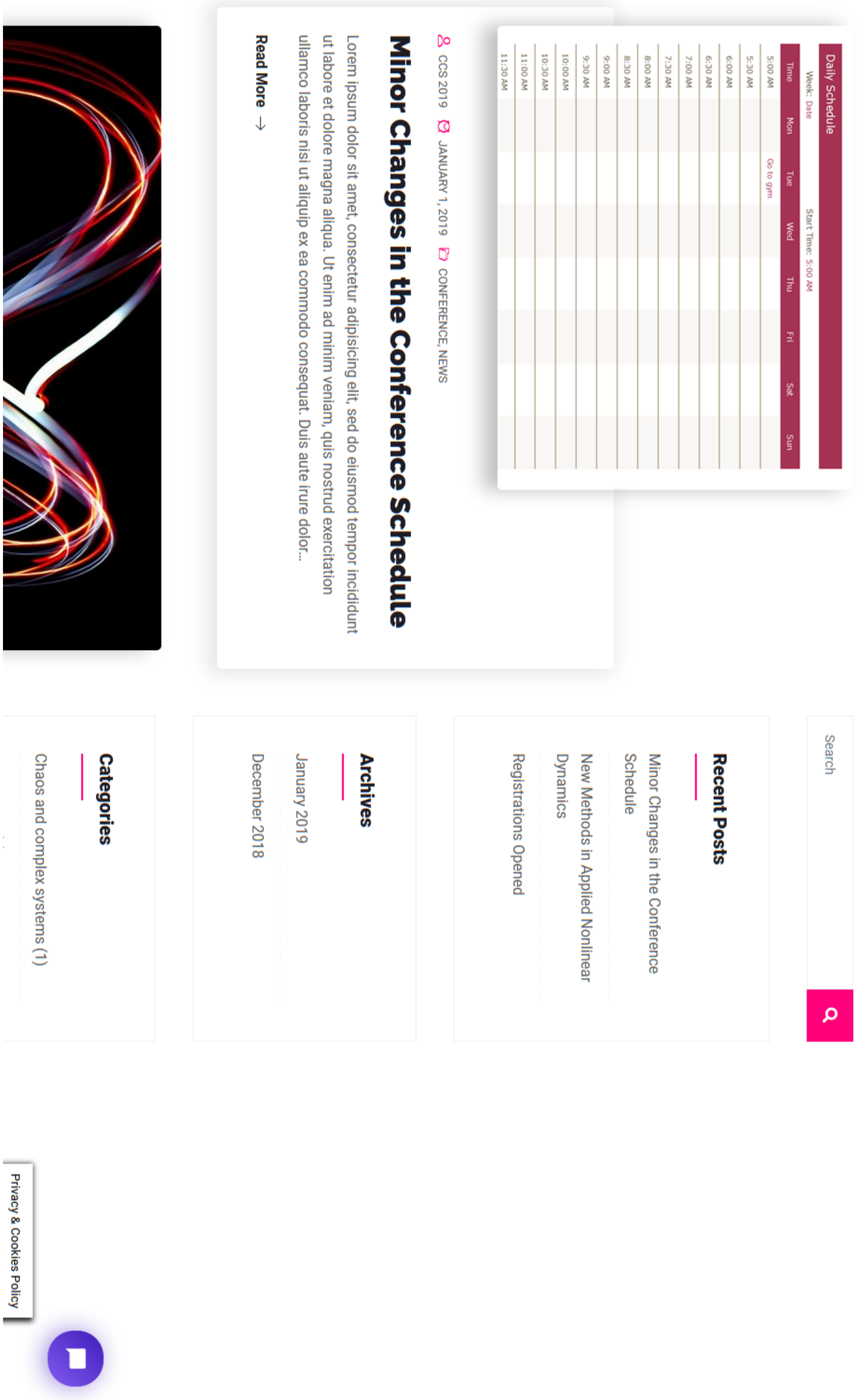
- ✓ Nonlinear Dynamics and Deterministic Chaos
- ✓ Complexity and Dynamical Processes
- ✓ Chaotic Systems and Maps
- ✓ Bifurcations
- ✓ Nonlinear Oscillations
- ✓ Control of Nonlinear Dynamical Systems
- ✓ Chaotic Synchronization and Synchronization Phenomena
- ✓ Time-Delay Control of Deterministic Chaotic Systems
- ✓ Adaptive Control of Chaos
- ✓ Fractional dynamical systems

Figure 43 Website Topics page



Conference venue is the 5-Star Hotel Papillon Ayscha, located near the city of Antalya. You'll feel really creative at Hotel Papillon Ayscha equipped with innovative, modern and practical ideas. They have carefully planned out every detail for your comfort, during your stay, at the new Hotel Papillon Ayscha where you can experience comfort and luxury in every corner.

Figure 44 Website Venue page



Daily Schedule

Week: Date Start Time: 5:00 AM

Time	Mon	Tue	Wed	Thu	Fri	Sat	Sun
5:00 AM		Go to gym					
5:30 AM							
6:00 AM							
6:30 AM							
7:00 AM							
7:30 AM							
8:00 AM							
8:30 AM							
9:00 AM							
9:30 AM							
10:00 AM							
10:30 AM							
11:00 AM							
11:30 AM							

CCS 2019 JANUARY 1, 2019 CONFERENCE, NEWS

Minor Changes in the Conference Schedule

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor...

Read More →

Recent Posts

Minor Changes in the Conference Schedule

New Methods in Applied Nonlinear Dynamics

Registrations Opened

Archives

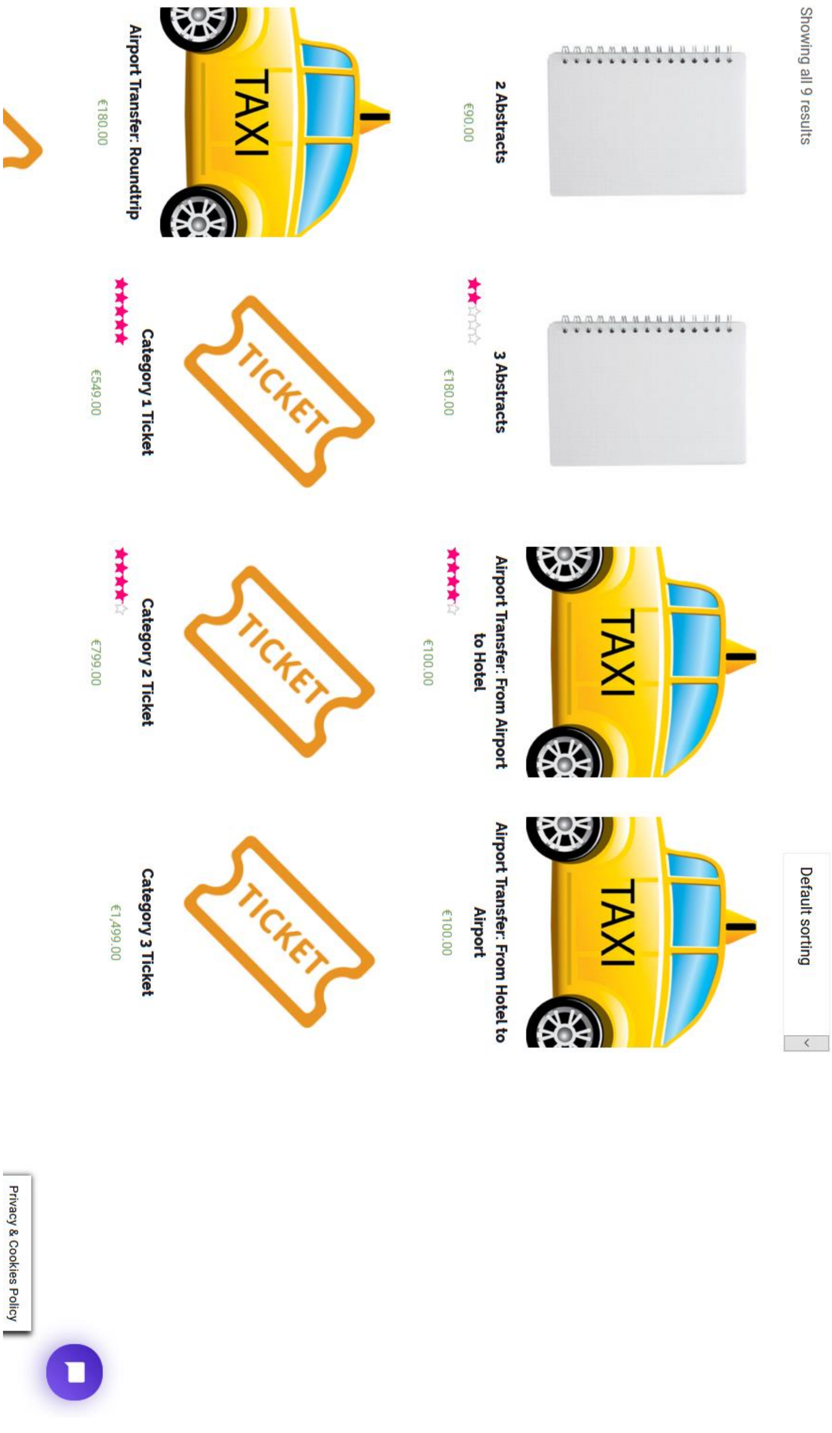
January 2019

December 2018

Categories

Chaos and complex systems (1)

Figure 45 Website Blog page



Showing all 9 results

Default sorting

2 Abstracts

€90.00

3 Abstracts

€180.00

Airport Transfer: From Airport to Hotel

€100.00

Airport Transfer: From Hotel to Airport

€100.00

Airport Transfer: Roundtrip

€180.00

Category 1 Ticket

€549.00

Category 2 Ticket

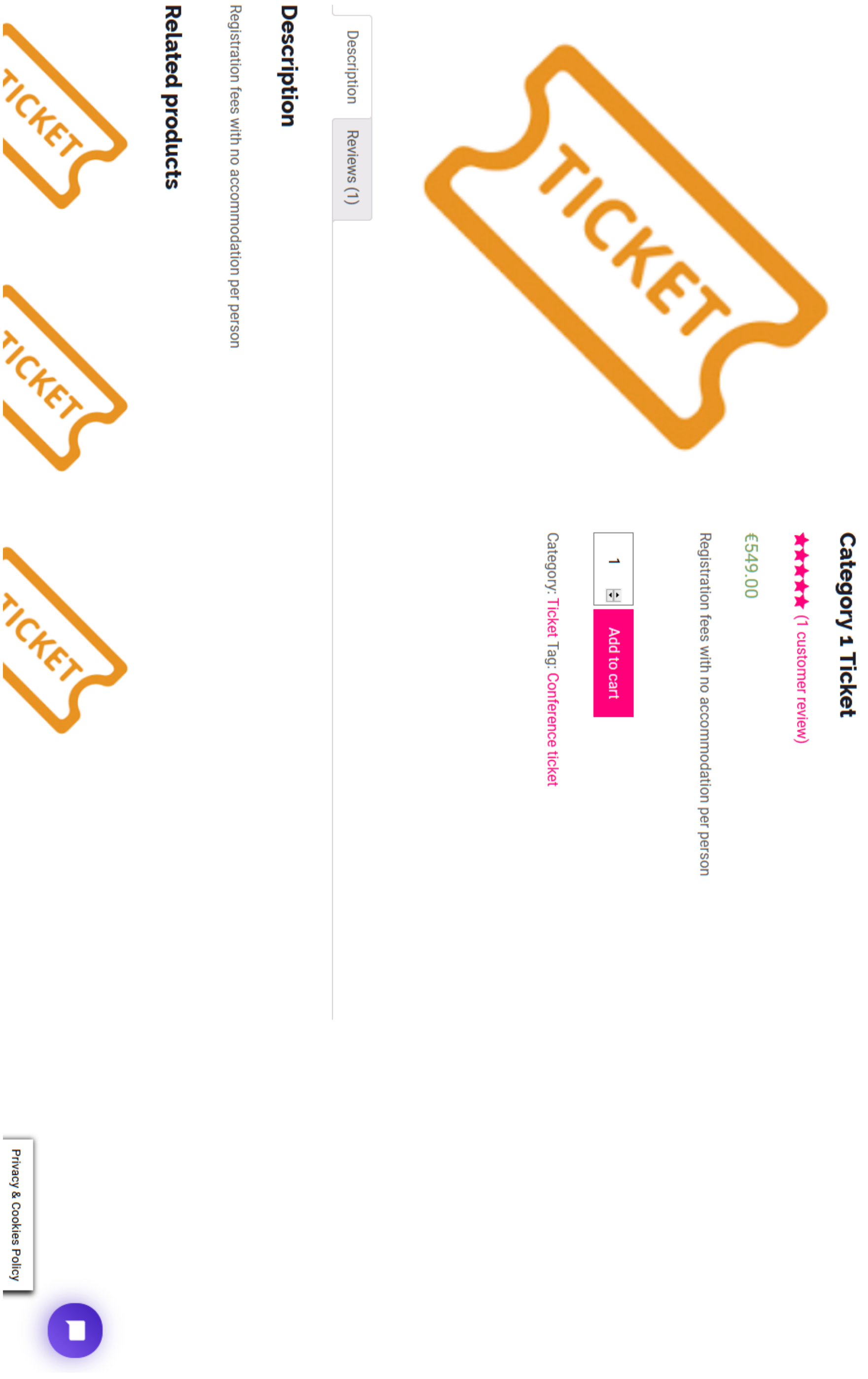
€799.00

Category 3 Ticket

€1,499.00

Privacy & Cookies Policy

Figure 46 Website Shop page



Category 1 Ticket

★★★★★ (1 customer review)

€549.00

Registration fees with no accommodation per person

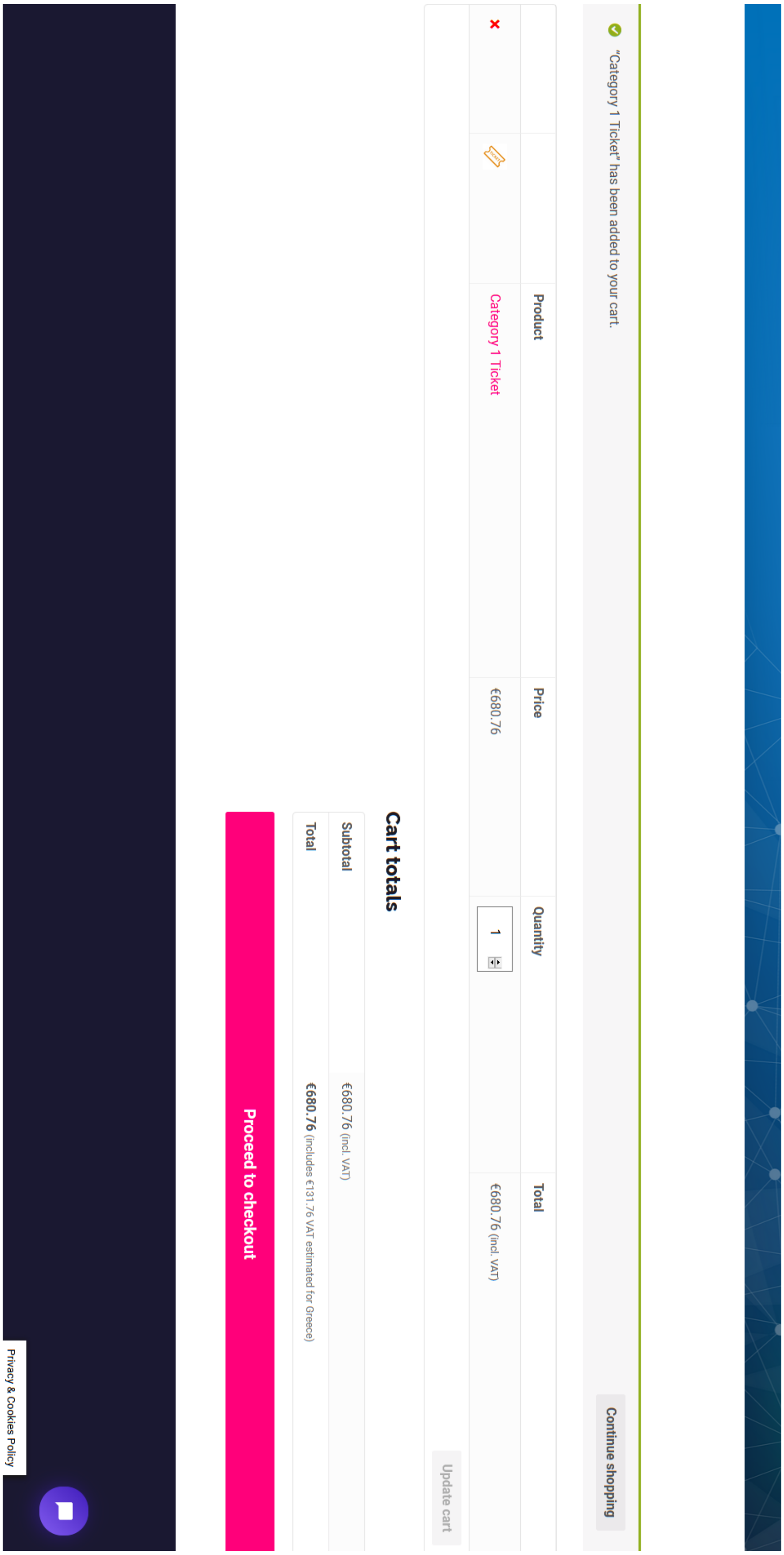
Category: Ticket Tag: Conference ticket

Description

Registration fees with no accommodation per person

Related products

Figure 47 Website Product Details page



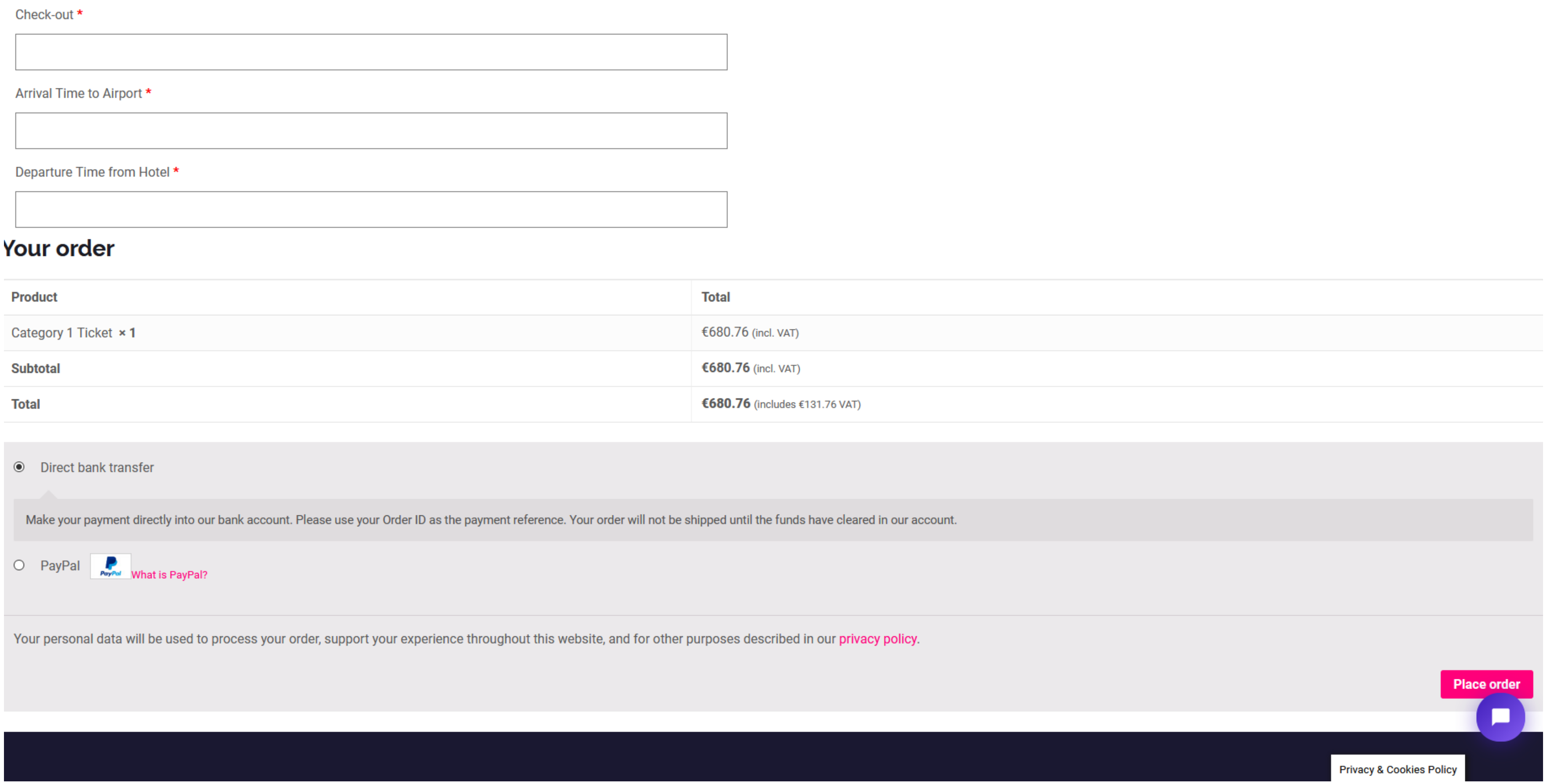
"Category 1 Ticket" has been added to your cart.

		Product	Price	Quantity	Total
×		Category 1 Ticket	€680.76	1	€680.76 (incl. VAT)

Cart totals

Subtotal	€680.76 (incl. VAT)
Total	€680.76 (includes €131.76 VAT estimated for Greece)

Figure 48 Website Cart page



Check-out *

Arrival Time to Airport *

Departure Time from Hotel *

Your order

Product	Total
Category 1 Ticket × 1	€680.76 (incl. VAT)
Subtotal	**€680.76** (incl. VAT)
Total	**€680.76** (includes €131.76 VAT)

● Direct bank transfer

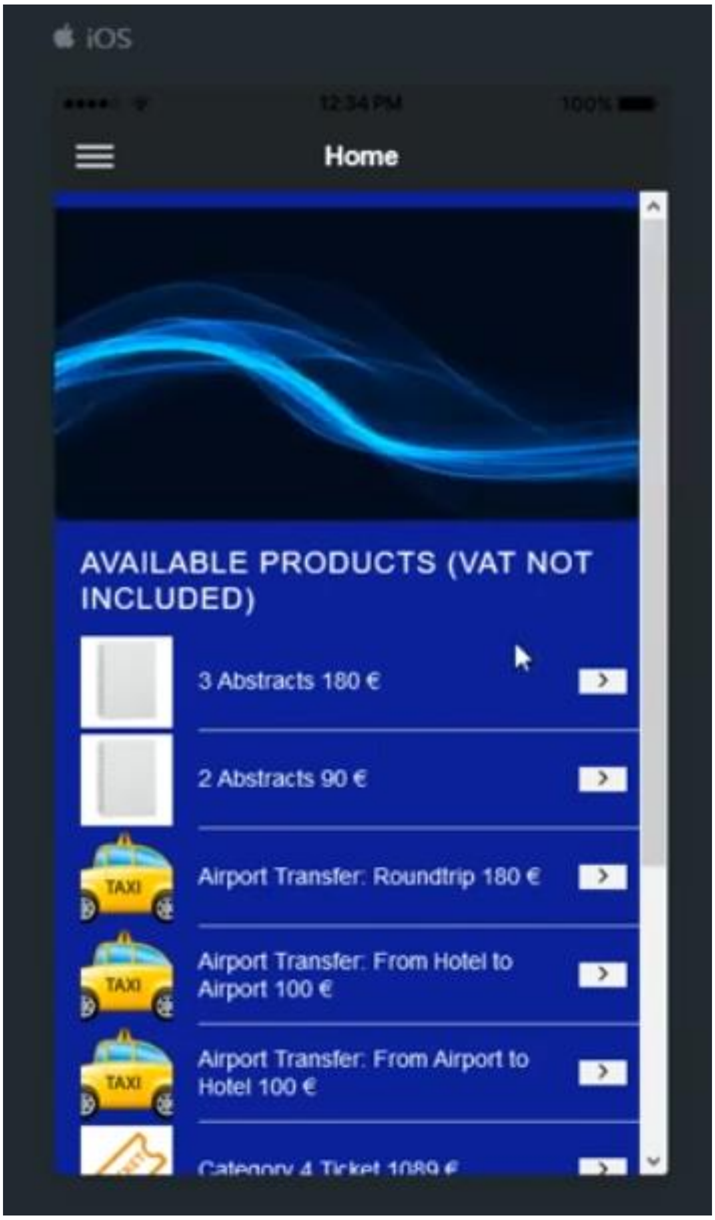
Make your payment directly into our bank account. Please use your Order ID as the payment reference. Your order will not be shipped until the funds have cleared in our account.

○ PayPal What is PayPal?

Your personal data will be used to process your order, support your experience throughout this website, and for other purposes described in our privacy policy.

