



Internship Report

Master in Computer Engineering-Mobile Computing

***Quantum Node Portal- Devices and Information
Management***

Sujane Natasha Lopez

Leiria, September 2017

This page was intentionally left blank



Internship report

Master in Computer Engineering-Mobile Computing

Quantum Node Portal- Devices and Information Management

Sujane Natasha Lopez

Internship Report developed under the supervision of Professor Doctor *Joao Pereira*, professor at the School of Technology and Management of the Polytechnic Institute of Leiria.

Leiria, *September 2017*

This page was intentionally left blank

Acknowledgments

I would like to take this opportunity to express my sincere gratitude to the people who helped me through this internship work.

I sincerely thank Professor Joao Pereira for guiding me throughout my Internship Period, Professor Carlos Grilo for giving me an opportunity to do an Internship in Domatica Global Solutions. Undoubtedly the main person CEO and Founder Samuel Silva who believed in me and made this Internship possible. The Director Pedro Pina for being a good team leader and guiding my work.

Besides them, a big thanks to my team members, my colleagues in Domatica Global Solutions.

I am thankful to my parents for being with me and supporting me unconditionally.

This page was intentionally left blank

Abstract

An Internship in a European Company for developing a Web application-Domatica Global Solutions, Lisbon was undertaken to complete the Master's Degree of Computer Engineering-Mobile Computing in the Polytechnic Institute of Leiria.

The team Domatica deals with providing IoT solutions used for monitoring, controlling and collecting the data from the IoT gateways. The present work aims to develop a Web application for client's side.

The Web application named Quantum Node Portal is developed for the Devices and Information management. It provides access to the clients to their IoT gateways. Clients can monitor their devices, get various information, also can access the Portal for claiming their IoT gateways.

The present work was developed using various technologies such as PHP framework named Laravel and several languages.

Key-Words: IoT Solutions, IoT gateways, Quantum Node Portal, PHP Framework, Laravel

This page was intentionally left blank

List of figures

Figure 1. Domatica Edge Server Architecture.....	2
Figure 2. Domatica Quantum Cloud.....	3
Figure 3. Quantum Node(IoT Gateway).....	4
Figure 4. Sentinel Web User Interface	10
Figure 5. User Interface: Readiness process class-scenario definition process.....	12
Figure 6. Laravel Routing.....	13
Figure 7. Web Interface for Tasks List.....	15
Figure 8. Web User Interface for Task Detail	16
Figure 9. Percentage of usage of PHP Frameworks	17
Figure 10. Dashboard page of the health portal	19
Figure 11. Deltanet using Laravel Framework.....	21
Figure 12. Neighborhood Lender using Laravel Framework	22
Figure 13. MyRank using Laravel Framework	23
Figure 14. Laravel Tricks	23
Figure 15. LaravelIO	24
Figure 16. Larasocial	25
Figure 17. Cachet.....	25
Figure 18. Orchestra Platform	26
Figure 19. Built with Laravel	28
Figure 20. Usage of PHP	30
Figure 21. M-V-C Architecture	45
Figure 22. Laravel framework Architecture	47
Figure 23. MySQL Database tables.....	48
Figure 24. Quantum Node Portal Architecture.....	49
Figure 25. REST API functionalities.....	50
Figure 26. Scrum Roles	53
Figure 27. Use Case Model	57
Figure 28. Project running locally	63
Figure 29. Laravel Project on Web browser.....	63
Figure 30. Root Directory.....	64
Figure 31. Basic Configuration	66
Figure 32. .env file.....	67
Figure 33. Routing mechanism.....	68
Figure 34. Registration page.....	79
Figure 35. Login Page	80
Figure 36. Password Recovery	80
Figure 37. Dashboard	82
Figure 38. Quantum Nodes List page.....	83
Figure 39. Quantum Nodes Replace.....	83
Figure 40. Quantum Nodes Release	84

Figure 41. Authentication page with Registration code	85
Figure 42. Authentication with Serial Number and Label Key	85
Figure 43. Quantum Node Creation page	86
Figure 44. Map View page with Drag and Drop Pin	87
Figure 45. Node Information page	88
Figure 46. Edit Node Information Modal	88
Figure 47. Status Tab	89
Figure 48. Configuration page	90
Figure 49. Data Management	90
Figure 50. Entity Information page	91
Figure 51. Edit Entity (a)	92
Figure 52. Edit Entity (b)	92
Figure 53. Add gToken	93
Figure 54. List gTokens	94
Figure 55. Database User data	96
Figure 56. Entity Information Response	97
Figure 57. Waiter page on the Web application	105
Figure 58. Cook page on the Web application	105
Figure 59. The 'View Analytics' feature of the Manager page	106
Figure 60. PHP Frameworks Comparison	107

This page was intentionally left blank

List of Tables

Table 1. Comparison of Scrum Roles with Our Project Members Roles..... 55

Table 2. Internship Tasks..... 104

This page was intentionally left blank

List of acronyms

AEP - Application Enablement Platform
AJAX - Asynchronous JavaScript and XML
AMQT - AmeriQuest Technologies
API - Application Programming Interface
ARM- Advanced RISC Machines
CLI - Command Line Interface
CoAP - Constrained Application Protocol
CSS - Cascading Style Sheets
CSRF - Cross Site Request Forgery
FTP - File Transfer Protocol
gTokens - Gateway Tokens
HTML - Hyper Text Mark-up Language
IoT - Internet of Things
IT - Information Technology
LK - Label Key
MVC - Model-View-Controller
MQTT - Message Queuing Telemetry Transport
PaaS - Platform as a Service
PDO - PHP Data Objects
PHP - Personal Home Page
RC - Registration Code
REST - Representational State Transfer
RTU – Remote Terminal Unit
SaaS - Software as a Service
SN - Serial Number
SQL - Structured Query Language
TLS - Transport Layer Security

This page was intentionally left blank

Table of Contents

ACKNOWLEDGEMENTS	V
ABSTRACT	VII
LIST OF FIGURES	IX
LIST OF TABLES	XII
LIST OF ACRONYMS	XIV
1. INTRODUCTION.....	1
1.1 Context of Project	1
1.2 Context of Internship	5
1.3 Technical Aspect	5
1.3.1 Languages Used	6
1.3.2 Tools Used	6
2. BIBLIOGRAPHY REVIEW	9
2.1 State of the art.....	9
2.1.1 Research Projects	9
2.1.2 Commercial Projects.....	20
3. FRAMEWORK REVIEW	29
3.1 Comparison of PHP frameworks and their advantages and disadvantages	29
3.1.1 Symfony framework	30
3.1.2 Yii framework.....	31
3.1.3 Code Igniter	32
3.1.4 CakePHP.....	33
3.1.5 Zend.....	34
3.1.6 Kohana.....	36
3.1.7 Aura	37
3.1.8 Slim	38
3.1.9 Silex.....	39

3.2	Choice of Technology	40
3.2.1	Laravel framework	40
4.	ARCHITECTURE	45
4.1	MVC Architecture	45
4.2	Laravel Architecture	47
4.3	Quantum Node Portal Architecture	49
5.	PROJECT METHODOLOGY	53
5.1	Scrum Methodology	53
6.	SYSTEM DESIGN	57
6.1	Use Case Model	57
6.2	Activity flowcharts	58
6.2.1	Login Activity	58
6.2.2	Claim Node activity	59
7.	IMPLEMENTATION	61
7.1	Functionalities	61
7.2	Laravel-Installation	62
7.3	Laravel-Application Structure	64
7.3.1	Root Directory	64
7.4	Configuration	65
7.4.1	Basic configuration	65
7.4.2	Environmental and Database configuration	66
7.5	Routing	67
7.6	Middleware	69
7.7	Controllers	71
7.7.1	Controller creation	71
7.7.2	Authentication Controllers	72
7.8	Views	74
7.8.1	Passing Data to the Views	74
7.9	Validation and Sending Email	75

7.9.1	Email Configuration	77
8.	USER INTERFACE	79
8.1	Account Management.....	79
8.1.1	Registration Page.....	79
8.1.2	Login page.....	79
8.1.3	Password Recovery.....	80
8.2	Nodes Management.....	81
8.2.1	Map.....	81
8.2.2	List	82
8.2.3	Claim	84
8.2.4	Node information	87
8.3	Entity Management	91
8.3.1	Entity Information	91
8.4	Tokens Management	93
8.4.1	Add Tokens	93
8.4.2	Remove Tokens	93
8.4.3	List Tokens	94
9.	TESTS	95
9.1	Authentication testing	95
9.2	Security testing	95
9.3	Application testing	96
10.	RESULTS.....	97
10.1	REST API responses:	97
11.	CONCLUSION.....	99
12.	FUTURE WORKS.....	101
	REFERENCES.....	102
	APPENDIX A.....	104
	APPENDIX B.....	105

APPENDIX C107

This page was intentionally left blank

1. Introduction

Advancements in technology have altered the methods and expectations of client service. Earlier dealing with the client service was the hectic job which required speaking to the client when needed, but now the self-service approach means surrendering clients with an ability to quickly access, receive and alter information. Portal is the Medium to the clients. Having a portal helps the clients to access it anytime and anywhere. It also gives the company to prove its credibility by ensuring the clients data security. This can earn a positive feedback for the service provided by the company. Companies that can effectively achieve this will have high client satisfaction. Portal allows a firm to provide organisations and people with a place where they can use a service in the manner that is time efficient and also to have a secure interaction with the service. Portal promises to deliver a more coherent information management platform.

1.1 Context of Project

The Team Domatica works for the development of a framework that enables monitoring and control of Internet of Things (IoT) solutions. It mainly focuses on device connectivity, management and collection of data. The devices are nothing but the Internet of Things (IoT) Gateways. This aim is fulfilled by Domatica Edge server. Domatica Edge server is an ultra-lightweight middleware for the IoT space that works on Application Enablement Platform (AEP).

Domatica Edge server has Northbound and Southbound connections as shown in Figure 1.

Northbound connections consist of Information Technology (IT) ecosystem that are the Third-Party Software as a Service (SaaS), Platform as a Service (PaaS), Apps.

Southbound connections are the Third-Party Sensors, Actuators, Equipment, Application Database.

Edge server includes Broker that is connected to Northbound connections through various services interface. The services such as Representational State Transfer (REST), Message Queuing Telemetry Transport (MQTT), AmeriQuest Technologies (AMQT), Constrained

Application Protocol (CoaP), File Transfer Protocol (FTP). The Broker translates the data information in the form of signals to data suitable for the IoT ecosystem.

Edge server Node is a software that can be installed in any IoT gateway that runs on Advanced RISC Machines (ARM), X86_32 and X86_64. This software uses Fog Computing architecture. It enables collection of data from various gateways. It enables configuration of Southbound interfaces through combination of drivers.

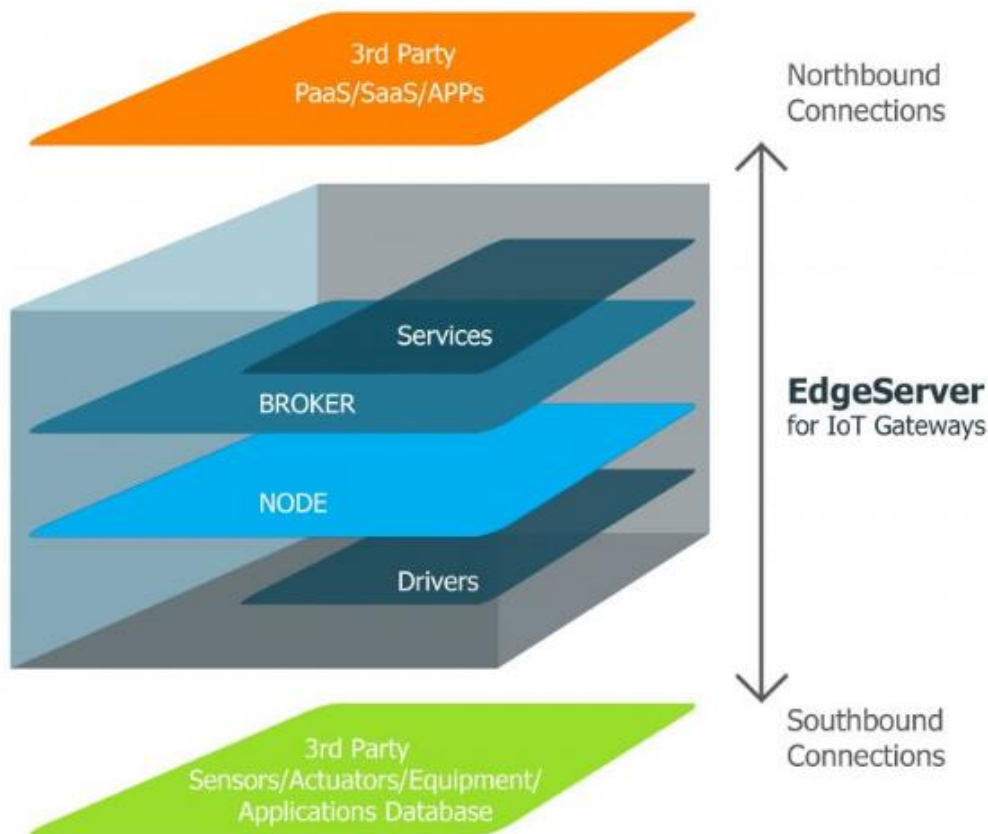


Figure 1. Domatiga Edge Server Architecture

Quantum Node Portal development is the project that I am working on. It is a fresh project. It consists of front-end developing and back-end developing. Quantum Node Portal is a web platform that works as Information and Device Management Portal. It provides the clients with facilities like monitoring their IoT Gateways considered as Nodes that are located at different places.

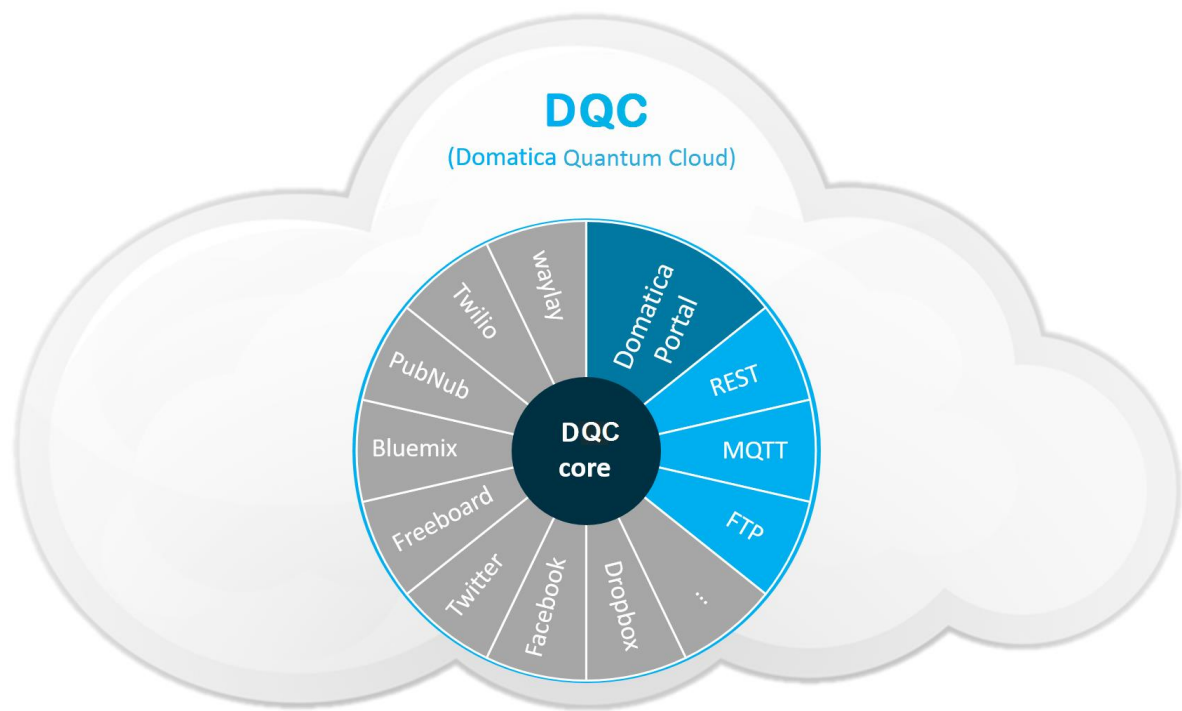


Figure 2. Domatiga Quantum Cloud

Quantum Nodes is the integration of various sensors from 0-10VDC, 4-20mA, current transformer meters, pulse meters (Energy, Gas, Water), digital, Pulse Width Modulation (PWM), and analog actuators, as well as protocol devices such as Modbus Remote Terminal Unit (RTU), Konnex (KNX), HDL Buspro, DMX512, and others by request.



Figure 3. Quantum Node(IoT Gateway)

Our project portal delivers information about the Quantum Nodes owned by the various clients. Quantum Nodes are placed at different locations for example farms, Information includes status(online/offline) of various nodes, location, view details of the nodes, to claim the nodes by the authentication process. It provides the owner/admin user of the device to distribute access to the other users, create and remove the user, make changes in the configuration as well.

There are Four types of Authentication methods to be implemented in Our Portal.

- User Authentication
- gTokens (gateway Tokens)
- Session Identity (Id)
- Serial Number and Label Key or Registration Code.

User Authentication is nothing but the user login information i.e. Email Id and Password.

gTokens authentication requires Token Id and vNode Id.

Authentication process is also done using Serial Number (SN) and Label Key (LK) of the Claiming device (as shown in figure 3) or the Registration Code of the Claiming device in order to Claim the Quantum Node.

Implementation of this project requires complex Front-end programming, Back-end programming and Data Management. Delivery of data to the clients in an understandable manner and having an access with a good User-interface is equally important.

1.2 Context of Internship

I have been assigned a task to develop the Front-end of the web application and provide a good design and User interface for the Users. I had to develop Account, Quantum Nodes, Entity, Tokens Management. It includes several types of Authentication process for the users, Registration and Login process, sending email that confirms the Registration of account to the Users, Forgot Password activity.

The Dashboard contains Menus, Map that displays various nodes location belonging to the user, the user can be (Admin/User). User can access the Node Information and Edit if required. The User can add/claim new Quantum Nodes by an authentication process. This process is to provide Serial Number and Label Key or Registration Code. The Authentication process infers security. Sending and Receiving the Data from the Database using REST services to complete the functionalities so that we have Real-time data.

More details about the tasks on Monthly basis can be found in Appendix A. Implementations of the project will be discussed further ahead in the report.

1.3 Technical Aspect

In order to complete the tasks, the following languages and scripts have been made use of: HTML for structuring the layout, Cascading Style Sheets (CSS) for the design, JavaScript for active functioning and jQuery. The Web application was developed in Laravel framework that eased the work of developing. Laravel framework includes useful features.

MySQL Database to store the Users information. The tests are performed in the Local environment.

1.3.1 Languages Used

HTML

HTML is a standard mark-up language that provides a structured way of developing web pages. A structure is provided by the HTML tags. Laravel provides a blade template engine that compiles HTML and PHP. Most of our Project view files are in Blade template.

CSS

CSS plays an important role in the design of the view. Using CSS language, we can describe the style of the layout. In Laravel, we use a folder named “Public” to store all the CSS files.

JavaScript

JavaScript language is used to develop the functionalities. The programming also includes jQuery and Asynchronous JavaScript and XML (AJAX). jQuery is quick and a minor JavaScript library that provides various helpful features for making event handling. AJAX library is the method of implementing existing standards. It helps to exchange data with the server and updating web pages. It is used to dynamically load the data.

1.3.2 Tools Used

Laravel Framework

A Web application framework is a software framework that eases the Web Application Development process and services. The framework offers the developers with the structured abstraction that covers the interfaces to functionalities mutual to the categories of applications to which the framework was constructed to develop.

Laravel is M-V-C framework. It has rich set of useful features that enables to develop the application faster. In order to use Laravel Framework it requires Composer and Xampp installed in the system.

Composer

Composer is a Dependency manager for PHP. Laravel makes use of this for managing dependencies. It provides the libraries required to be installed or updated as required by the project.

Xampp

Xampp is a server that includes Apache, MySQL Database and interpretation done by PHP or Perl Languages. In our project, we make use PHP language. The local tests were first conducted using Apache server and MySQL Database.

2. Bibliography review

2.1 State of the art

This chapter introduces the related projects that use Laravel framework. Also, related projects similar to our project that deals with managing (Internet of Things) IoT devices using Web Portal and summary of the other similar Personal Home Page (PHP) frameworks with comparison done between them, with the advantages and disadvantages of the PHP frameworks and choice of technology for our project.

2.1.1 Research Projects

This section summarizes the research projects which uses Laravel Framework to build Web applications.

Sentinel Web: Implementation of Laravel Framework in Web Based Temperature and Humidity Monitoring System

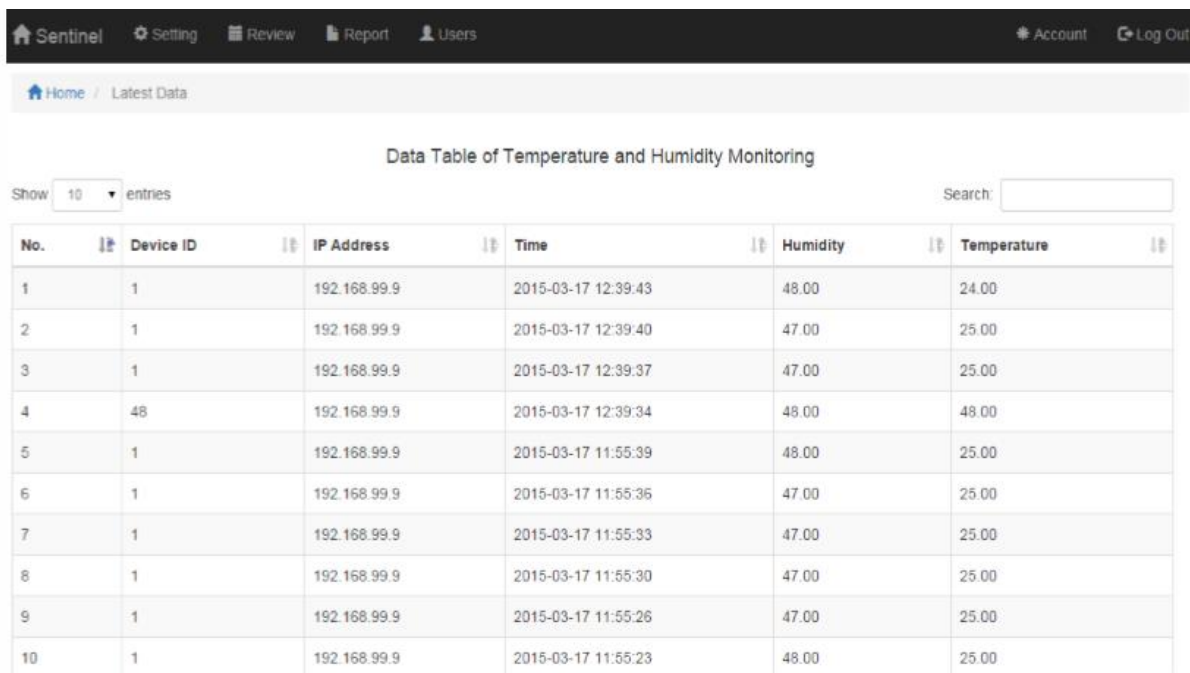
Sentinel Web: Implementation of Laravel framework [1] is a web-based temperature and humidity monitoring system. The environmental parameters such as temperature and humidity has influence on information technology infrastructure. This paper focuses on monitoring such parameters. A web based monitoring system as an online application media is developed to simplify recapitulation and documentation of temperature and humidity data. The development is fastened with the implementation of web technology such as framework for scripting. The Laravel Model View Controller (MVC) framework is employed.

In this project, Sentinel web is presented through the object-oriented modelling diagram. Unified Modelling Language (UML) is chosen as a modelling tool. The project has two types of classes, Classes in Model and Classes in Controller. Controller class has a function to control how information will flow after brought by Model.

Bootstrap framework, Asynchronous JavaScript and XML (Ajax) and jQuery are used to build User interface. Sentinel Web has six menus. It derives from Home, Setting, Review, Report, Users, and Account. Home page will generate the latest monitoring data. Setting page provide page to set site properties. Review page can automatically download our error report to pdf version document. In Users page, user can see others user profile. Account page show user information page that is displayed to other users.

This project is similar to our project because it monitors the real-time data as the Quantum Portal monitors the Quantum Nodes. Another similarity is it is developed using Laravel framework.

This figure below represents the Online Sentinel Web Application User Interface monitoring the temperature and humidity.



The screenshot displays the Sentinel Web application interface. At the top, there is a dark navigation bar with links for Sentinel, Setting, Review, Report, Users, Account, and Log Out. Below this is a breadcrumb trail: Home / Latest Data. The main content area is titled "Data Table of Temperature and Humidity Monitoring". It features a "Show 10 entries" dropdown and a search bar. The table below contains 10 rows of monitoring data.

No.	Device ID	IP Address	Time	Humidity	Temperature
1	1	192.168.99.9	2015-03-17 12:39:43	48.00	24.00
2	1	192.168.99.9	2015-03-17 12:39:40	47.00	25.00
3	1	192.168.99.9	2015-03-17 12:39:37	47.00	25.00
4	48	192.168.99.9	2015-03-17 12:39:34	48.00	48.00
5	1	192.168.99.9	2015-03-17 11:55:39	48.00	25.00
6	1	192.168.99.9	2015-03-17 11:55:36	47.00	25.00
7	1	192.168.99.9	2015-03-17 11:55:33	47.00	25.00
8	1	192.168.99.9	2015-03-17 11:55:30	47.00	25.00
9	1	192.168.99.9	2015-03-17 11:55:26	47.00	25.00
10	1	192.168.99.9	2015-03-17 11:55:23	48.00	25.00

Figure 4. Sentinel Web User Interface

Towards a Prototype for Guidance and Implementation of a Standardized Digital Forensic Investigation Process

This paper introduces a Web Application that performs a digital forensic investigation and guides a digital forensic investigator to efficiently implement the process [2]. This Web application has two main functionalities. The first would be to act as an expert system that can be used for guidance and training of novice investigators. The second functionality would be to enable the implementation of the investigation process while reliably logging of all actions taken within the processes proposed in a comprehensive and harmonized digital forensic investigation process model.

The Prototype built is Web-based as this allows ease of use across multiple platforms and from any location. It enables collaboration of multiple users, from multiple organizations and even from multiple jurisdictions and countries. It is also better to provide the prototype as a Software as a Service (SaaS) because the user has no control over the server thus it can be optimized for complete security by the owner. Also, SaaS can provide better cost effectiveness for the user and it enables user to concentrate on core activity: the digital forensic investigation.

The Application is implemented using the PHP coding language and the Laravel Framework. Laravel is a free, open source PHP web application framework, designed for the development of MVC (Model-View-Controller) web applications. The Laravel framework was chosen because of its MVC and REST (Representational state transfer) capabilities, as well as its database support and available add-ons and libraries. Laravel also has a number of important security related functionality such as encryption and authentication. Sentry is an add-on to Laravel that provides configurable authentication, authorization, user activation, groups and group permissions, login throttling, user suspension, user banning and it is interface driven. Sentry also encrypts all passwords and allows an easy way to authenticate a user and to prevent access to pages based on the user that is logged in.

The figure below represents a Graphical User Interface showing the readiness processes class and in the specific scenario definition process. It is intended that one can follow the processes, as per the standardized process model, while software provides guidance (on the left side of the user pane) and possibility to implement the process (on the right side of the user pane). The user can also choose to generate reports through selecting the 'Reports' tab.

The screenshot displays a web application interface for the 'Readiness Processes Class'. At the top, there are navigation tabs: 'Readiness Processes Class' (active), 'Initialization Processes Class', 'Acquisitive Processes Class', 'Investigative Processes Class', and 'Reports'. A 'Logout' link is in the top right corner. The main content area is titled 'Readiness Processes Class' and features a vertical sidebar on the left with a list of process steps: 'Scenario definition' (highlighted in blue), 'Identification of potential digital evidence sources', 'Planning pre-incident collection, storage and handling of data representing potential digital evidence', 'Planning pre-incident analysis of data representing potential digital evidence', 'Planning incident detection', 'Defining system architecture', 'Implementing system architecture', and 'Implementing pre-incident collection, storage and handling of data representing potential digital evidence'. The central panel, titled 'Scenario definition', contains a detailed description of the process. To the right is a form for defining a scenario, including a text input for 'Scenario Definition Name', two file upload sections for 'Scenario Definition Document (pdf)' and 'Risk Assessment Document (pdf)' (each with a 'Choose File' button and 'No file chosen' text), and a 'Next' button at the bottom right.

Figure 5. User Interface: Readiness process class-scenario definition process

A review study on secure web application development using PHP with Laravel Framework

This paper explains the various features of Laravel framework, what makes Laravel different from other frameworks, and how we can build a secure web application using this framework. [3] Laravel offers well tested set of libraries to develop fast and secure web application.

Request Life Cycle gives us the exact understanding of how Laravel is different from other php frameworks. “/public/index.php” file is the entry point for all the requests made in Laravel. “Index.php” file is having the auto generated code which creates an object of Kernel class to perform other activity. Kernel class makes Laravel routing a bit different from other MVC framework. The following diagram explains the Laravel routing more clearly.

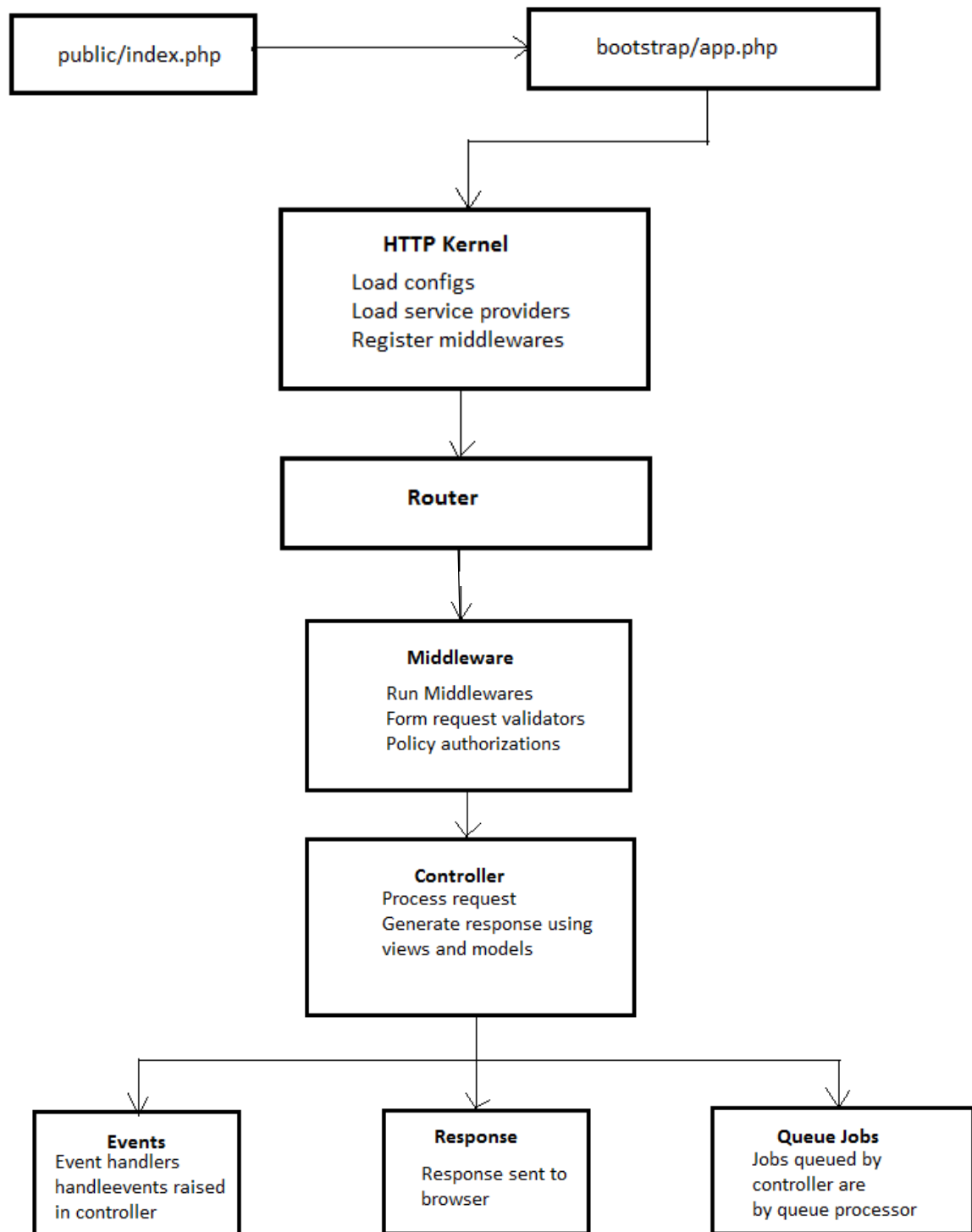


Figure 6. Laravel Routing

According to the figure 6 “public/index.php” is the entry point for all the requests made in Laravel. After that it gets an instance of Laravel application from “bootstrap/app.php”, loads the HTTP Kernel. Kernel loads configuration, middleware and service provider and so on.

Laravel has more features that makes it more secured such as:

Eloquent Object Relational Mapper (ORM): It uses PHP Data Object (PDO) parameter binding to avoid Structured Query Language (SQL) injection. Parameter binding ensures that malicious users can't pass in query data which could modify the query's intent. The Eloquent ORM provides security with an Active Record implementation to deal with database.

CSRF (Cross Site Request Forgery): It is a request that the user did not intend to make. So, an attacker somehow creates a forged request, and submits it on the user's behalf. Cross-site request forgery, also known as one-click attack or session riding and abbreviated as CSRF or XSRF, is a type of malicious exploit of a website where unauthorized commands are transmitted from a user that the website trusts.

Authentication: The authentication configuration implemented in auth.php file, contains several authentication services. Laravel authentication facilitates are made up of "guards" and "providers". Guards define how users are authenticated for each request. Providers define how users are retrieved from your persistent storage.

Route: Laravel can scaffold all the routes and views required for authentication of users with a single command.

Authentication Throttling: It limits the user's login attempts. By default, the user will not be able to login for one minute if they fail to provide the correct credentials after several attempts. The throttling is unique to the user's username/ e-mail address and their IP address.

```
protected $middlewareGroups = [  
    'api' => [  
        'throttle:60,1',  
    ],  
];
```

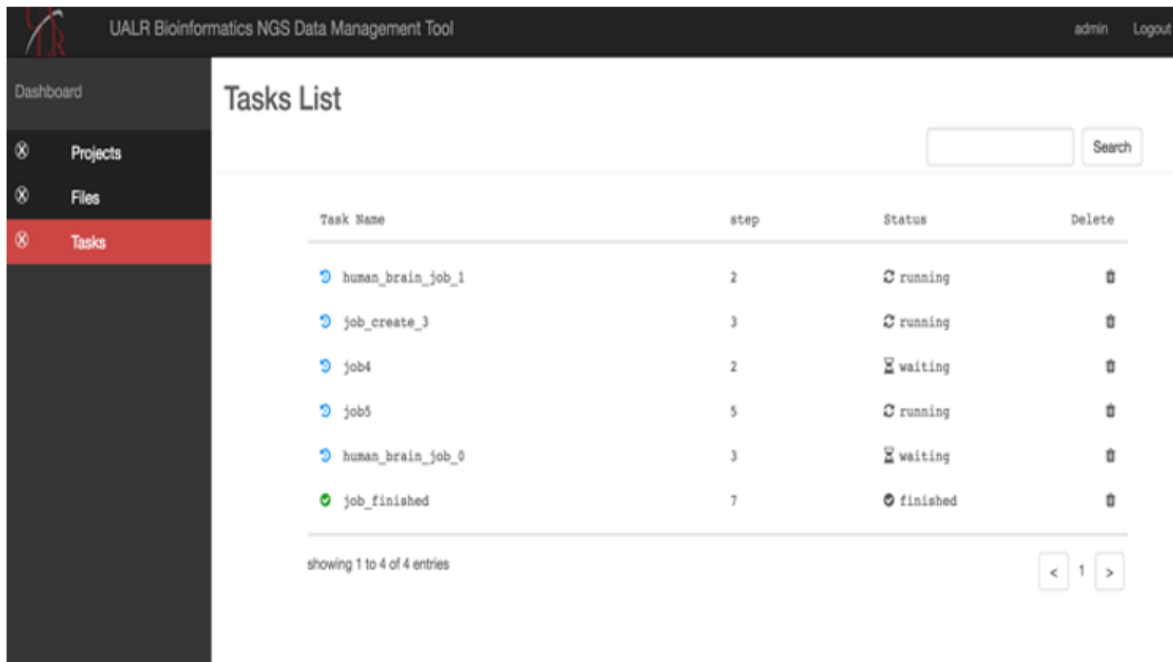
The above code snippet from the Kernel.php file tells that after sixty failed attempts of login, user will not be able to login for one minute and it can be customized.

Hashing: The Laravel hashing provides secure Bcrypt hashing for storing user passwords.

Building genomics foundation for precision medicine research: A portable multitask data management system

This project is about Next Generation Sequencing(NGS) Data Management system [4] for automating the process of large-scale multidivisional genomic big data and permits the efficient utilization of resources. It is a web application developed using Laravel framework.

The Users can track and analyse their data via the web interface. This system contains an easy to use web interface that allows users to create an account so that they can login and manage their projects. This provides users a functionality to upload their data files, choose available pipelines and submit the job. All jobs are executed on the server and the results are stored in the File Transfer Protocol (FTP) server of the Systems Genomics Laboratory of University of Arkanas at Little Rock. Users can see all the information about their jobs including the status of their submitted jobs and the results of completed jobs.



The screenshot displays the 'Tasks List' page of the 'UALR Bioinformatics NGS Data Management Tool'. The interface includes a sidebar with navigation links: Dashboard, Projects, Files, and Tasks (which is highlighted in red). The main content area shows a table of tasks with columns for Task Name, step, Status, and Delete. The tasks listed are: human_brain_job_1 (step 2, running), job_create_3 (step 3, running), job4 (step 2, waiting), job5 (step 5, running), human_brain_job_0 (step 3, waiting), and job_finished (step 7, finished). A search bar is located at the top right of the table. At the bottom, it indicates 'showing 1 to 4 of 4 entries' and has pagination controls for page 1.

Task Name	step	Status	Delete
human_brain_job_1	2	running	
job_create_3	3	running	
job4	2	waiting	
job5	5	running	
human_brain_job_0	3	waiting	
job_finished	7	finished	

showing 1 to 4 of 4 entries

Figure 7. Web Interface for Tasks List

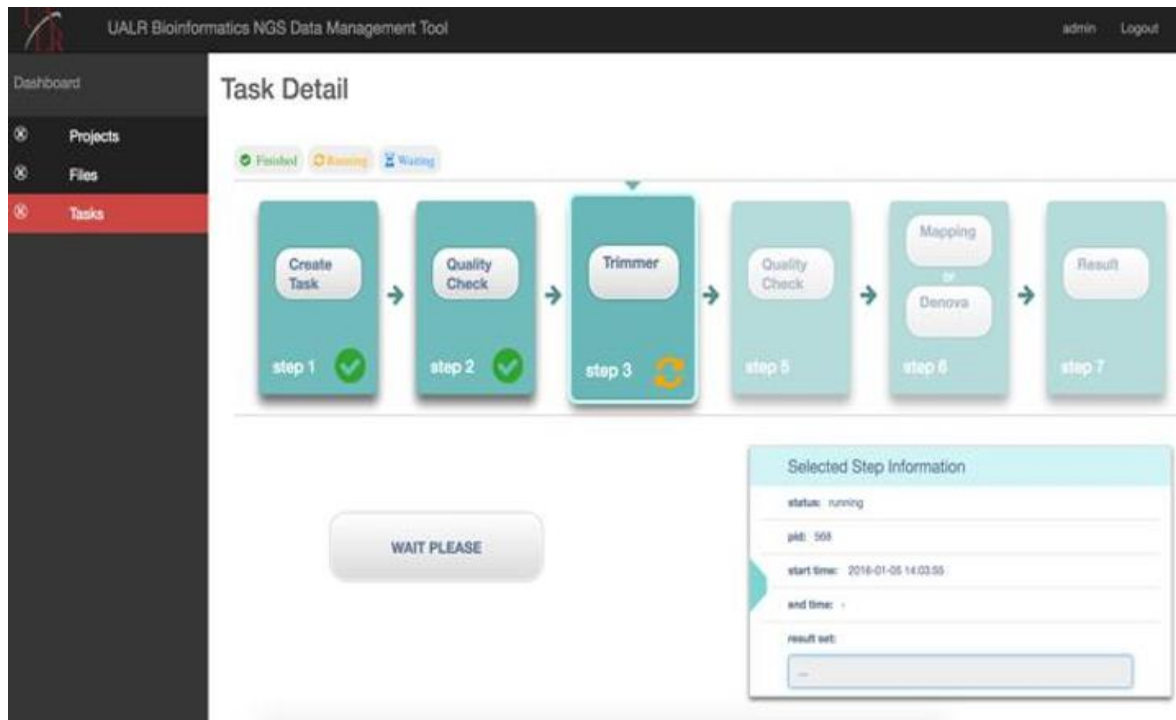


Figure 8. Web User Interface for Task Detail

Design and implementation of web based on Laravel framework

This paper presents the [5] design and implementation method of a web-based on Laravel framework as it makes the development process standardized, processing non-business logic relationship automatically, allowing programmers to focus on the implementation of business logic.

This paper concludes that Laravel framework has various advantages such as:

- Calculated by network, reducing the load on the server and the pressure for the network, so that the large number of calculations are distributed on each node in the network.
- It can be inserted. When the business need to be expanded, simply to insert the corresponding function in original software modules, the compiler can be applied without recompiling the entire system deployment.

- Design can be completed better. For software architecture, it enables software structure more clearly, the affinity for users and programmers is strengthened, and also the structure and function of the software is more reasonable.
- The module can be reused, the modules of same or similar function, just need to be copied to the destination and can be used after compiling the deployment.
- Easy to update.
- Implementation of separation from interface.

Android Application Development using Android Studio and PHP Framework

This paper [6] is about an Android application developed using Android studio and Laravel Framework. Laravel framework gives a layered approach to android. From the Pie chart below, it is clear that Laravel framework is widely used compared with other frameworks.

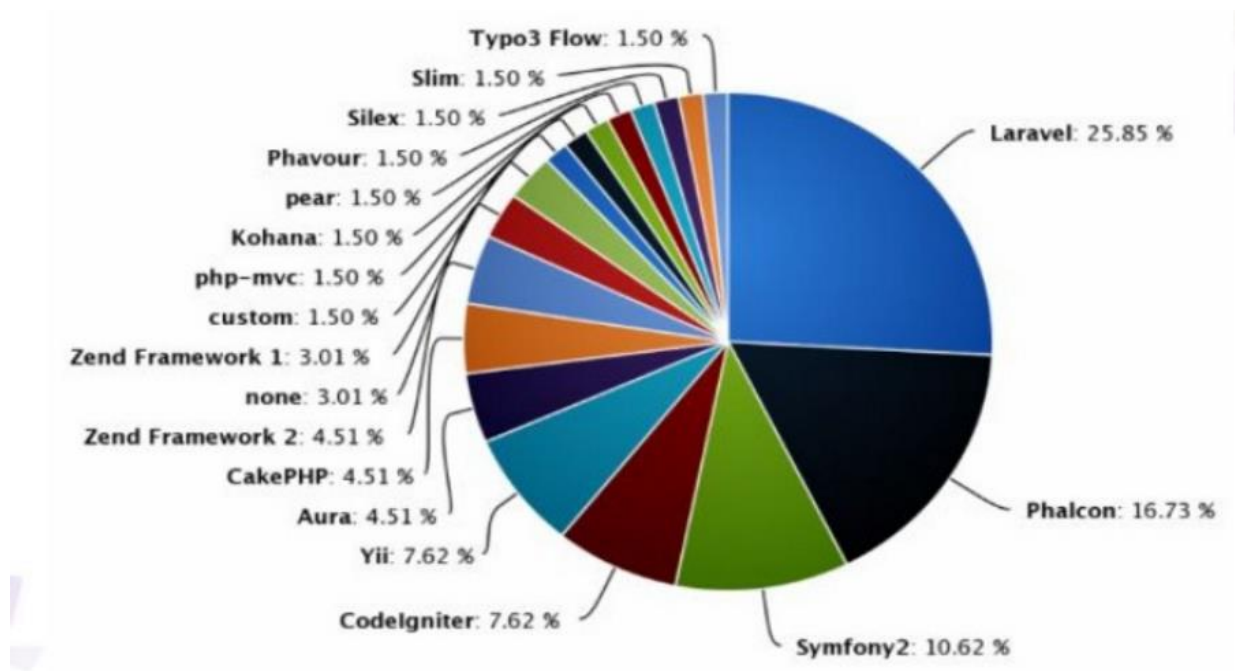


Figure 9. Percentage of usage of PHP Frameworks

The reasons for choosing Laravel as the PHP Framework is as follows:

- Popularity

- Community Support
- Security
- Backend with Zend Foundation which is maintained by IBM and Google
- OAuth 2.0 Implementation
- Multiple Database Support
- Command Line Tools
- Strict MVC Based framework
- PHP Auto Load Feature
- Composer Dependency Manager
- Modular Approach
- Less Learning Curve

Mobile Health Platform for Diabetes Management Based on the Internet-of-Things

This paper [7] introduces a new way of monitoring Diabetic Patients. It involves various wireless nodes that are the medical sensors such as Weight Scale, Blood pressure, Blood Glucose Monitor, Pulse Rate Monitor. They are connected to the Mobile Application via Bluetooth Connectivity. Smart Phone integrates this physical layer of sensors and the Health Portal Application. Smart phone is a medium for patient interaction with Health Portal Application. Smart Phone is connected to the Portal through GSM network (3G/LTE) and /or Wi-Fi network.

The Application Portal interfaces the various objects of the physical layer (e.g. sensors) to the other objects (e.g. healthcare professionals, hospitals, and other systems). It is responsible for remote data collection and storage, data processing and monitoring, and making decisions based on constraints specified by individual patient treatment plans. It also handles all use/device requests and generates responses.

The Application Portal is built on using M-V-C Architecture for the security purposes. And also, to have scalability as it is primary requirement when it comes to IoT systems. On the server side, PHP language is used. The application is developed using Hyper-Text Markup Language (HTML5), Cascading Style Sheets (CSS3) and JavaScript. It is supported by PostgreSQL database to store the system-level configurations, system static and dynamic data, and patient's data. Data synchronization between the local smartphone database and

the remote health portal database is performed through sending HTTP requests from the smartphone to the health portal server which in turn responds with JSON formatted acknowledgement message.

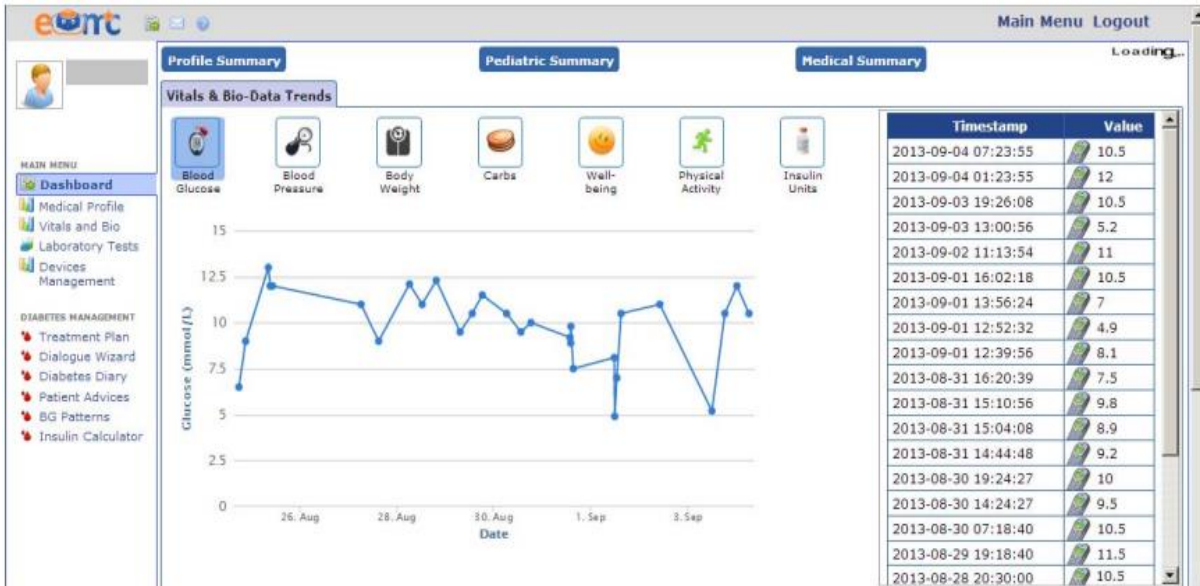


Figure 10. Dashboard page of the health portal

Near-Field Communication Sensors and Cloud-Based Smart Restaurant Management System

This paper [8] introduces a project developed to solve various problems faced in Restaurants. Smart Restaurant Management System (SRMS) makes use of various technologies such as Internet of Things (IoT), Near-Field Communications (NFC) sensors, Mobile and Web Applications and Cloud computing. Mobile application is used by the Customers whereas Web Application is used by the Restaurant staffs.

Mobile Application was developed by implementing Java and Extensible Markup Language (XML). Android studio was used as the IDE for the Android application. The Web application was developed using JavaScript, HTML and PHP. The Web application was deployed on an Apache server installed on a virtual machine in the Cloud. Both the software applications interact with each other by having access to the Apache HTTP web server hosted. Both the applications access, query and update the database via the web server.

Interface for Restaurant Staff Interaction for SRMS is given by the three web pages: Waiter page, Cook page and the Manager page.

Waiter page as shown in Appendix B, notifies the waiter as soon as customers request to see them through their Android application or when orders are finished in the kitchen and ready to be served.

The Cook page as shown in Appendix B, shows every food item ordered by customers and also the Cook can update the status of the dishes.

The Manager page as shown in Appendix B, allows the manager with the option to ‘View Analytics’. This feature displays real-time analytics and data about the performance of the restaurant for the current day, including information such as how much revenue has been earned, number of orders placed in the day, most popular dish as well as the busiest and slowest hours.

2.1.2 Commercial Projects

This section summarizes the Commercial projects which uses Laravel Framework to build Web applications [9].