Figure 49 Website Checkout page

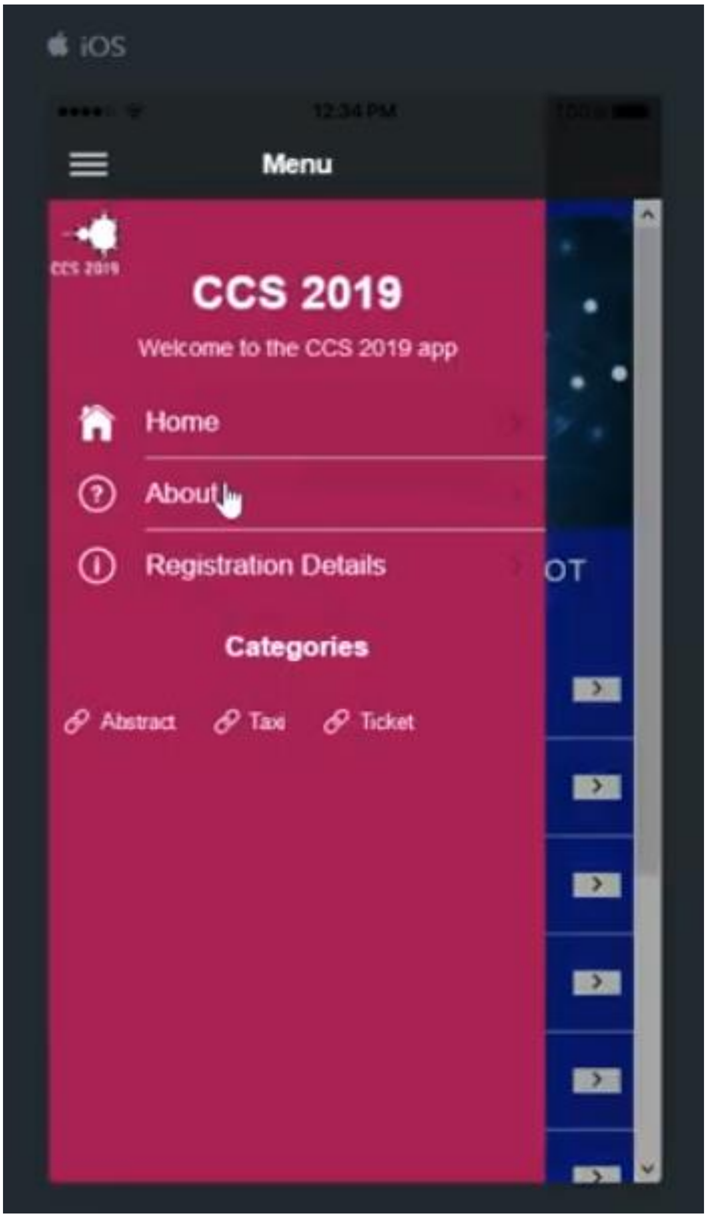
Appendix B



AVAILABLE PRODUCTS (VAT NOT INCLUDED)

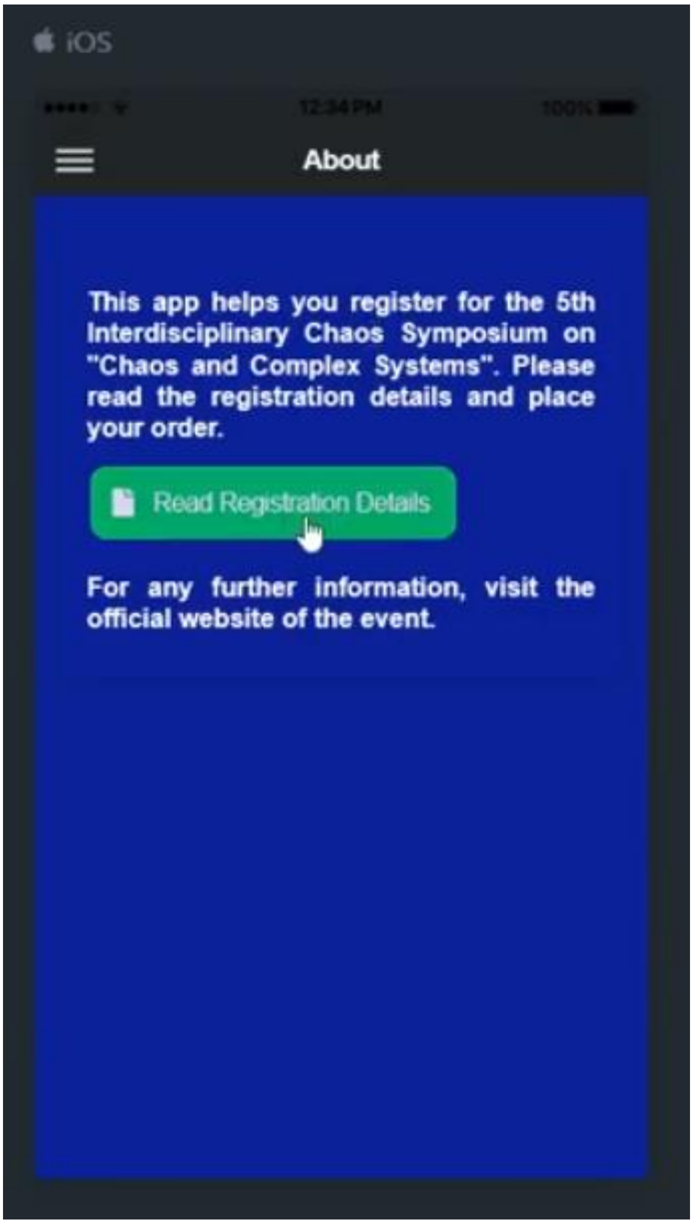
- 3 Abstracts 180 €
- 2 Abstracts 90 €
- Airport Transfer: Roundtrip 180 €
- Airport Transfer: From Hotel to Airport 100 €
- Airport Transfer: From Airport to Hotel 100 €
- Category 4 Ticket 1089 €

Figure 50 Mobile application Home page



Welcome to the CCS 2019 app

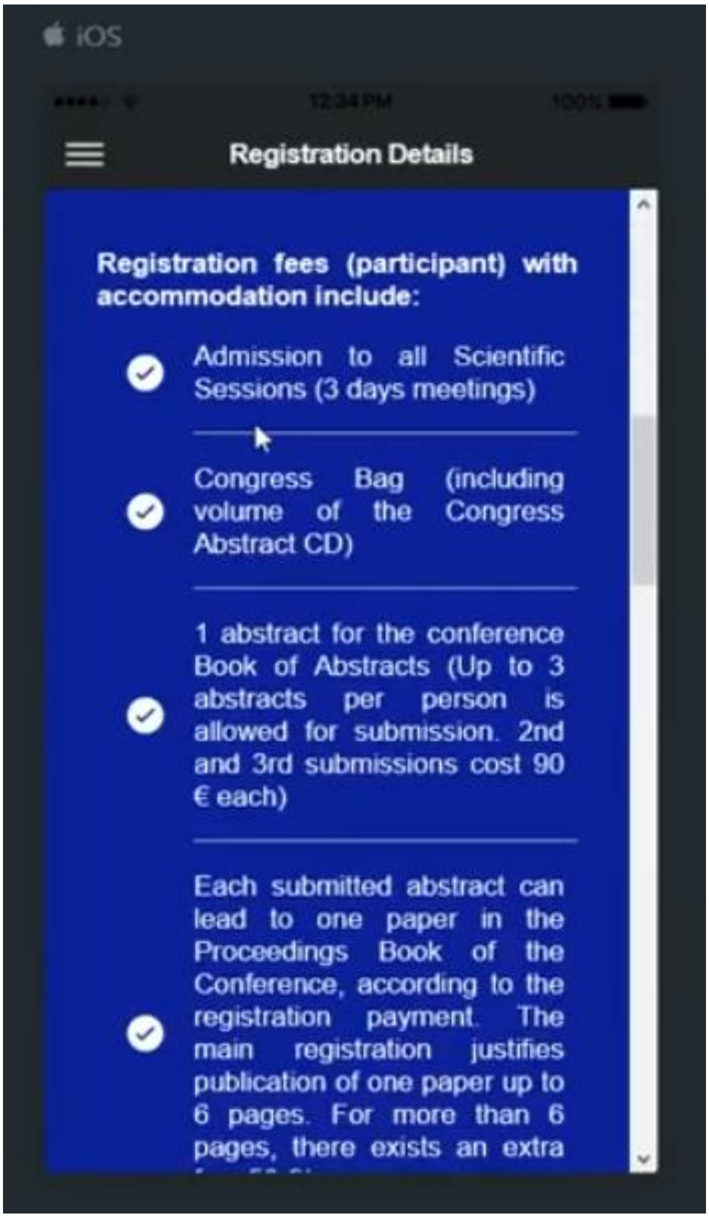
Figure 51 Mobile application side menu



This app helps you register for the 5th Interdisciplinary Chaos Symposium on "Chaos and Complex Systems". Please read the registration details and place your order.

For any further information, visit the official website of the event.

Figure 52 Mobile application About page

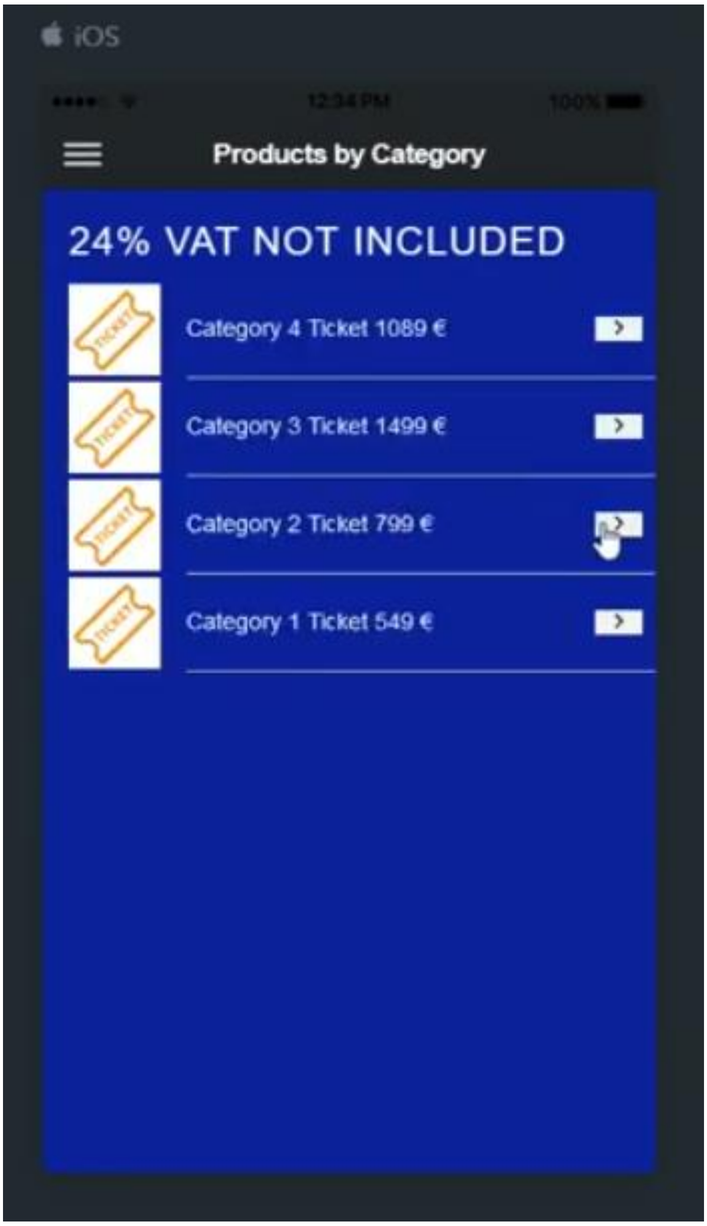


Registration Details

Registration fees (participant) with accommodation include:

- ✓ Admission to all Scientific Sessions (3 days meetings)
- ✓ Congress Bag (including volume of the Congress Abstract CD)
- ✓ 1 abstract for the conference Book of Abstracts (Up to 3 abstracts per person is allowed for submission. 2nd and 3rd submissions cost 90 € each)
- ✓ Each submitted abstract can lead to one paper in the Proceedings Book of the Conference, according to the registration payment. The main registration justifies publication of one paper up to 6 pages. For more than 6 pages, there exists an extra

Figure 53 Mobile application Registration Details page



Products by Category

24% VAT NOT INCLUDED

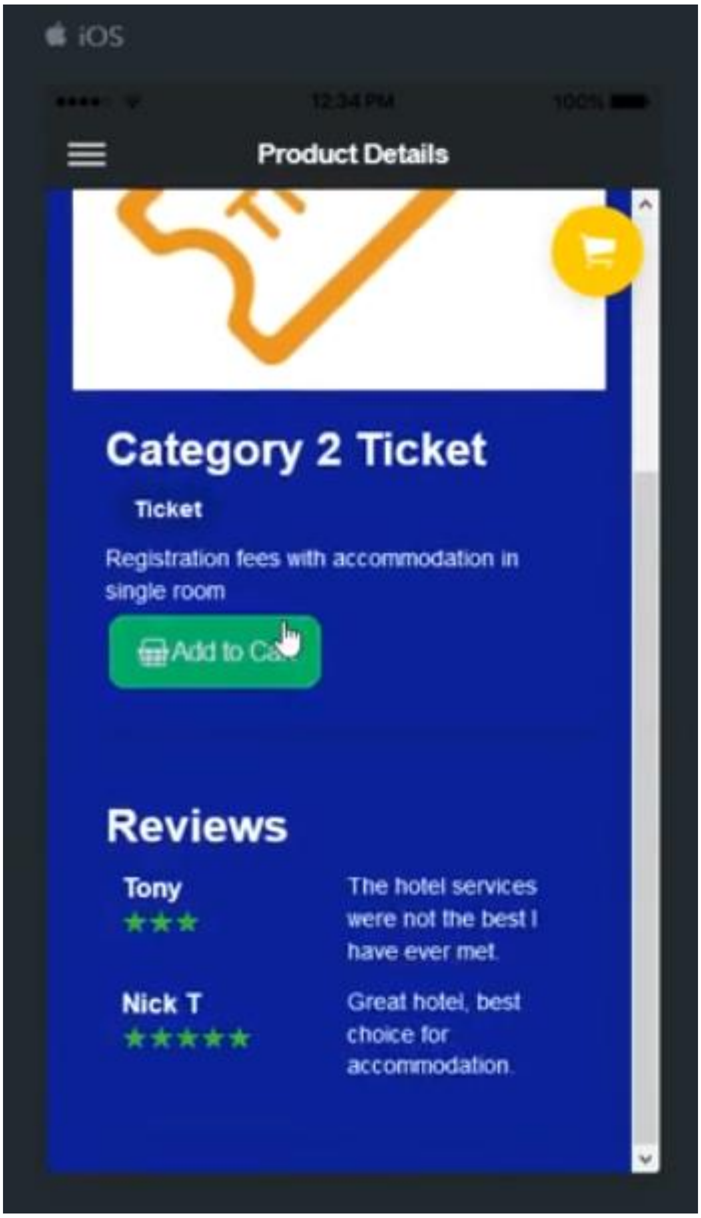
Category 4 Ticket 1089 €

Category 3 Ticket 1499 €

Category 2 Ticket 799 €

Category 1 Ticket 549 €

Figure 54 Mobile application Products by Category page



Category 2 Ticket

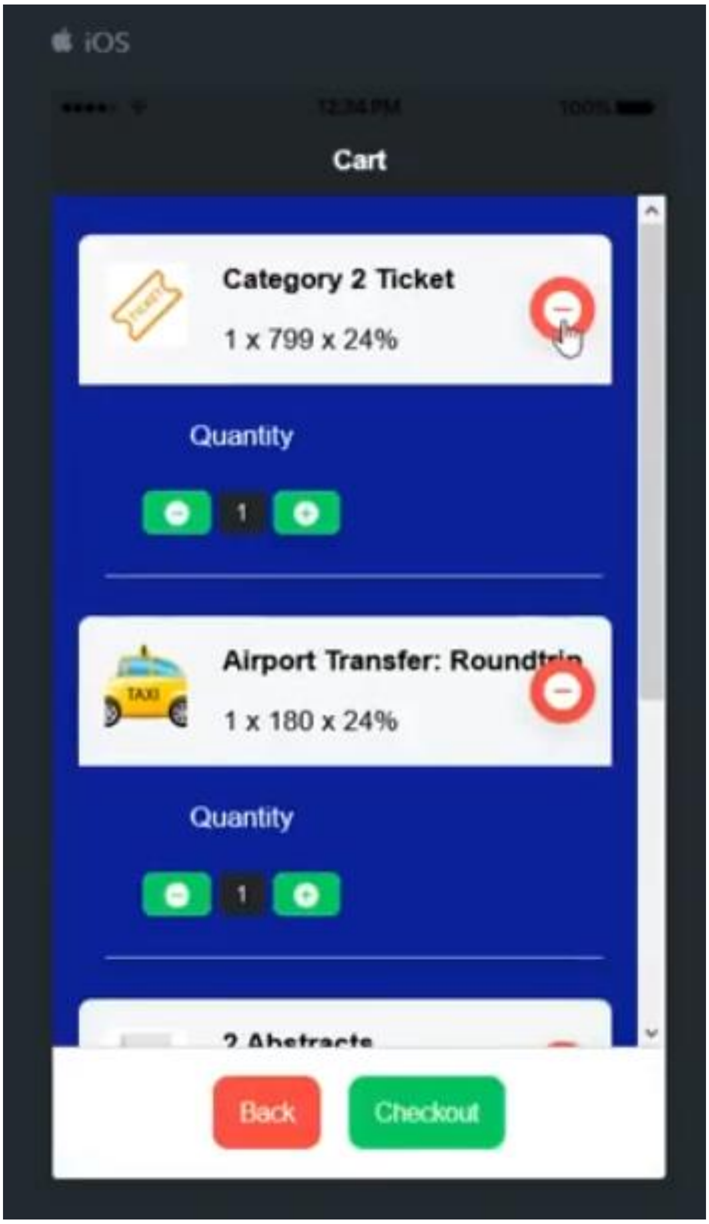
Ticket

Registration fees with accommodation in single room

Reviews

Tony ★★★	The hotel services were not the best I have ever met.
Nick T ★★★★★	Great hotel, best choice for accommodation.

Figure 55 Mobile application Product Details page



Cart

Category 2 Ticket

1 x 799 x 24%

Quantity

1

Airport Transfer: Roundtrip

1 x 180 x 24%

Quantity

1

2 Abstracts

Figure 56 Mobile application Cart page

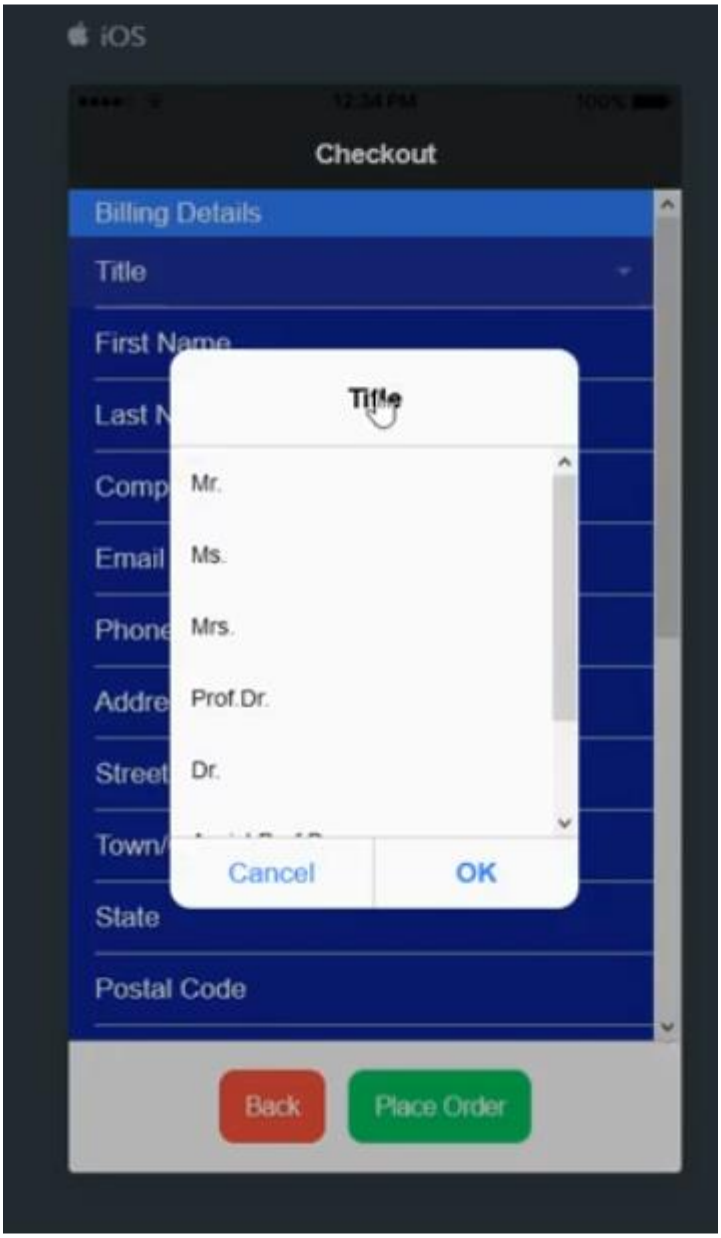


Figure 57 Mobile application Checkout page

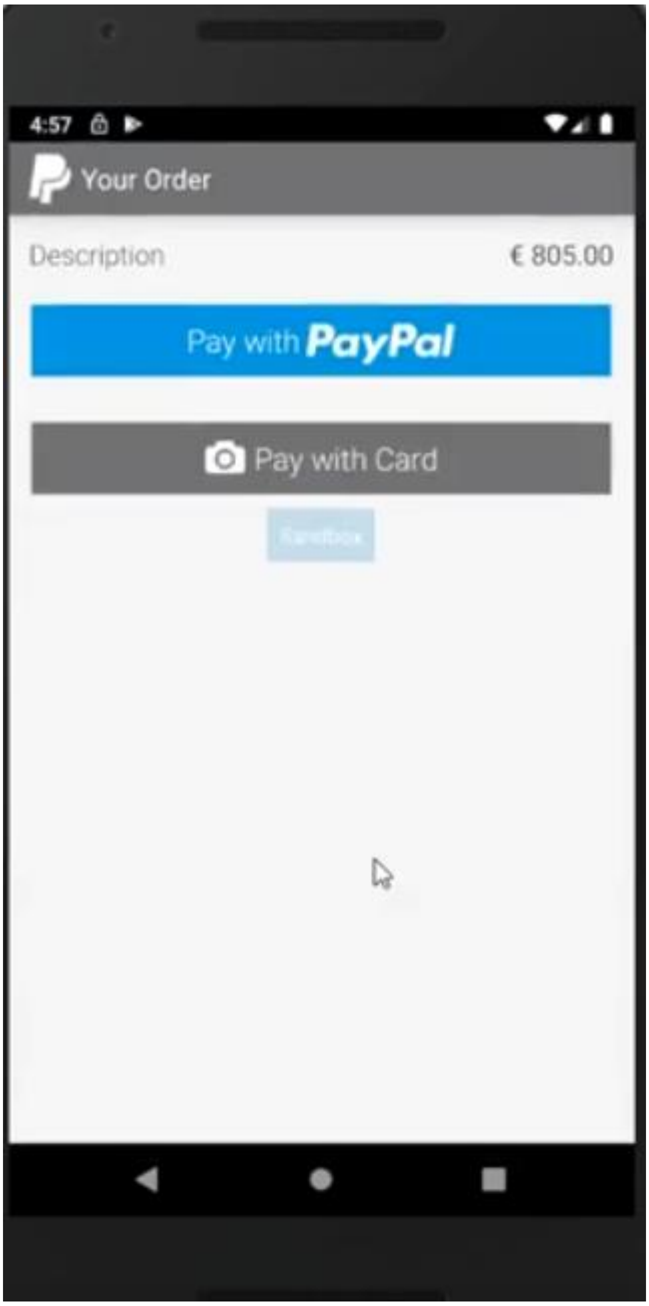


Figure 58 PayPal payment gateway page

Appendix C

App.component.ts

```
import { Component, ViewChild } from '@angular/core';

import { Platform,MenuController} from '@ionic/angular';
import { SplashScreen } from '@ionic-native/splash-screen/ngx';
import { StatusBar } from '@ionic-native/status-bar/ngx';
import { ApiService } from './api.service';
import { Observable } from 'rxjs';
import { Router } from '@angular/router';



@Component({
  selector: 'app-root',
  templateUrl: 'app.component.html',
  styleUrls: ['app.component.scss']
})
export class AppComponent {
  categories : any = [];

  public appPages = [
    {
      title: 'Home',
      url: '/home',
      icon: 'home'
    },
    {
      title: 'About',
      url: '/about',
      icon: 'help-circle-outline'
    },
    {
      title: 'Registration Details',
      url: '/registration-details',
      icon: 'information-circle-outline'
    }

  ];

  constructor(
    private platform: Platform,
    private splashScreen: SplashScreen,
    private statusBar: StatusBar,
    public api:ApiService,
    private router: Router,
    public menuCtrl: MenuController
    )

   {
    let temp:Observable<any> = this.api.get('products/categories');
    temp.subscribe(result => {this.categories = result, console.log(this.categories)});

    this.initializeApp();

  }

  initializeApp() {
    this.platform.ready().then(() => {
      this.statusBar.styleDefault();
      this.splashScreen.hide();
    });
  }

  openCategoryPage(category:String) { //get the selected category and pass it to the "products by category" page
    this.router.navigate(['/products-by-category', category]);

  }

  closeMenu() {
    this.menuCtrl.close();
  }



}
```