Deltanet Travel

Deltanet Travel [9] is a Web application that manages destinations. It serves to manage tourist services in Greece. It is developed using Laravel Framework and has smooth browsing. The website is filled with latest new feeds about travel. It contains a column that make a search for hotels at the specific destinations. And this search can be done without getting logged on to the website.

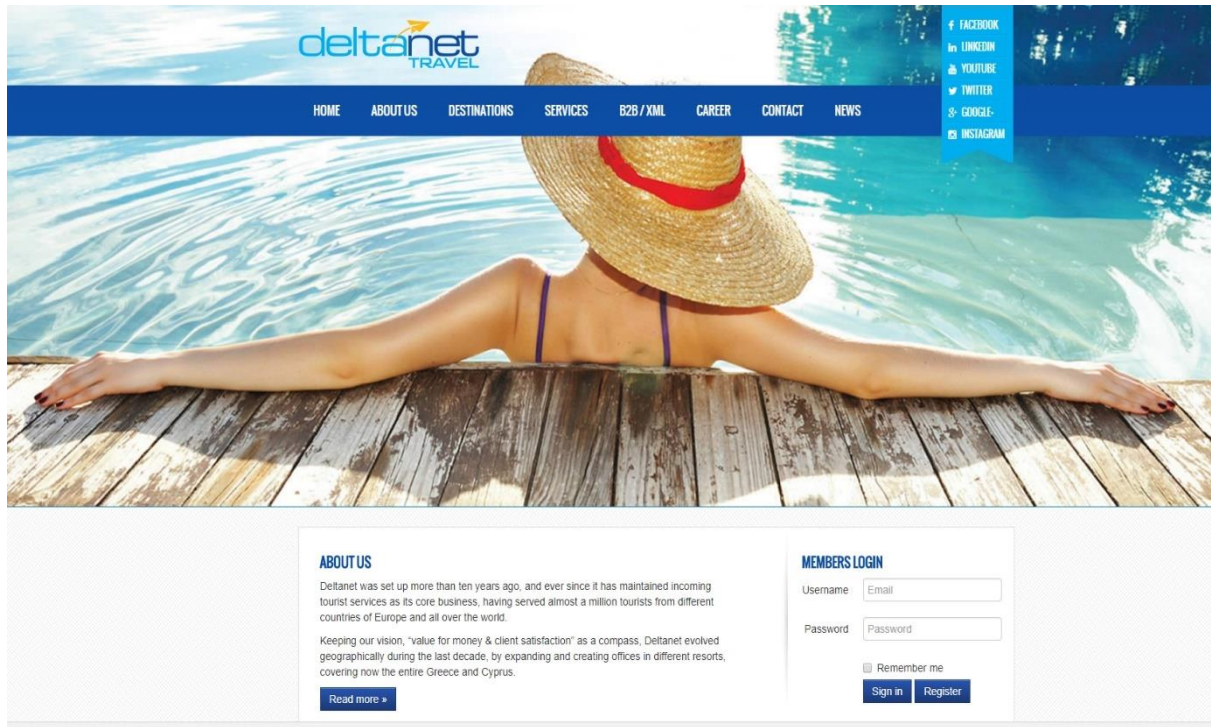


Figure 11. Deltanet using Laravel Framework

Neighborhood Lender

It is business website [9] that provides suggestions and plan of loan.

Depending on the capability of a person, various loan plans are suggested for buying a house in specific regions of United States. It provides a mortgage calculator on subscription which shows the wide range of functionality that is offered by Laravel.

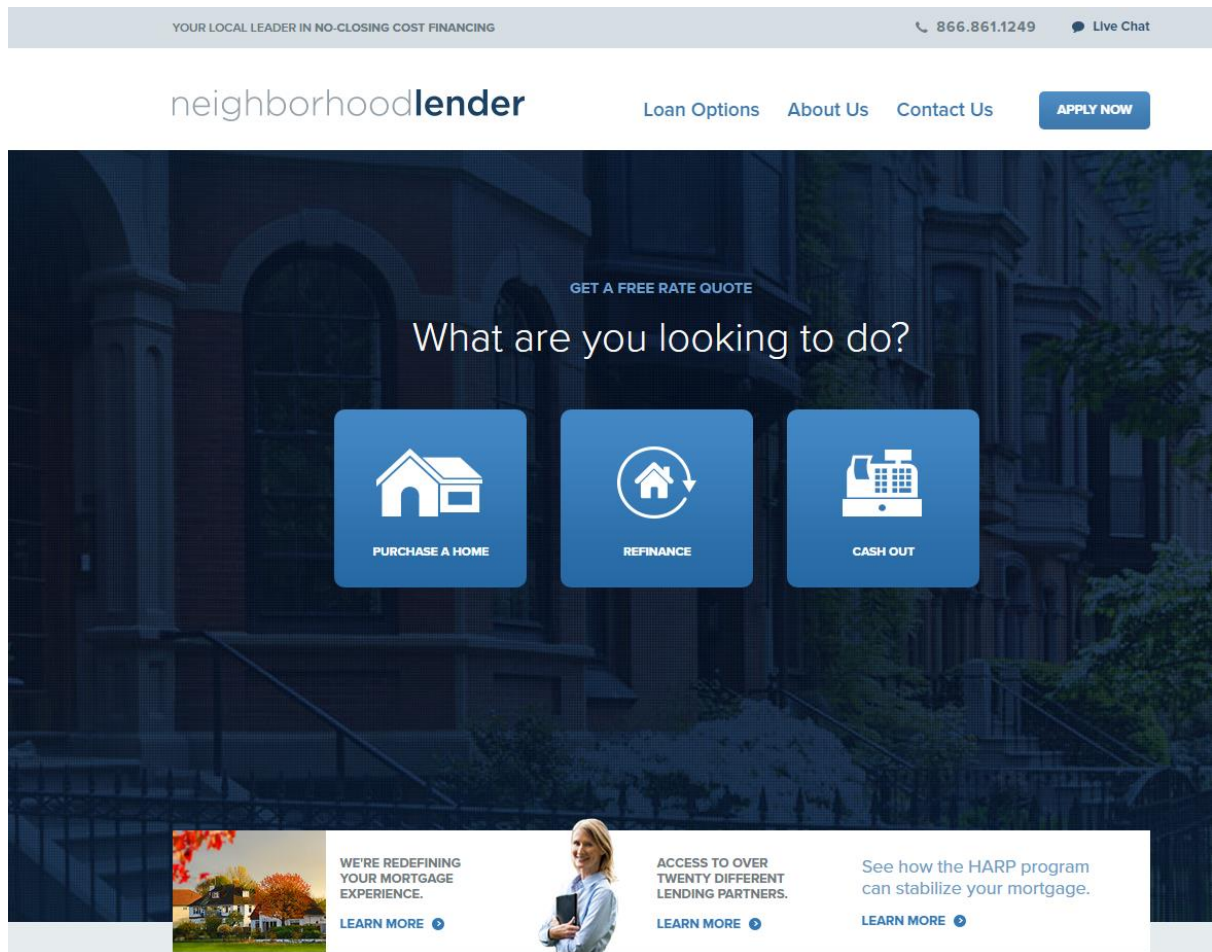


Figure 12. Neighborhood Lender using Laravel Framework

My Rank

It is a full scale online learning website [9] for students of under graduation in India developed with Laravel Framework. It is a self-assessment platform for students.

It helps the students with various subjects like Banking, Physics, Chemistry, Mathematics and Biology. It provides recorded classes and live classes depending upon the fees paid. The design of this website is amazing with option to view notifications such as exam details, counselling dates in different colleges.

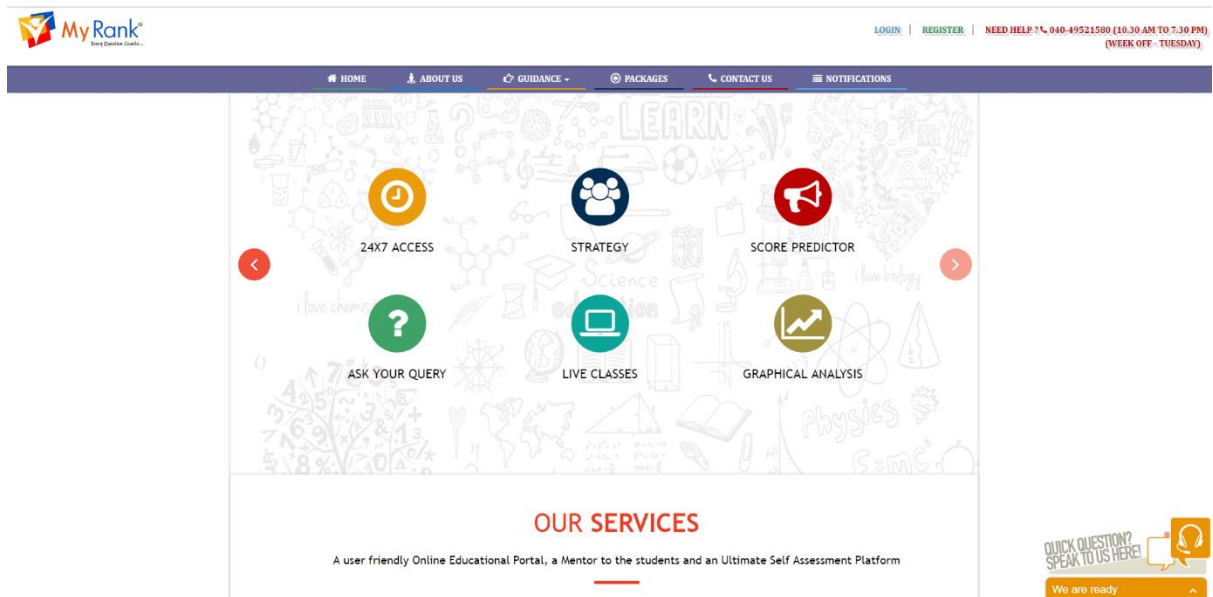


Figure 13. MyRank using Laravel Framework

Laravel Tricks

Laravel Tricks website [9] is built on Laravel has a great design and option panel. The Logged user can create his own trick by providing the code and sharing it with Laravel community. The tricks are nothing but the different ways of using Laravel.

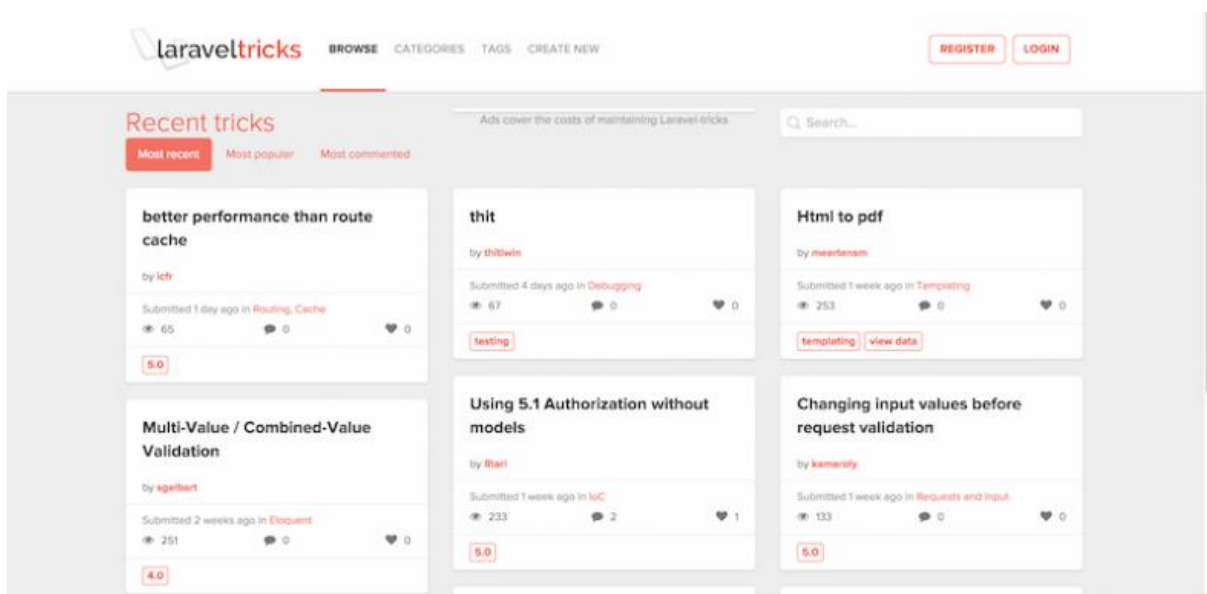


Figure 14. Laravel Tricks

LaravelIO

LaravelIO [9] is a forum community for Laravel developers, with platform built using Laravel framework. It is a forum that helps developers to share their code, share their queries with other developers who can provide solutions for it. It works similar to Stack overflow. The user need to have a GitHub account.

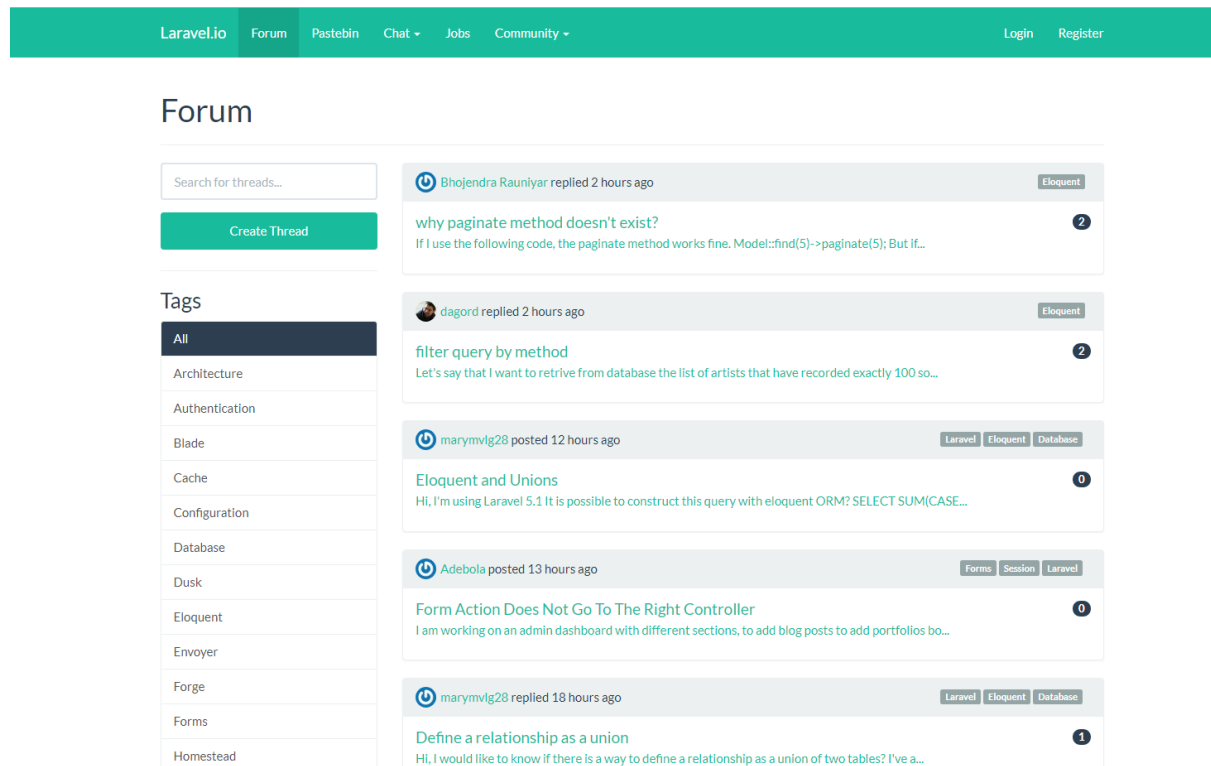


Figure 15. LaravelIO

Larasocial

Larasocial is a lightweight social networking platform [9] built using Laravel. It's an open-source project that's built upon a rich code foundation, and as it is a social network, it expects to find features such as send/decline friend requests, to be able to send messages between users (chatting), as well as private messaging and more.

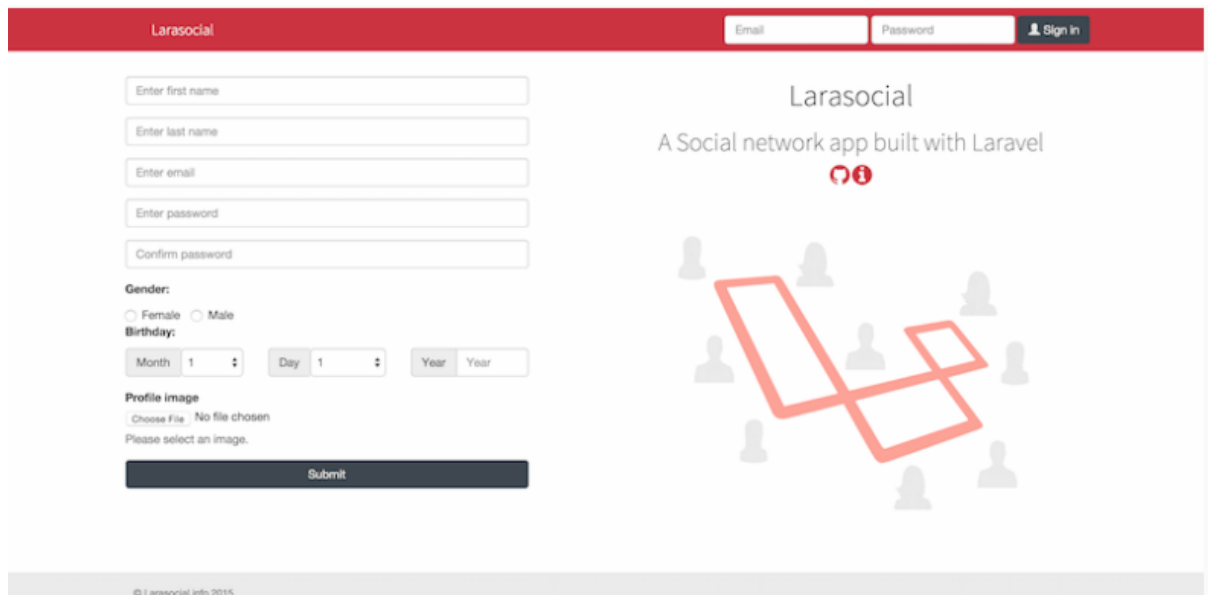


Figure 16. Larasocial

Cachet

Cachet [9] is built using Laravel that helps in solving various problems like website slowing down and improper functioning, by enabling status pages that can be used to see what services are performing optimally, and which aren't.

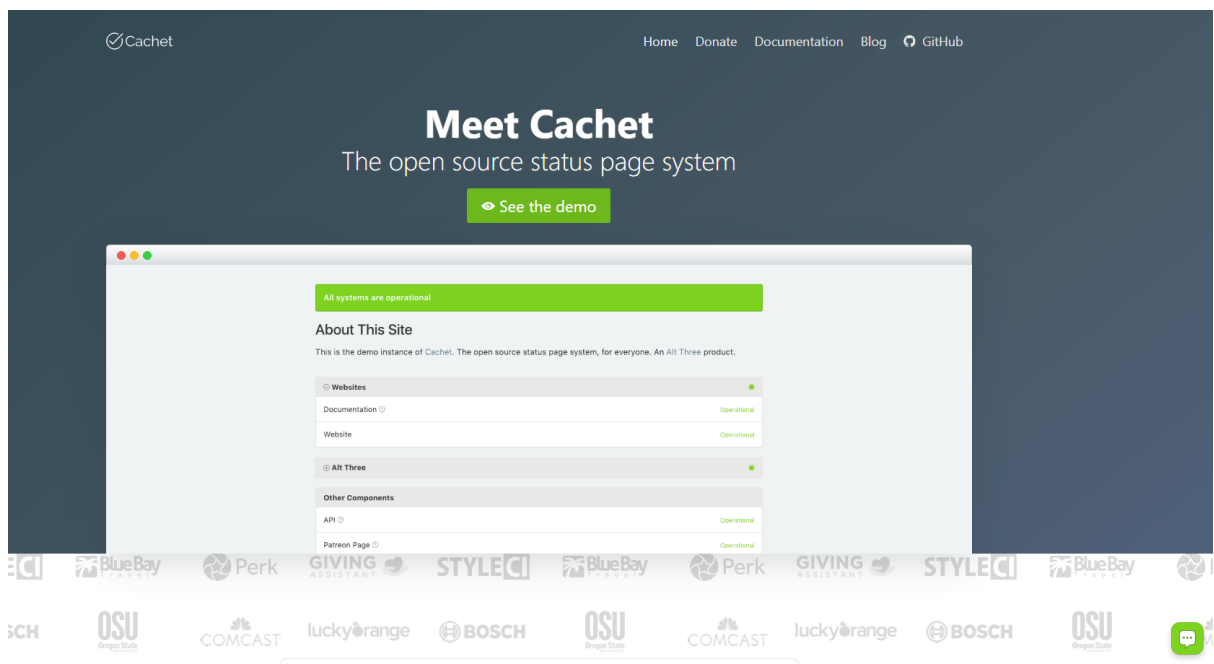


Figure 17. Cachet

Orchestra Platform

Orchestra Platform [9] is an admin panel that handles Laravel extensions and user management. This curation of Laravel extensions provide the boilerplate for any web application development to ensure a speedy deployment using the best tools available on Laravel.

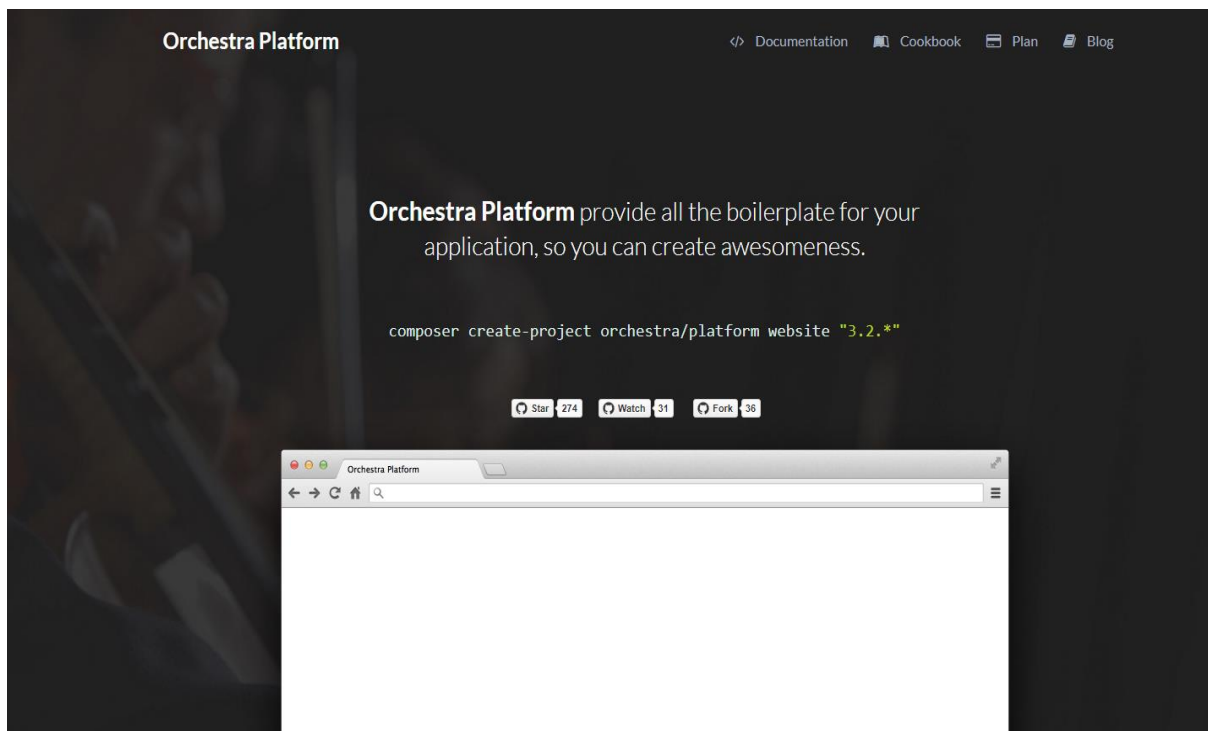


Figure 18. Orchestra Platform

Built with Laravel

This website has collection of open-source projects built with Laravel framework [10]

1. **Eloquent Sluggable:** A slug is a simplified version of a string, typically URL-friendly. The act of “slugging” a string usually involves converting it to one case, and removing any non-URL-friendly characters (spaces, accented letters,

ampersands, etc.). Slugs tend to be unique as well. Eloquent-Sluggable package for Laravel 5 aims to handle it automatically with minimum configuration.

2. **Laravel Vimeo:** Laravel vimeo is a Vimeo bridge for Laravel 5. It gives an easy way to handle Vimeo configuration like the client identifier and secret keys. The package includes a Facade and a contract which can be integrated with a project.
3. **Podcastwala:** Podcast web app is built with Laravel 5. This web app enables to manage Rich Site Summary (RSS) feeds for favourite podcasts and listen to the episodes in a seamless User Interface (UI). New episodes are automatically updated and can mark the items listened to as read and much more.
4. **Laravel Pusher:** A Pusher bridge for Laravel 5
5. **Laravel MongoDB:** An Eloquent Model and Query builder with support for MongoDB, using the original Laravel API. This library extends the original Laravel classes.
6. **AbuseIO:** Open Source abuse management for network operators, A toolkit to receive, process, correlate and notify end-users about abuse reports received by the network operators, typically hosting and access providers. AbuseIO's purpose is to consolidate efforts by various companies and individuals to automate and improve the abuse handling process.
7. **Scrumwala:** Scrumwala is a scrum/agile web app that lets you create and manage projects with plan and work views, group issues in a project into sprints, set deadlines for issues, active sprints and projects and get reminders by email listing issues nearing deadline.
8. **StyleCI:** It is powered by PHP CS Fixer, and is totally configurable via an optional style.yml file, committed to each repo analysed. StyleCI also provides some stand-alone packages that can be used outside the system, and is available as a free hosted service.
9. **Bugsnag Laravel:** The Bugsnag Notifier for Laravel gives you instant notification of errors and exceptions in Laravel PHP applications. It supports Laravel 5 and also lower versions of Laravel. Bugsnag captures errors in real-time from web, mobile and desktop applications, helping to understand and resolve them as fast as possible.
10. **Laravel Generators:** Custom Laravel generators is an easy way to speed up the workflow. Rather than opening the model's directory, creating a new file, saving it,

and adding the class. Some of the features include commands like `make:migration:schema`, `make:migration:pivot`, etc.

11. **Laravel Tagging:** This is the handy package that helps to add tagging ability to Laravel models using traits easily. Its flexible functionality means adding, removing and retrieving tags is a breeze.
12. **Laroute:** This package allows to port routes over to JavaScript, with familiar helper functions to use.
13. **Laravel Messenger:** Laravel Messenger allows to add a full user messaging system into Laravel application. It features multiple conversations per user, looping in additional users with each new message, returning the users unread message count easily and is flexible enough that we can implement our own access control.
14. **Hashids:** It is a Laravel 5 wrapper for the Hashids API. Hashids is a small open-source library that generates short, unique, non-sequential ids from numbers.
15. **Setting:** Persistent configuration settings allows to create, read, update and delete settings stored in files using JSON.
16. **Paperwork:** Paperwork aims to be an open-source alternative to service like Evernote, Microsoft OneNote or Google Keep. It is built on Laravel 4, also makes use of AngularJS and Bootstrap 3.
17. **Bootstrap CMS:** Bootstrap CMS pulls together packages including Laravel Core and Laravel Credentials to bring a Laravel powered CMS.

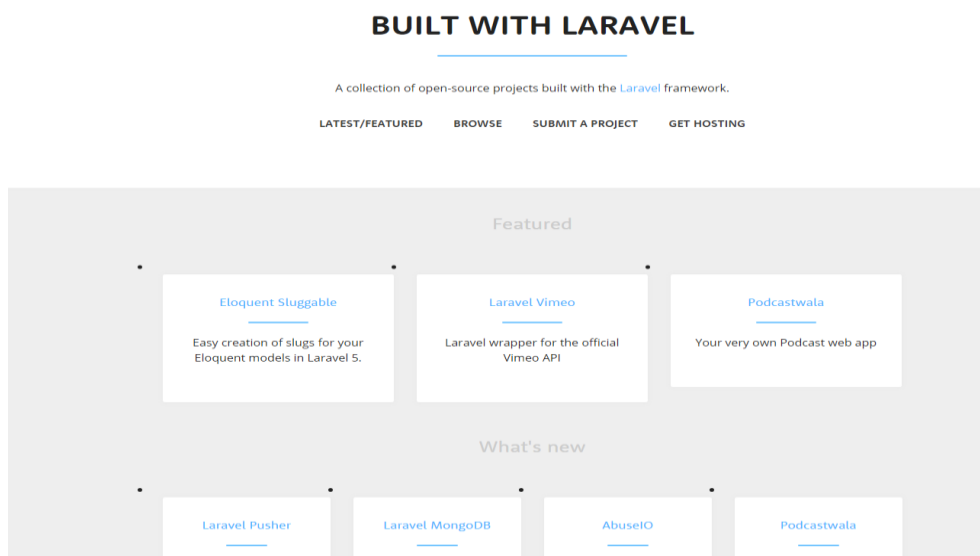


Figure 19. Built with Laravel

3. Framework review

3.1 Comparison of PHP frameworks and their advantages and disadvantages

Personal Home Page (PHP) is one of the most widely used programming languages, where recently PHP 7 has been made a server-side programming language. PHP is widely used in major projects. PHP framework makes the development faster with the Model View Controller (MVC) model and also considered to be more secure.

[11] In the past, PHP was simply mixed with HTML, which was then pre-processed on the server before being sent to the client's browser. It was known only for small projects. Today, PHP has been extended and embedded into various templating engines and web frameworks.

[12] More and more new projects began to use PHP frameworks and successful completion of these projects made PHP frameworks even more popular. Popular programming languages for website development are PHP, Ruby, Python and Java. Selecting from these languages PHP was chosen for framework analysis and web site development, because it is an open source scripting language with many advantages such as support for different database types, many language structures were taken from C and Perl languages. Since PHP is a popular and widely used language, it has a lot of documentation on the Web. From the figure below we can infer that PHP is used by 81.7% of all the websites whose server-side programming languages are known.

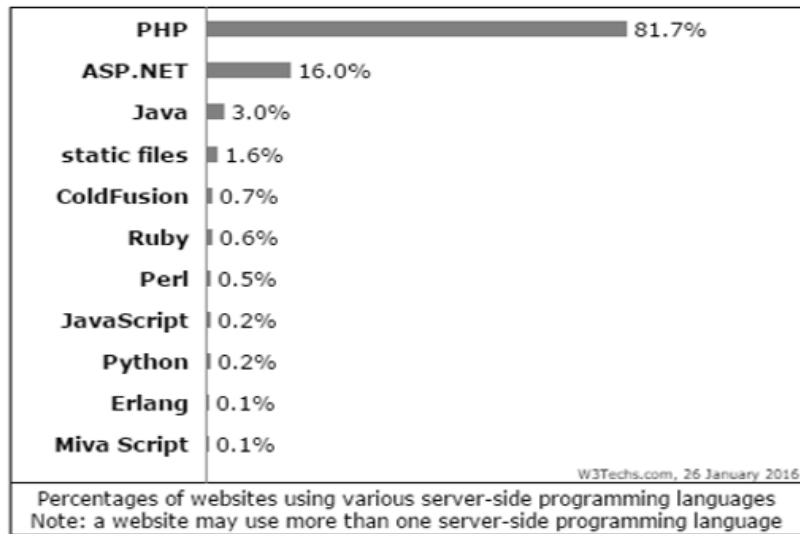


Figure 20. Usage of PHP

PHP frameworks helps for the speed development. It provides a well-organized, reusable and maintainable code. Frameworks can decrease the risk of low-level security of a site. The MVC (Model-View-Controller) pattern guarantees the separation of presentation and logic. The best PHP frameworks considered are Laravel, Symfony, CodeIgniter, Yii 2, CakePHP, Zend framework, Slim, FuelPHP, PHPPixie etc. The comparison is being made between few PHP frameworks (see Appendix C).

3.1.1 Symfony framework

Symfony is a set of recyclable PHP components that helps developer to make scalable, high-performance applications. The Symfony components are becoming a standard foundation for building the PHP applications. The developer is independent to use any components on their applications. [13] Symfony doesn't integrate any component to work with databases, it provides tight integration with a third-party library called Doctrine. Doctrine provides powerful tools to make database interactions easy and flexible. Symfony framework consists of multiple tools such as code generation tools, testing tools. It has its templating language known as Twig. It follows M-V-C Architecture, with Object oriented programming and has large library. It has Open source MIT Licence.

Advantages

- Symfony framework is a backbone of the application that defines the architecture, style, and the method for solving problems.
- It has a collection of ready to use development code.
- Code generation tools are used for prototyping and one-click back-end administration.
- It consists of tools for unit and functional testing and hence provides test-driven development.
- Simple to use, in most cases, but still flexible enough to adapt to complex cases.
- Easy installation and configuration on all platforms.
- Easy to extend, allowing for integration with other vendor libraries.

Disadvantages

- Users find it difficult to use security mechanisms of this framework.
- Need to spend lot of time on documentation before working with this framework.
- Parsing of a file is hard.

3.1.2 Yii framework

Created by Qiang Xue in 2008, Yii stands for Yes, it is! [14] Yii is a secure, fast, high-performance application/web-development framework. Yii utilizes the Composer dependency manager for PHP for handling different dependencies and installations. Yii is the fastest PHP framework with the existence of Code generator that can benefit speed up the work process developers in creating applications. Yii is equipped with a Web-based code generator called Gii. It utilizes components for making the development faster. It follows M-V-C Architecture with Object oriented programming. It has a powerful console tool Yii CLI (yiic). It practices Database Access Objects(DAO), Active Record (AR) for the Object Relational Mapping(ORM) layer. It has Open source (BSD) license.

Advantages

- It has Caching support and hence it is appropriate for developing all types of web applications such as portals, forums, (Representational State Transfer) RESTful services, content management systems, e-commerce websites etc.
- It offers maximum reusability.
- It has a great feature of jQuery integration which enables the developers to implement this framework.
- Uses scaffolding to generate code.
- HTML helper generates tags from the code.
- It is explicitly designed and has Asynchronous JavaScript and XML (AJAX) helpers to work efficiently with AJAX.
- Yii framework provides a great security. It includes input validation, SQL injection, Output filtering and Cross-site scripting prevention.

Disadvantages

- AR doesn't support many doctrines like AR queries
- The ajax functions for various things, like the grid is not that well documented so often you need to explore the JavaScript code.
- It lacks support of needed extensions to create complex applications

3.1.3 Code Igniter

CodeIgniter [15] is an Application Development Framework, a toolkit to build websites using PHP. It has a rich set of functionalities that turns out the development to be faster. It also has a rich set of libraries for commonly needed tasks, as well as a simple interface and logical structure to access these libraries. It minimizes the amount of code needed for a given task. It also provides security by preventing various attacks. It has Open Source License but not complete Open source. It is Object Oriented and follows M-V-C Architecture.

Advantages

- Flexible with M-V-C Architecture and provides good performance.
- Good documentation making easier to learn, adopt and deploy.
- Active Record Implementation for Object Relational Mapping(ORM) layer.
- Code generated using Combustor.

- Migrations from one server hosting to another is done with no much difficulty.
- Good community support.

Disadvantages

- It is company driven.
- Irregular releases.
- Framework does not have its own ORM, Only via third-party.
- No console tools.
- No automatic generation for any functionality.
- Has several libraries, from which a set of classes that would compose the library responsible with the authorizing is absent.

3.1.4 CakePHP

Using CakePHP framework [16], the developers can build web applications simpler and faster, with less code requirement. CakePHP has number of libraries to reduce the overload of most common tasks. This web application helps in reducing the overall development costs. It has Open source (MIT) license. It also has M-V-C Architecture.

Advantages

- Scaffolding code generation.
- Working with classes is made easier using this framework.
- Its automated configuration process auto-detects preferred settings and hence there is no need to configure LAMP setup.
- Plugins available that helps to keep the code clean.
- ORM helps developers to create excellent queries and codes.
- Console tool: cake.
- Cake schema is used for database backup/restore functionality.
- Good community support.

Disadvantages

- Documentation lacks clarity.
- Codes are bulky.

- Slow in performance.
- The developer needs to update the default routes for creating fancy URLs.
- One-way routing
- Does not support natively SQLite3

3.1.5 Zend

Zend framework [17] is an open source, object oriented web application framework for PHP5. Zend framework is often called as ‘component library’, because it has many loosely coupled components that you can use more or less independently. But It also provides an advanced Model-View Controller implementation that can be used to establish a basic structure for this framework applications. It has a console tool for Rapid development.

- SQL is not generated automatically
- CRUD functions are not created automatically
- Poor documentation

Advantages [18]

- Facilitates the organisation of the source code into three distinct structures: presentation, logic, and data, which leads to a fair tasks distribution in the process of developing a software application.
- Allows code reusability, by dividing it into components with little or no dependencies, which offers the possibility of including them into other applications.
- Offers high quality code, due to its strict standards and to the permanent testing and reviewing by the large community of users, according to the Contributor Guide, which states that code **MUST** be tested and covered by a unit test using PHPUnit and must adhere to strict Coding Standards.
- Helps developers in producing scalable and secure web applications and services, due to its components that incorporate essential functions for a site and to the ones that have a high degree of customization, providing tools and measures to prevent and protect data.
- Zend Framework is a fully object-oriented framework, which makes most of all the ZF’s components extensible, according to the inheritance concept. It allows

developers to customize components in order to create functionality that is unique to their projects and also to use them in other projects as well.

- ZF is a “glue” framework, which unlike the full-stack frameworks that impose strict conventions on a project, is more flexible, allowing the use of components as individual libraries instead of the whole framework and also the integration of other libraries.
- ZF uses test-driven development, a programming technique that supposes writing the tests for a function before writing the code, which ensures the coverage of all the use case scenarios.
- Supports multiple database systems, including: MySQL, Oracle, IBMDB, MSSQL Server, PostgreSQL, SQLLITE etc.

Disadvantages

[18] At first sight, Zend Framework doesn't seem to be very friendly, mostly because:

- It is big- it is composed of a large collection of classes, useful, but very hard to understand. It may be even impossible to be learned by a novice in web development
- It takes quite enough resources (for example, the last version, that is 2.2.5 in ZIP format has 3.04MB). This disadvantage gives more credit to the affirmation “software is more expensive than hardware is”.
- It requires previous knowledges about design patterns – it is known the fact that Singleton Pattern was the foundation of the first version of this PHP Framework. Singleton is implemented by the Front Controller, which is used inside the Model-View-Controller applications.
- It also requires previous Object-Oriented-Programming knowledges because all its components are entirely object-oriented and to be more precise, all its components are entirely PHP5 object-oriented.
- It has a substantial documentation, which is not such a bad thing because it is always better to have a choice than to have no information to use or to help whenever it is needed. The large number of pages from manuals, that are available on official websites, can be a sure source of intimidation and they can even make somebody turn around and give away the attempt to learn this PHP and MVC framework

- The update is not automatic - the website administrator or a web developer has to be careful when making the update whenever is needed. In Windows for example, the update and even the install needs to be made in command prompt(console)
- The update requires a lot of attention because sometimes certain aspects of the framework's behaviour from the previous version can suffer major changes or even completely disappear in the current version. For example: Zend Framework 2 was completely rewritten and its Model-View-Controller layer suffered several changes, but despite these things, a projected built in the first version of Zend Framework must be able, and it does, to migrate to the latest one, which is currently 2.2.5.
- The web developers are creative people and they are often solving the same problem with different choices because they like variety; sometimes the web developers chose to ignore the framework's suggestions and to follow their instinct taking decisions on their own and writing the code that they want. In some cases, the framework considers these decisions strange and rejects them; In conclusion, a framework isn't really a suitable for people who want to write their own solutions for things that Zend already has;
- Some developers consider Zend Framework to be more a View-Controller rather than a Model-View-Controller and this is due to the fact that Zend doesn't come with any model implemented letting the developers to be the first ones to implement one.
- It isn't suitable for any kind of application- it is rather appropriate for large websites because it has a large number of classes, which are included and used very often, for everything.

3.1.6 Kohana

Kohana framework [19] is an Open source with(BSD) License which was originally based on Code Igniter. It is Object oriented with Hierarchical model-view-controller(H-M-V-C) Architecture. Kohana Module System is intended to make modules portable. It can be considered at any level in the Cascading Filesystem. The process of importing module consists of Copy, Paste and Adding definition to bootstrap.php. Development is fast with Kohana as it offers the basic constructs of inheritance, polymorphism, and encapsulation from object-oriented programming and applies them to presentation layer development. Components are encapsulated. They are self-contained and reusable units

of an application. Kohana comprises a set of prebuilt components. Components can be assembled and configured to form new components.

Advantages

- Routing offers an easy technique to direct requests to explicit controllers.
- Supports regex pattern matching.
- Sub patterns on Routes can similarly be made non-compulsory making Routes more dynamic.
- CLI (command line interface) script creation with minion module.
- Robust ORM Engine.
- In Built templating.
- It does not limit access to global variables (GET, POST, COOKIE, and SESSION).
- Scalability- Cascading Filesystem is overwhelming, but also very taxing on high traffic sites. Pre-caching the file lookups as part of the production release script will intensely upsurge the performance.

Disadvantages

- Does not support SQLite3 natively.
- No console tools.
- No automatic generation of any functionality.
- Documentation is poor.

3.1.7 Aura

Aura [20] is a UI framework for developing dynamic web applications for mobile and desktops devices. Aura provides a scalable long-lived lifecycle to support building applications engineered for growth. Aura supports partitioned multi-tier component development that bridges the client and server. It uses JavaScript on the client side and Java on the server side.

Advantages

- Out of Box Components
- Event driven

- Device aware and Cross Browser Compatibility
- Supports multi-tier component development for bridging the gap between client and server.
- Optimizes apps for different devices
- Performance- Stateful Client and Stateless Server-side
- Uses JSON to communicate between client and the server
- Client calls the server only if necessary to get metadata

Disadvantages

- Documentation is not good.

Events in the frameworks are of two types. Component events are held by the component itself or a component that instantiates or holds the component. Application events are primarily a traditional publish-subscribe model. Entire components that provide a handler for the event are notified once the event is fired.

3.1.8 Slim

Slim [21] is one of the most popular PHP microframeworks which is suitable for developing small web applications as it does not have all the components, libraries, helpers and structures like found in full-stack frameworks.

Advantages

- Built with excellently robust routing system with all HTTP methods like GET, POST, DELETE, PUT
- Plenty of helper methods for easier caching, and session support.
- Works on PHP 5.2+ version and above and better known for as lightweight and micro-framework
- Supports PSR-7 HTTP message implementation
- Provides complete control of external tools to users

Disadvantages

- When working with REST API (Application Programming Interface), this framework forces to use a single .php file which can put load on the server if the .php file is large

3.1.9 Silex

Silex framework [22] belongs to microframeworks. It is likely to be extended into a full M-V-C framework by picking one of the versions given below:

Fat- Derived of all features, a template engine and database abstraction

Slim- Only comprises of routing engine.

- It is based on Symfony and a Twig template engine
- It is extremely lightweight and permits to include extra features as required

Advantages

- It is light weight.
- Creation of REST URL is easier.
- It provides security.
- The code is well structured.
- Good performance.
- It has Platform independence, cloud support, window support, NoSQL support.
- It is easy to use.
- It has dependency injection pimple and console component symfony2 and hence testing applications is made easier.

Disadvantages

- Config file should be modified.

3.2 Choice of Technology

Choosing a framework is a chief task because the quality and speed of the application built depends on it. When choosing a framework, we should consider the main features [12] like license, usage context, software pattern, hosting requirements, ease of installation, core library, learning curve, DB abstraction and ORM, included JS libraries etc.

From my point of view, Laravel makes developing easier for the person who does not have enough knowledge in developing Web application. The tutorials provided for using Laravel is very helpful to understand and learn.

Laravel has several security features and methods such as Auth :: Attempt method which takes user email and password as arguments, check () method is used to check if the data exists in the database, and thus makes authentication process easier.

Laravel uses middleware, they are created using artisan command. 'auth' middleware is created when default authentication is created. It helps to block the access to the pages inside the Web application if the user is not logged in. There are several other middleware's such as 'web' that takes care of sessions, cookies, verification of Cross Site Request Forgery (CSRF) tokens etc. Laravel has a default way of creating Login and Register, which can be customized in a required way.

3.2.1 Laravel framework

Laravel framework is an (Model View Controller) M-V-C framework with an expressive and elegant syntax. It is an Open source framework with very rich set of features which will boost the speed of Web Development. It gives solution to the development by facilitating general tasks.

Laravel was chosen because it has various functions like Bundles, Migrations, Eloquent ORM, Application Logic, Reverse Routing, Class Auto Loading, etc. It also has its own Command Line Interface called Artisan. Laravel assures that an application built on it is secure as it prevents various attacks on web sites.

[23] Laravel features

- Authorization technique

The implementation of authentication techniques is made very simple and also a simple way to organize authorization logic and control access to resources is provided by Laravel.

- Object Oriented libraries

Object oriented libraries and many pre-installed libraries which are usually not found in other PHP frameworks are found in Laravel. One of them is the Authentication library with many advanced features like checking active users, Bcrypt hashing, Password reset, CSRF (Cross-site Request Forgery) protection and encryption.

This framework is separated into individual modules that implement modern PHP principles consenting developers to build responsive, modular and convenient web applications.

- Artisan

Laravel includes a built-in tool named as Artisan. A developer needs to interact with a Laravel framework using the command line called as Artisan that creates and handles the project in the environment. This tool permits us to accomplish the common repetitive and tedious programming tasks that most of the developers bypass to perform them manually.

Artisan can also be applied to make the database structure, a skeleton code, and build the migrations that turn out to be easier to manage the database system.

- MVC support

It supports MVC architecture similar to Symfony framework confirming clearness between logic and presentation. MVC assists in improving the performance, permits improved documentation, and has multiple built-in functionalities.

- Security

Laravel takes care of security inside the framework. It makes use of salted and hashed passwords, which means that the passwords would never be saved as the plain text in database. It practises Bcrypt hashing algorithm for the encryption of a password. Laravel uses prepared SQL statements which make injection incredible. Laravel also provides a simple way to escape user input to avoid user injection of <script> tag.

- Database Migration

Keeping database in sync with the development machines is a hectic job for the developers, which is made easier by the Laravel framework. After a long working hour, the changes may have been made to the database, MySQL workbench is not a great approach to sync databases between the development machines. As far as all the database work is in migrations and seeds, it is easy to migrate the changes into any other development machine. This is another reason which makes Laravel the best PHP framework.

- Blade Templating Engine

The Blade templating engine is very intuitive and assists working with the typical PHP/ HTML combination in a better way.

- Query Builder

The database query builder offers a suitable, smooth interface for creating and running database queries.

- HTTP Middleware

It affords to be an appropriate mechanism for filtering HTTP requests entering the application.

- Eloquent ORM (object relational mapping)

Laravel framework proposes the Eloquent framework that comprises of a simple PHP Active Record implementation. It allows the Web Application developers issue database queries with PHP syntax rather than writing SQL code. An ORM is quite faster than the other PHP frameworks.

[24] **Laravel Framework solves common Web Development Problems:**

- It makes the implementation of authentication a simple process. Almost all the things are configured out-of-the-box. It also offers a simple method to organize authorization logic and control access to resources. Hence it solves Authentication and authorization complexity.
- Laravel delivers a clean, simple API over the widespread SwiftMailer library. Laravel also affords drivers for SMTP, Mailgun, Mandrill, SparkPost, Amazon SES, PHP's "mail" function, and "sendmail", letting an application to rapidly start sending mail via a local or cloud based service.

- Laravel integrates with tools and makes the development of web applications quick. It also supports prevalent cache backends like Memcached and Redis out-of-the-box. Multiple cache configurations can be done using this Framework.
- Common Technical Vulnerabilities are fixed as it protects from the most common and serious security risks like SQL injection, cross-site request forgery, and cross-site scripting. Hence this Framework is considered to be secure one.
- Configuration of Error and Exception handling is done by default in a new Laravel-based project. It is integrated with the Monolog logging library, that delivers support for a variety of powerful log handlers.
- Provision for testing with PHPUnit is comprised out-of-the-box, and a phpunit.xml file is previously setup for the application. It provides easy ways for simulating basic behaviour of users (making requests to the application and examining the output, for example, clicking links, filling out forms).
- All the routes defined in the routes file are automatically loaded by Laravel. It provides a simple method for defining routes, basic routes simple accepts a URI and a Closure.
- M-V-C architecture separates the Business Logic Code from Presentation Code.
- Message Queue System (Delayed Delivery) Configuration in Laravel, where the queue service provides a unified API across a variety of different queue backends.
- The command scheduler in Laravel permits for the easy and expressive defining of command schedule within Laravel itself, and only a Single Cron entry is needed on the server.

4. Architecture

4.1 MVC Architecture

The Model-View Controller (MVC) pattern is a technique for categorising an application into three separate components: Model, View and Controller. This architecture solved the difficulties of accomplishing with the static Hyper-Text Markup Language (HTML) pages. It is appropriate for separating the modelling of the domain, presentation and the actions into three isolated classes built on user inputs. The aim of MVC design pattern is to isolate the application object (model) from the manner it is represented to the user (view) and from the manner in which the user controls it (controller). This Architecture makes it easier for the developer.