App.component.html

```
<ion-app>
  <ion-split-pane>
    <ion-menu type="overlay">
      <ion-header>
        <ion-toolbar color = "dark">
            <ion-buttons slot="start">
                <ion-menu-button></ion-menu-button>
              </ion-buttons>
          <ion-title>Menu</ion-title>
        </ion-toolbar>
      </ion-header>
      <ion-content class="card-background-page" >
        <ion-card>
          <ion-thumbnail slot="start">
          <img src="./assets/images/logo.png" />
        </ion-thumbnail>
          <div class="card-
title" style = "color :white;">CCS 2019</div> <br/><br/>
          <div class="subtitle" style = "color :white;">Welcome to the CC
S 2019 app</div> <br/>
        </ion-card>

        <ion-list>
          <ion-menu-toggle auto-hide="true" *ngFor="let p of appPages">
            <ion-item [routerDirection]="'root'" [routerLink]="[p.url]">
              <ion-
icon slot="start" [name]="p.icon" style = "color :white;"></ion-icon>
              <ion-label style = "color :white;">
                {{p.title}}
              </ion-label>
            </ion-item>
          </ion-menu-toggle>
        <ion-list> <br/>

          <ion-label ><h2 style = "color :white; text-
align: center; font-weight : bold;">Categories</h2></ion-label> <br/>
            <ion-
button color = "#9F1053" size = "small" menuClose *ngFor = "let category
of categories" (click)="closeMenu(); openCategoryPage(category.name)" >
              <ion-icon ios="ios-link" md="md-link" slot = "start"></ion-
icon>
              <ion-label item-right>
                 {{category.name}}
              </ion-label>
            </ion-button>
        </ion-list>
        </ion-list>
      </ion-content>
    </ion-menu>
    <ion-router-outlet main></ion-router-outlet>
  </ion-split-pane>
</ion-app>
```

App.component.scss

```
ion-content{
    --ion-background-color: #9F1053;
}
.card-background-page {

    ion-card {
        position: relative;
        text-align: center;
        margin: 0px !important;
        width: 100% !important;
        background-color: #9F1053;
    }

    .card-title {
        position: absolute;
        top: 36%;
        font-size: 2em;
        width: 100%;
        font-weight: bold;
        color: #000000
    }

    .card-subtitle {
        font-size: 1em;
        position: absolute;
        top: 60%;
        width: 100%;
        color: #000000
    }
}
```

App.module.ts

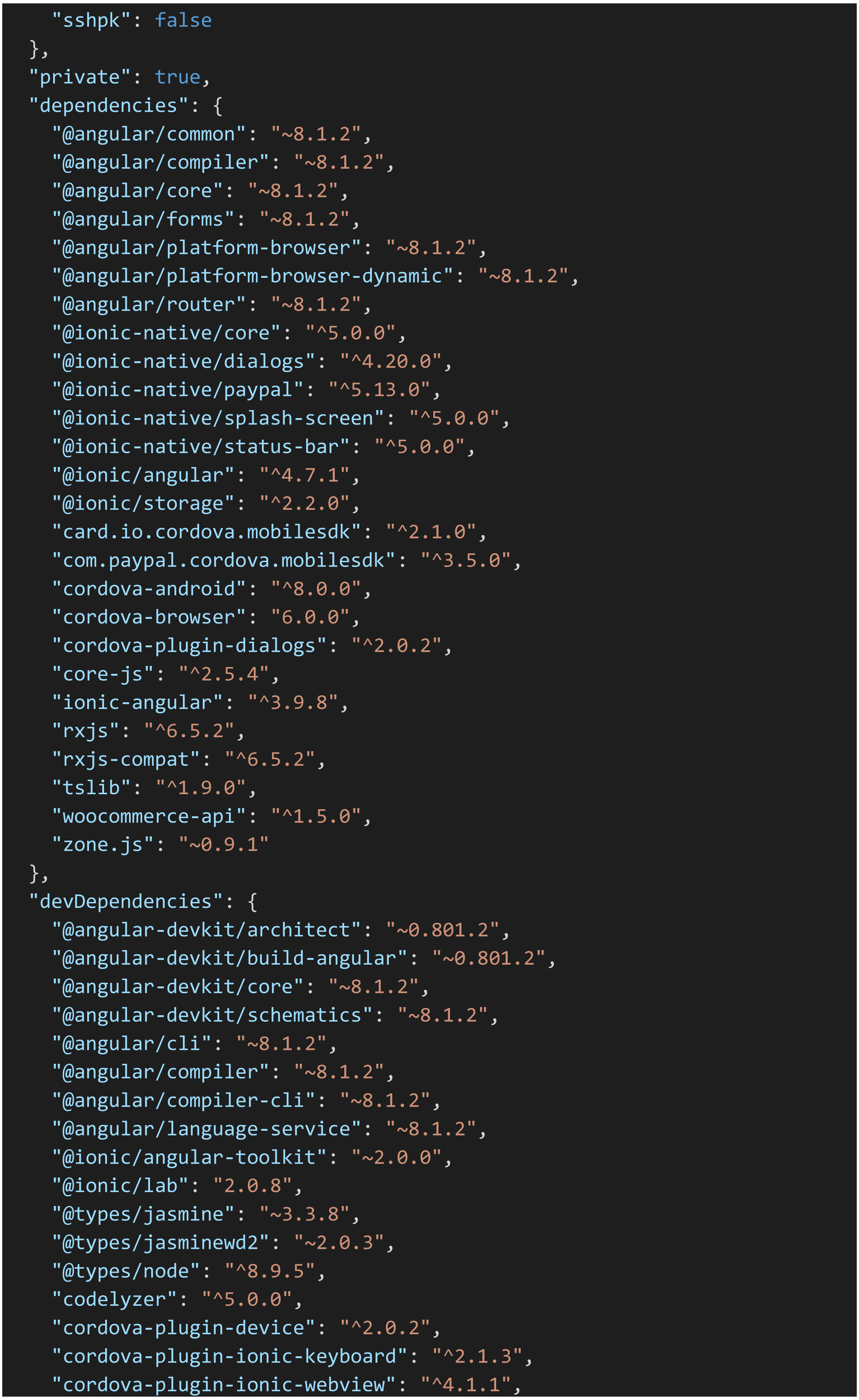
```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { RouteReuseStrategy } from '@angular/router';
import { IonicModule, IonicRouteStrategy } from '@ionic/angular';
import { SplashScreen } from '@ionic-native/splash-screen/ngx';
import { StatusBar } from '@ionic-native/status-bar/ngx';
import { AppComponent } from './app.component';
import { AppRoutingModule } from './app-routing.module';
import { HttpClientModule } from '@angular/common/http';
import { CartPage } from './cart/cart.page';
import { IonicStorageModule } from '@ionic/storage';
import { CheckoutPage } from './checkout/checkout.page';
import { FormsModule } from '@angular/forms';
import { PayPal, PayPalPayment, PayPalConfiguration } from '@ionic-native/paypal/ngx';

@NgModule({
  declarations: [
    AppComponent,
    CartPage,
    CheckoutPage

  ],
  entryComponents: [
    AppComponent,
    CartPage,
    CheckoutPage
  ],
  imports: [
    BrowserModule,
    IonicModule.forRoot(),
    IonicStorageModule.forRoot(),
    AppRoutingModule,
    HttpClientModule,
    FormsModule
  ],
  providers: [
    StatusBar,
    SplashScreen,
    PayPal,
    { provide: RouteReuseStrategy, useClass: IonicRouteStrategy }
  ],
  bootstrap: [AppComponent]
})
export class AppModule {}
```

Api.service.ts

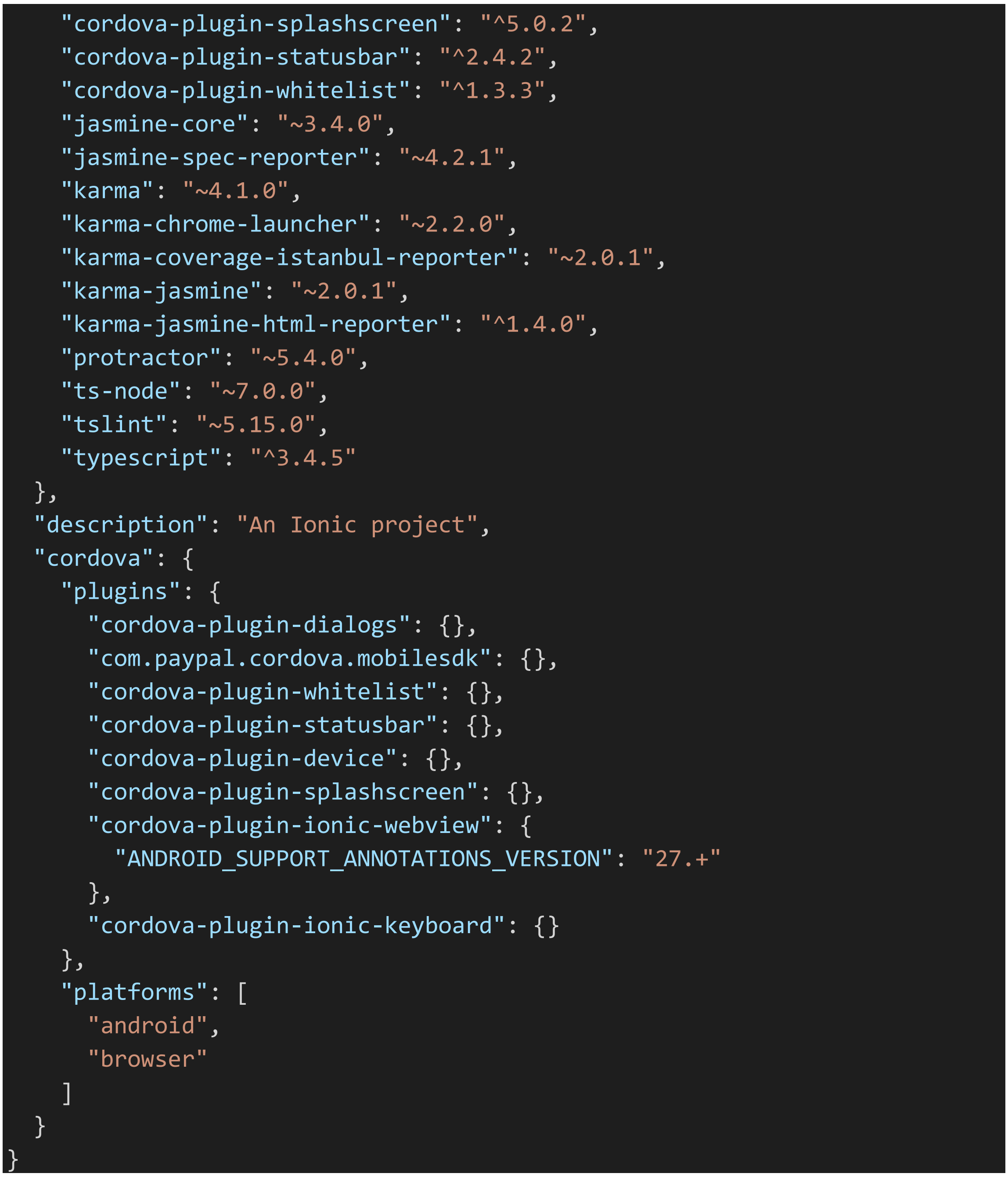
```
import { Injectable } from '@angular/core';
import {HttpClient} from '@angular/common/http'

@Injectable({
  providedIn: 'root'
})
export class ApiService {

  private API_URL:string = 'https://conference1.geowebsite.net/wp-
json/public-woo/v1/';

  constructor(public http: HttpClient) {
    console.log('This is the Api Service');
   }

get(query:string = '') {
  return this.http.get(this.API_URL + query);
}

}
```

Package.json

```
{
  "name": "CC2019",
  "version": "0.0.1",
  "author": "Ionic Framework",
  "homepage": "https://ionicframework.com/",
  "scripts": {
    "ng": "ng",
    "start": "ng serve",
    "build": "ng build",
    "test": "ng test",
    "lint": "ng lint",
    "e2e": "ng e2e"
  },
  "browser": {
    "http": false,
    "https": false,
    "net": false,
    "path": false,
    "stream": false,
    "tls": false,
    "fs": false,
    "crypto": false,
    "zlib": false,
```



```
    "sshpk": false
  },
  "private": true,
  "dependencies": {
    "@angular/common": "~8.1.2",
    "@angular/compiler": "~8.1.2",
    "@angular/core": "~8.1.2",
    "@angular/forms": "~8.1.2",
    "@angular/platform-browser": "~8.1.2",
    "@angular/platform-browser-dynamic": "~8.1.2",
    "@angular/router": "~8.1.2",
    "@ionic-native/core": "^5.0.0",
    "@ionic-native/dialogs": "^4.20.0",
    "@ionic-native/paypal": "^5.13.0",
    "@ionic-native/splash-screen": "^5.0.0",
    "@ionic-native/status-bar": "^5.0.0",
    "@ionic/angular": "^4.7.1",
    "@ionic/storage": "^2.2.0",
    "card.io.cordova.mobilesdk": "^2.1.0",
    "com.paypal.cordova.mobilesdk": "^3.5.0",
    "cordova-android": "^8.0.0",
    "cordova-browser": "6.0.0",
    "cordova-plugin-dialogs": "^2.0.2",
    "core-js": "^2.5.4",
    "ionic-angular": "^3.9.8",
    "rxjs": "^6.5.2",
    "rxjs-compat": "^6.5.2",
    "tslib": "^1.9.0",
    "woocommerce-api": "^1.5.0",
    "zone.js": "~0.9.1"
  },
  "devDependencies": {
    "@angular-devkit/architect": "~0.801.2",
    "@angular-devkit/build-angular": "~0.801.2",
    "@angular-devkit/core": "~8.1.2",
    "@angular-devkit/schematics": "~8.1.2",
    "@angular/cli": "~8.1.2",
    "@angular/compiler": "~8.1.2",
    "@angular/compiler-cli": "~8.1.2",
    "@angular/language-service": "~8.1.2",
    "@ionic/angular-toolkit": "~2.0.0",
    "@ionic/lab": "2.0.8",
    "@types/jasmine": "~3.3.8",
    "@types/jasminewd2": "~2.0.3",
    "@types/node": "^8.9.5",
    "codelyzer": "^5.0.0",
    "cordova-plugin-device": "^2.0.2",
    "cordova-plugin-ionic-keyboard": "^2.1.3",
    "cordova-plugin-ionic-webview": "^4.1.1",
```



```
    "cordova-plugin-splashscreen": "^5.0.2",
    "cordova-plugin-statusbar": "^2.4.2",
    "cordova-plugin-whitelist": "^1.3.3",
    "jasmine-core": "~3.4.0",
    "jasmine-spec-reporter": "~4.2.1",
    "karma": "~4.1.0",
    "karma-chrome-launcher": "~2.2.0",
    "karma-coverage-istanbul-reporter": "~2.0.1",
    "karma-jasmine": "~2.0.1",
    "karma-jasmine-html-reporter": "^1.4.0",
    "protractor": "~5.4.0",
    "ts-node": "~7.0.0",
    "tslint": "~5.15.0",
    "typescript": "^3.4.5"
  },
  "description": "An Ionic project",
  "cordova": {
    "plugins": {
      "cordova-plugin-dialogs": {},
      "com.paypal.cordova.mobilesdk": {},
      "cordova-plugin-whitelist": {},
      "cordova-plugin-statusbar": {},
      "cordova-plugin-device": {},
      "cordova-plugin-splashscreen": {},
      "cordova-plugin-ionic-webview": {
        "ANDROID_SUPPORT_ANNOTATIONS_VERSION": "27.+"
      },
      "cordova-plugin-ionic-keyboard": {}
    },
    "platforms": [
      "android",
      "browser"
    ]
  }
}
```

About.page.ts

```
import { Component, OnInit } from '@angular/core';
import { Router} from '@angular/router';

@Component({
  selector: 'app-about',
  templateUrl: './about.page.html',
  styleUrls: ['./about.page.scss'],
})
export class AboutPage implements OnInit {

  constructor(private router: Router) { }

  ngOnInit() {
  }


  openRegDetails() {
    this.router.navigateByUrl('/registration-details')

  }

}
```

About.page.html

```
<ion-header>
    <ion-toolbar color = "dark">
      <ion-buttons slot="start">
        <ion-menu-button></ion-menu-button>
      </ion-buttons>
      <ion-title style ="color: #ffffff;">
        About
      </ion-title>
    </ion-toolbar>
  </ion-header>

<ion-content>

      <ion-card padding>
        <ion-row>
        <ion-text>
            <h6 style = "text-align : justify; font-
weight: bold; color: #ffffff;">This app helps you register for the 5th In
terdisciplinary Chaos Symposium on "Chaos and Complex Systems". Please re
ad the registration details and place your order.</h6>
```

```
        </ion-text>
        </ion-row>
        <ion-row>
          <ion-
button color = "success" expand = "block" (click) = "openRegDetails()"><i
on-icon name="document" slot="start"></ion-
icon>Read Registration Details</ion-button>
        </ion-row>
        <ion-row><ion-text><h6 style = "text-align : justify; font-
weight: bold; color: #ffffff;">For any further information, visit the off
icial website of the event.</h6></ion-text></ion-row>
      </ion-card>

</ion-content>
```

About.page.scss

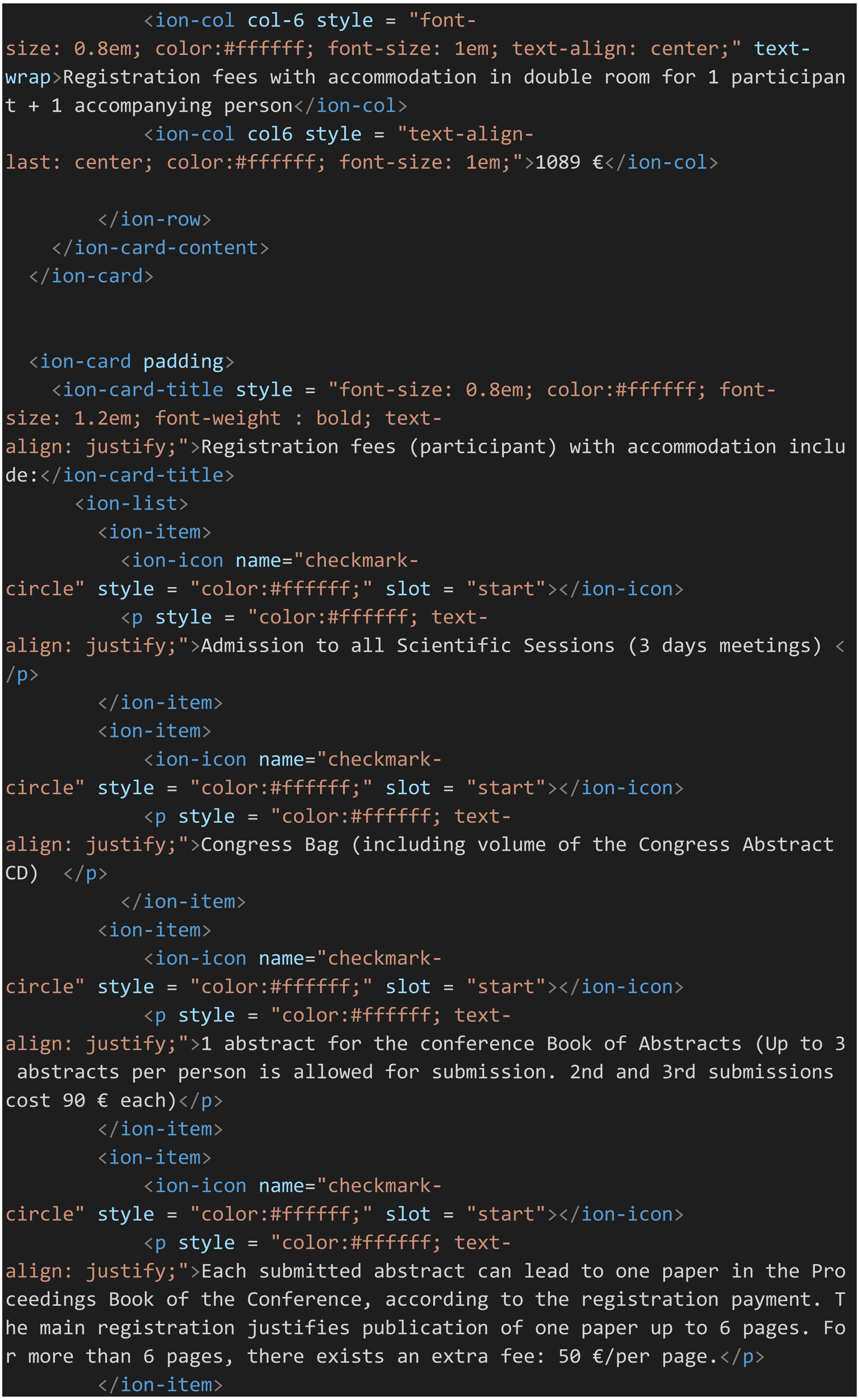
```
ion-content{
    --ion-background-color:#0E1C95;
}
```

Registration-details.page.ts

```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-registration-details',
  templateUrl: './registration-details.page.html',
  styleUrls: ['./registration-details.page.scss'],
})
export class RegistrationDetailsPage implements OnInit {

  constructor() { }

  ngOnInit() {
  }

}
```