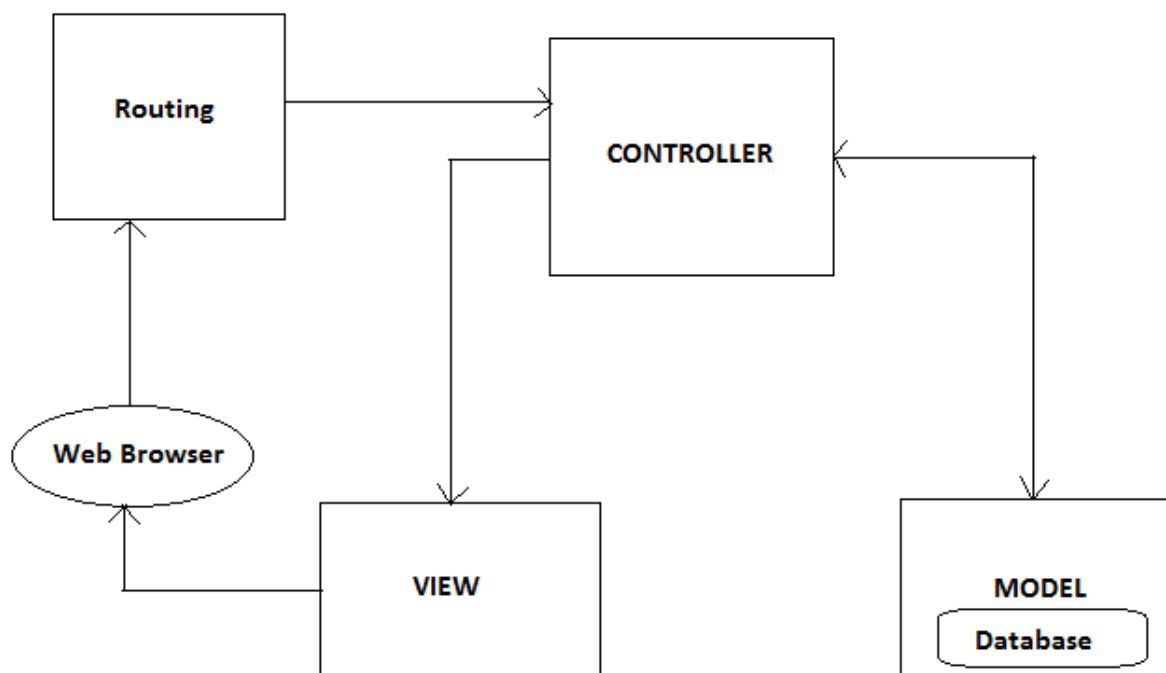


Figure 21. M-V-C Architecture

Model: The model manages the actions and data of the application domain. Model is usually found outside of an Application which consists of a Database. The model recognises all the data that need to be displayed. Also knows the application of operations in order to transform

that object. But, the data are retrieved and operated through approaches independent of the Graphical User Interface (GUI). i.e. it does not know about the GUI that manipulated the data which are to be displayed. The model signifies enterprise data and the business rules that rule access to and updates of this data.

View: The view manages the information in the display. The view object makes use of the query methods to obtain data from the model and then displays the information. It extracts the model contents through the Controller. Controller accesses the enterprise data through the model and View postulates the data presentation. It sustains the consistency in its presentation during the changes in the model.

Controller: The controller reads the mouse and keyboard inputs from the user in the View, and provides the information to the model. Model then sends the response back. It translates the relations with the view into actions to be achieved by the model. The user interactions could be the button clicks or menu selections or GET and POST HTTP requests as in Web applications. Grounded on the user interactions and the model actions, the controller returns by selecting an appropriate view. As in the controller is answerable for updating a particular parameter in the model that is then displayed in the view. Each controller-view pair is linked with only one model. However, a particular model can have several view-controller pairs.

From the above figure, we can infer that Controller and View depends on the Model. In Web applications, the separation between view (browser) and controller (server-side components handling HTTP request) is well-defined.

Routing: In Laravel, the routes define the connection between the controller and the view. When the User sends a request through the Web browser, Web server receives the request which is then passed on to the Laravel Routing Engine. The Router receives the request and redirects to the suitable controller class method.

4.2 Laravel Architecture

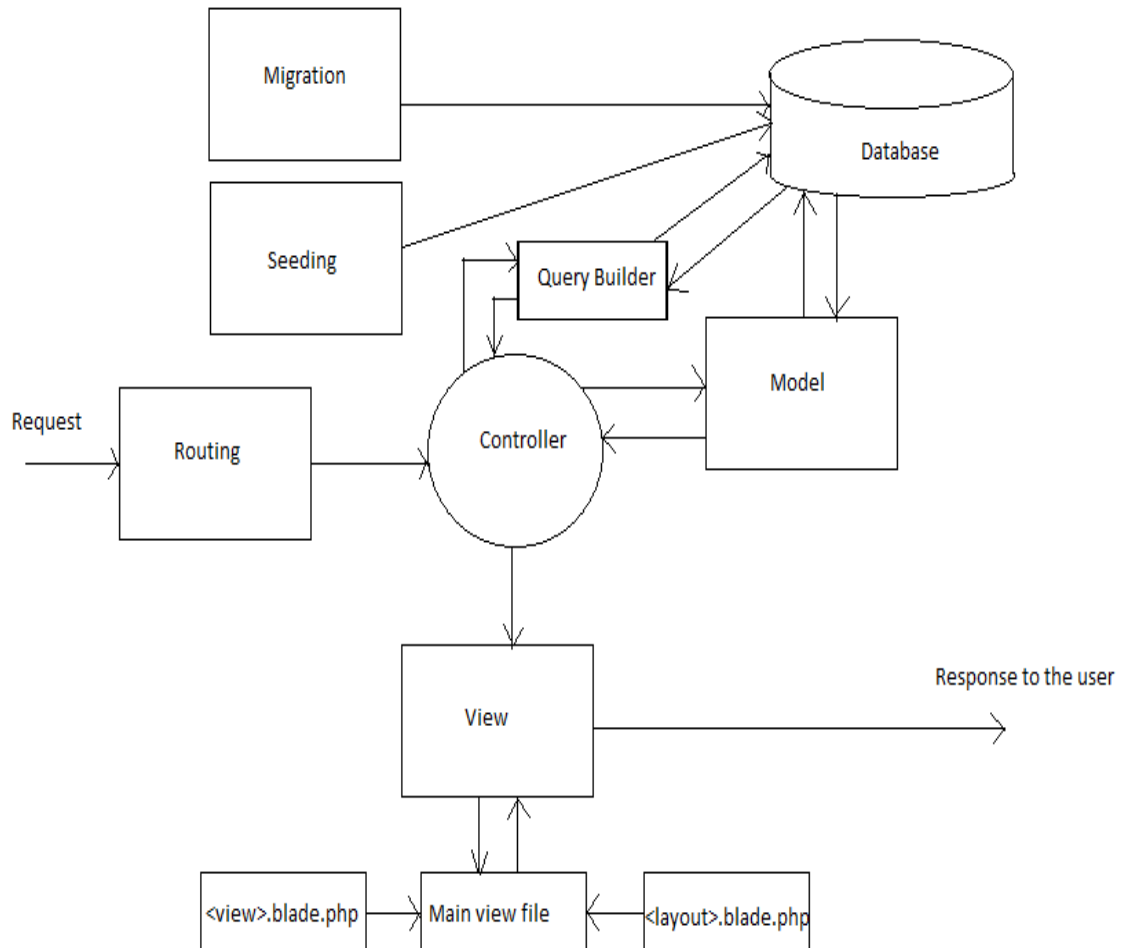


Figure 22. Laravel framework Architecture

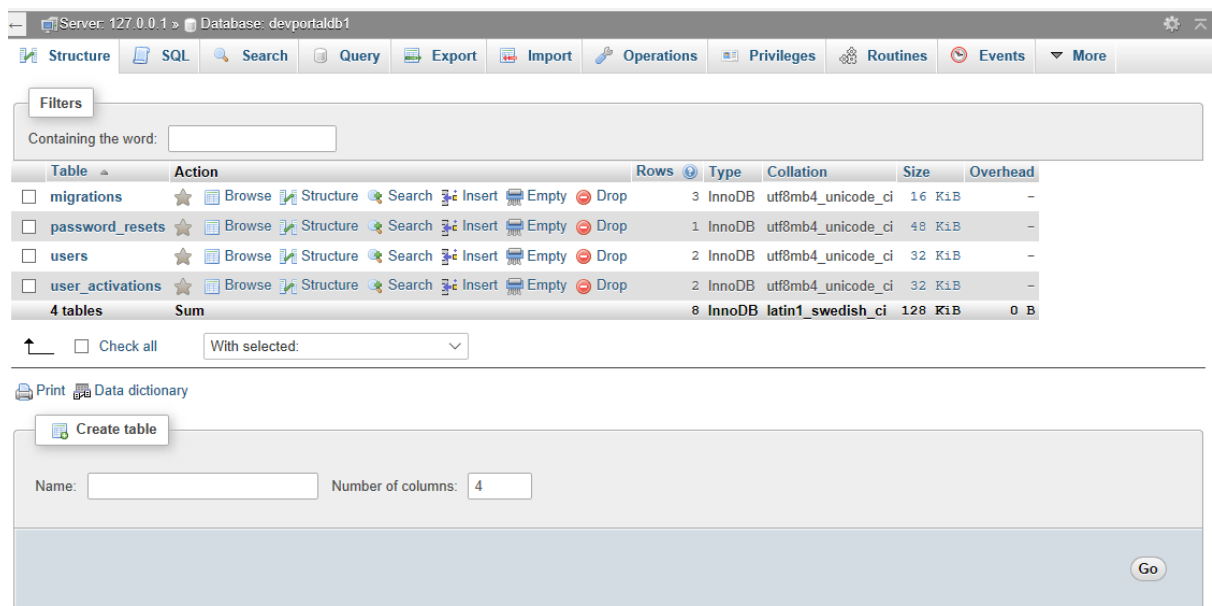
Migration: It helps to create schema [27] for database using schema builder. Interactions with migration is made through the artisan command. Each of the migration command has up and down functions to migrate and rollback respectively. Up function performs the required operations in order to put the database schema in its new state. Down function performs the reverse operations of up function. Migrations provides version control to the database and has other benefits like it permits to add and remove columns while maintaining control. Transition to production environment becomes smooth. Automatic database seeding can be done.

For the User Authentication in the Quantum Portal, Three Tables were created and migrated.

1. Users table
2. Password Resets table
3. User Activation table

User table included the User data such as First name, Last name, Telephone, Email, Password, Remember Token, Time stamps etc. Password Resets contained Email, Token and Created at Time stamp. User Activation table contained User Id, Token, Created at Time stamp and Is Activated column with Boolean value to check whether the User is activated or not.

Using the artisan command for Migration, the required Columns were added to the MySQL Database.



The screenshot shows the MySQL Database Structure tool interface. At the top, it displays 'Server: 127.0.0.1' and 'Database: devportaldb1'. Below the navigation tabs (Structure, SQL, Search, Query, Export, Import, Operations, Privileges, Routines, Events, More), there is a 'Filters' section with a search box. The main table lists the database tables with columns: Table, Action, Rows, Type, Collation, Size, and Overhead. The tables listed are 'migrations', 'password_resets', 'users', and 'user_activations'. Below the table, there is a 'Check all' checkbox and a 'With selected:' dropdown. At the bottom, there is a 'Create table' button and a form with 'Name:' and 'Number of columns:' fields, with '4' entered in the latter. A 'Go' button is at the bottom right.

Table	Action	Rows	Type	Collation	Size	Overhead
migrations	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_unicode_ci	16 KiB	-
password_resets	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	48 KiB	-
users	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode_ci	32 KiB	-
user_activations	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode_ci	32 KiB	-
4 tables	Sum	8	InnoDB	latin1_swedish_ci	128 KiB	0 B

Figure 23. MySQL Database tables

Seeding: It helps to populate testing data to database. Seeds are used to insert dummy data into the tables, so that we are able to create test users, test products etc in the development environment.

Query builder: The query builder offers a direct database access. Laravel at present supports following 4 databases:

- MySQL
- SQLite

- Postgres
- SQL Server

The query can be excited to the database using the raw SQL, which is the fluent query builder, and the Eloquent ORM considered to be effective.

We made use of MySQL Database for testing the Portal in the Local Environment.

Routing: It provides simple routing. Handles the request based on URLs.

4.3 Quantum Node Portal Architecture

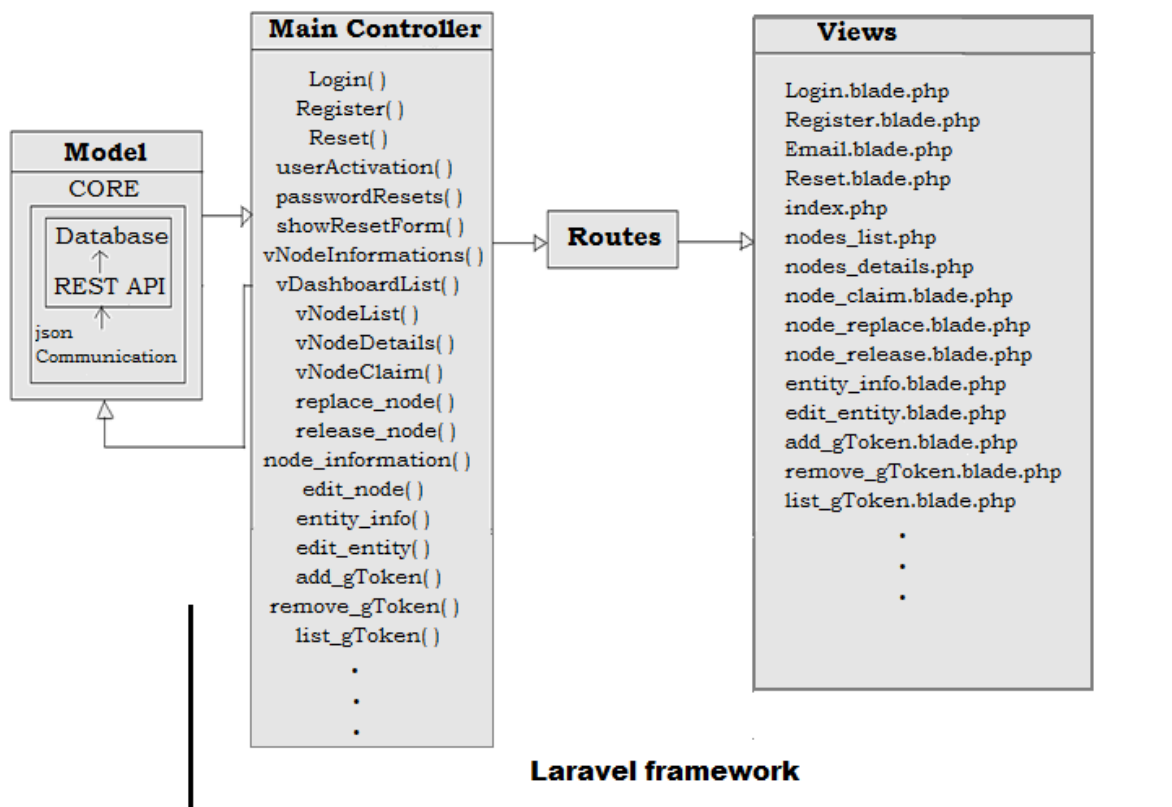


Figure 24. Quantum Node Portal Architecture

Quantum Node Portal is defined by the Three main components such as: Model, Main Controller and Views.

Model: It is the Model Domain around which the Quantum Node Portal is built. It consists of the CORE elements required to make our Portal a real-time application. Our Database has all the data required by the clients of the Portal. We use Representation State Transfer (REST) API services to send and retrieve the data from the database. The communication is in the form of JSON.

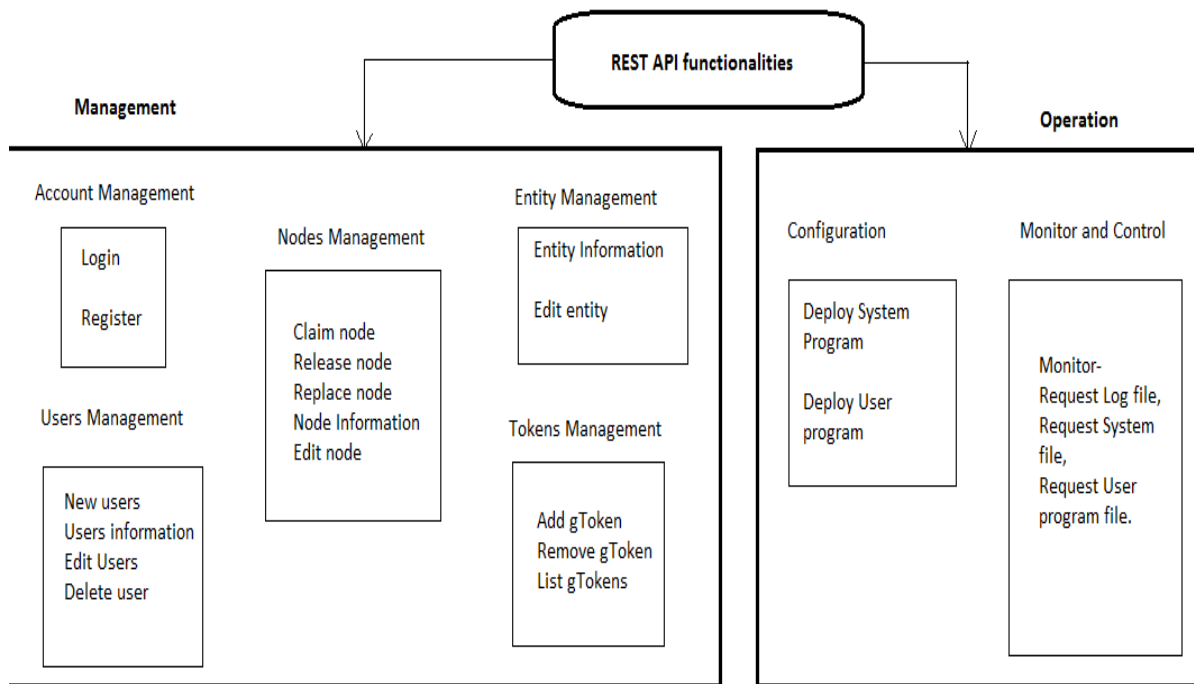


Figure 25. REST API functionalities

The REST API (Application Programming Interface) consists of various functionalities for Management and Operation.

All of the functionalities include at least one of the Authentication methods as discussed earlier.

- Account Management- Login and Register.
- Users Management- New Users, Users Information, Edit Users, Delete Users.
- Nodes Management- Claim, Release, Replace, Node Information, Edit Node.
- Entity Management- Edit entity and entity information.
- Tokens Management- Add, Remove and List gTokens (gateway Tokens).

- Configuration Operation- Deploy System program and User Program.
- Monitor and Control-Requesting the Log file, System File, User program file.

Main Controller: Main Controller organises the connection between the View and the Model. The Main Controller class consists of all the methods required by the Quantum Portal to work as shown in figure 26.

View: It provides the User Interface for the clients. The Client user makes a request through the Web browser. Routes Engine of Laravel connects the Main Controller with the View. Router manages the routes and the requests made by the Clients are redirected to the appropriate Controller method.

5. Project Methodology

Managing the project is an important part while building a project and to complete it. This chapter introduces the methodology used, the work flow and how it is adopted in our project. In order to develop the project in a systematic way, it is necessary to implement a method that helps to reach the target within the predefined time duration, and to meet the requirements.

5.1 Scrum Methodology

Scrum is an Agile-based methodology that comprises of an iterative, incremental, and empiric process. It is implemented to have a focus, clarity and transparency to the activities in the project.

Scrum [25] has three main roles: Product Owner, Scrum Master and Scrum Team.

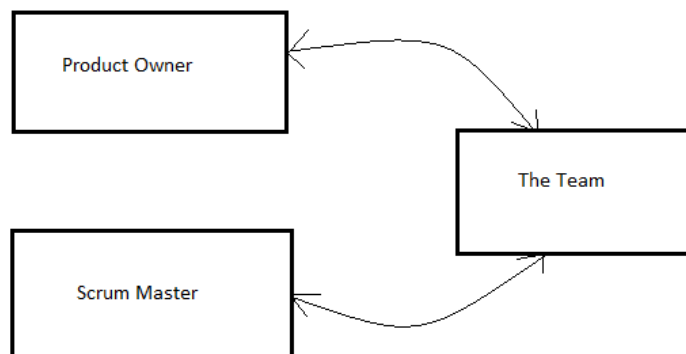


Figure 26. Scrum Roles

A. Product Owner

It is a person responsible for taking the inputs from the customers, team members, end users and stakeholders and defining features of the product, desired outcomes. He takes the responsibility of the complete project.

In Our project, Our Company CEO is the Product Owner. He keeps the requirements of our Web application Quantum Portal in front.

B. Scrum Master

Scrum master helps the team to accomplish the assigned task. He removes interferences and blocks to the team's success, facilitates meetings and oversees the implementation of Scrum.

In Our project, Our Team Leader has played a role of a Scrum Master. We have team meetings, and the tasks are divided among the team members. The time is given to complete the tasks on weekly basis.

C. Team Members

The members of the team self-organize and work towards implementation of the assigned task.

We are a team of three members, with each one contributing on the Project. I am responsible for developing the front-end of the application using Laravel Technology, Making use of Xampp with Apache and MySQL database. I develop and conduct the tests locally. I am accompanied by another member. The third member dealt with back-end developing dealing with Representational State Transfer (REST) services and database.

Comparing the Scrum Roles with Proposed Web application development Roles.

The table describes the Scrum roles, Proposed roles responsible for the development of Quantum Portal and the responsibilities of each member in team.

Scrum Roles	Responsibilities	Proposed Roles in Quantum Portal development	Responsibilities
Product Owner	Responsible for the success and failure of the project. He is the decision maker who controls the product backlog.	CEO of the Company	Gives ideas, requirements of project to the Team Leader. Creates product backlog with list of subjects/content.
Scrum Master	Responsible for an implementation that is whole the whole process	Team Leader	Removes barriers. Conducts meetings and assigns tasks on weekly basis. He is

			responsible for the whole process.
The development team	The success of scrum is dependent on then development team	Interns	Has the right to do everything within the boundaries of the project guidelines to reach the goal. They can teach the subject in detailed manner.

Table 1. Comparison of Scrum Roles with Our Project Members Roles

6. System design

6.1 Use Case Model

The User has to make the registration. The account is activated and verified by sending Email. The User can then Login to the Quantum Node Portal. The User has the access to the all pages in the Portal. The User activities are shown in the figure below. The Portal also has authentication process involved for the security purposes.

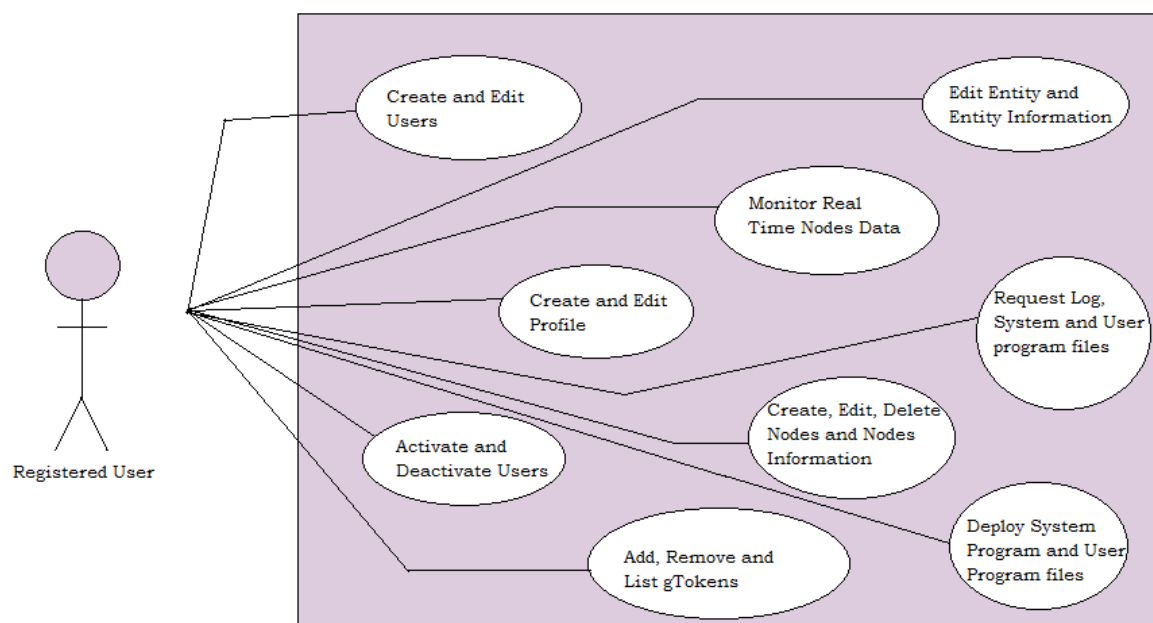


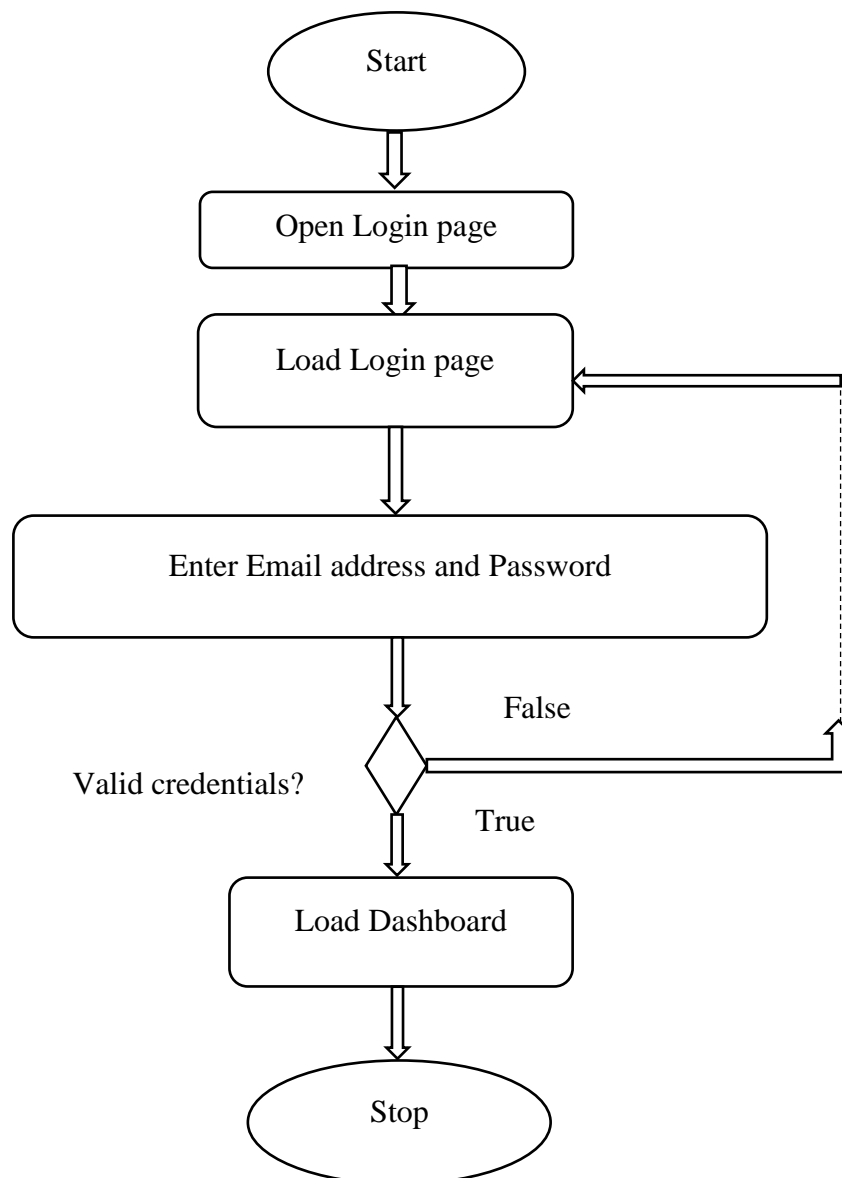
Figure 27. Use Case Model

In the future, The Portal will have two types of users such as Admin user and the Normal user. Admin will have access to User management i.e., creating new normal users, editing normal user information and also will have access to the Nodes management, Tokens Management and Entity Management. The User at present has access to all the Managements and Operations. The restriction for the access will be done by using middleware. The middleware is created and will be able to separate the Admin and Normal users.

6.2 Activity flowcharts

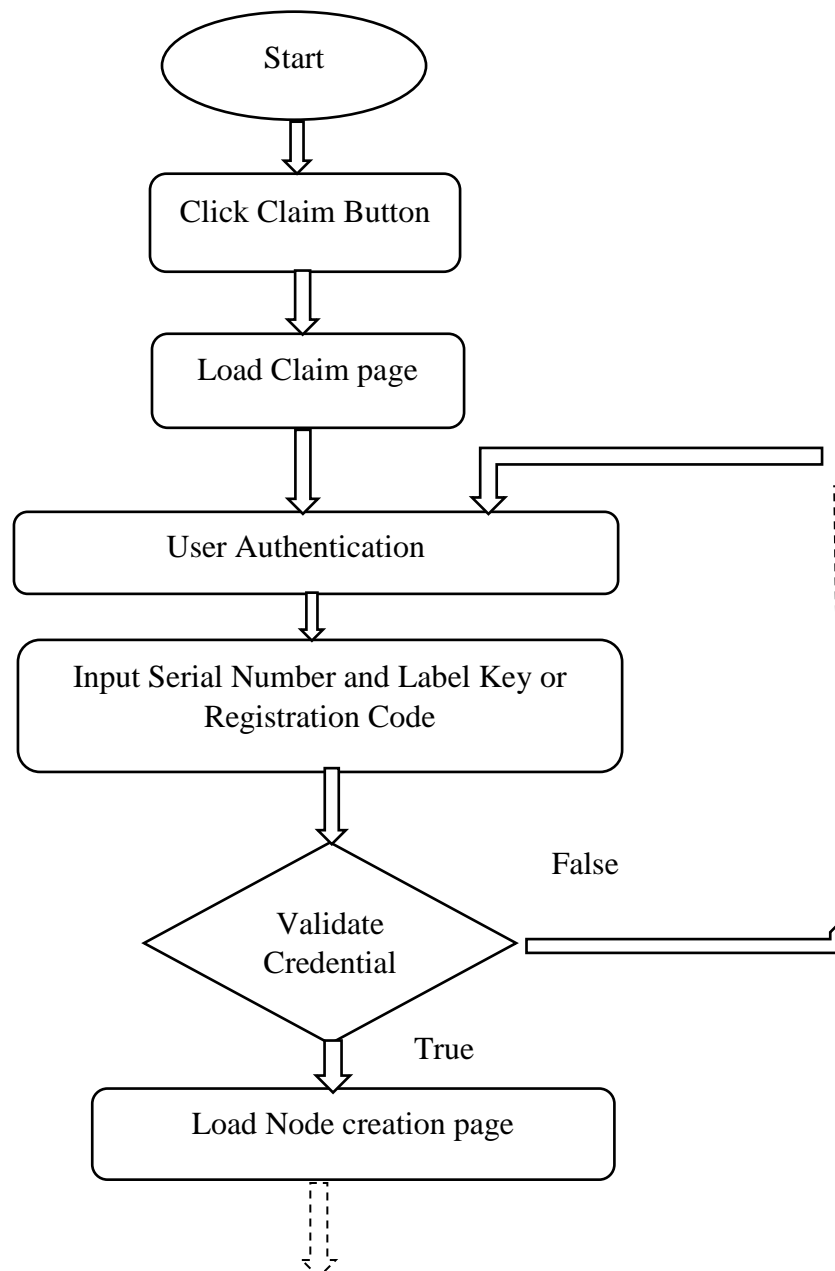
6.2.1 Login Activity

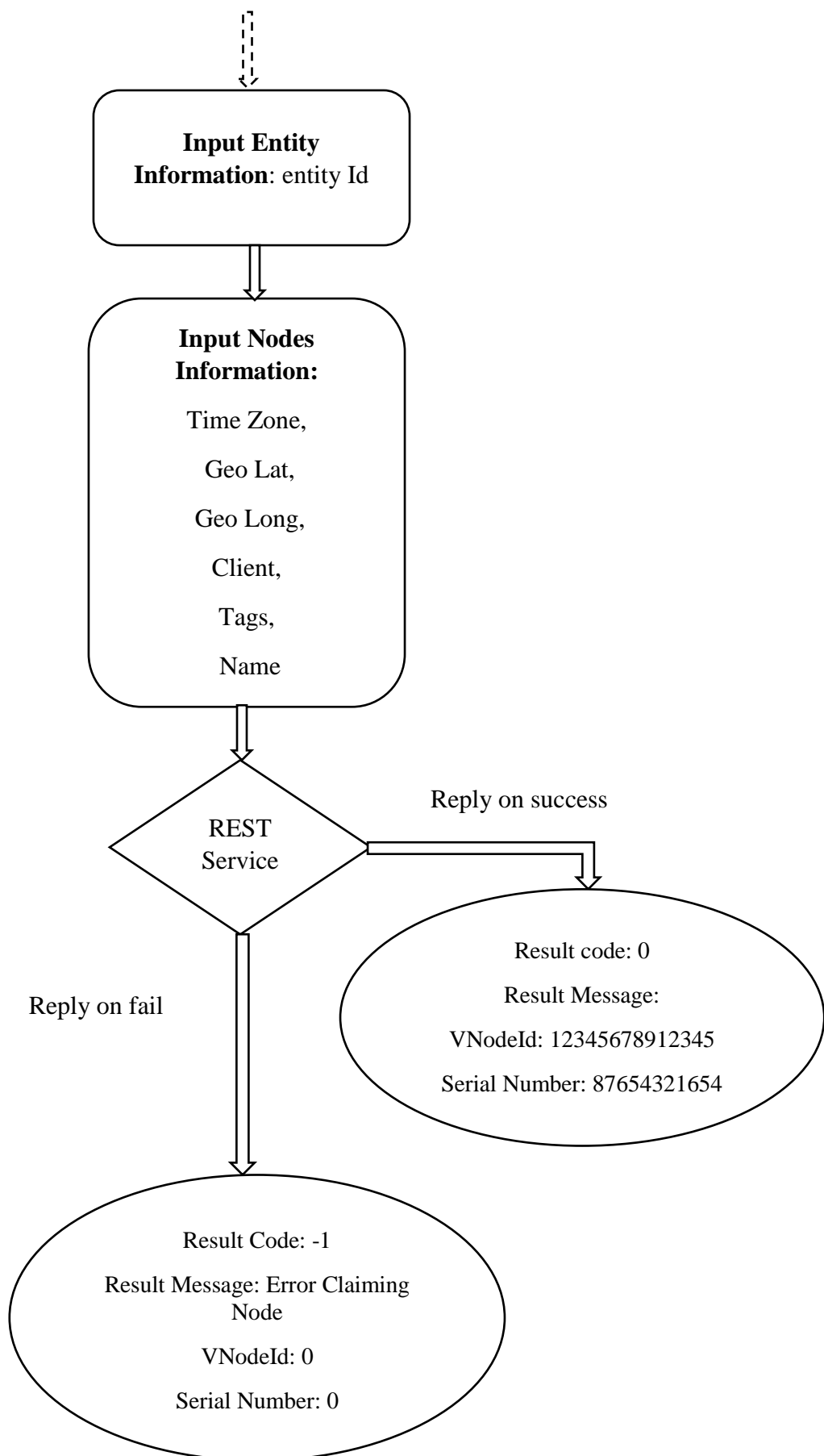
Login activity began with opening the login page. Once the login page is loaded, the credentials namely, email address and password are inserted. These credentials are then checked with the database registered users, if the credentials are valid, the user is directed to the home page otherwise the user is redirected to the login page to perform registration process.



6.2.2 Claim Node activity

Claim Node activity starts by clicking on claim button, claim page is loaded. Claim page contains authentication process for claiming a Quantum Node. Authentication method is done using User Authentication, and Node Validation Method is done with Registration code or Serial Number and Label Key. If authentication method and Node Validation are successful, the user is directed to Quantum Node Creation page, else redirected to Claim page in order to insert valid authentication details.





7. Implementation

Firstly, the starting stage of internship was to learn the new technologies and languages used to develop the Web application. The next stage was to understand the requirements for the Web Application, Planning the design and to implement what we learnt. Quantum Node Portal was implemented using Laravel Framework. Hence the implementation started with installing the necessary requirements. The work is not a continuation of previous work. It is the new project.

Laravel [26] is a Personal Home Page (PHP) framework for the development of web applications using Model View Controller (MVC) Its features are simple and to the point syntax. It's a modular system with devoted dependency manager, several forms of relational database access. Laravel is an open source framework. It comes with bundles, migrations, and Artisan Command Line Interface (CLI). Laravel offers a strong set of tools and an application architecture that comprises many of the optimum features of frameworks like CodeIgniter, Yii, and others. It contributes to the speed of the development. Websites built in Laravel are considered to be more secure. It prevents the various attacks that can take place on websites.

7.1 Functionalities

- Clients can register on the portal and receive an activation email.
- Clients register using user information that includes email/password combination, Entity information that includes company information and also the authentication of the Quantum Node owned by the client.
- Authentication of the Quantum Node is done using the combination of Serial Number and Label key or the Registration code.
- Clients can confirm their account through the activation email that contains an activation link.
- Clients can request the portal system to resend the activation code.
- Clients can login after they have confirmed their account by activation.

- Clients can reset their passwords if they have forgotten by a reset link sent to their corresponding email.
- After Login, Clients are directed to the Dashboard. It contains the map view of Quantum nodes located.
- Clients can Claim, View the List of Quantum Nodes.
- Clients have the access to the User information and Node information. They can Edit, Delete, Update depending upon whether the client is Admin User or Normal User.
- Clients can deploy System Program file and User Program file.
- Clients can also request Log file, System Program file and User Program file.
- Clients have access to Token Management.
- Clients have access to Entity Management.

7.2 Laravel-Installation

Laravel makes use of Composer to manage its dependencies [26].

Step 1: Download and install Composer on the system.

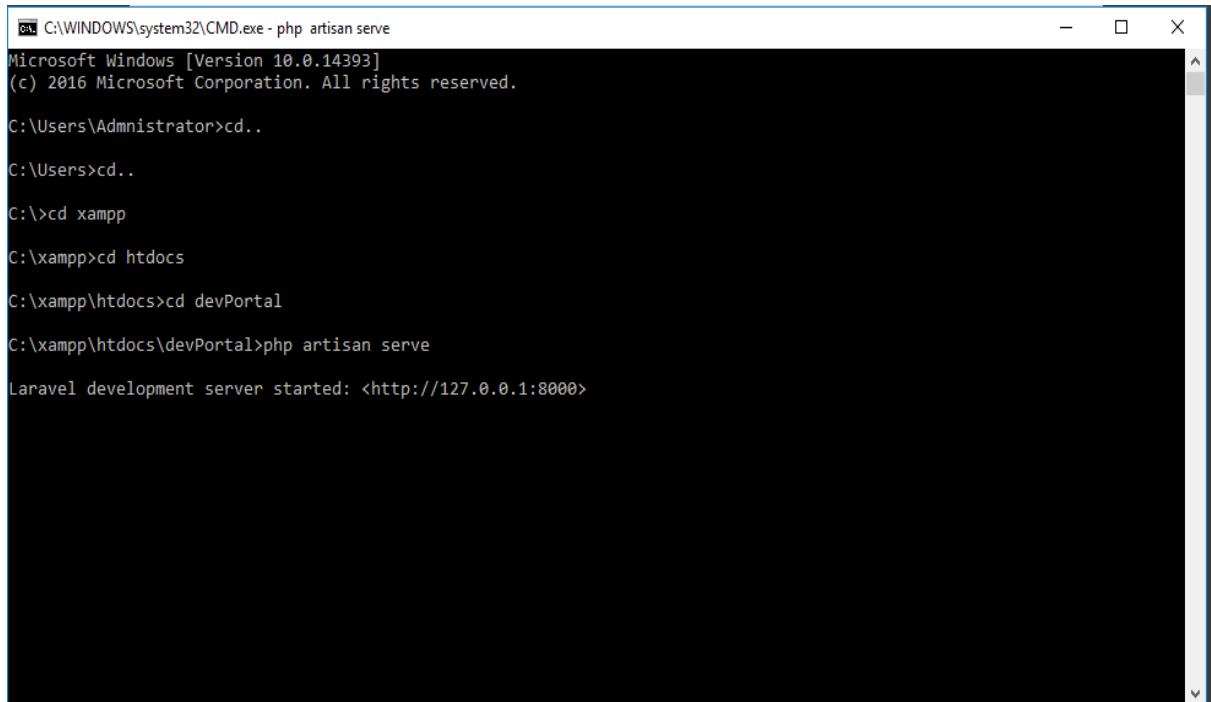
Step 2: Download and Install Xampp on the system.

Step 3: Using command prompt, direct the path C:\xampp\htdocs\

Step 4: Create the Laravel project using the command, **composer create-project --prefer-dist laravel/laravel project name**. The project is created via Composer.

Step 5: The above command will install Laravel in the current directory. Start the Laravel service by executing the following command, **php artisan serve**

Step 6: After executing the above command, screen will appear as below:



```
C:\WINDOWS\system32\CMD.exe - php artisan serve
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd..
C:\Users>cd..
C:\>cd xampp
C:\xampp>cd htdocs
C:\xampp\htdocs>cd devPortal
C:\xampp\htdocs\devPortal>php artisan serve
Laravel development server started: <http://127.0.0.1:8000>
```

Figure 28. Project running locally

Step 7: Open the given URL in the browser. If the following screen appears, it implies that Laravel has been installed successfully.



Figure 29. Laravel Project on Web browser

7.3 Laravel-Application Structure

Laravel provides an organised way for developing applications. It enables the structured method where the files are destined at their designated place [26].

7.3.1 Root Directory

The Root Directory of Laravel contains various folders and files as follows:

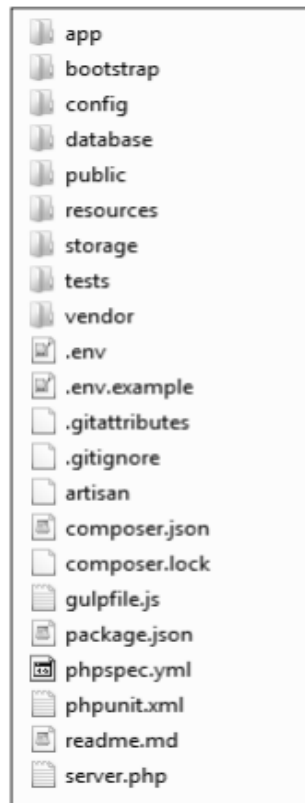


Figure 30. Root Directory

App: This directory covers the core code of the application. It contains all the Controllers. There are additional directories within it, such as Console, Exceptions, Http, Providers etc. Console directory has the artisan commands stored. Exceptions has a Handler file. Http directory contains all the Controller files, Middleware files and the kernel file. Providers contains the various service providers.

Bootstrap: This directory covers the application bootstrapping script.

Config: This directory covers configuration files of application.

Database: This folder covers the database migrations and seeds.

public: This is the root of the application's document. It is responsible for the start of the Laravel application. It also includes the assets of the application like JavaScript, Cascading Style Sheets (CSS), Images, etc.

resources: This directory covers raw assets such as the Leaner CSS (LESS) and Syntactically Awesome Style Sheets (SASS) files, localization and language files, and Templates that are rendered as HTML.

Storage: This directory covers App storage such as file uploads etc. It contains Framework storage(cache), and application-generated logs.

Test: This directory comprises of various test cases.

Vendor: This directory holds composer dependencies.

7.4 Configuration

7.4.1 Basic configuration

Generated an Application key to secure session and other encrypted data. The newly generated application key is obtained in .env file.

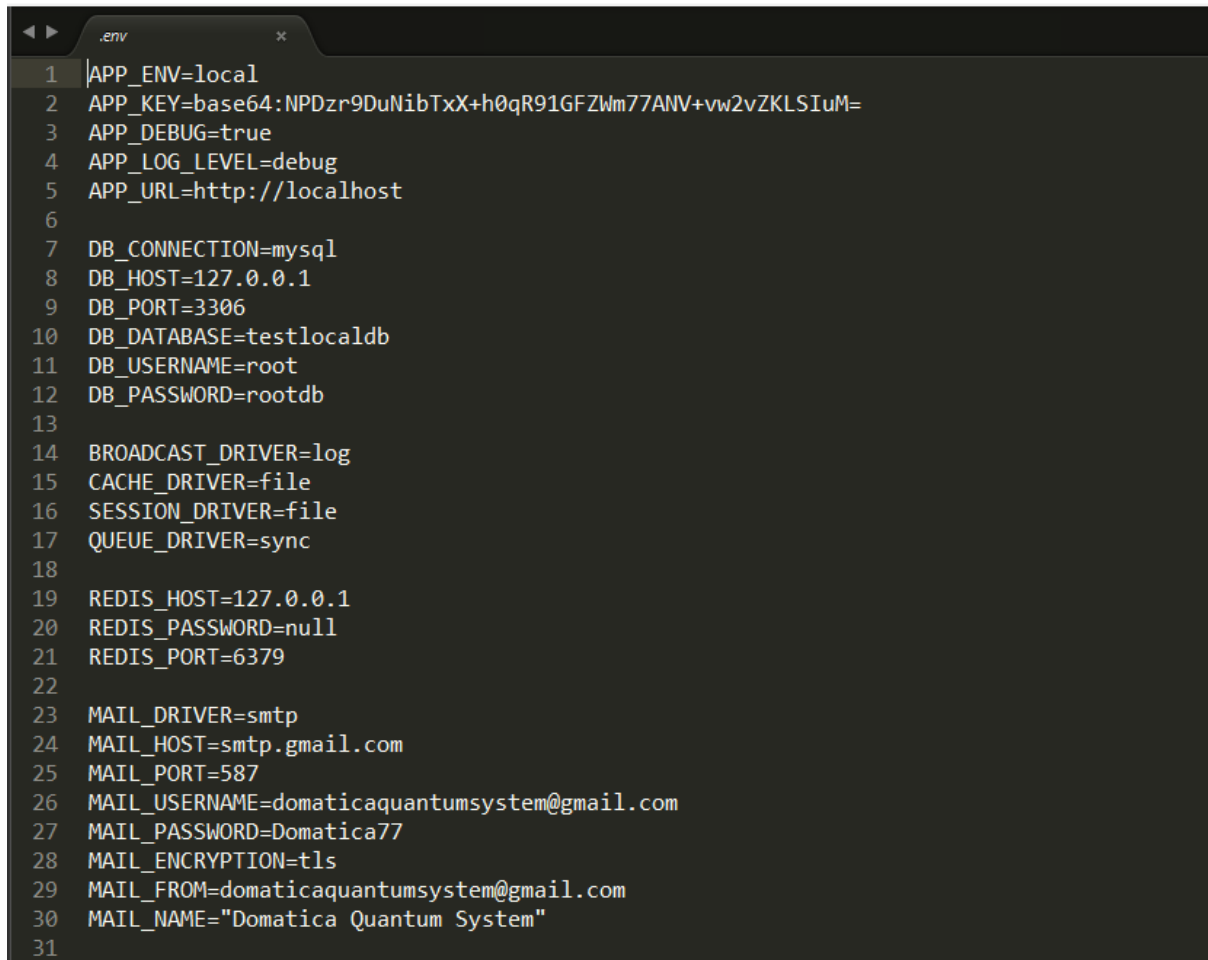
```
CA\WINDOWS\system32\CMD.exe - php artisan serve
)
laravel/framework suggests installing nexmo/client (Required to use the Nexmo transport (~1.0).)
laravel/framework suggests installing pda/pheanstalk (Required to use the beanstalk queue driver (~3.0).)
laravel/framework suggests installing predis/predis (Required to use the redis cache and queue drivers (~1.0).)
laravel/framework suggests installing pusher/pusher-php-server (Required to use the Pusher broadcast driver (~2.0).)
laravel/framework suggests installing symfony/dom-crawler (Required to use most of the crawler integration testing tools (~3.2).)
laravel/framework suggests installing symfony/psr-http-message-bridge (Required to psr7 bridging features (0.2.*).)
sebastian/global-state suggests installing ext-uopz (*)
phpunit/phpunit-mock-objects suggests installing ext-soap (*)
phpunit/php-code-coverage suggests installing ext-xdebug (^2.5.1)
phpunit/phpunit suggests installing phpunit/php-invoker (~1.1)
phpunit/phpunit suggests installing ext-xdebug (*)
Writing lock file
Generating autoload files
> Illuminate\Foundation\ComposerScripts::postUpdate
> php artisan optimize
Generating optimized class loader
The compiled services file has been removed.
> php artisan key:generate
Application key [base64:iEvvfVOZPZirHtEhmjAbuJiJPluKb/iNURNGWOYmpd4=] set successfully.
C:\xampp\htdocs>cd blo
C:\xampp\htdocs\blo>php artisan serve
Laravel development server started: <http://127.0.0.1:8000>
[Tue Aug 8 09:59:12 2017] 127.0.0.1:49673 [200]: /favicon.ico
[Tue Aug 8 10:02:09 2017] 127.0.0.1:49674 Invalid request (Unexpected EOF)
[Tue Aug 8 10:02:09 2017] 127.0.0.1:49675 Invalid request (Unexpected EOF)
[Tue Aug 8 10:02:09 2017] 127.0.0.1:49676 Invalid request (Unexpected EOF)
```

Figure 31. Basic Configuration

7.4.2 Environmental and Database configuration

Laravel has the facility to run the Laravel application in various environments like testing, production etc. The configuration for this requirement was done with the environment file (.env file) of the root directory. As the installed Laravel used composer, it was automatically created. Also, the Database configurations is done here.

As the Quantum Node Portal was first developed locally, the local environment was used. And the database was created in Xampp -MySQL.

A screenshot of a code editor showing a .env file. The file contains 31 lines of environment variables for a local development environment. The variables are listed in a numbered format from 1 to 31. The editor has a dark theme and a tab labeled '.env' at the top.

```
1 APP_ENV=local
2 APP_KEY=base64:NPDzr9DuNibTxX+h0qR91GFZWm77ANV+vw2vZKLSIU=
3 APP_DEBUG=true
4 APP_LOG_LEVEL=debug
5 APP_URL=http://localhost
6
7 DB_CONNECTION=mysql
8 DB_HOST=127.0.0.1
9 DB_PORT=3306
10 DB_DATABASE=testlocaldb
11 DB_USERNAME=root
12 DB_PASSWORD=rootdb
13
14 BROADCAST_DRIVER=log
15 CACHE_DRIVER=file
16 SESSION_DRIVER=file
17 QUEUE_DRIVER=sync
18
19 REDIS_HOST=127.0.0.1
20 REDIS_PASSWORD=null
21 REDIS_PORT=6379
22
23 MAIL_DRIVER=smtp
24 MAIL_HOST=smtp.gmail.com
25 MAIL_PORT=587
26 MAIL_USERNAME=domaticaquantumsystem@gmail.com
27 MAIL_PASSWORD=Domatica77
28 MAIL_ENCRYPTION=tls
29 MAIL_FROM=domaticaquantumsystem@gmail.com
30 MAIL_NAME="Domatica Quantum System"
31
```

Figure 32. .env file

The above image shows the local environment which can be changed to production or testing environment.