Registration-details.page.html

```html
<ion-header>
    <ion-toolbar color = "dark">
      <ion-buttons slot="start">
        <ion-menu-button></ion-menu-button>
      </ion-buttons>
      <ion-title style ="color: #ffffff;">
        Registration Details
      </ion-title>
    </ion-toolbar>
  </ion-header>

<ion-content>

  <ion-card>
    <ion-card-content padding>
        <ion-row>
        <ion-col col-6 style = "color:#ffffff; font-weight: bold; text-align: center; font-size: 1.3em;">Registration Fees</ion-col>
        <ion-col col6 style = "color:#ffffff; font-weight: bold; text-align: center; font-size: 1.3em;">Regular Deadline: April 22, 2019</ion-col>
        </ion-row>
        <ion-row>
        <ion-col col-6 style = "font-size: 0.8em; color:#ffffff; font-size: 1em; text-align: center;" text-wrap>Registration fees with no accommodation per person</ion-col>
        <ion-col col6 style = "text-align-last: center; color:#ffffff; font-size: 1em;">549 €</ion-col>

        </ion-row>
        <ion-row>
            <ion-col col-6 style = "font-size: 0.8em; color:#ffffff; font-size: 1em; text-align: center; " text-wrap>Registration fees with accommodation in single room</ion-col>
            <ion-col col6 style = "text-align-last: center; color:#ffffff; font-size: 1em;">799 €</ion-col>
        </ion-row>
        <ion-row>
            <ion-col col-6 style = "font-size: 0.8em; color:#ffffff; font-size: 1em; text-align: center;" text-wrap>Registration fees with accommodation in double room for 2 participants</ion-col>
            <ion-col col6 style = "text-align-last: center; color:#ffffff; font-size: 1em;">1499 €</ion-col>

        </ion-row>
        <ion-row>
```



```
            <ion-col col-6 style = "font-
size: 0.8em; color:#ffffff; font-size: 1em; text-align: center;" text-
wrap>Registration fees with accommodation in double room for 1 participan
t + 1 accompanying person</ion-col>
            <ion-col col6 style = "text-align-
last: center; color:#ffffff; font-size: 1em;">1089 €</ion-col>

        </ion-row>
    </ion-card-content>
  </ion-card>


  <ion-card padding>
    <ion-card-title style = "font-size: 0.8em; color:#ffffff; font-
size: 1.2em; font-weight : bold; text-
align: justify;">Registration fees (participant) with accommodation inclu
de:</ion-card-title>
      <ion-list>
        <ion-item>
          <ion-icon name="checkmark-
circle" style = "color:#ffffff;" slot = "start"></ion-icon>
          <p style = "color:#ffffff; text-
align: justify;">Admission to all Scientific Sessions (3 days meetings) <
/p>
        </ion-item>
        <ion-item>
            <ion-icon name="checkmark-
circle" style = "color:#ffffff;" slot = "start"></ion-icon>
            <p style = "color:#ffffff; text-
align: justify;">Congress Bag (including volume of the Congress Abstract
CD)  </p>
          </ion-item>
        <ion-item>
            <ion-icon name="checkmark-
circle" style = "color:#ffffff;" slot = "start"></ion-icon>
            <p style = "color:#ffffff; text-
align: justify;">1 abstract for the conference Book of Abstracts (Up to 3
 abstracts per person is allowed for submission. 2nd and 3rd submissions
cost 90 € each)</p>
        </ion-item>
        <ion-item>
            <ion-icon name="checkmark-
circle" style = "color:#ffffff;" slot = "start"></ion-icon>
            <p style = "color:#ffffff; text-
align: justify;">Each submitted abstract can lead to one paper in the Pro
ceedings Book of the Conference, according to the registration payment. T
he main registration justifies publication of one paper up to 6 pages. Fo
r more than 6 pages, there exists an extra fee: 50 €/per page.</p>
        </ion-item>
        <ion-item>
            <ion-icon name="checkmark-circle" style = "color:#ffffff;" slot = "start"></ion-icon>
            <p style = "color:#ffffff; text-align: justify;">Coffee breaks during the Meetings and High class all-inclusive food and drinks (for more information please visit “Venue” page)</p>
        </ion-item>
        <ion-item>
            <ion-icon name="checkmark-circle" style = "color:#ffffff;" slot = "start"></ion-icon>
            <p style = "color:#ffffff; text-align: justify;">High class all-inclusive accommodation for 3 nights / 4 days (for more information please visit “Venue” page)</p>
        </ion-item>
        <ion-item>
            <ion-icon name="checkmark-circle" style = "color:#ffffff;" slot = "start"></ion-icon>
            <p style = "color:#ffffff; text-align: justify;">Welcome Cocktail (The event is an excellent opportunity to meet colleagues in a convivial atmosphere) </p>
        </ion-item>
        <ion-item>
            <ion-icon name="checkmark-circle" style = "color:#ffffff;" slot = "start"></ion-icon>
            <p style = "color:#ffffff; text-align: justify;">After Party (Only for participants staying in the Papillon Ayscha Hotel Resort & SPA) </p>
        </ion-item>

    </ion-list>
  </ion-card>

  <ion-card padding>
      <ion-card-title style = "font-size: 0.8em; color:#ffffff; font-size: 1.2em; font-weight : bold; text-align: justify;">Registration fees (participant) with no accommodation include:</ion-card-title>
      <ion-list>
        <ion-item>
          <ion-icon name="checkmark-circle" style = "color:#ffffff;" slot = "start"></ion-icon>
          <p style = "color:#ffffff; text-align: justify;">Congress Bag (including volume of the congress Abstract CD)</p>
        </ion-item>
        <ion-item>
```

```
            <ion-icon name="checkmark-
circle" style = "color:#ffffff;" slot = "start"></ion-icon>
            <p style = "color:#ffffff; text-
align: justify;">1 abstract for the conference Book of Abstracts (Up to 3
 abstracts per person is allowed for submission. 2nd and 3rd submissions
cost 90 € each)</p>
          </ion-item>
        <ion-item>
            <ion-icon name="checkmark-
circle" style = "color:#ffffff;" slot = "start"></ion-icon>
            <p style = "color:#ffffff; text-
align: justify;">Each submitted abstract can lead to one paper in the Pro
ceedings Book of the Conference, according to the registration payment. T
he main registration justifies publication of one paper up to 6 pages. Fo
r more than 6 pages, there exists an extra fee: 50 €/per page.</p>
        </ion-item>
        <ion-item>
            <ion-icon name="checkmark-
circle" style = "color:#ffffff;" slot = "start"></ion-icon>
            <p style = "color:#ffffff; text-
align: justify;">Coffee breaks during the Meetings and High class all-
inclusive food and drinks (between 9:00-
19:00) (for more information please visit “Venue” page)</p>
        </ion-item>
        <ion-item>
            <ion-icon name="checkmark-
circle" style = "color:#ffffff;" slot = "start"></ion-icon>
            <p style = "color:#ffffff; text-
align: justify;">Welcome Cocktail (The event is an excellent opportunity
to meet colleagues in a convivial atmosphere)</p>
        </ion-item>
        <ion-item>
            <ion-icon name="checkmark-
circle" style = "color:#ffffff;" slot = "start"></ion-icon>
            <p style = "color:#ffffff; text-
align: justify;">Accompanying person, registration fee (50 €/per day) is
only for guests not staying in the hotel. Includes: High class all-
inclusive food and drinks between 9:00-19:00</p>
        </ion-item>

    </ion-list>
  </ion-card>

  <ion-card padding>
    <ion-card-title style = "font-size: 0.8em; color:#ffffff; font-
size: 1.2em; font-weight : bold; text-
align: justify;">Direct Bank Transfer</ion-card-title>
    <ion-list>
      <ion-item>
```

```
        <ion-icon name="checkmark-circle" style = "color:#ffffff;" slot = "start"></ion-icon>
        <p style = "color:#ffffff; text-align: justify;">In case you select Direct Bank Transfer as payment method, our bank details are the following: </p>

      </ion-item>
      <ion-row  style = "color:#ffffff; font-size: 1.2em;">Bank : Bank<br/><br/> Account Number : 46216541654 <br/><br/> IBAN : GR18918618468462165416540</ion-row>
  </ion-list>

</ion-card>

</ion-content>
```

Registration-details.page.scss

```
ion-content{
    --ion-background-color:#0E1C95;
}
```

Home.page.ts

```
import { Component, OnInit, ViewChild } from '@angular/core';
import { ApiService } from '../api.service';
import { Observable } from 'rxjs';
import { Router } from '@angular/router';
import { ProductDetailsPage } from '../product-details/product-details.page';

@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
})

export class HomePage {
  products : any = [];

 constructor(public api:ApiService, private router: Router) {
  //this.api.get('products').subscribe((data) => {console.log((data)) });
  let data:Observable<any> = this.api.get('products');
  data.subscribe(result => {this.products = result, console.log(this.products)});
  }
  openProductPage(product: String) {
    this.router.navigate(['product-details', product]);

  }

}
```

Home.page.html

```
<ion-header>
  <ion-toolbar color = "dark">
    <ion-buttons slot="start">
      <ion-menu-button></ion-menu-button>
    </ion-buttons>
    <ion-title style ="color: #ffffff;">
      Home
    </ion-title>
  </ion-toolbar>
</ion-header>

<ion-content no-padding>
  <ion-card>
  <ion-slides [options] = " {loop:true, autoplay:true} " >
    <ion-slide *ngFor="let number of [1,2,3]">
      <img style="margin-
top: 10px;" src="./assets/images/{{number}}.jpg" />
    </ion-slide>
  </ion-slides>
</ion-card>

<ion-list>
    <ion-list-
header><h5 style ="color: #ffffff;">Available Products (VAT not included)
</h5></ion-list-header>
    <ion-item *ngFor = "let product of products">
      <ion-thumbnail slot="start">
          <img [src]="product.images[0].src"/>
      </ion-thumbnail>
      <ion-label text-
wrap> <h4 style ="color: #ffffff;">{{product.name}} <span [innerHTML]= "p
roduct.price" style ="color: #ffffff;"></span><span style ="color: #fffff
f;"> €</span></h4>
       <!--
 <p><span style ="color: #ffffff;" [innerHTML]= "product.description"></s
pan></p> --> </ion-label>
```

```
        <button ion-button icon clear item-right (click)=openProductPage(product.name)>
          <ion-icon name="arrow-forward"></ion-icon>
        </button>
    </ion-item>
</ion-list>

</ion-content>
```

Home.page.scss

```
ion-card {
  margin: 0;
  width: 100;
  height: 100;
}

ion-content{
  --ion-background-color:#0E1C95;
}
```

Products-by-category.page.ts

```
import { Component, OnInit } from '@angular/core';
import { ApiService } from '../api.service';
import { Observable } from 'rxjs';
import { ActivatedRoute, Router } from '@angular/router';

@Component({
  selector: 'app-products-by-category',
  templateUrl: './products-by-category.page.html',
  styleUrls: ['./products-by-category.page.scss'],
})
export class ProductsByCategoryPage implements OnInit {

  category: any;
  products: any [];
  ps: any [];

  constructor(public api:ApiService, private route: ActivatedRoute, private router: Router) {

  }
  ngOnInit() {

    this.category = this.route.snapshot.paramMap.get('category');
    let d:Observable<any> = this.api.get('products');
    d.subscribe(result => { this.products = [], this.ps = result; let j=0; console.log(this.category); for (let x of this.ps) { //filter fetched products by category
      if (x.categories[0].name === this.category) {
        this.products.push(x);
      }
    }});

  }

  openProductPage(product: String) {
    this.router.navigate(['product-details', product]);

  }

}
```

Products-by-category.page.html

```
<ion-header>
  <ion-toolbar color = "dark">
    <ion-buttons slot="start">
      <ion-menu-button></ion-menu-button>
    </ion-buttons>
    <ion-title style ="color: #ffffff;">
      Products by Category
    </ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>

    <ion-list>
        <ion-list-header><h2 style ="color: #ffffff;">24% VAT not included</h2></ion-list-header>
        <ion-item *ngFor = "let product of products">
          <ion-thumbnail slot="start">
              <img [src]="product.images[0].src"/>
          </ion-thumbnail>
          <ion-label text-wrap> <h4 style ="color: #ffffff;" >{{product.name}} <span style ="color: #ffffff;" [innerHTML]= "product.price"></span><span> €</span></h4>
```

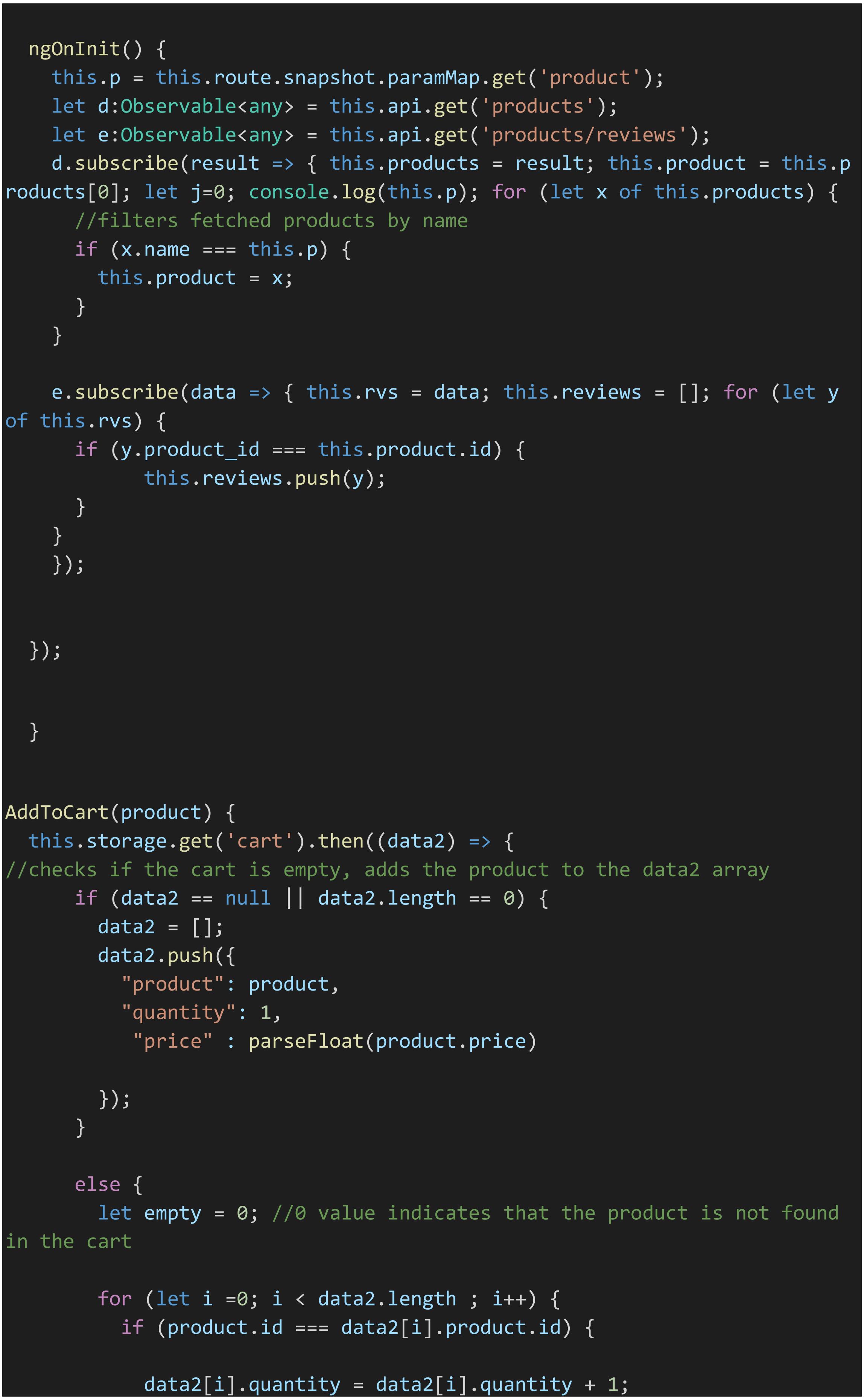
```
            <!--
<p><span style ="color: #ffffff;" [innerHTML]= "product.description"></sp
an></p>--> </ion-label>
            <button ion-button icon clear item-
right (click)="openProductPage(product.name)">
              <ion-icon name="arrow-forward"></ion-icon>
            </button>
        </ion-item>
    </ion-list>

</ion-content>
```

Products-by-category.scss

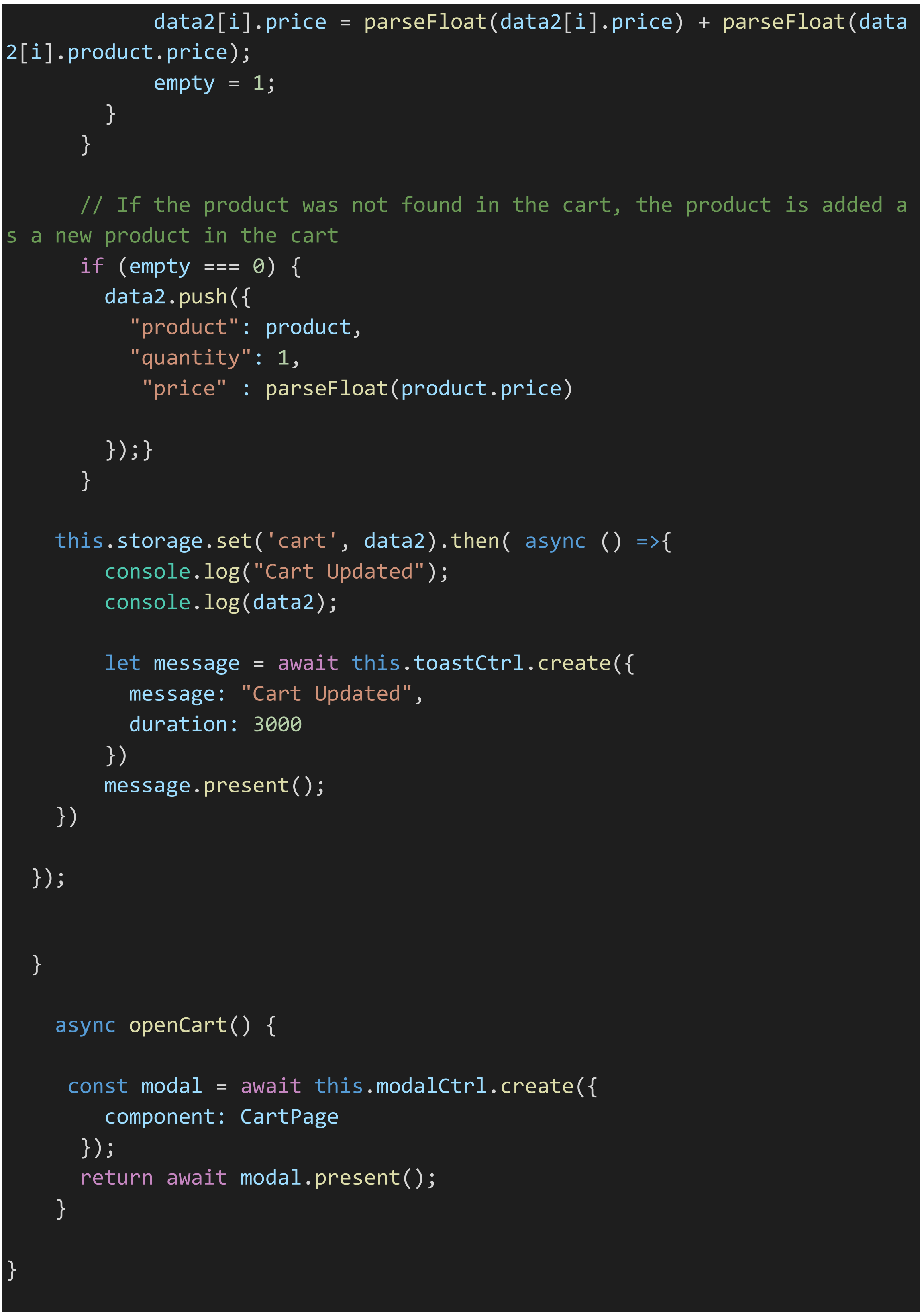
```
ion-content{
    --ion-background-color:#0E1C95;
  }
```

Product-details.page.ts

```
import { Component, OnInit } from '@angular/core';
import { ActivatedRoute } from '@angular/router';
import { ApiService } from '../api.service';
import { Observable } from 'rxjs';
import {Storage} from '@ionic/storage'
import { ToastController, ModalController } from '@ionic/angular';
import { CartPage } from '../cart/cart.page';

@Component({
  selector: 'app-product-details',
  templateUrl: './product-details.page.html',
  styleUrls: ['./product-details.page.scss'],
})
export class ProductDetailsPage implements OnInit {

  product : any;
  products : any[] ;
  p : any; //name of the selected product which is passed to the product
details
  reviews : any[] = [];
  rvs : any[];



  constructor(public api:ApiService, private route: ActivatedRoute, publi
c storage: Storage, public toastCtrl: ToastController, public modalCtrl:
ModalController) { }
```



```
 ngOnInit() {
  this.p = this.route.snapshot.paramMap.get('product');
  let d:Observable<any> = this.api.get('products');
  let e:Observable<any> = this.api.get('products/reviews');
  d.subscribe(result => { this.products = result; this.product = this.products[0]; let j=0; console.log(this.p); for (let x of this.products) {
   //filters fetched products by name
   if (x.name === this.p) {
    this.product = x;
   }
  }

  e.subscribe(data => { this.rvs = data; this.reviews = []; for (let y of this.rvs) {
   if (y.product_id === this.product.id) {
      this.reviews.push(y);
   }
  }
  });


 });


 }


AddToCart(product) {
 this.storage.get('cart').then((data2) => {
//checks if the cart is empty, adds the product to the data2 array
   if (data2 == null || data2.length == 0) {
    data2 = [];
    data2.push({
     "product": product,
     "quantity": 1,
      "price" : parseFloat(product.price)

    });
   }

   else {
    let empty = 0; //0 value indicates that the product is not found in the cart

    for (let i =0; i < data2.length ; i++) {
     if (product.id === data2[i].product.id) {

      data2[i].quantity = data2[i].quantity + 1;
```



```
            data2[i].price = parseFloat(data2[i].price) + parseFloat(data2[i].product.price);
            empty = 1;
        }
      }

      // If the product was not found in the cart, the product is added as a new product in the cart
      if (empty === 0) {
        data2.push({
          "product": product,
          "quantity": 1,
           "price" : parseFloat(product.price)

        });}
      }

    this.storage.set('cart', data2).then( async () =>{
        console.log("Cart Updated");
        console.log(data2);

        let message = await this.toastCtrl.create({
          message: "Cart Updated",
          duration: 3000
        })
        message.present();
    })

  });


  }

    async openCart() {

     const modal = await this.modalCtrl.create({
        component: CartPage
      });
      return await modal.present();
    }

}
```

Product-details.page.html

```
<ion-header>
    <ion-toolbar color = "dark">
      <ion-buttons slot="start">
        <ion-menu-button></ion-menu-button>
      </ion-buttons>
      <ion-title style ="color: #ffffff;">
        Product Details
      </ion-title>
    </ion-toolbar>
  </ion-header>


<ion-content>

  <ion-fab vertical="top" horizontal="end" slot="fixed" (click) = "openCart()">
      <ion-fab-button color = "warning">
          <ion-icon name="cart"></ion-icon>
      </ion-fab-button>
  </ion-fab>


  <ion-card *ngIf = "product">
      <img [src]="product.images[0].src"/>



    <ion-card-content>
      <ion-card-title style ="color: #ffffff;">
        {{product.name}}
        <ion-chip style="margin-left: 5px;"><ion-label style ="color: #ffffff;">{{product.categories[0].name}}</ion-label></ion-chip>
      </ion-card-title>

      <p style ="color: #ffffff;" [innerHTML]="product.description"></p>
      <ion-button color="success" icon-left (click) = "AddToCart(product)"> <ion-icon name="basket"></ion-icon> Add to Cart</ion-button>

    </ion-card-content>
  </ion-card>

  <ion-card *ngIf = "reviews.length > 0">
    <ion-card-content>
        <ion-card-title style ="color: #ffffff;">Reviews</ion-card-title>
       <ion-grid>
```

```
         <ion-row *ngFor = "let review of reviews">
          <ion-col col-4>
          <b style ="color: #ffffff;">{{review.reviewer}}</b><br/>
          <span *ngIf = "review.rating >=1"><ion-
icon style="color:#32CD32;" small name="star"></ion-icon></span>
          <span *ngIf = "review.rating >=2"><ion-
icon style="color:#32CD32;" small name="star"></ion-icon></span>
          <span *ngIf = "review.rating >=3"><ion-
icon style="color:#32CD32;" small name="star"></ion-icon></span>
          <span *ngIf = "review.rating >=4"><ion-
icon style="color:#32CD32;" small name="star"></ion-icon></span>
          <span *ngIf = "review.rating >=5"><ion-
icon style="color:#32CD32;" small name="star"></ion-icon></span>
        </ion-col>
          <ion-col col-8>
            <p style ="color: #ffffff;" [innerHTML] = "review.review" tet
xt-wrap></p>
          </ion-col>
          </ion-row>
      </ion-grid>
    </ion-card-content>
  </ion-card>


</ion-content>
```

Product-details.page.scss

```
ion-content{
    --ion-background-color:#0E1C95;
  }
```

Cart.page.ts

```
import { Component, OnInit} from '@angular/core';
import { Storage } from '@ionic/storage';
import {ModalController, AlertController} from '@ionic/angular';
import { CheckoutPage } from '../checkout/checkout.page';

@Component({
  selector: 'app-cart',
  templateUrl: './cart.page.html',
  styleUrls: ['./cart.page.scss'],
})
export class CartPage implements OnInit {

  cartItems: any[] = [];
  total: any;
  emptyCart = false;

  constructor(public storage: Storage, private modalCtrl: ModalController, private modalCtrl2: ModalController, private alertCtrl: AlertController) {

    this.total = 0;
    this.storage.ready().then( () => {

      this.storage.get('cart').then( (data) => {
        this.cartItems = data;
        console.log(this.cartItems);

        if (this.cartItems.length > 0) { // if the cart is not empty, calculate the total price of products in the cart
          this.cartItems.forEach( (item, index) => {
            this.total = Math.round( this.total + (1.24*item.product.price * item.quantity))
          } )
        }
        if (this.cartItems.length === 0 ) {
          this.emptyCart = true;
        }
      })

  })

 }

  ngOnInit() {
  }

  removeCardItem(item:any, i:any) {
    let price = item.product.price;
    let quantity = item.quantity;
    this.cartItems.splice(i, 1); //remove 1 item starting from carditems[
i]
    this.storage.set("cart", this.cartItems).then( ()=> {
    if (this.cartItems.length > 0) {
        this.total = Math.round(this.total - 1.24*price * quantity);}
    if (this.cartItems.length === 0) {
      this.total = 0;
    }
    });

    if (this.cartItems.length === 0) {
      this.emptyCart = true;
      this.total = 0;
    }
  }



  closeModal() {
      this.modalCtrl.dismiss();
    }

  async checkout() {
    const modal = await this.modalCtrl2.create({
       component: CheckoutPage
     });
     return await modal.present();
   }

   async presentAlert() {
    const alert = await this.alertCtrl.create({
      header : "No Items in the Cart",
      message : "You have to add products to the cart before proceeding t
o checkout.",
      buttons : ['OK']
    });

    await alert.present();
  }

  changeQuantity(item: any, i: any, value: any) {
    let price = 0;
    let q = 0;
    q = item.quantity;
    if (value < 0 && item.quantity === 1) {
      return; }
    q = q + value;
    item.quantity = q; //Change quantity
    this.storage.set("cart", this.cartItems).then( ()=> {
      for (let z=0; z<this.cartItems.length; z++) {
        if (this.cartItems[z] === item) {
          this.cartItems === item;
        }
      }
      });

    this.storage.ready().then( () => {
      this.storage.get('cart').then( (data) => {
        this.total = 0;
        this.cartItems = data;
        console.log(this.cartItems);

        if (this.cartItems.length > 0) { // if the cart is not empty, cal
culate the total price of products in the cart
          this.cartItems.forEach( (item, index) => {
            this.total = Math.round( this.total + (1.24*item.product.pric
e * item.quantity))
          } )
        }
        if (this.cartItems.length === 0 ) {
          this.emptyCart = true;
        }
      })
    })

  }


}
```

Cart.page.html

```
<ion-header>
    <ion-toolbar color = "dark">
      <ion-title style ="color: #ffffff;">
        Cart
      </ion-title>
    </ion-toolbar>
  </ion-header>

<ion-content>

  <ion-card [hidden] = "!emptyCart">
    <ion-grid>
```

```
      <ion-row ><ion-
col style ="color: #ffffff;">There are no products in the cart!</ion-
col></ion-row>
    </ion-grid>
  </ion-card>


    <ion-card *ngFor = "let item of cartItems; let i = index">
      <ion-item color = "light">
          <ion-thumbnail slot="start">
              <img [src]="item.product.images[0].src" style = "width :50p
x !important; height: 50px !important;  " />
          </ion-thumbnail>
          <div>
          <p><strong>{{item.product.name}}</strong></p>
          <p>{{item.quantity}} x {{item.product.price}} x 24% </p>
          </div>

          <ion-fab  horizontal="end" (click) = "removeCardItem(item, i)">
              <ion-fab-
button color = "danger" style = "height :40px; width : 40px">
                <ion-icon name="remove-circle"></ion-icon>
              </ion-fab-button>
          </ion-fab>

      </ion-item>

      <ion-item>
          <ion-row padding>
            <ion-col col-6> <ion-label style ="color: #ffffff; text-
align: center;">Quantity</ion-label> <br/>
            <ion-
button color="success" (click) = "changeQuantity(item, i , -1)" ><ion-
icon name="remove-circle"></ion-icon></ion-button>
            <ion-button color="dark">{{item.quantity}}</ion-button>
            <ion-
button color="success" (click) = "changeQuantity(item, i , 1)"><ion-
icon name="add-circle" ></ion-icon></ion-button>
           </ion-col>

          </ion-row>

      </ion-item>

    </ion-card>

    <ion-grid>
      <ion-card color="light">
        <ion-grid>
```

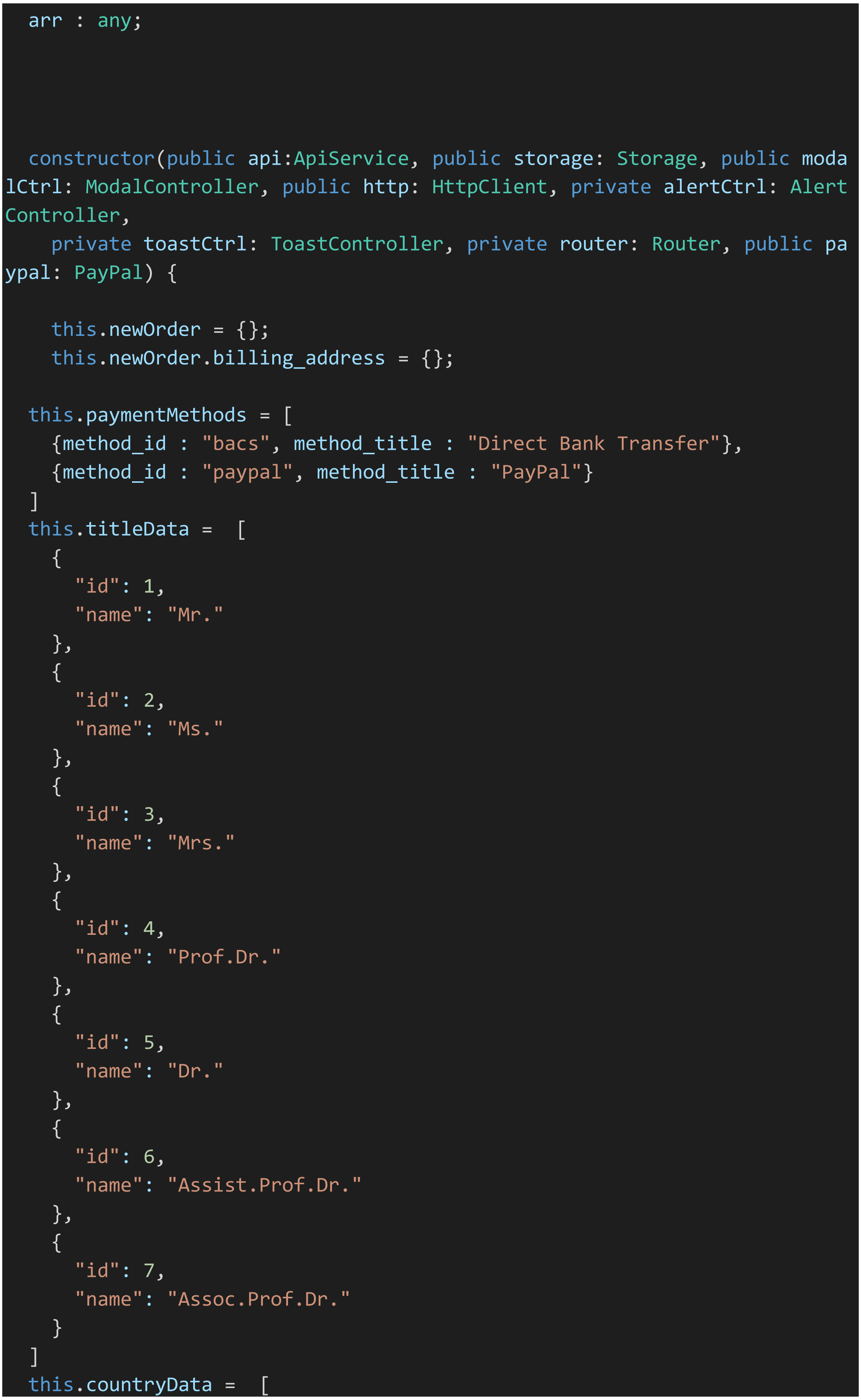
```
          <ion-row>
            <ion-col col-2>
            </ion-col>
            <ion-col col-4><strong>TOTAL</strong></ion-col>
            <ion-col col-3>VAT included</ion-col>
            <ion-col col-3><strong>{{total}}</strong> €</ion-col>
          </ion-row>
        </ion-grid>
      </ion-card>
    </ion-grid>

</ion-content>

<ion-footer>
  <ion-toolbar>
    <ion-grid>
      <ion-row>
        <ion-col col-2></ion-col>
        <ion-col col-4>
          <ion-button color="danger" (click)= "closeModal()">Back</ion-button>
        </ion-col>
        <ion-col col-4>
            <div *ngIf="cartItems.length > 0; then proceed else error"></div>

            <ng-template #proceed><ion-button color="success" (click) = "checkout()">Checkout</ion-button></ng-template>
            <ng-template #error><ion-button color="success" (click) = "presentAlert()">Checkout</ion-button></ng-template>

        </ion-col>
        <ion-col col-2></ion-col>
      </ion-row>
    </ion-grid>
  </ion-toolbar>
</ion-footer>
```

Cart.page.scss

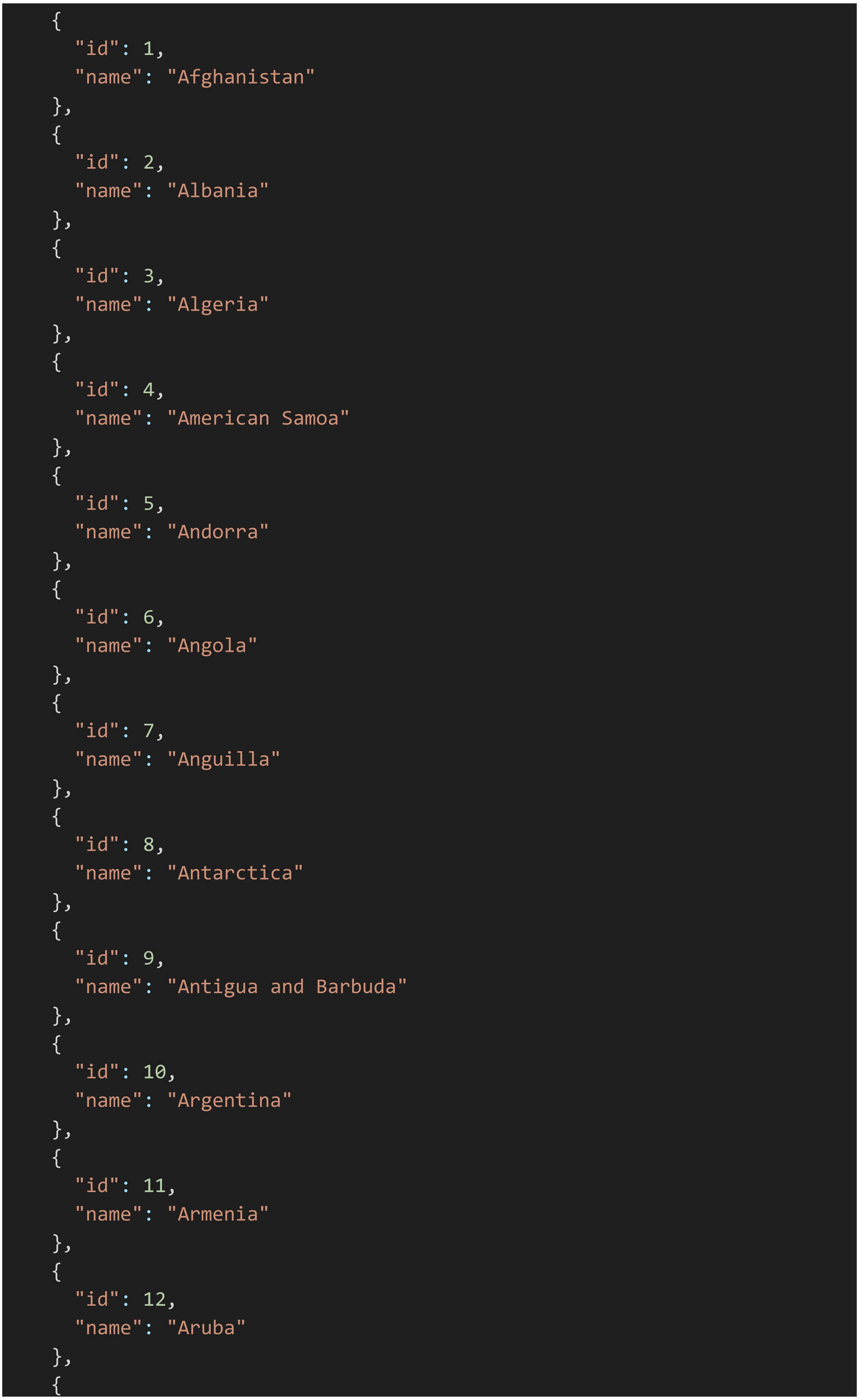
```
ion-content{
    --ion-background-color:#0E1C95;
  }
```

Checkout.page.ts

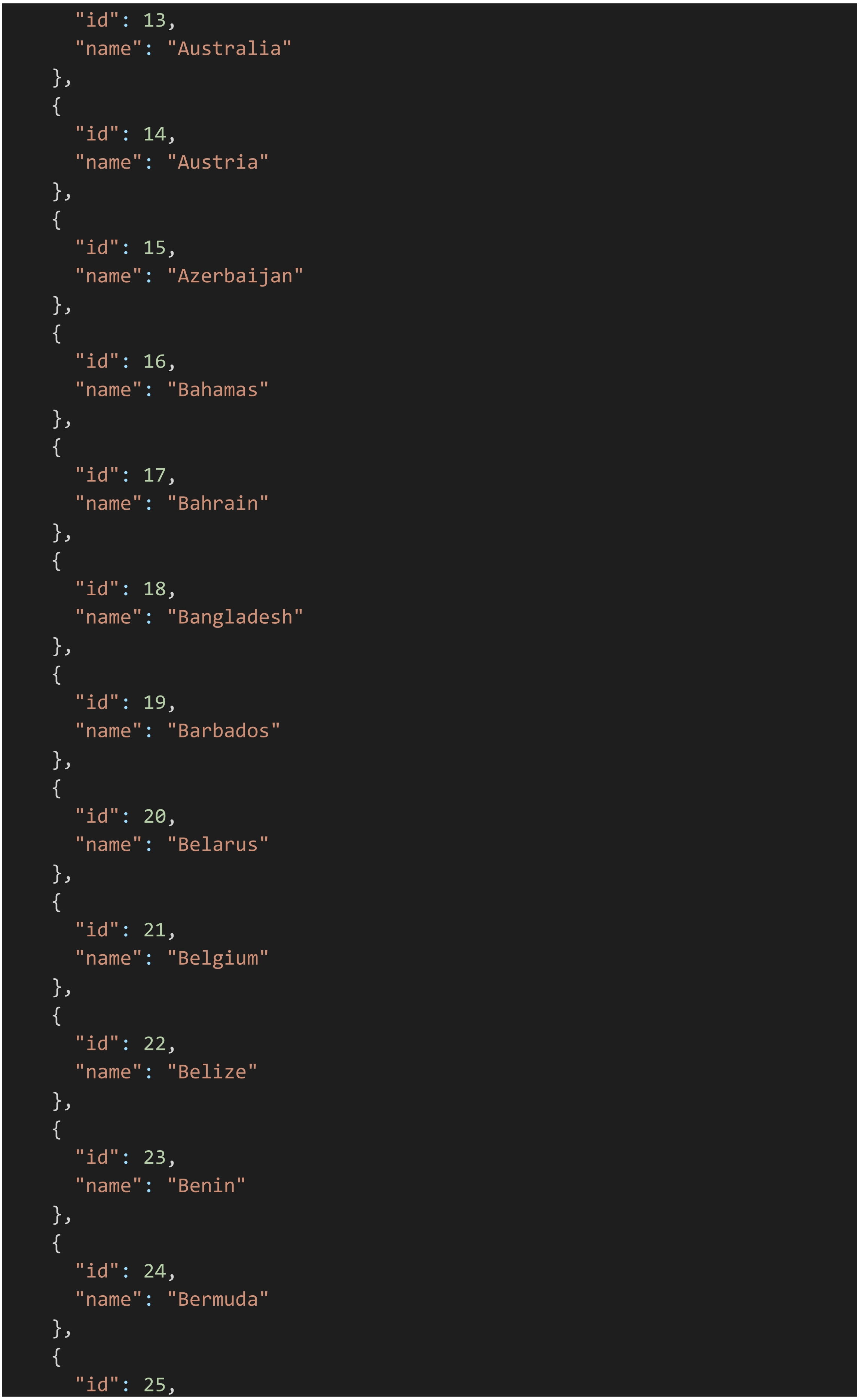
```
import { Component, OnInit } from '@angular/core';
import { ApiService } from '../api.service';
import { Storage } from '@ionic/storage';
import { ModalController, AlertController, ToastController } from '@ionic
/angular';
import { HttpClient, HttpHeaders } from '@angular/common/http';
import { Router } from '@angular/router';
import { PayPal, PayPalPayment, PayPalConfiguration } from '@ionic-
native/paypal/ngx'


@Component({
  selector: 'app-checkout',
  templateUrl: './checkout.page.html',
  styleUrls: ['./checkout.page.scss'],
})
export class CheckoutPage implements OnInit {

  cin: String;
  newOrder: any;
  paymentMethods : any[];
  countryData : any[];
  country : any;
  paymentMethod : any;
  titleData : any[];
  title : any;
  meta_data1 : any[] = [];
  orderItems: any[] = [];
  temp : any[] = [];
  data3 : any;
  n: String;
  total_tax : any;
  flag = 0;
  orderNumber : any;
  paymentData: any = {};
  fn :any;
  ln : any;
  ad1 : any;
  ad2: any;
  ct : any;
  st : any;
  pc : any;
  ctr : any;
  em : any;
  ph :any;
  co : any;
  customer_notes : any;
```



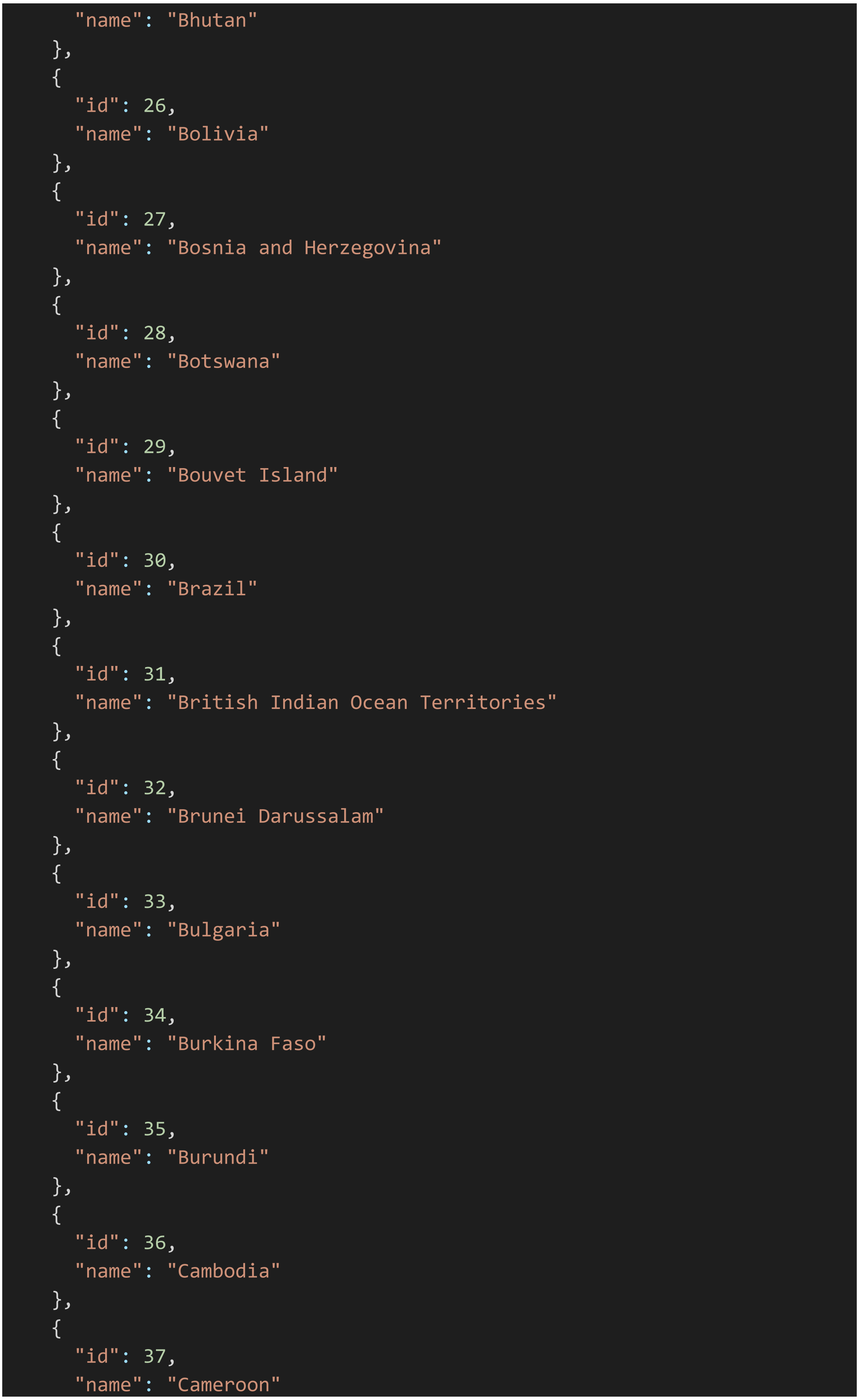
```
  arr : any;




  constructor(public api:ApiService, public storage: Storage, public modalCtrl: ModalController, public http: HttpClient, private alertCtrl: AlertController,
    private toastCtrl: ToastController, private router: Router, public paypal: PayPal) {

    this.newOrder = {};
    this.newOrder.billing_address = {};

  this.paymentMethods = [
    {method_id : "bacs", method_title : "Direct Bank Transfer"},
    {method_id : "paypal", method_title : "PayPal"}
  ]
  this.titleData =  [
    {
      "id": 1,
      "name": "Mr."
    },
    {
      "id": 2,
      "name": "Ms."
    },
    {
      "id": 3,
      "name": "Mrs."
    },
    {
      "id": 4,
      "name": "Prof.Dr."
    },
    {
      "id": 5,
      "name": "Dr."
    },
    {
      "id": 6,
      "name": "Assist.Prof.Dr."
    },
    {
      "id": 7,
      "name": "Assoc.Prof.Dr."
    }
  ]
  this.countryData =  [
```



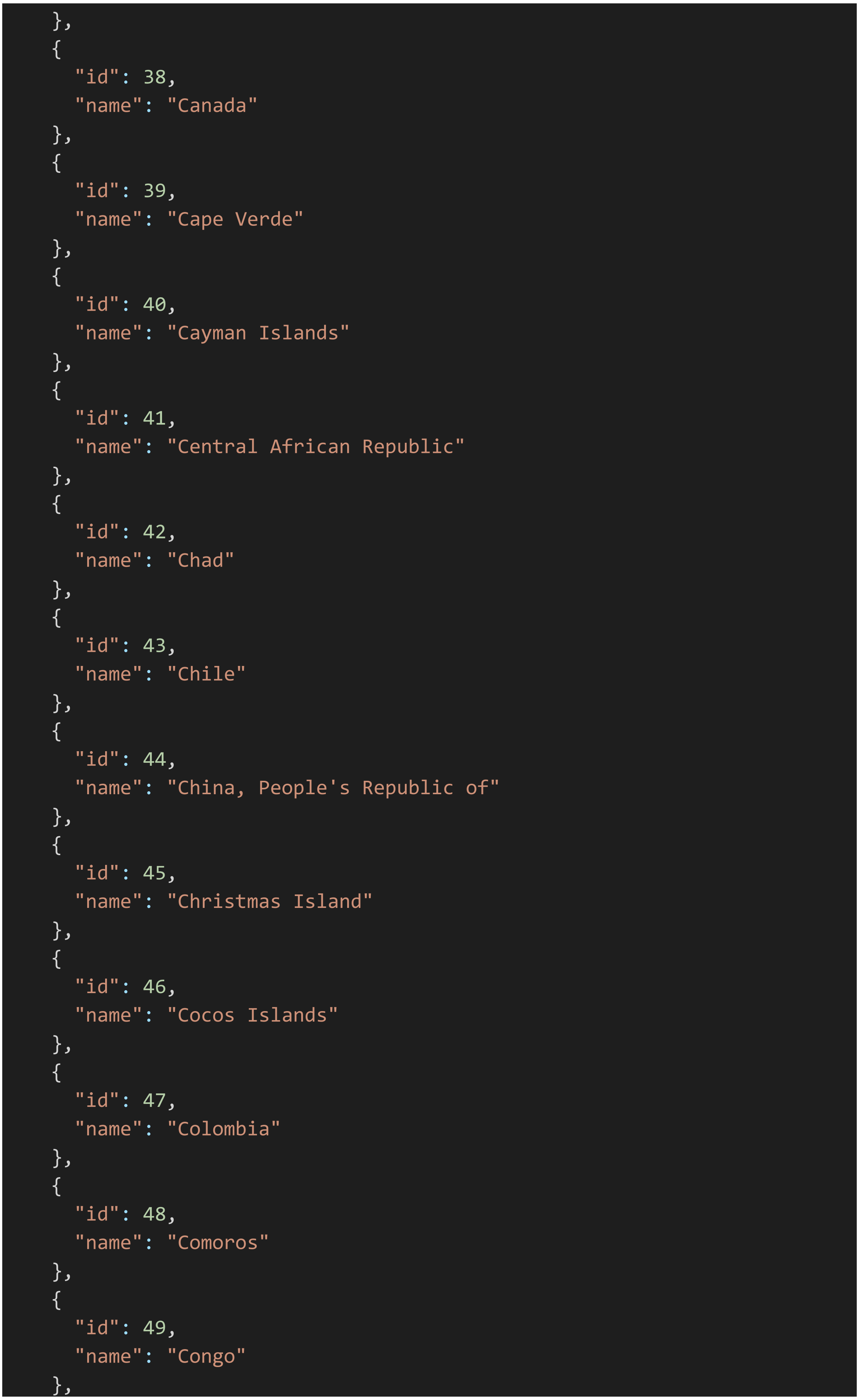
```
    {
      "id": 1,
      "name": "Afghanistan"
    },
    {
      "id": 2,
      "name": "Albania"
    },
    {
      "id": 3,
      "name": "Algeria"
    },
    {
      "id": 4,
      "name": "American Samoa"
    },
    {
      "id": 5,
      "name": "Andorra"
    },
    {
      "id": 6,
      "name": "Angola"
    },
    {
      "id": 7,
      "name": "Anguilla"
    },
    {
      "id": 8,
      "name": "Antarctica"
    },
    {
      "id": 9,
      "name": "Antigua and Barbuda"
    },
    {
      "id": 10,
      "name": "Argentina"
    },
    {
      "id": 11,
      "name": "Armenia"
    },
    {
      "id": 12,
      "name": "Aruba"
    },
    {
```