7.5 Routing

Routing in a simple way can be shown like routing request to an appropriate controller. The routes are defined in the routes file, located in the routes directory. The routes/web.php file defines routes that are for web interface [26].

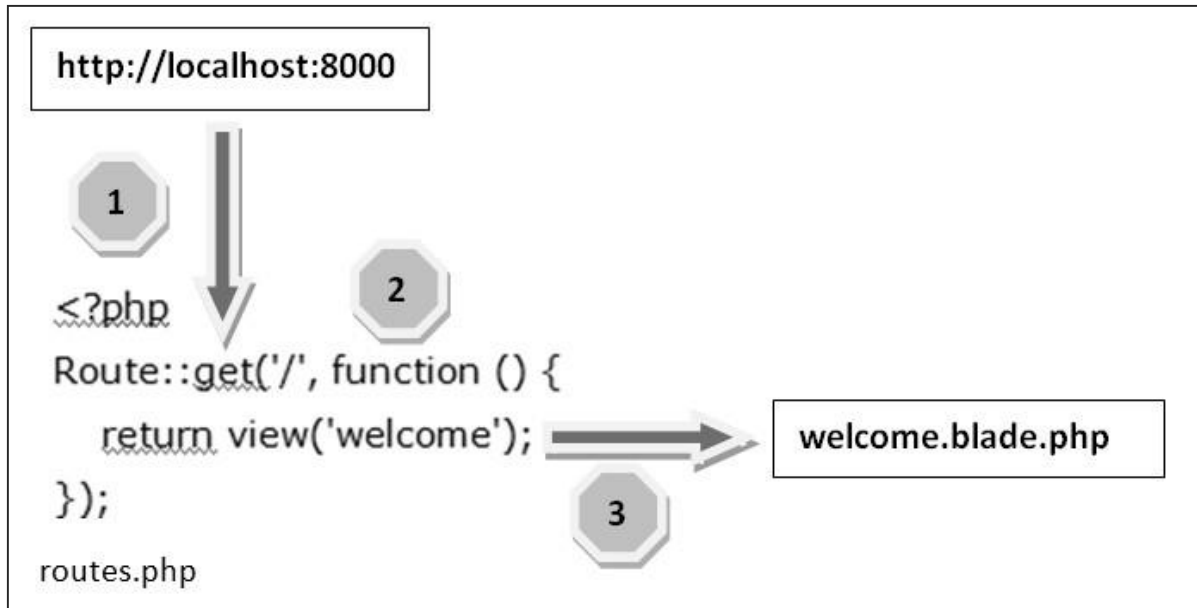


Figure 33. Routing mechanism

Step 1: Executes the root URL of the application.

Step 2: The executed URL will match with the appropriate method in the route.php file. In our case, it will match to get the method and the root (‘/’) URL. This will execute the related function.

Step 3: The function calls the template file **resources/views/welcome.blade.php**. The function later calls the **view ()** function with argument ‘**welcome**’ without using the **blade.php**. It will produce the following HTML output.

Consider this route from the project,


```
Route::get('/', function () {  
    if(Auth::check())  
    {  
        return redirect ('/Dashboard');  
    }  
    else  
    {  
        return view('welcome');  
    }  
});
```

The executed local host URL will match with the get method in the routes file with the path ('/') which will execute the function. The function will check for the user authentication condition and will direct the user either to the home page which has the path ('/Dashboard') or to the login page having the path ('welcome') for the user to login to the website.

7.6 Middleware

Middleware behaves like a middle man among request and response. It is a type of filtering mechanism. It helps for securing the application. For example, Laravel includes a middleware that verifies whether user of the application is authenticated or not. If the user is authenticated, he will be redirected to the home page otherwise, he will be redirected to the login page. It can also be used to restrict or permit the user from accessing the Web pages. This way it is possible to have different types of users accessing the Portal.

Middleware directory is registered in the path app/Http/Middleware directory.

Middleware is created by executing the following command:

```
php artisan make:middleware <middleware-name>
```

There are two types of Middleware in Laravel.

- Global Middleware: It will run on every HTTP request of the application.
- Route Middleware: It will be assigned to a specific route.

Middleware contains two properties `$middleware` and `$routeMiddleware`. `$middleware` property is used to register Global Middleware and `$routeMiddleware` property is used to register route specific middleware.

It can also be assigned to controllers's route or the controller's constructor.

Assigning Middleware to Route

```
Route::get('profile', [  
    'middleware' => 'auth',  
    'uses' =>  
    'MainController@showProfile'  
]);
```

Middleware within Controller's constructor:

```
<?  
  
namespace App\Http\Controllers;  
    use Illuminate\Http\Request;  
    use App\Http\Requests;  
    use App\Http\Controllers\Controller;  
class MainController extends Controller  
    {  
        public function __construct(){  
            $this->middleware('auth');  
        }  
    }
```

7.7 Controllers

Laravel MVC framework [26], C stands for Controller. It acts as a directing traffic between Views and Models.

7.7.1 Controller creation

In the Terminal, the following command is used to create controller [26].

```
php artisan make: controller <controller-name> --plain
```

The created constructor will be in the app/Http/Controllers. This constructor can then be called in routes as follows:

```
Route::get('Base URI', 'Controller@method');
```

```
<? php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Http\Requests;
use App\Http\Controllers\Controller;

class nameController extends Controller
{
    //
}
```

7.7.2 Authentication Controllers

Laravel provides ready-made code and configuration for authentication method by the following command:

```
php artisan make:auth
```

The various default Auth Controllers will be created which are located in `App\Http\Controllers\Auth` namespace. The `RegisterController` handles the new user registration, the `LoginController` handles authentication, the `ForgotPasswordController` handles e-mailing links for resetting passwords, and the `ResetPasswordController` contains the logic to reset passwords. The default logic is located inside `vendor\laravel\framework\src\Illuminate`. This command also installs the layout view, registration and login views, as well as routes for all authentication end-points.

For routes, a new method is added to the `routes/web.php` file called `Auth::routes()`. This method basically loads laravel's default Auth routes.

The routes in the `Auth::routes()` method are defined as follows:

// Authentication Routes...

```
Route::get('login', 'Auth\LoginController@showLoginForm')->name('login');  
Route::post('login', 'Auth\LoginController@login');  
Route::post('logout', 'Auth\LoginController@logout');
```

// Registration Routes...

```
Route::get('register', 'Auth\RegisterController@showRegistrationForm');  
Route::post('register', 'Auth\RegisterController@register');
```

// Password Reset Routes...

```
Route::get('password/reset', 'Auth\ForgotPasswordController@showLinkRequestForm');  
Route::post('password/email', 'Auth\ForgotPasswordController@sendResetLinkEmail');  
Route::get('password/reset/{token}',  
'Auth\ResetPasswordController@showResetForm');  
Route::post('password/reset', 'Auth\ResetPasswordController@reset');
```

But, in our project, we have a single controller named `MainController` that consists of all the methods required for the Web Application to run as shown in fig. 26. Hence, I have added all the authentication methods within the `MainController`. This required the alteration of automatic routes and methods provided by Laravel.

The required Traits are added to the `MainController` such as `Mail`, `DB`, `Validator`, `SendsPasswordResetEmails`, `Password`, `Str`, `Input`.

I created Confirmation Email trait which contains the Email format to be sent to the users. We discuss about `Validator`, `Mail`, `DB` and other traits in the next section.

7.8 Views

The views of Laravel makes use of Blade Templating Engine. Blade view files have the extension .blade.php which are located in the resources/views directory.

The views folder in Laravel consists three sub folders named auth, email, and includes. It also contains the other view pages of the Quantum Node Portal such as Login, Dashboard, List Nodes etc.

Auth folder contains the views for forgot password page, reset link page and register page.

Email folder contains activation view, a format that is sent as an email for activating the account.

Includes folder contains header and footer views. We make use of @include directive to include the header and footer content, @include('includes.header') and @include('includes.footer') which helps not to rewrite the code on every page.

Creating Loops and applying conditions is easier when we use blade template as it provides constructs such as @foreach, @for, @if, @while, @elseif. This helps to avoid more number of code lines.

7.8.1 Passing Data to the Views

Laravel makes passing data simpler. Consider the following example.

In the Controller method, the data array is returned.

```
public function vNodeList()
{
    $dataNode = $this->vNodeInformations();
    return view('nodes_list')->with(['dataNode' =>
    $dataNode]);
}
```

In the routes,

```
Route::get('nodes_list' , ['as' => 'nodes_list', 'uses' => 'MainController@vNodeList']);
```

Here, nodes_list is the path of the view file and it makes use of vNodeList function from the Main Controller which returns the data array.

To display the data in the View file,

```
<? php foreach($dataNode as $listnode):?>
    <a href="node_details"><? =$listnode['vNodeId'];?></a>
    <a href="node_details"><? =$listnode['serialNumber'];?></a>
    <? php endforeach;?>
```

We also made use of Asynchronous JavaScript and XML (AJAX) Post method to send the data from the view files to the controller methods by specifying the path.

7.9 Validation and Sending Email

Laravel framework is considered to be one of the best framework for authentication and authorization. [26] The authentication and authorization is done by creating the register, login and also creating to retrieve the forgotten passwords in our application. The clients can register to the portal. Registration includes the User information, Entity information as well as Authentication of the Quantum nodes owned by the client. As the client is registered, email is sent to their respective mail account that enables them to activate their Portal account. Laravel makes use of default controllers for these tasks by default, but in the project, I have implemented the login and register to one controller named as Main Controller. Hence it is required to understand the functionalities and classes that each default controller makes use of, also route the files to the Main controller.

Laravel validation helps to validate the incoming data with the data existing in the Database. The HTTP requests are validated with the rules. The Validator class has a make method under which the arguments to be validated are passed. The necessary rules are appended on them. The code example for Validation that is used in our project is given below:

```
protected function validator (array $data)
{
    return Validator::make($data, [
        'firstName' => 'required|max:255',
        'lastName' => 'required|max:255',
        'telephone' => 'required|min:11|numeric',
        'email' => 'required|email|max:255|unique: users',
        'password' => 'required|min:6|confirmed',
    ]);
}
```

Laravel makes use of library named “SwiftMailer” to send emails. This library helps to send emails without too many bothers. It provides drivers for SMTP, Mailgun, SparkPost, Amazon SES, PHP’s mail function, and sendmail. In this project, I make use of SMTP driver.

Email templates are loaded in the similar way as views are loaded, which is done using Blade syntax and injecting data into the templates.

```
Mail::send(['text' => 'view'], $data, $callback);
```

`$view(string|array)`-name and path of the file that contains the email message template.

`$data(array)`- array of data to pass to the view, the data is nothing but the user information.

`$callback`- Closure callback receives the message instance. I included the recipients, subject and other aspects of the mail message.

- `$message->to($user['email'])->from('domaticaquantumsystem@gmail.com');`

- `$message->subject (' Activation Code');`

It is necessary to add new table and one column in users table. We need to have Boolean field 'is_activated' in users table to keep track of whether to determine user is active or deactive. We also need to create a new table "user_activations" that will store token of user activation code so we can check the email verification link time.

```
Route::get('user/activation/{token}', 'Auth\AuthController@userActivation');
```

This route helps to create activation link. A mail is sent after registration with activation link that way user can click on the link and activate their account.

7.9.1 Email Configuration

Configuration of the .env file[26] for sending an email from Gmail account is done as follows:

```
MAIL_DRIVER=smtp
MAIL_HOST=smtp.gmail.com
MAIL_PORT=587
MAIL_USERNAME=abc@gmail.com
MAIL_PASSWORD=xxxxxxx
MAIL_ENCRYPTION=tls
MAIL_FROM=abc@gmail.com
MAIL_NAME="abc"
```

Configuration of the config/Mail.php file is done as follows:

```
        return [  
            'driver' => env('MAIL_DRIVER', 'smtp'),  
            'host' => env('MAIL_HOST', 'smtp.gmail.com'),  
            'port' => env('MAIL_PORT', 587),  
            'from' => [  
                'address' => env('MAIL_FROM'),  
                'name' => env('MAIL_NAME')],  
            'encryption' => env('MAIL_ENCRYPTION', 'tls'),  
            'username' => env('MAIL_USERNAME'),  
            'password' => env('MAIL_PASSWORD'),  
            'sendmail' => '/usr/sbin/sendmail -bs',  
        ];
```

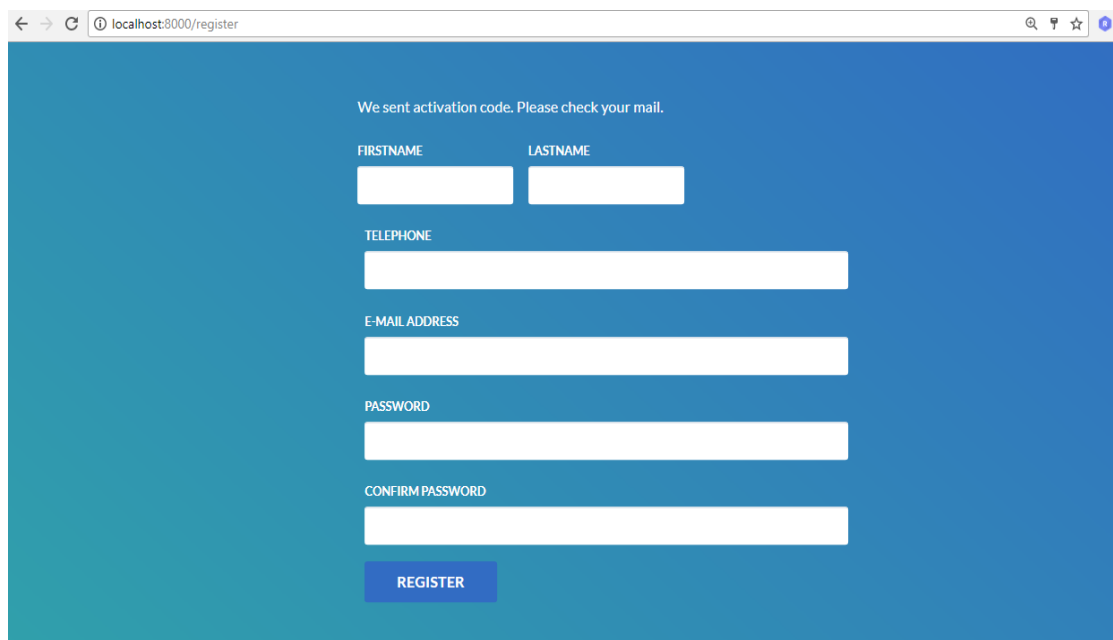
Simple Mail Transfer Protocol (SMTP) driver is used in our project. Host address of the server is provided. We make use of 'smtp.gmail.com'. The port used by our application to send emails is 587. In order to send emails to the clients from the same address, we specify the name and the address that is used globally for all e-mails. In our case, the company name and email address. E-mail encryption protocol should be specified. We make use of Transport Layer Security (TLS). Server requires the username and password for the authentication purpose. Hence it will get used to authenticate with our server on connection.

8. User Interface

8.1 Account Management

8.1.1 Registration Page

The Registered User will get an email with Activation Code. This ensures only Registered Users can access the Quantum Portal.



The screenshot shows a web browser window with the address bar displaying 'localhost:8000/register'. The page has a blue gradient background. At the top, a message reads 'We sent activation code. Please check your mail.' Below this, there are input fields for 'FIRSTNAME' and 'LASTNAME', followed by a 'TELEPHONE' field, an 'E-MAIL ADDRESS' field, a 'PASSWORD' field, and a 'CONFIRM PASSWORD' field. A blue 'REGISTER' button is positioned at the bottom of the form.

Figure 34. Registration page

8.1.2 Login page

On clicking the link in the email with Activation code, the user is directed to Login page indicating the account is activated.

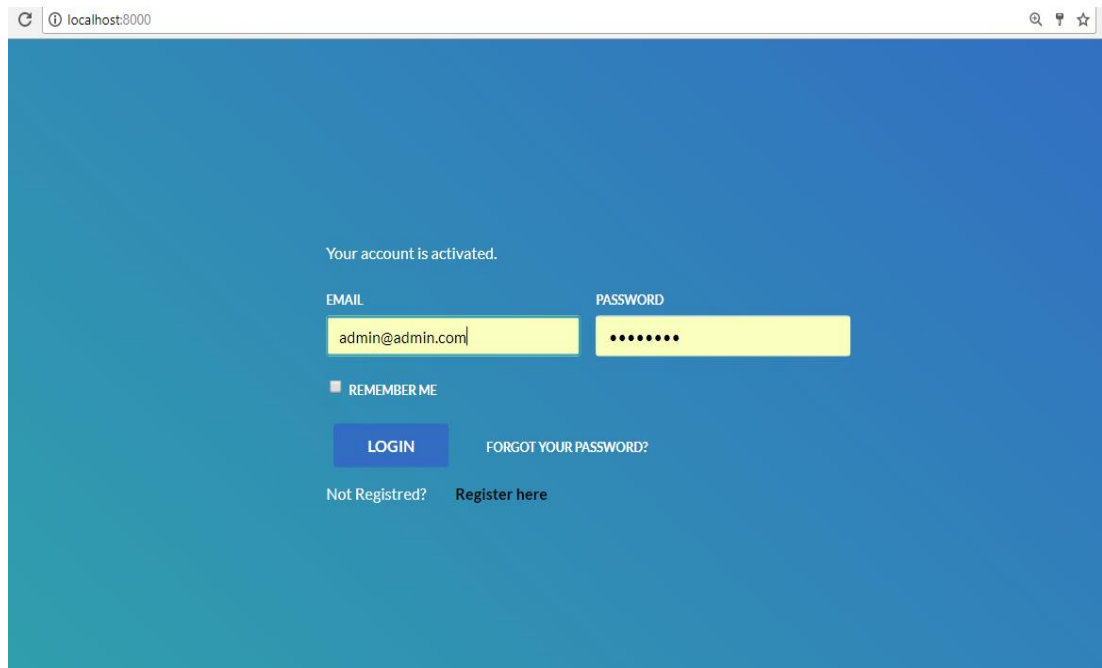


Figure 35. Login Page

8.1.3 Password Recovery

This page asks for the email address to which the recovery password has to be sent.

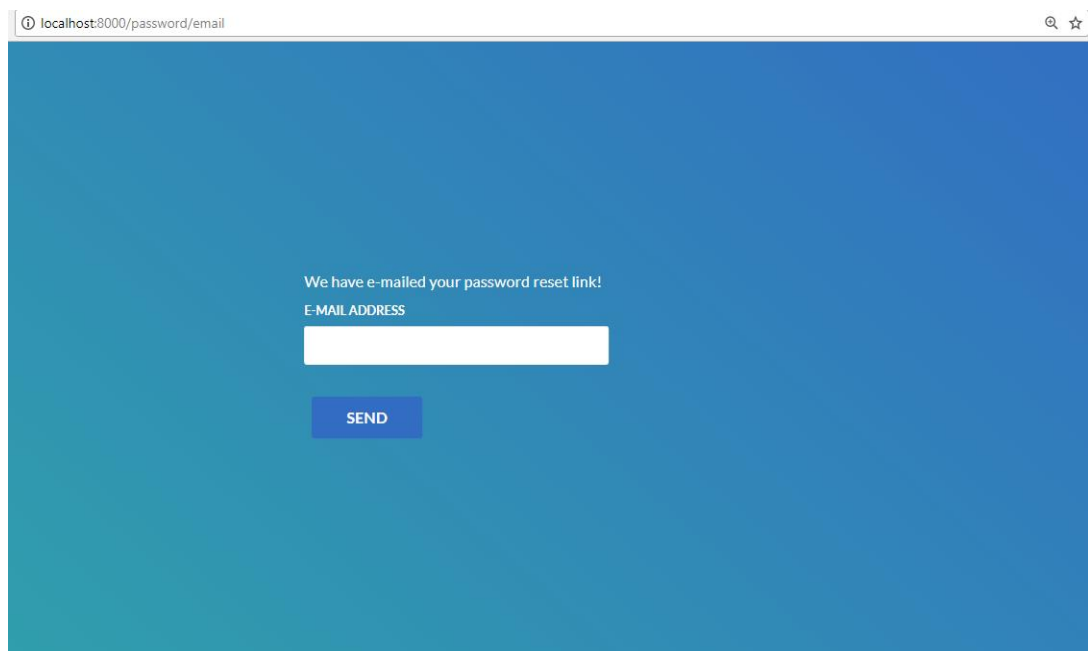


Figure 36. Password Recovery

8.2 Nodes Management

Initially the Quantum Portal was completely different design, it had the side bar menu with sub menus for Nodes Management, User Management, Account Management. But in the middle of the Internship, the changes had to be made. In the new design, the Dashboard contains menu on Top of the Web page with Map and List options. After the User has logged in, The User is directed to Dashboard.

8.2.1 Map

Map menu displays the home page. Home page displays the total Quantum Nodes, and also the number of Online and Offline Nodes belonging to that user. A User can be a Normal user or an Admin user. It also contains a Map view that displays Quantum Nodes location on map. Map contains markers of two different colours, green marker belongs to online status of nodes and red marker belongs to offline status of the nodes. On clicking the marker, the tool tip displays the information about the Quantum node. Information includes Serial Number, Node Id, Name of the node, Name of the Client. It also includes a button to view more information about that specific Quantum Node. Home page has a claim button, as explained earlier in the Activity flowcharts section, claim button loads a page with Authentication method required to claim the Quantum Node.

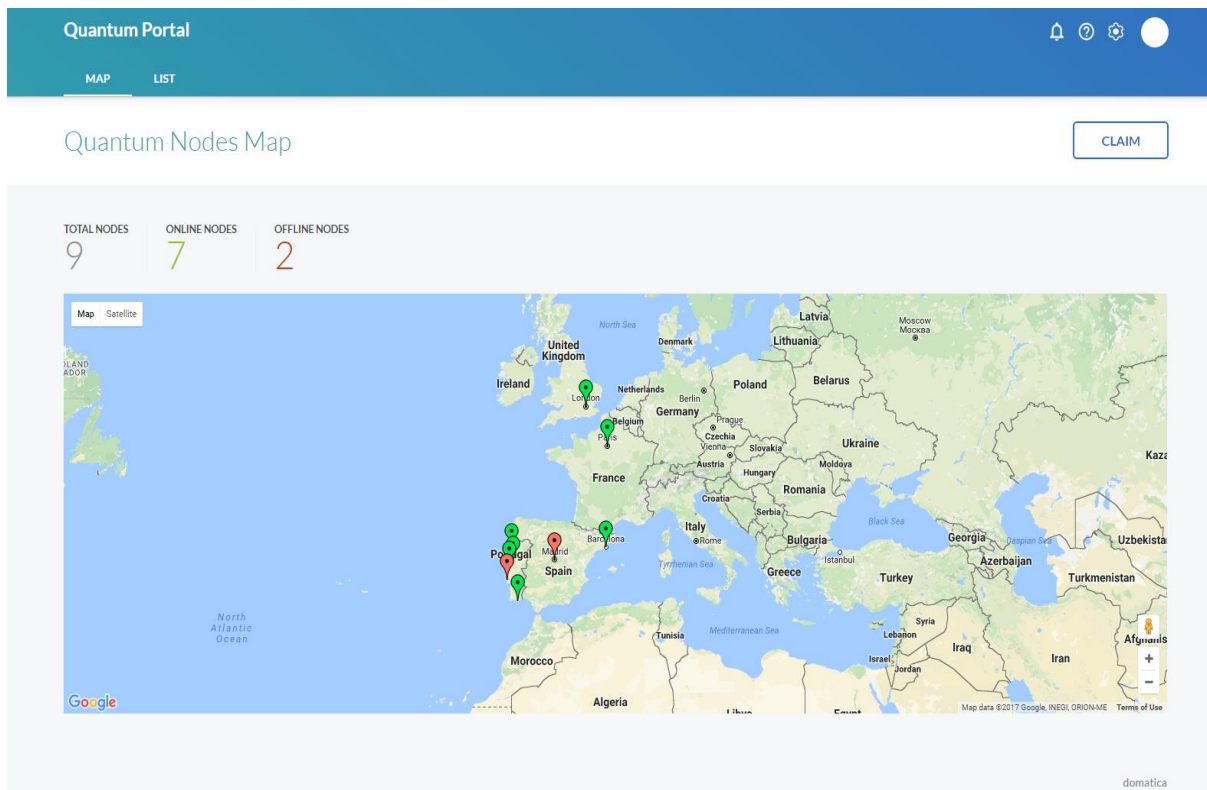


Figure 37. Dashboard

8.2.2 List

List menu displays Quantum Nodes list belonging to that User (Admin/User). It displays a table with 7 columns namely, VNode Id, Serial Number, Name, End Points, Monthly Traffic, Since, Status, Actions. Actions column contains replace node and release node. There is a search option which helps to sort the Quantum Node search. On clicking any of the column content, it loads a page that contains a respective Node information.