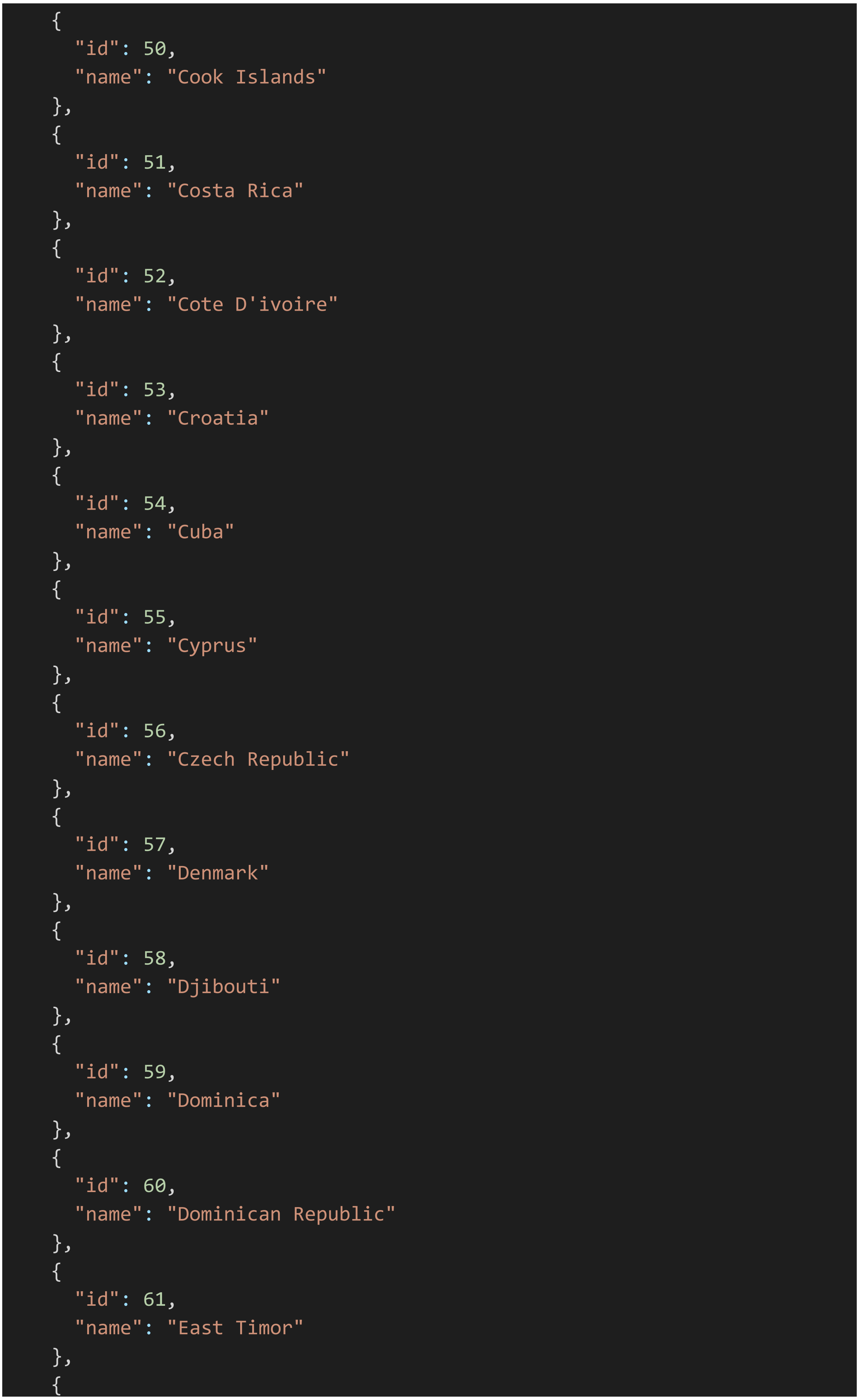
```
      "id": 13,
      "name": "Australia"
    },
    {
      "id": 14,
      "name": "Austria"
    },
    {
      "id": 15,
      "name": "Azerbaijan"
    },
    {
      "id": 16,
      "name": "Bahamas"
    },
    {
      "id": 17,
      "name": "Bahrain"
    },
    {
      "id": 18,
      "name": "Bangladesh"
    },
    {
      "id": 19,
      "name": "Barbados"
    },
    {
      "id": 20,
      "name": "Belarus"
    },
    {
      "id": 21,
      "name": "Belgium"
    },
    {
      "id": 22,
      "name": "Belize"
    },
    {
      "id": 23,
      "name": "Benin"
    },
    {
      "id": 24,
      "name": "Bermuda"
    },
    {
      "id": 25,
```



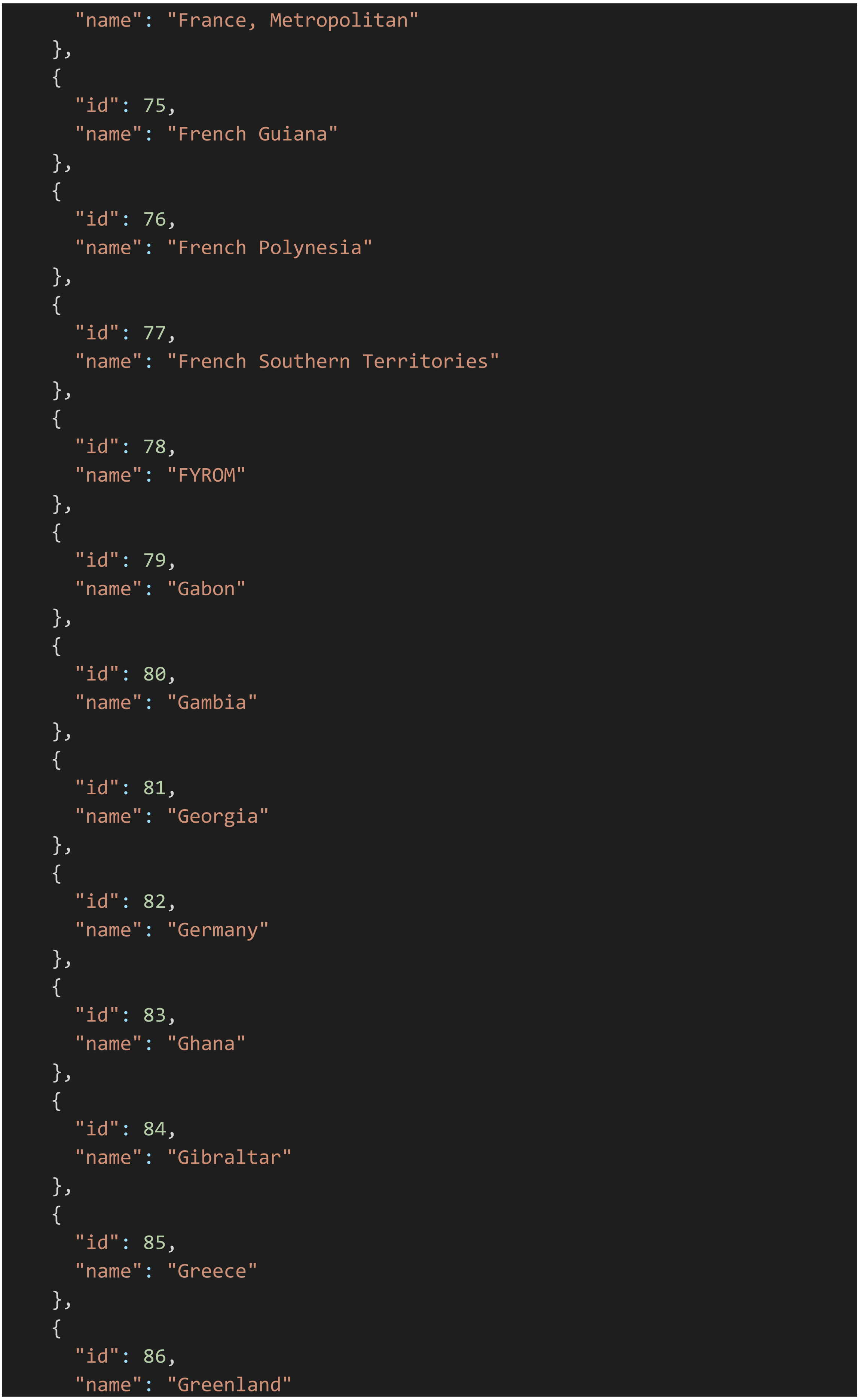
```
      "name": "Bhutan"
    },
    {
      "id": 26,
      "name": "Bolivia"
    },
    {
      "id": 27,
      "name": "Bosnia and Herzegovina"
    },
    {
      "id": 28,
      "name": "Botswana"
    },
    {
      "id": 29,
      "name": "Bouvet Island"
    },
    {
      "id": 30,
      "name": "Brazil"
    },
    {
      "id": 31,
      "name": "British Indian Ocean Territories"
    },
    {
      "id": 32,
      "name": "Brunei Darussalam"
    },
    {
      "id": 33,
      "name": "Bulgaria"
    },
    {
      "id": 34,
      "name": "Burkina Faso"
    },
    {
      "id": 35,
      "name": "Burundi"
    },
    {
      "id": 36,
      "name": "Cambodia"
    },
    {
      "id": 37,
      "name": "Cameroon"
```



```
    },
    {
      "id": 38,
      "name": "Canada"
    },
    {
      "id": 39,
      "name": "Cape Verde"
    },
    {
      "id": 40,
      "name": "Cayman Islands"
    },
    {
      "id": 41,
      "name": "Central African Republic"
    },
    {
      "id": 42,
      "name": "Chad"
    },
    {
      "id": 43,
      "name": "Chile"
    },
    {
      "id": 44,
      "name": "China, People's Republic of"
    },
    {
      "id": 45,
      "name": "Christmas Island"
    },
    {
      "id": 46,
      "name": "Cocos Islands"
    },
    {
      "id": 47,
      "name": "Colombia"
    },
    {
      "id": 48,
      "name": "Comoros"
    },
    {
      "id": 49,
      "name": "Congo"
    },
```

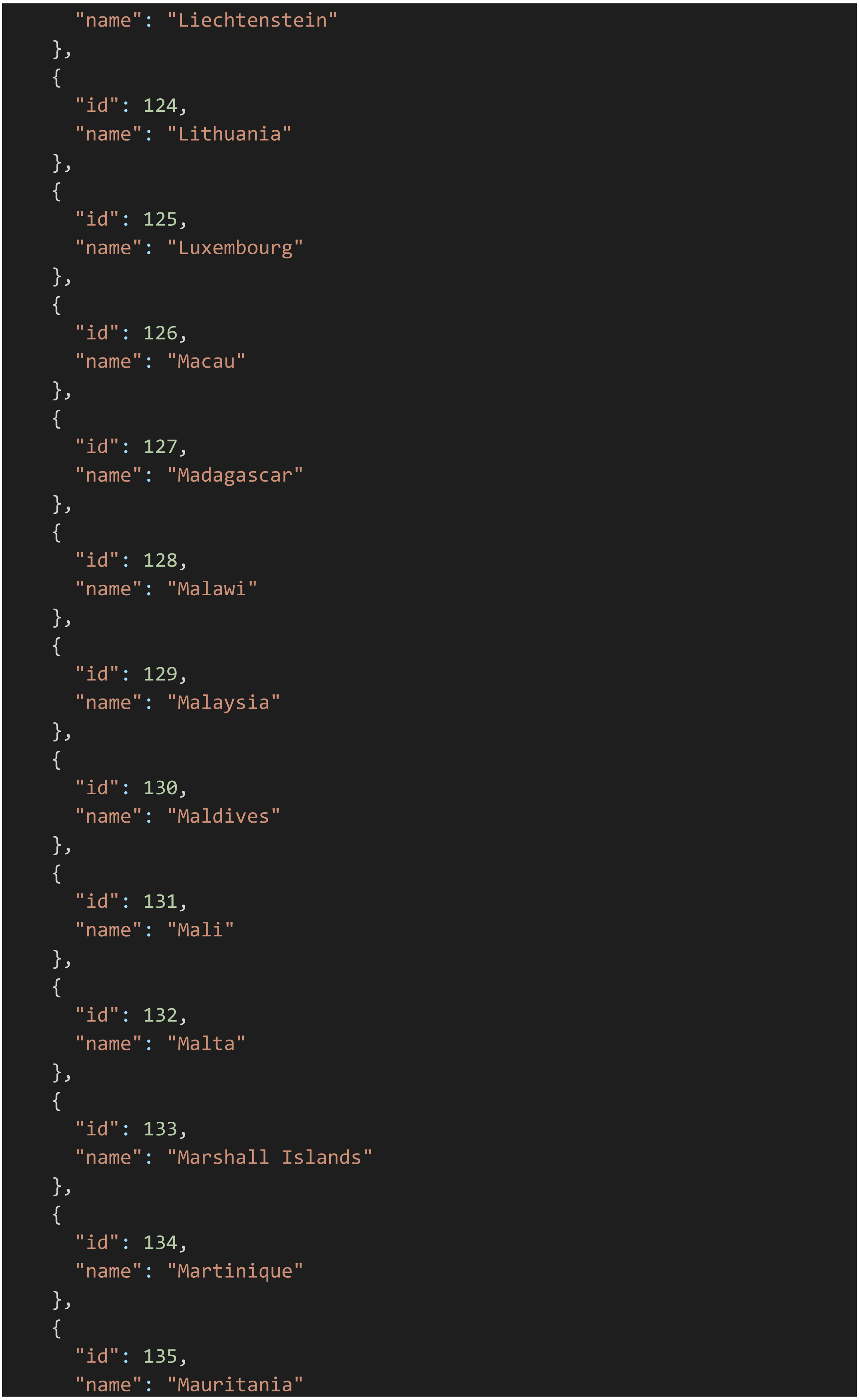


```
    {
      "id": 50,
      "name": "Cook Islands"
    },
    {
      "id": 51,
      "name": "Costa Rica"
    },
    {
      "id": 52,
      "name": "Cote D'ivoire"
    },
    {
      "id": 53,
      "name": "Croatia"
    },
    {
      "id": 54,
      "name": "Cuba"
    },
    {
      "id": 55,
      "name": "Cyprus"
    },
    {
      "id": 56,
      "name": "Czech Republic"
    },
    {
      "id": 57,
      "name": "Denmark"
    },
    {
      "id": 58,
      "name": "Djibouti"
    },
    {
      "id": 59,
      "name": "Dominica"
    },
    {
      "id": 60,
      "name": "Dominican Republic"
    },
    {
      "id": 61,
      "name": "East Timor"
    },
    {
      "id": 62,
      "name": "Ecuador"
    },
    {
      "id": 63,
      "name": "Egypt"
    },
    {
      "id": 64,
      "name": "El Salvador"
    },
    {
      "id": 65,
      "name": "Equatorial Guinea"
    },
    {
      "id": 66,
      "name": "Eritrea"
    },
    {
      "id": 67,
      "name": "Estonia"
    },
    {
      "id": 68,
      "name": "Ethiopia"
    },
    {
      "id": 69,
      "name": "Falkland Islands"
    },
    {
      "id": 70,
      "name": "Faroe Islands"
    },
    {
      "id": 71,
      "name": "Fiji"
    },
    {
      "id": 72,
      "name": "Finland"
    },
    {
      "id": 73,
      "name": "France"
    },
    {
      "id": 74,
```

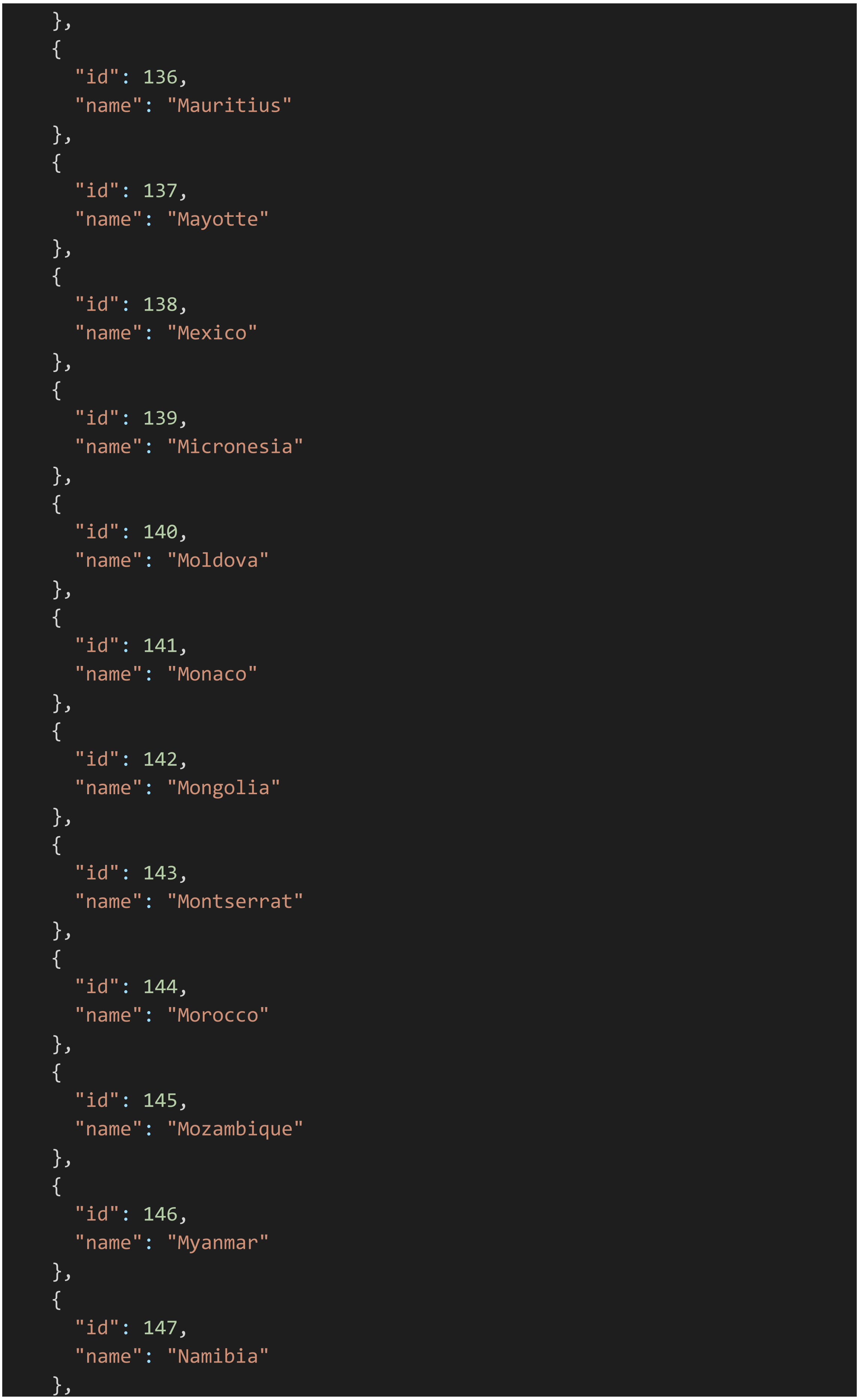


```
      "name": "France, Metropolitan"
    },
    {
      "id": 75,
      "name": "French Guiana"
    },
    {
      "id": 76,
      "name": "French Polynesia"
    },
    {
      "id": 77,
      "name": "French Southern Territories"
    },
    {
      "id": 78,
      "name": "FYROM"
    },
    {
      "id": 79,
      "name": "Gabon"
    },
    {
      "id": 80,
      "name": "Gambia"
    },
    {
      "id": 81,
      "name": "Georgia"
    },
    {
      "id": 82,
      "name": "Germany"
    },
    {
      "id": 83,
      "name": "Ghana"
    },
    {
      "id": 84,
      "name": "Gibraltar"
    },
    {
      "id": 85,
      "name": "Greece"
    },
    {
      "id": 86,
      "name": "Greenland"
```

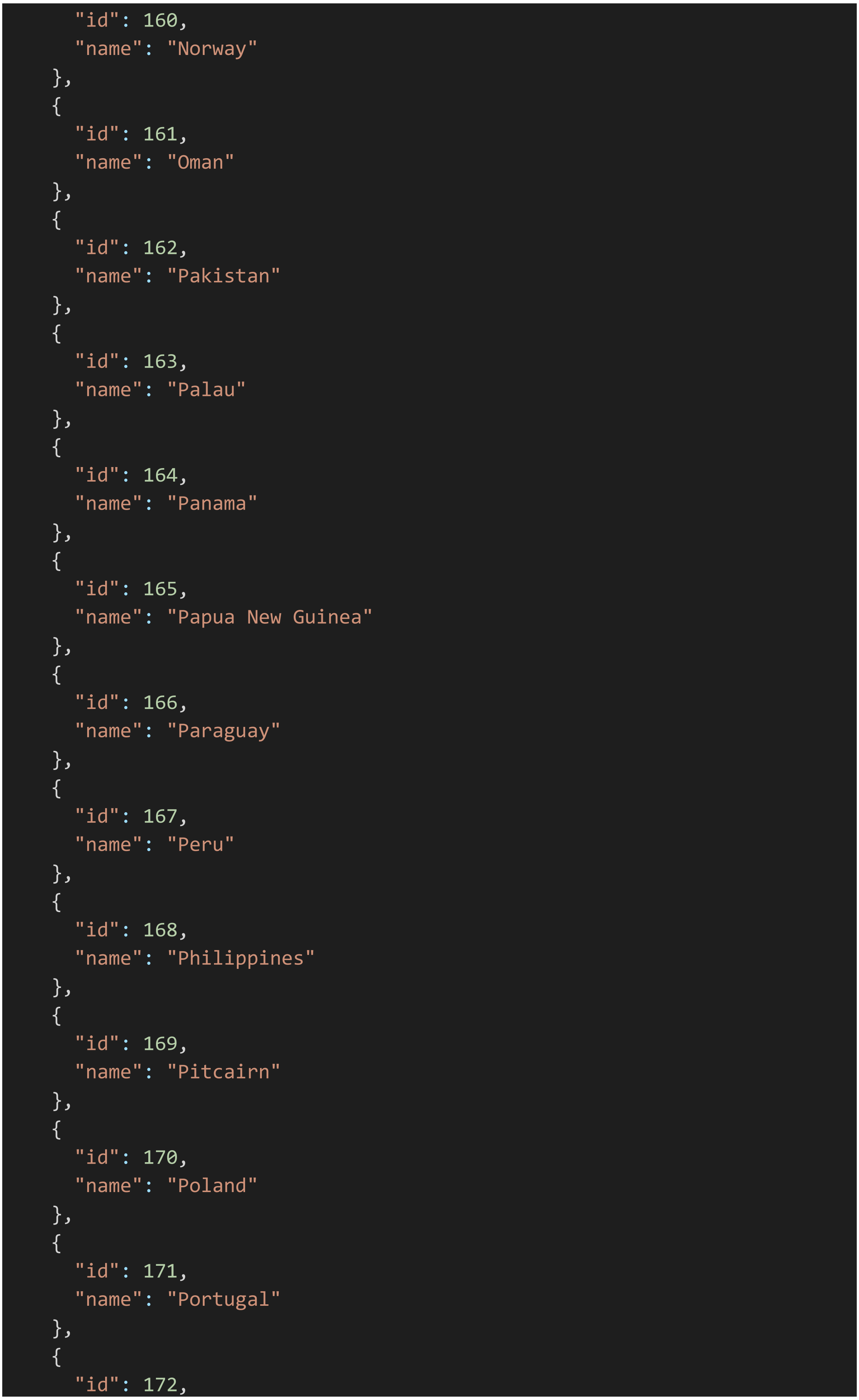
```
    },
    {
      "id": 87,
      "name": "Grenada"
    },
    {
      "id": 88,
      "name": "Guadeloupe"
    },
    {
      "id": 89,
      "name": "Guam"
    },
    {
      "id": 90,
      "name": "Guatemala"
    },
    {
      "id": 91,
      "name": "Guinea"
    },
    {
      "id": 92,
      "name": "Guinea-Bissau"
    },
    {
      "id": 93,
      "name": "Guyana"
    },
    {
      "id": 94,
      "name": "Haiti"
    },
    {
      "id": 95,
      "name": "Heard Island And Mcdonald Islands"
    },
    {
      "id": 96,
      "name": "Honduras"
    },
    {
      "id": 97,
      "name": "Hong Kong"
    },
    {
      "id": 98,
      "name": "Hungary"
    },
    {
      "id": 99,
      "name": "Iceland"
    },
    {
      "id": 100,
      "name": "India"
    },
    {
      "id": 101,
      "name": "Indonesia"
    },
    {
      "id": 102,
      "name": "Iran"
    },
    {
      "id": 103,
      "name": "Iraq"
    },
    {
      "id": 104,
      "name": "Ireland"
    },
    {
      "id": 105,
      "name": "Israel"
    },
    {
      "id": 106,
      "name": "Italy"
    },
    {
      "id": 107,
      "name": "Jamaica"
    },
    {
      "id": 108,
      "name": "Japan"
    },
    {
      "id": 109,
      "name": "Jordan"
    },
    {
      "id": 110,
      "name": "Kazakhstan"
    },
    {
      "id": 111,
      "name": "Kenya"
    },
    {
      "id": 112,
      "name": "Kiribati"
    },
    {
      "id": 113,
      "name": "North Korea"
    },
    {
      "id": 114,
      "name": "South Korea"
    },
    {
      "id": 115,
      "name": "Kuwait"
    },
    {
      "id": 116,
      "name": "Kyrgyzstan"
    },
    {
      "id": 117,
      "name": "Lao Peoples Democratic Republic"
    },
    {
      "id": 118,
      "name": "Latvia"
    },
    {
      "id": 119,
      "name": "Lebanon"
    },
    {
      "id": 120,
      "name": "Lesotho"
    },
    {
      "id": 121,
      "name": "Liberia"
    },
    {
      "id": 122,
      "name": "Libyan Arab Jamahiriya"
    },
    {
      "id": 123,
```



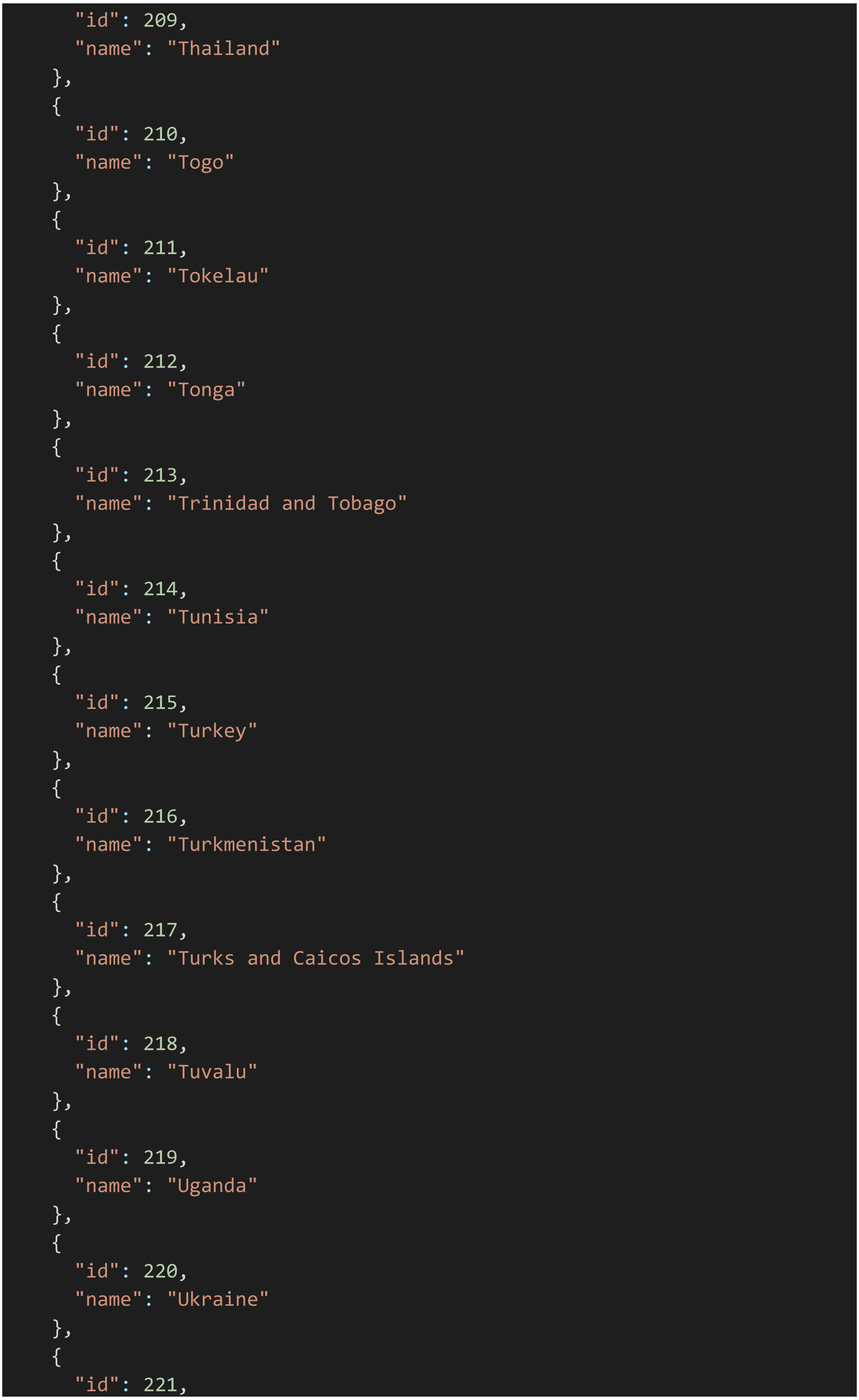
```
      "name": "Liechtenstein"
    },
    {
      "id": 124,
      "name": "Lithuania"
    },
    {
      "id": 125,
      "name": "Luxembourg"
    },
    {
      "id": 126,
      "name": "Macau"
    },
    {
      "id": 127,
      "name": "Madagascar"
    },
    {
      "id": 128,
      "name": "Malawi"
    },
    {
      "id": 129,
      "name": "Malaysia"
    },
    {
      "id": 130,
      "name": "Maldives"
    },
    {
      "id": 131,
      "name": "Mali"
    },
    {
      "id": 132,
      "name": "Malta"
    },
    {
      "id": 133,
      "name": "Marshall Islands"
    },
    {
      "id": 134,
      "name": "Martinique"
    },
    {
      "id": 135,
      "name": "Mauritania"
```



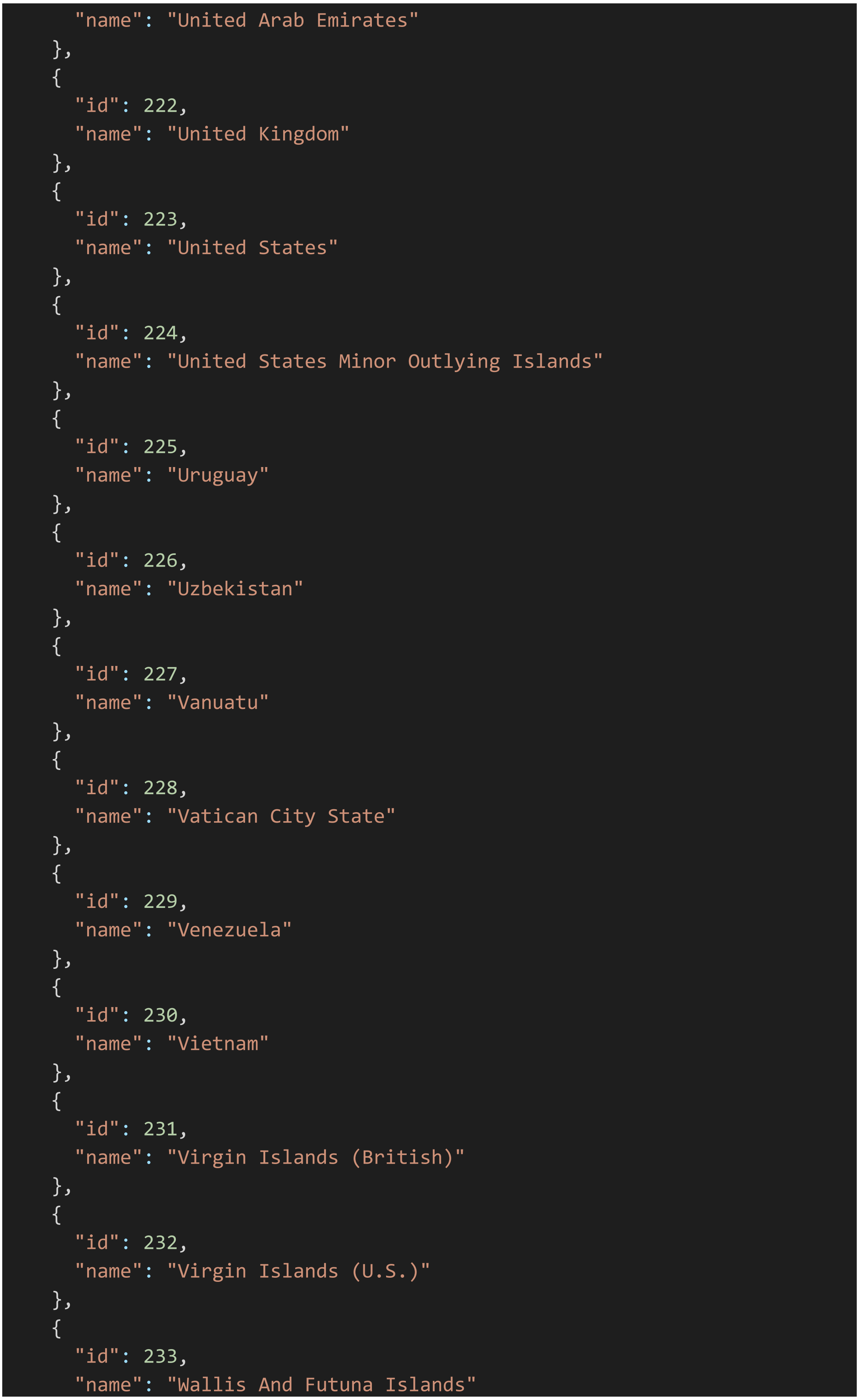
```
    },
    {
      "id": 136,
      "name": "Mauritius"
    },
    {
      "id": 137,
      "name": "Mayotte"
    },
    {
      "id": 138,
      "name": "Mexico"
    },
    {
      "id": 139,
      "name": "Micronesia"
    },
    {
      "id": 140,
      "name": "Moldova"
    },
    {
      "id": 141,
      "name": "Monaco"
    },
    {
      "id": 142,
      "name": "Mongolia"
    },
    {
      "id": 143,
      "name": "Montserrat"
    },
    {
      "id": 144,
      "name": "Morocco"
    },
    {
      "id": 145,
      "name": "Mozambique"
    },
    {
      "id": 146,
      "name": "Myanmar"
    },
    {
      "id": 147,
      "name": "Namibia"
    },
    {
      "id": 148,
      "name": "Nauru"
    },
    {
      "id": 149,
      "name": "Nepal"
    },
    {
      "id": 150,
      "name": "Netherlands"
    },
    {
      "id": 151,
      "name": "Netherlands Antilles"
    },
    {
      "id": 152,
      "name": "New Caledonia"
    },
    {
      "id": 153,
      "name": "New Zealand"
    },
    {
      "id": 154,
      "name": "Nicaragua"
    },
    {
      "id": 155,
      "name": "Niger"
    },
    {
      "id": 156,
      "name": "Nigeria"
    },
    {
      "id": 157,
      "name": "Niue"
    },
    {
      "id": 158,
      "name": "Norfolk Island"
    },
    {
      "id": 159,
      "name": "Northern Mariana Islands"
    },
    {
```



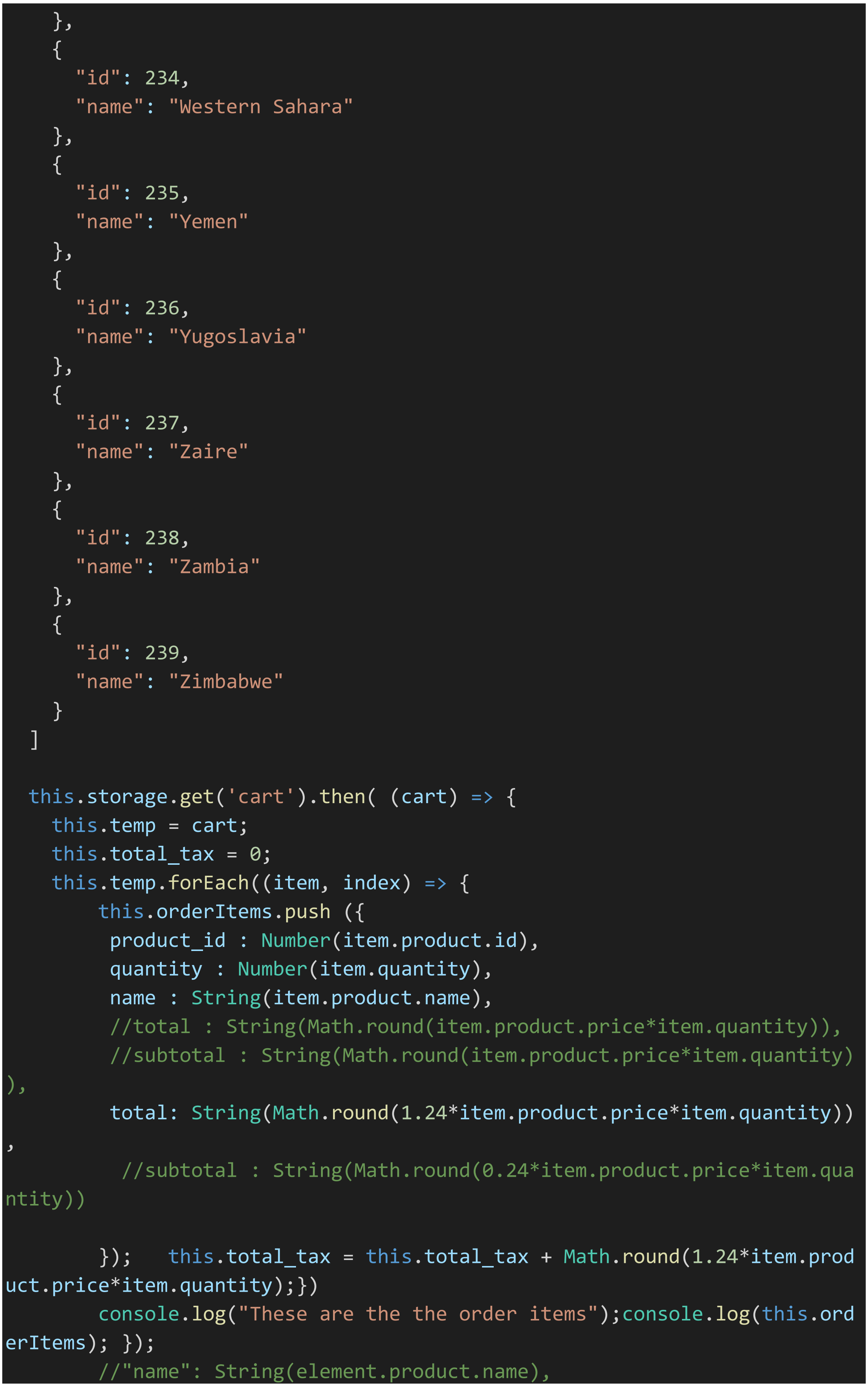
```
      "id": 160,
      "name": "Norway"
    },
    {
      "id": 161,
      "name": "Oman"
    },
    {
      "id": 162,
      "name": "Pakistan"
    },
    {
      "id": 163,
      "name": "Palau"
    },
    {
      "id": 164,
      "name": "Panama"
    },
    {
      "id": 165,
      "name": "Papua New Guinea"
    },
    {
      "id": 166,
      "name": "Paraguay"
    },
    {
      "id": 167,
      "name": "Peru"
    },
    {
      "id": 168,
      "name": "Philippines"
    },
    {
      "id": 169,
      "name": "Pitcairn"
    },
    {
      "id": 170,
      "name": "Poland"
    },
    {
      "id": 171,
      "name": "Portugal"
    },
    {
      "id": 172,
    "name": "Puerto Rico"
  },
  {
    "id": 173,
    "name": "Qatar"
  },
  {
    "id": 174,
    "name": "Reunion"
  },
  {
    "id": 175,
    "name": "Romania"
  },
  {
    "id": 176,
    "name": "Russian Federation"
  },
  {
    "id": 177,
    "name": "Rwanda"
  },
  {
    "id": 178,
    "name": "Saint Helena"
  },
  {
    "id": 179,
    "name": "Saint Kitts and Nevis"
  },
  {
    "id": 180,
    "name": "Saint Lucia"
  },
  {
    "id": 181,
    "name": "Saint Pierre and Miquelon"
  },
  {
    "id": 182,
    "name": "Saint Vincent and The Grenadines"
  },
  {
    "id": 183,
    "name": "Samoa"
  },
  {
    "id": 184,
    "name": "San Marino"
    },
    {
      "id": 185,
      "name": "Sao Tome and Principe"
    },
    {
      "id": 186,
      "name": "Saudi Arabia"
    },
    {
      "id": 187,
      "name": "Senegal"
    },
    {
      "id": 188,
      "name": "Seychelles"
    },
    {
      "id": 189,
      "name": "Sierra Leone"
    },
    {
      "id": 190,
      "name": "Singapore"
    },
    {
      "id": 191,
      "name": "Slovakia"
    },
    {
      "id": 192,
      "name": "Slovenia"
    },
    {
      "id": 193,
      "name": "Solomon Islands"
    },
    {
      "id": 194,
      "name": "Somalia"
    },
    {
      "id": 195,
      "name": "South Africa"
    },
    {
      "id": 196,
      "name": "South Georgia and Sandwich Islands"
    },
    {
      "id": 197,
      "name": "Spain"
    },
    {
      "id": 198,
      "name": "Sri Lanka"
    },
    {
      "id": 199,
      "name": "Sudan"
    },
    {
      "id": 200,
      "name": "Suriname"
    },
    {
      "id": 201,
      "name": "Svalbard and Jan Mayen"
    },
    {
      "id": 202,
      "name": "Swaziland"
    },
    {
      "id": 203,
      "name": "Sweden"
    },
    {
      "id": 204,
      "name": "Switzerland"
    },
    {
      "id": 205,
      "name": "Syrian Arab Republic"
    },
    {
      "id": 206,
      "name": "Taiwan"
    },
    {
      "id": 207,
      "name": "Tajikistan"
    },
    {
      "id": 208,
      "name": "Tanzania"
    },
    {
```



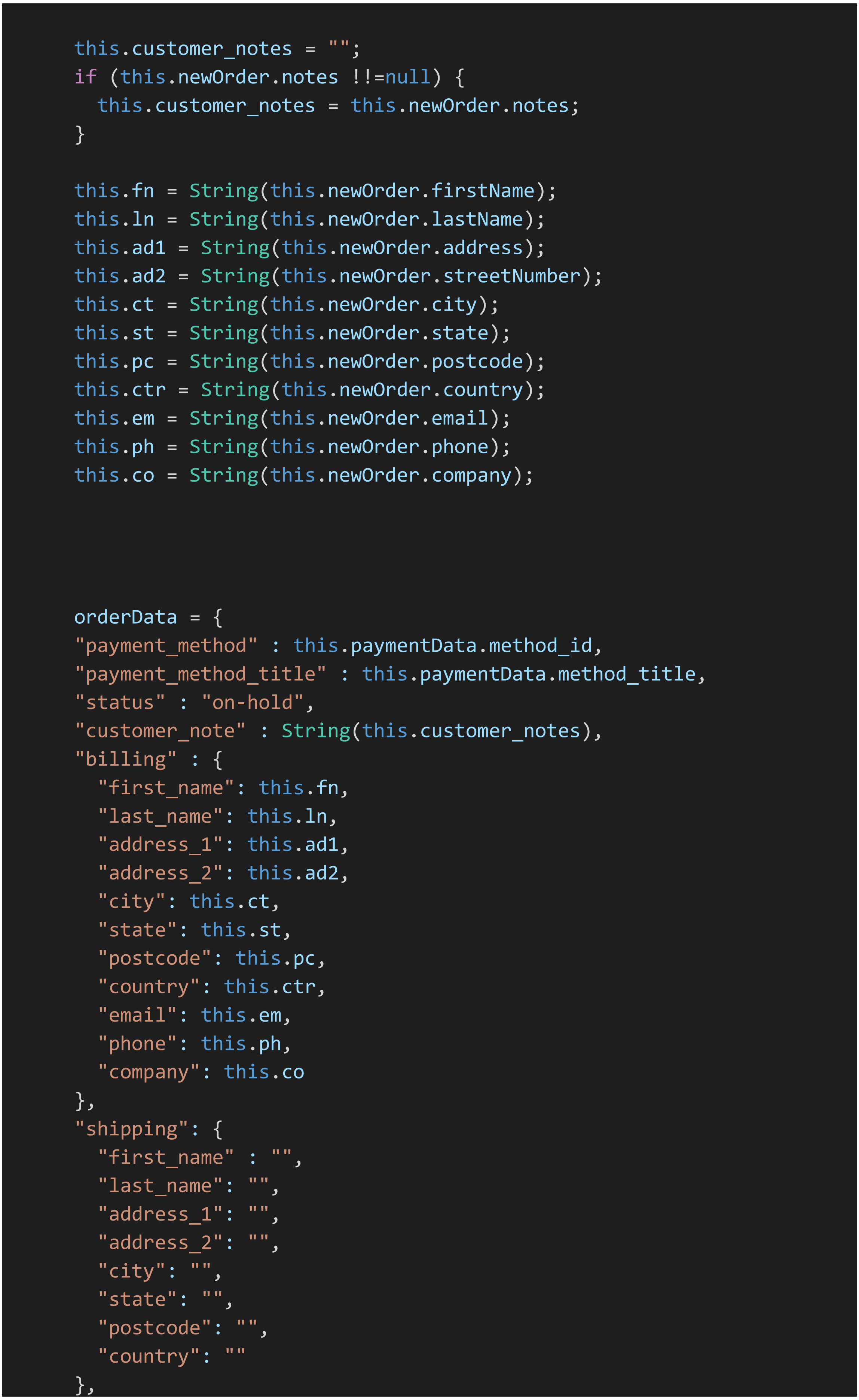
```
      "id": 209,
      "name": "Thailand"
    },
    {
      "id": 210,
      "name": "Togo"
    },
    {
      "id": 211,
      "name": "Tokelau"
    },
    {
      "id": 212,
      "name": "Tonga"
    },
    {
      "id": 213,
      "name": "Trinidad and Tobago"
    },
    {
      "id": 214,
      "name": "Tunisia"
    },
    {
      "id": 215,
      "name": "Turkey"
    },
    {
      "id": 216,
      "name": "Turkmenistan"
    },
    {
      "id": 217,
      "name": "Turks and Caicos Islands"
    },
    {
      "id": 218,
      "name": "Tuvalu"
    },
    {
      "id": 219,
      "name": "Uganda"
    },
    {
      "id": 220,
      "name": "Ukraine"
    },
    {
      "id": 221,
```



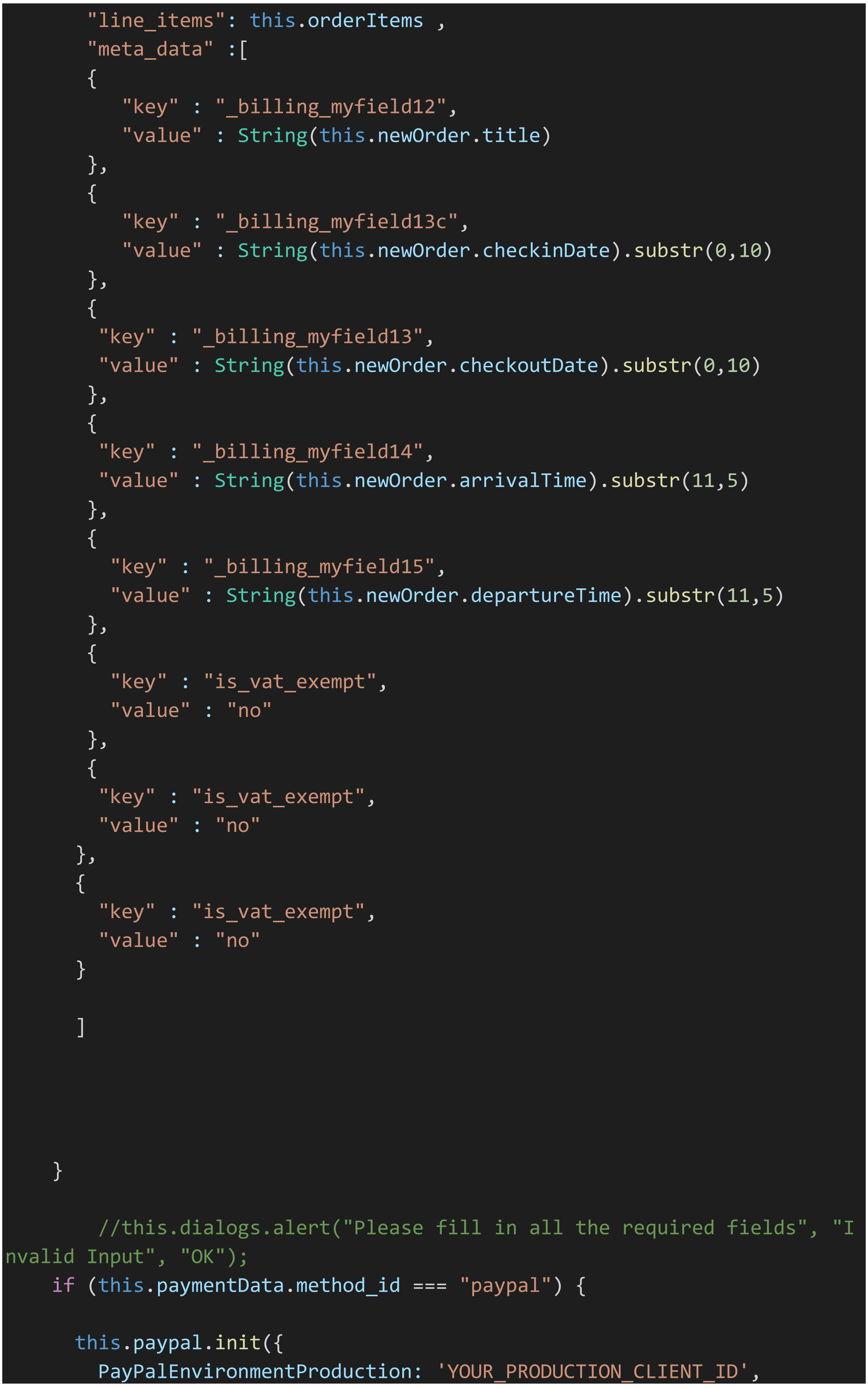
```
      "name": "United Arab Emirates"
    },
    {
      "id": 222,
      "name": "United Kingdom"
    },
    {
      "id": 223,
      "name": "United States"
    },
    {
      "id": 224,
      "name": "United States Minor Outlying Islands"
    },
    {
      "id": 225,
      "name": "Uruguay"
    },
    {
      "id": 226,
      "name": "Uzbekistan"
    },
    {
      "id": 227,
      "name": "Vanuatu"
    },
    {
      "id": 228,
      "name": "Vatican City State"
    },
    {
      "id": 229,
      "name": "Venezuela"
    },
    {
      "id": 230,
      "name": "Vietnam"
    },
    {
      "id": 231,
      "name": "Virgin Islands (British)"
    },
    {
      "id": 232,
      "name": "Virgin Islands (U.S.)"
    },
    {
      "id": 233,
      "name": "Wallis And Futuna Islands"
```



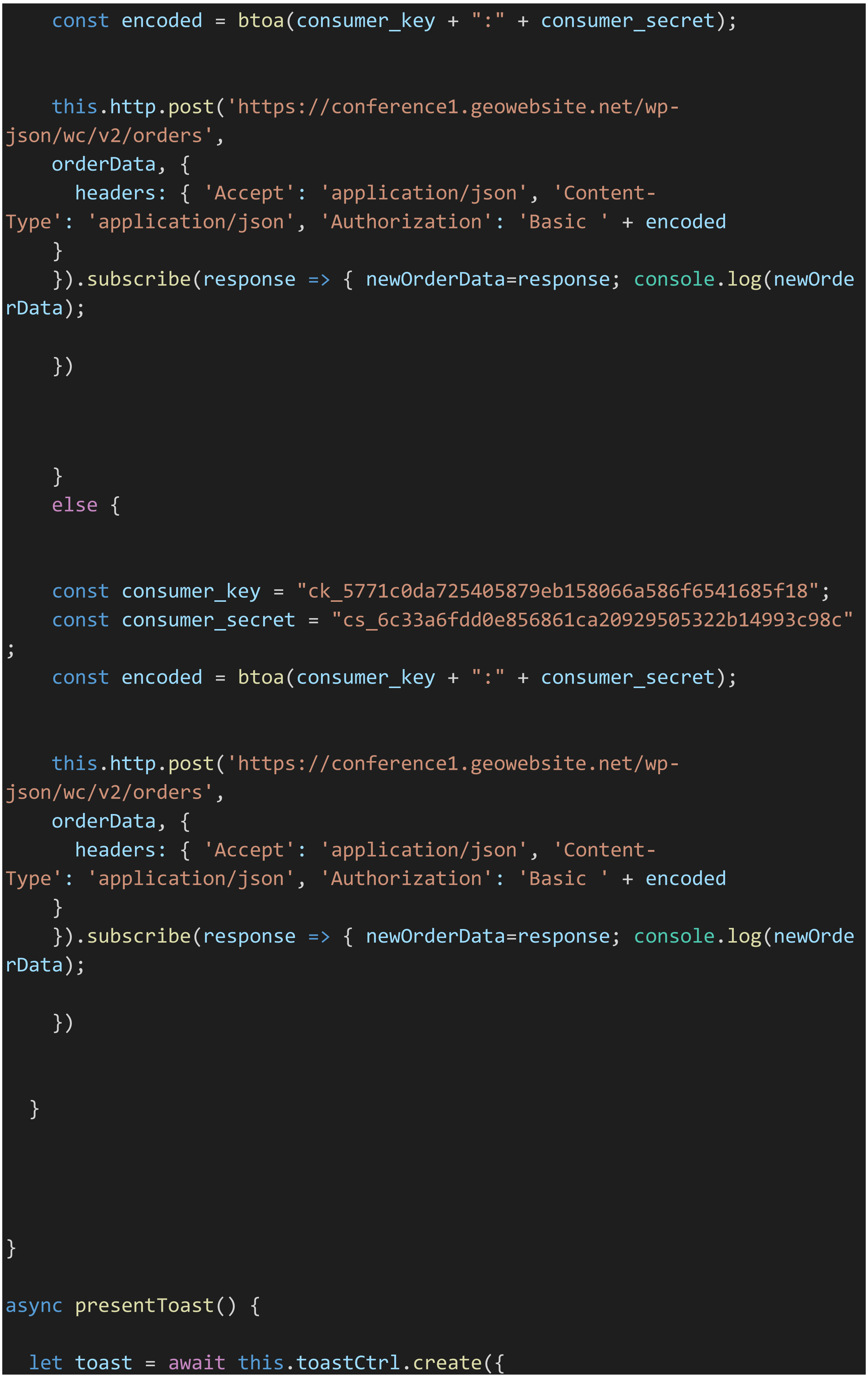
```
    },
    {
      "id": 234,
      "name": "Western Sahara"
    },
    {
      "id": 235,
      "name": "Yemen"
    },
    {
      "id": 236,
      "name": "Yugoslavia"
    },
    {
      "id": 237,
      "name": "Zaire"
    },
    {
      "id": 238,
      "name": "Zambia"
    },
    {
      "id": 239,
      "name": "Zimbabwe"
    }
  ]

  this.storage.get('cart').then( (cart) => {
    this.temp = cart;
    this.total_tax = 0;
    this.temp.forEach((item, index) => {
        this.orderItems.push ({
         product_id : Number(item.product.id),
         quantity : Number(item.quantity),
         name : String(item.product.name),
         //total : String(Math.round(item.product.price*item.quantity)),
         //subtotal : String(Math.round(item.product.price*item.quantity)
),
         total: String(Math.round(1.24*item.product.price*item.quantity))
,
          //subtotal : String(Math.round(0.24*item.product.price*item.qua
ntity))

        });   this.total_tax = this.total_tax + Math.round(1.24*item.prod
uct.price*item.quantity);})
        console.log("These are the the order items");console.log(this.ord
erItems); });
        //"name": String(element.product.name),
        //"total": String(Math.round(element.product.price*element.quantity)),
        //"total_tax": String(Math.round(0.24*element.product.price*element.quantity))


      this.http.get('https://conference1.geowebsite.net/wp-json/wc/v2/orders?consumer_key=ck_5771c0da725405879eb158066a586f6541685f18&consumer_secret=cs_6c33a6fdd0e856861ca20929505322b14993c98c').subscribe(res => {
        this.orderNumber = res[0].id; this.orderNumber = this.orderNumber + 1;
      })


 }

  ngOnInit() {


  }

  placeOrder() {

    let orderData: any = {};
    let newOrderData :any;
    this.flag = 0;


    //if(paymentData.method_id === 'paypal') {
      //TODO
    //}

    this.meta_data1 = [[{value: JSON.stringify(this.newOrder.title)}],
      [{value: JSON.stringify(this.newOrder.checkinDate)}],
      [{value: JSON.stringify(this.newOrder.checkoutDate)}],
      [{value: JSON.stringify(this.newOrder.arrivalTime)}],
      [{value: JSON.stringify(this.newOrder.departureTime)}]
      ]

      this.paymentData.method_id = "";
      this.paymentData.method_title = "";

      for (let pm of this.paymentMethods) {
        if (pm.method_title === this.newOrder.paymentMethod) {
          this.paymentData.method_id = pm.method_id;
          this.paymentData.method_title = pm.method_title;
        }
      }
```



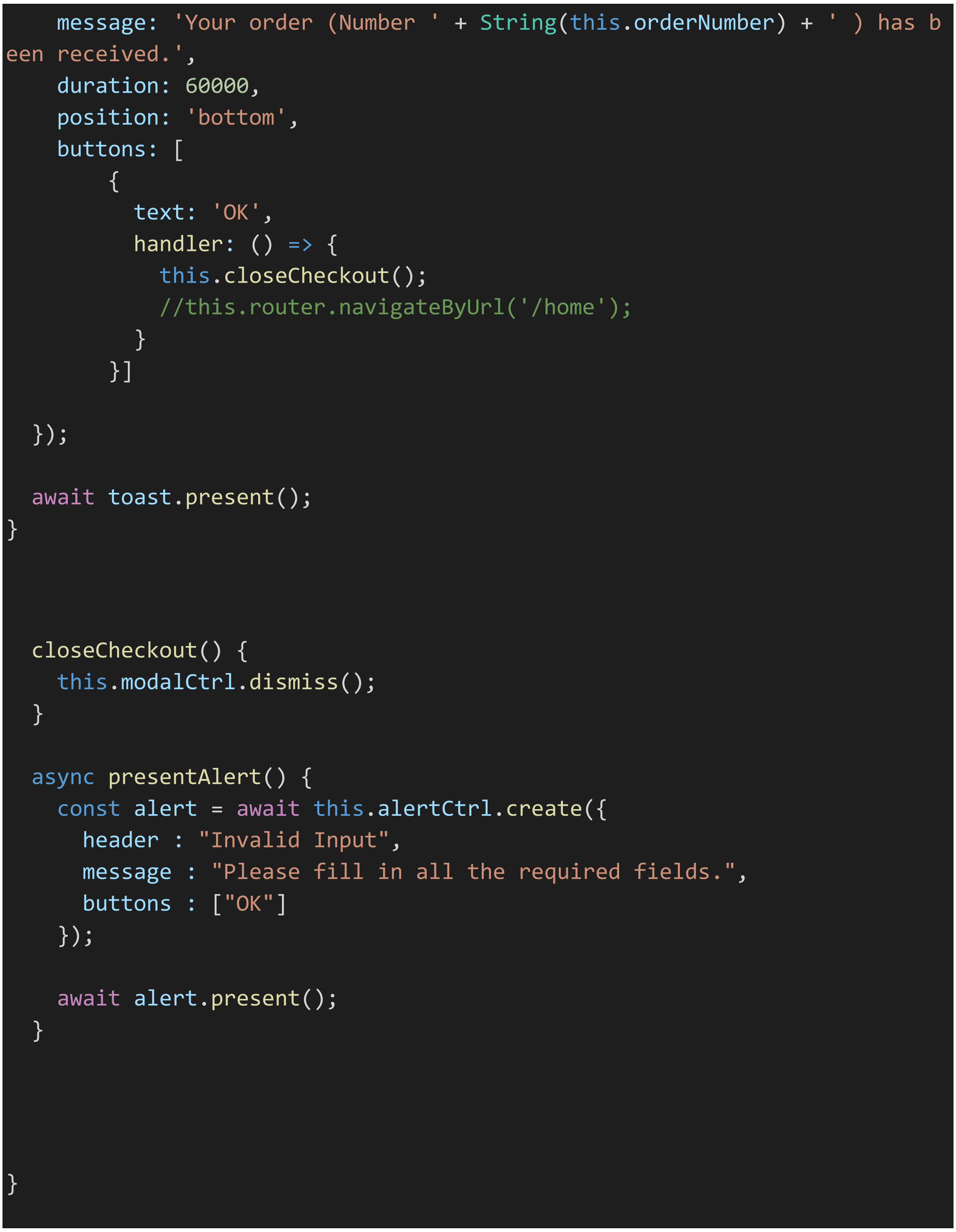
```
this.customer_notes = "";
if (this.newOrder.notes !!=null) {
  this.customer_notes = this.newOrder.notes;
}

this.fn = String(this.newOrder.firstName);
this.ln = String(this.newOrder.lastName);
this.ad1 = String(this.newOrder.address);
this.ad2 = String(this.newOrder.streetNumber);
this.ct = String(this.newOrder.city);
this.st = String(this.newOrder.state);
this.pc = String(this.newOrder.postcode);
this.ctr = String(this.newOrder.country);
this.em = String(this.newOrder.email);
this.ph = String(this.newOrder.phone);
this.co = String(this.newOrder.company);




orderData = {
"payment_method" : this.paymentData.method_id,
"payment_method_title" : this.paymentData.method_title,
"status" : "on-hold",
"customer_note" : String(this.customer_notes),
"billing" : {
  "first_name": this.fn,
  "last_name": this.ln,
  "address_1": this.ad1,
  "address_2": this.ad2,
  "city": this.ct,
  "state": this.st,
  "postcode": this.pc,
  "country": this.ctr,
  "email": this.em,
  "phone": this.ph,
  "company": this.co
},
"shipping": {
  "first_name" : "",
  "last_name": "",
  "address_1": "",
  "address_2": "",
  "city": "",
  "state": "",
  "postcode": "",
  "country": ""
},
```



```
       "line_items": this.orderItems ,
       "meta_data" :[
       {
          "key" : "_billing_myfield12",
          "value" : String(this.newOrder.title)
       },
       {
          "key" : "_billing_myfield13c",
          "value" : String(this.newOrder.checkinDate).substr(0,10)
       },
       {
        "key" : "_billing_myfield13",
        "value" : String(this.newOrder.checkoutDate).substr(0,10)
       },
       {
        "key" : "_billing_myfield14",
        "value" : String(this.newOrder.arrivalTime).substr(11,5)
       },
       {
         "key" : "_billing_myfield15",
         "value" : String(this.newOrder.departureTime).substr(11,5)
       },
       {
         "key" : "is_vat_exempt",
         "value" : "no"
       },
       {
        "key" : "is_vat_exempt",
        "value" : "no"
      },
      {
        "key" : "is_vat_exempt",
        "value" : "no"
      }

      ]




    }

        //this.dialogs.alert("Please fill in all the required fields", "I
nvalid Input", "OK");
    if (this.paymentData.method_id === "paypal") {

      this.paypal.init({
        PayPalEnvironmentProduction: 'YOUR_PRODUCTION_CLIENT_ID',
    PayPalEnvironmentSandbox: 'AfXcoKIBOIopqnVxKAv-
eiATANujnI8ICSXKPH3KZN5bsvSrpicIRPgthwDuxAfMc4nxvldZjyGLaxnS'
   }).then(() => {
    // Environments: PayPalEnvironmentNoNetwork, PayPalEnvironmentSan
dbox, PayPalEnvironmentProduction
    this.paypal.prepareToRender('PayPalEnvironmentSandbox', new PayPa
lConfiguration({
     // Only needed if you get an "Internal Service Error" after Pay
Pal login!
     //payPalShippingAddressOption: 2 // PayPalShippingAddressOption
PayPal
    })).then(() => {

     this.storage.get('cart').then( (cart) => {
      let total = 0;
      cart.forEach((element,index) => {
       this.orderItems.push({product_id : element.product.id, quan
tity : element.quantity});
       total = Math.round(total + 1.24*(element.product.price * el
ement.quantity));
      })

      let payment = new PayPalPayment(String(total), 'EUR', 'Descri
ption', 'sale');
      this.paypal.renderSinglePaymentUI(payment).then((response) =>
 {
       // Successfully paid

      //alert(JSON.stringify(response));

     })




     }, () => {
      // Error or render dialog closed without being successful
     });
    }, () => {
     // Error in configuration
    });
   }, () => {
    // Error in initialization, maybe PayPal isn't supported or somet
hing else
   });

  const consumer_key = "ck_5771c0da725405879eb158066a586f6541685f18";
  const consumer_secret = "cs_6c33a6fdd0e856861ca20929505322b14993c98c"
;
```



```
    const encoded = btoa(consumer_key + ":" + consumer_secret);


    this.http.post('https://conference1.geowebsite.net/wp-
json/wc/v2/orders',
    orderData, {
      headers: { 'Accept': 'application/json', 'Content-
Type': 'application/json', 'Authorization': 'Basic ' + encoded
    }
    }).subscribe(response => { newOrderData=response; console.log(newOrde
rData);

    })



    }
    else {


    const consumer_key = "ck_5771c0da725405879eb158066a586f6541685f18";
    const consumer_secret = "cs_6c33a6fdd0e856861ca20929505322b14993c98c"
;
    const encoded = btoa(consumer_key + ":" + consumer_secret);


    this.http.post('https://conference1.geowebsite.net/wp-
json/wc/v2/orders',
    orderData, {
      headers: { 'Accept': 'application/json', 'Content-
Type': 'application/json', 'Authorization': 'Basic ' + encoded
    }
    }).subscribe(response => { newOrderData=response; console.log(newOrde
rData);

    })


  }




}

async presentToast() {

  let toast = await this.toastCtrl.create({
```



```
    message: 'Your order (Number ' + String(this.orderNumber) + ' ) has been received.',
    duration: 60000,
    position: 'bottom',
    buttons: [
        {
          text: 'OK',
          handler: () => {
            this.closeCheckout();
            //this.router.navigateByUrl('/home');
          }
        }]

  });

  await toast.present();
}



  closeCheckout() {
    this.modalCtrl.dismiss();
  }

  async presentAlert() {
    const alert = await this.alertCtrl.create({
      header : "Invalid Input",
      message : "Please fill in all the required fields.",
      buttons : ["OK"]
    });

    await alert.present();
  }




}
```

Checkout.page.html

```
<ion-header>
  <ion-toolbar color = "dark">
   <ion-title style ="color: #ffffff;">
    Checkout
   </ion-title>
  </ion-toolbar>
 </ion-header>

<ion-content>

 <ion-list>
   <ion-item-divider color = "primary"> Billing Details  </ion-item-
divider>

   <ion-item>
     <ion-label style ="color: #ffffff;">Title</ion-label>
     <ion-
select [(ngModel)] = "newOrder.title" style ="color: #ffffff;">
      <ion-select-
option *ngFor = "let t of titleData" value="{{t.name}}">{{t.name}}</ion-
select-option>
     </ion-select>
   </ion-item>
   <ion-item>
    <ion-label style ="color: #ffffff;">First Name</ion-label>
     <ion-
input style ="color: #ffffff;"type="text" [(ngModel)] = "newOrder.firstNa
me"></ion-input>
   </ion-item>
   <ion-item>
     <ion-label style ="color: #ffffff;">Last Name</ion-label>
      <ion-
input style ="color: #ffffff;" type="text" [(ngModel)] = "newOrder.lastNa
me"></ion-input>
   </ion-item>
   <ion-item>
     <ion-label style ="color: #ffffff;">Company</ion-label>
      <ion-
input style ="color: #ffffff;"type="text" [(ngModel)] = "newOrder.company
"></ion-input>
   </ion-item>
   <ion-item>
     <ion-label style ="color: #ffffff;">Email</ion-label>
     <ion-
input style ="color: #ffffff;" type="email" [(ngModel)] = "newOrder.email
"></ion-input>
   </ion-item>
   <ion-item>
     <ion-label style ="color: #ffffff;">Phone</ion-label>
      <ion-
input style ="color: #ffffff;" type="text" [(ngModel)] = "newOrder.phone"
></ion-input>
   </ion-item>
   <ion-item>
     <ion-label style ="color: #ffffff;">Address</ion-label>
      <ion-
input style ="color: #ffffff;" type="text" maxlength="80" [(ngModel)] = "
newOrder.address"></ion-input>
   </ion-item>
   <ion-item >
   <ion-label style ="color: #ffffff;">Street Number</ion-label>
      <ion-
input style ="color: #ffffff;" type="text" maxlength="80" [(ngModel)] = "
newOrder.streetNumber"></ion-input>
   </ion-item>
   <ion-item>
     <ion-label style ="color: #ffffff;">Town/City</ion-label>
      <ion-
input style ="color: #ffffff;" type="text" [(ngModel)] = "newOrder.city">
</ion-input>
   </ion-item>
   <ion-item>
     <ion-label style ="color: #ffffff;">State</ion-label>
      <ion-
input style ="color: #ffffff;" type="text" [(ngModel)] = "newOrder.state"
></ion-input>
   </ion-item>
   <ion-item>
     <ion-label style ="color: #ffffff;">Postal Code</ion-label>
     <ion-
input style ="color: #ffffff;" type="text" [(ngModel)] = "newOrder.postco
de"></ion-input>
   </ion-item>
   <ion-item>
     <ion-label style ="color: #ffffff;">Country</ion-label>
     <ion-
select style ="color: #ffffff;" [(ngModel)] = "newOrder.country">
      <ion-select-
option *ngFor = "let c of countryData" value="{{c.name}}">{{c.name}}</ion
-select-option>
     </ion-select>
   </ion-item>
   <ion-item>
    <ion-label style ="color: #ffffff;">Check-in</ion-label>
    <ion-datetime style ="color: #ffffff;" displayFormat= "YYYY-MM-
DD" value="2019-05-
```

```
09" placeholder="Select Date" [(ngModel)] = "newOrder.checkinDate" ></ion
-datetime>
      </ion-item>
      <ion-item>
              <ion-label style ="color: #ffffff;">Check-out</ion-label>
              <ion-
datetime style ="color: #ffffff;" displayFormat= "YYYY-MM-
DD" value="2019-05-
12" placeholder="Select Date" [(ngModel)] = "newOrder.checkoutDate" ></io
n-datetime>
      </ion-item>
      <ion-item>
          <ion-
label style ="color: #ffffff;">Arrival Time to Airport</ion-label>
          <ion-
datetime style ="color: #ffffff;" displayFormat="HH:mm" picker-
format="HH:mm" [(ngModel)] = "newOrder.arrivalTime" placeholder="Select T
ime" ></ion-datetime>
      </ion-item>
      <ion-item>
          <ion-
label style ="color: #ffffff;">Departure Time from Hotel</ion-label>
          <ion-
datetime style ="color: #ffffff;" displayFormat="HH:mm" picker-
format="HH:mm"  [(ngModel)] = "newOrder.departureTime" placeholder="Selec
t Time"  ></ion-datetime>
      </ion-item>

      <ion-item>
          <ion-
label style ="color: #ffffff;" position="floating">Order Notes</ion-
label>
          <ion-
textarea style ="color: #ffffff;" maxlength="80" [(ngModel)] = "newOrder.
notes" placeholder = "Special notes about the order (Optional)"> </ion-
textarea>
      </ion-item>


     <ion-item-divider color = "primary">Payment Methods</ion-item-
divider>
        <ion-item>
          <ion-label style ="color: #ffffff;">Payment Method</ion-label>
          <ion-
select style ="color: #ffffff;" [(ngModel)] = "newOrder.paymentMethod">
            <ion-select-
option *ngFor = "let p of paymentMethods" value="{{p.method_title}}">{{p.
method_title}}</ion-select-option>
          </ion-select>
        </ion-item>
  </ion-list>


</ion-content>

<ion-footer>
    <ion-toolbar>
      <ion-grid>
        <ion-row>
          <ion-col col-2></ion-col>
          <ion-col col-4>
            <ion-button color="danger" (click)= "closeCheckout()">Back</ion-button>
          </ion-col>
          <ion-col col-4>

              <div *ngIf="this.newOrder.firstName === undefined || this.newOrder.lastName === undefined || this.newOrder.company === undefined || this.newOrder.email === undefined ||
              this.newOrder.phone === undefined || this.newOrder.address === undefined || this.newOrder.streetNumber === undefined || this.newOrder.postcode === undefined ||
              this.newOrder.city === undefined|| this.newOrder.state === undefined|| this.newOrder.country === undefined|| this.newOrder.title === undefined||
              this.newOrder.checkinDate === undefined || this.newOrder.checkoutDate === undefined || this.newOrder.arrivalTime === undefined|| this.newOrder.departureTime === undefined||
              this.newOrder.paymentMethod === undefined; then alert else placetheorder"></div>
              <ng-template #alert><ion-button color="success" (click) = "presentAlert()">Place Order</ion-button></ng-template>
              <ng-template #placetheorder><ion-button color="success" (click) = "placeOrder(); presentToast();">Place Order</ion-button></ng-template>


          </ion-col>
          <ion-col col-2></ion-col>
        </ion-row>
      </ion-grid>
    </ion-toolbar>
</ion-footer>
```

Checkout.page.scss

```
ion-content{
    --ion-background-color:#0E1C95;
  }
```