Quantum Portal

MAPLIST

Quantum Nodes ListCLAIM

Search for vNodeid..	Search for serialNumber.	Search for names..					
VNODEID	SERIAL NUMBER	NAME	ENDPOINTS	MONTHLY TRAFFIC	SINCE	STATUS	ACTIONS
11664	1234587219	Best Node	276	5.9 MB	June 12-2016 11:28:59		
12737	6647566693	Amazing Node	272	5.1 MB	June 28-2016 14:34:25		
13848	9944539117	Domatica Node	278	5.5 MB	November 09 2016 18:58:09		
14847	7563091849	Google Node	254	4 MB	July 22-2016 22:48:39		
15183	1234568910	Domatica Node Leiria	255	2 MB	Dezember 12-2016 11:28:59		
16881	8825330983	Domatica Node 2	255	2 MB	January 12-2016 11:28:59		
17009	9917293602	Co2	245	3 MB	October 12-2016 11:28:59		
18738	1260138845	Special Node	243	9 MB	November 18-2016 18:18:19		

<< Prev12Next >>

Figure 38. Quantum Nodes List page

Quantum Portal

MAPLIST

Quantum Nodes ReplaceCLAIM

OLD SERIAL NUMBER

NEW VNODE ID

SEND

domatica

Figure 39. Quantum Nodes Replace

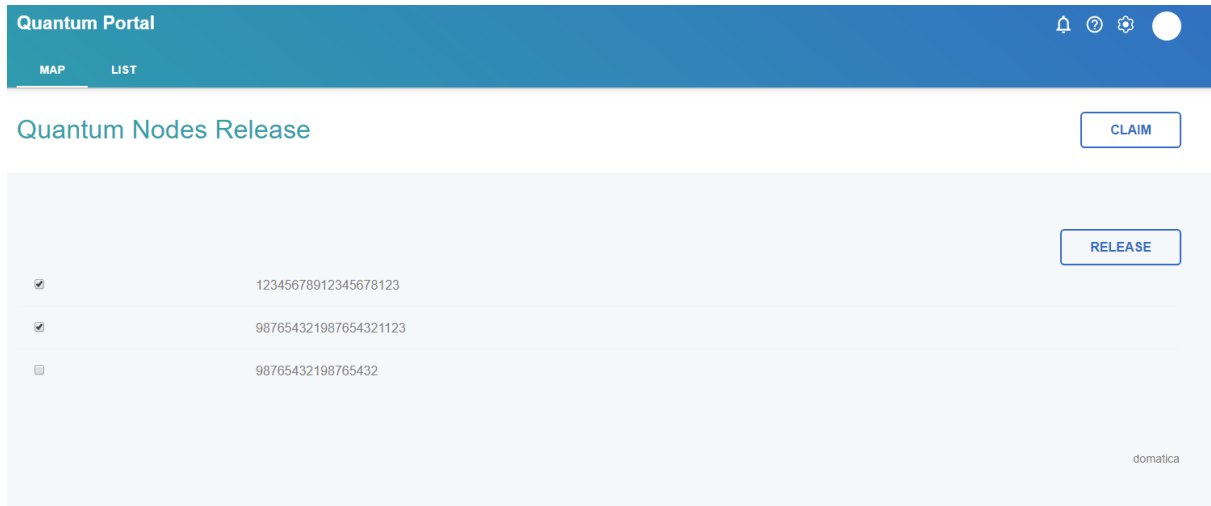
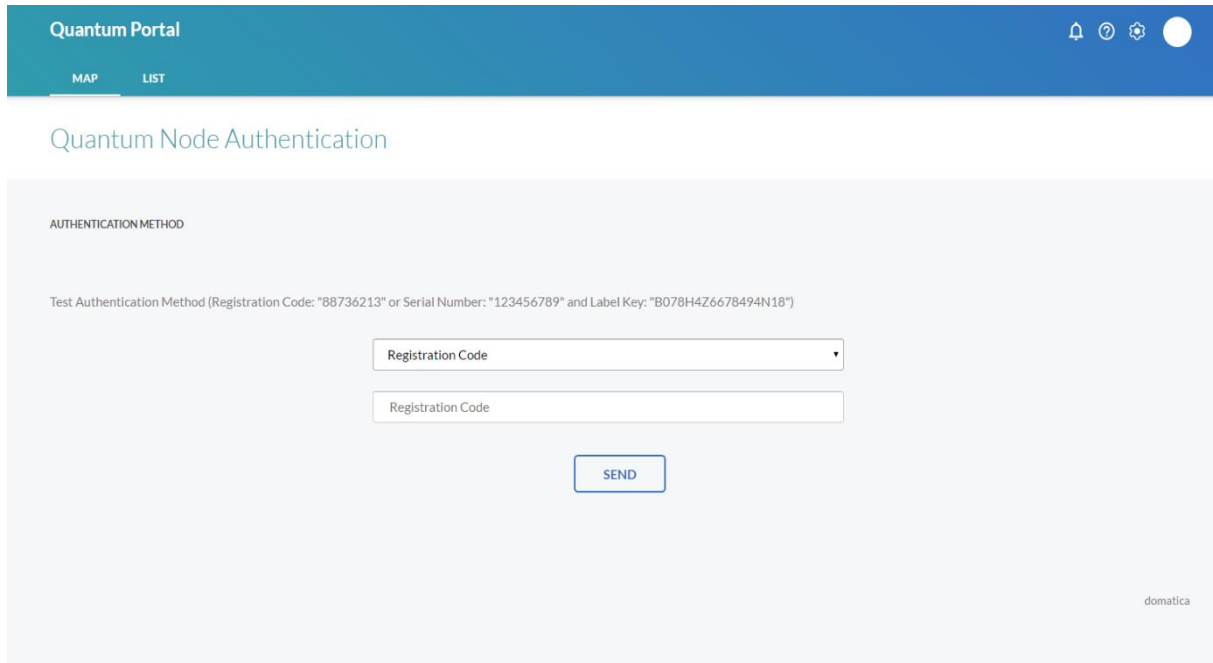


Figure 40. Quantum Nodes Release

8.2.3 Claim

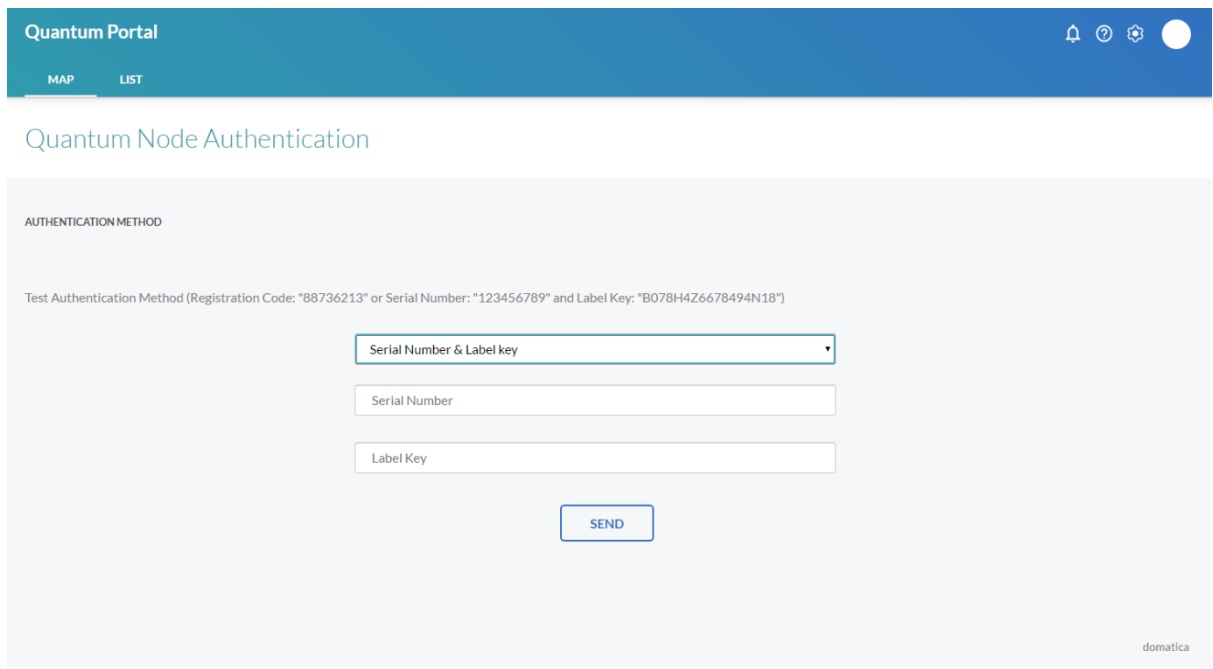
Claim page generates Authentication process. Authentication process includes Registration method and Serial Number and Label Key combination method. Node Claiming requires to send information such as Entity Id and Node Information, which in response will return the claim is succesful or not.

The location of the node can be added precisely by clicking the marker button. The button loads a page with Map View. The Map has the unique feature of Drag and Drop the Marker Pin to locate the Latitude and Longitude Cordinates precisely.



The image shows a web interface for 'Quantum Portal'. At the top, there is a blue header bar with the text 'Quantum Portal' on the left and three icons (a bell, a question mark, and a gear) on the right. Below the header, there are two tabs: 'MAP' and 'LIST'. The main content area is titled 'Quantum Node Authentication'. Underneath this title, it says 'AUTHENTICATION METHOD'. A line of text provides test data: 'Test Authentication Method (Registration Code: "88736213" or Serial Number: "123456789" and Label Key: "B078H4Z6678494N18")'. There are two input fields, both labeled 'Registration Code'. The first is a dropdown menu, and the second is a text box. Below these fields is a blue button labeled 'SEND'. In the bottom right corner of the main content area, the word 'domatica' is written.





Figure 41. Authentication page with Registration code



The image shows a web interface for 'Quantum Portal', similar to Figure 41. It has the same blue header bar with 'Quantum Portal' and icons, and the same 'MAP' and 'LIST' tabs. The main content area is titled 'Quantum Node Authentication'. Underneath this title, it says 'AUTHENTICATION METHOD'. A line of text provides test data: 'Test Authentication Method (Registration Code: "88736213" or Serial Number: "123456789" and Label Key: "B078H4Z6678494N18")'. There are three input fields. The first is a dropdown menu labeled 'Serial Number & Label key'. Below it are two text boxes, one labeled 'Serial Number' and one labeled 'Label Key'. Below these fields is a blue button labeled 'SEND'. In the bottom right corner of the main content area, the word 'domatica' is written.

Figure 42. Authentication with Serial Number and Label Key

Quantum Portal




MAP

LIST

Quantum Node Creation

CLAIM

NODE ID	11664		
SERIAL NUMBER	43467435774532		
LABEL KEY	12345678678		
NAME	<input type="text" value="name"/>		
CLIENT	<input type="text" value="client name"/>		
TAGS	<input type="text" value="tags"/>		
TIME ZONE	<div>(GMT) Western Europe Time, London, I ▾</div>		
LOCATION	<input type="text" value="latitude"/>	<input type="text" value="longitude"/>	
<div>SAVE</div>			

domatica

Figure 43. Quantum Node Creation page

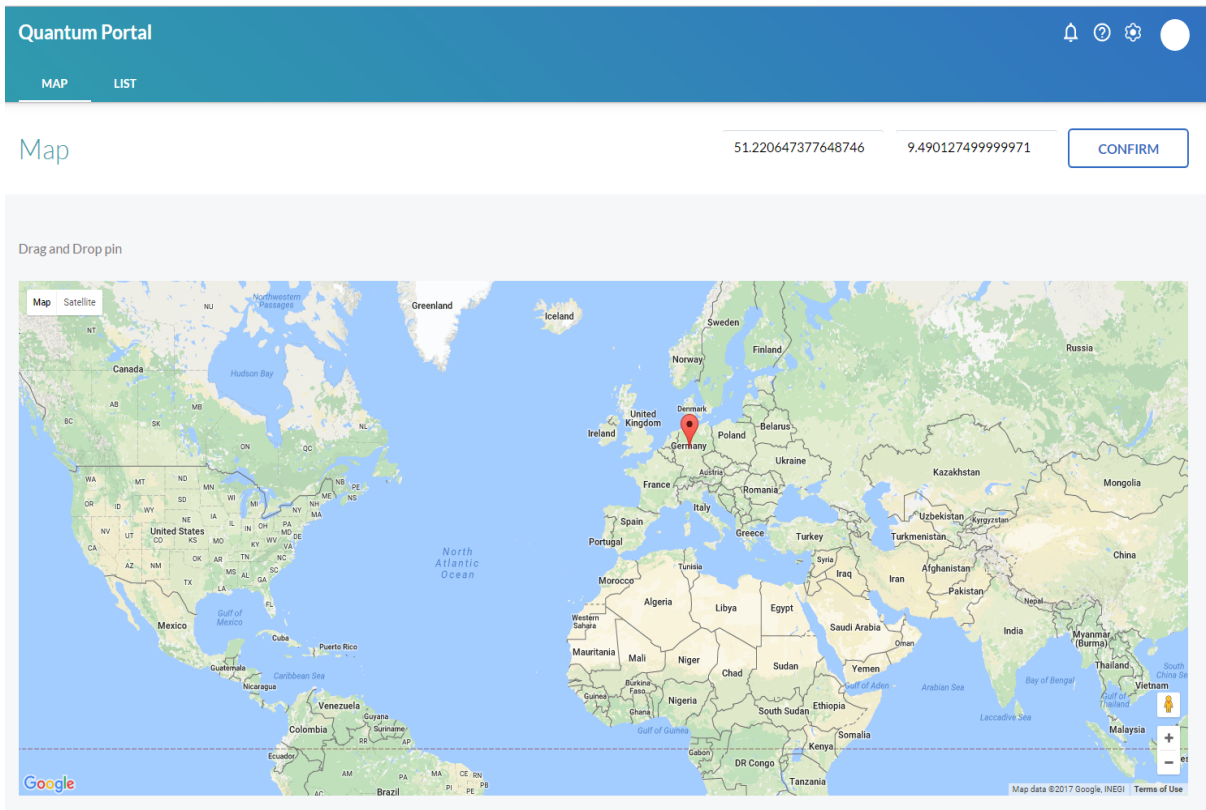


Figure 44. Map View page with Drag and Drop Pin

8.2.4 Node information

This page includes six tabs namely, Information, Status, Programming, Data Management, Tools and Services.

Information Tab contains a table of informations such as Node Id, Serial Number, Label Key, Name, Client name, Tags, Time Zone, Latitue and Longitude Cordinates. Also has an edit option that opens a display modal to edit all the following details and a map with a drag and

Drop pin.

Quantum Portal

MAP

LIST

Quantum Nodes

11664

Information

Status

Programming

Data Management

Tools

Services

NODE ID

11664

SERIAL NUMBER

1234587219

LABEL KEY

12345678

NAME

Node Name Best Node

CLIENT

Cisco

TIME ZONE

LOCATION

LAT:

LONG:

EDIT

domatica

Figure 45. Node Information page

Quantum Portal

MAP

LIST

Quantum Nodes

11664

Information

Status

Programming

Data Management

Tools

Services

NODE ID

SERIAL NUMBER

LABEL KEY

NAME

CLIENT

TIME ZONE

LOCATION

VNODEID :

SERIAL NUMBER :

LABEL KEY :

NAME :

CLIENT :

TIME ZONE : (GMT) Western Europe Time, London, I

GEO :


Latitude

Longitude

Drag and Drop pin

Map

Satellite



CONFIRM

EDIT

domatica

Figure 46. Edit Node Information Modal

Status Tab displays the core information like Serial Number, Module Name, Module description, Hardware version, Firmware version, Project date, Up time, Received messages, Sent messages, Completed processes, System program, Settings storage, Run time memory, Timers, Variables, Mobile communication, Status. Also includes Sim Card Management button.

Information	SERIAL NUMBER	0x23000769(Activated: 2013-12-13 16:14:17)
Status	MODULE NAME	iDomGateway3_1
Programming	MODULE DESCRIPTION	
Data Management	HARDWARE VERSION	3.00
Tools	FIRMWARE VERSION	3.00(Build: 5340)
Services	PROJECT DATE	2016-07-03 21:34:42
	UP TIME	7 weeks, 3 days, 19 hours, 47 minutes and 45 seconds
	RECEIVED MESSAGES	11956495
	SENT MESSAGES	1586590
	COMPLETED PROCESSES	29583605
	SYSTEM PROGRAM	1%
	SETTINGS STORAGE	21%
	RUNTIME MEMORY	27%
	TIMERS	0/512
	VARIABLES	0/8192
	MOBILE COMMUNICATION	Yes
	SIM CARD MANAGEMENT	
	STATUS	

Programming tab provides users make Configuration by deploying the User program files and System program files.

Figure 47. Status Tab

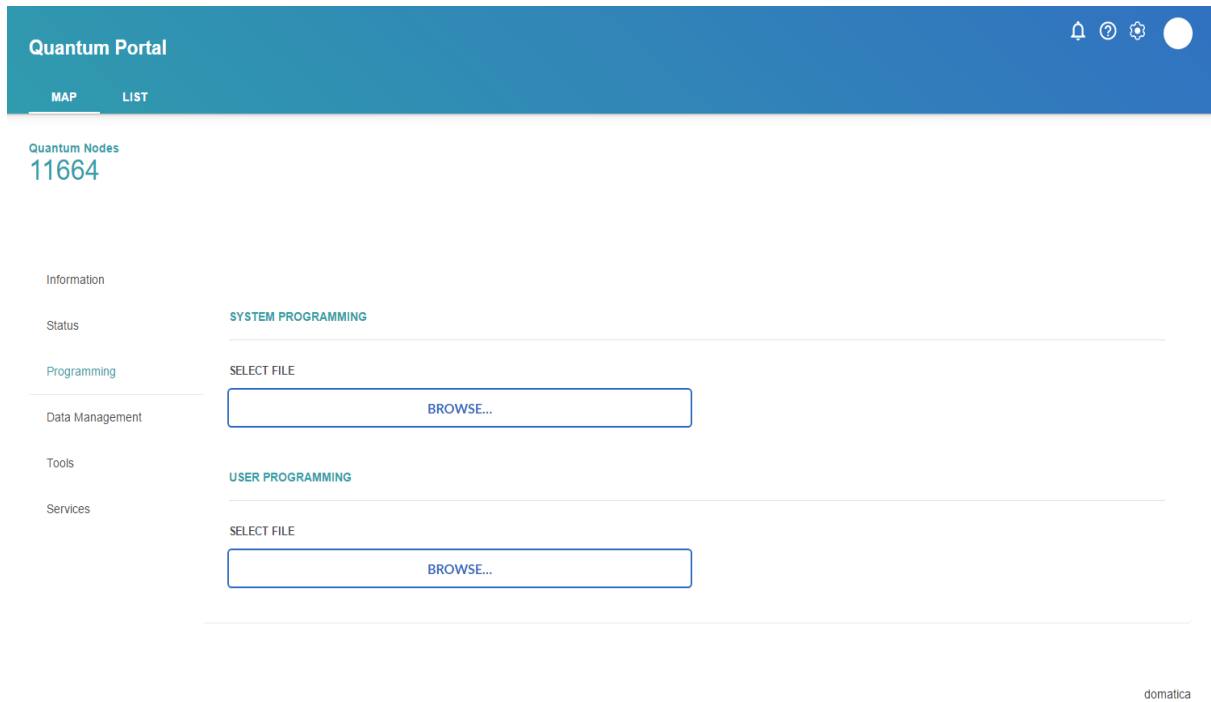


Figure 48. Configuration page

Data Management Tab provides the users to request files from the Database such as Log files, System files and User program files.

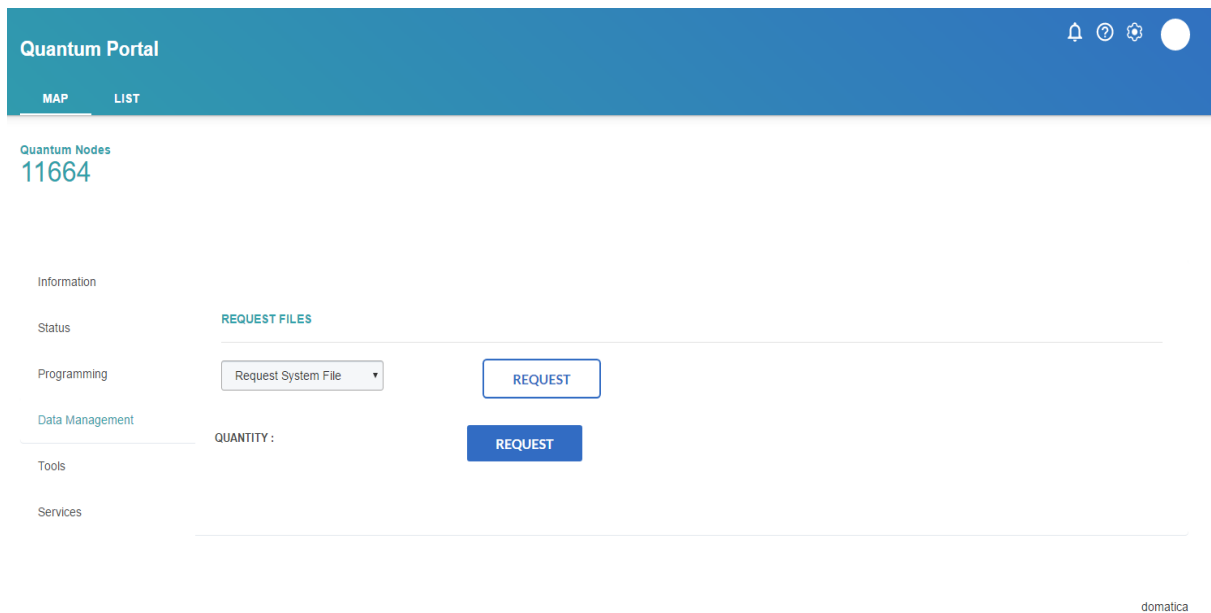


Figure 49. Data Management

8.3 Entity Management

8.3.1 Entity Information

On requesting the Entity Information, this page returns the Information of the Entity.

The screenshot shows the 'Entity Information' page in the Quantum Portal. The page has a blue header with the 'Quantum Portal' logo and navigation links 'MAP' and 'LIST'. Below the header, the title 'Entity Information' is displayed next to a 'CLAIM' button. The main content area contains a table with the following data:

VAT ID	12345678
COMPANY NAME	Domatiga2
PHONE NUMBER	244828282
CITY	Leiria 2
COUNTRY	country
EMAIL	email
WEB SITE	www.domatiga2.com
ACTIVITY SECTOR	IoT Master
STATE	Leiria
ZIP CODE	2400
ADDRESS	urb. Santa clara

Figure 50. Entity Information page

The user is able to edit the entity by providing the valid Entity Information which is then saved in the Database.

Quantum Portal

MAP

LIST

Edit Entity

CLAIM

ENTITY ID

COMPANY NAME

ACTIVITY SECTOR

PHONE NUMBER

CITY

Json Response:

Figure 51. Edit Entity (a)

ADDRESS

STATE

ZIP CODE

COUNTRY

ACCOUNT CONTACT

WEB SITE

VAT ID

SEND

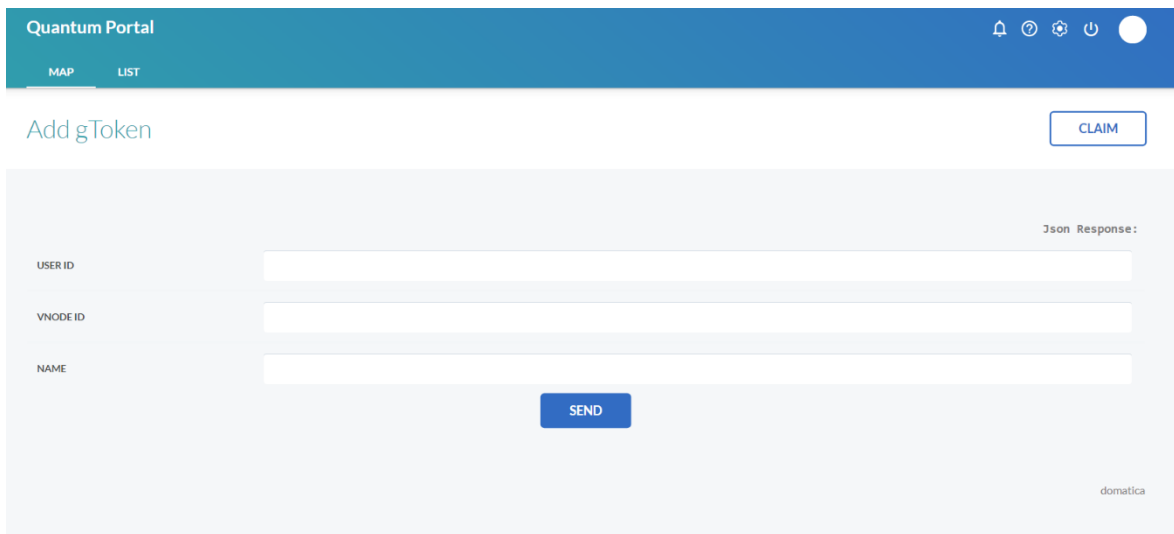
domatica

Figure 52. Edit Entity (b)

8.4 Tokens Management

8.4.1 Add Tokens

The user can add the tokens by their user authentication, by providing User Id, vNodeId, and Token Name. gToken is added in the Database and the message is received in response.



The screenshot shows the 'Quantum Portal' interface. At the top, there is a blue header bar with the text 'Quantum Portal' on the left and a series of icons (bell, question mark, gear, power, and a profile circle) on the right. Below the header, there are two tabs: 'MAP' and 'LIST'. The main content area is titled 'Add gToken' on the left and has a 'CLAIM' button on the right. The form itself is a light gray box containing three input fields labeled 'USER ID', 'VNODE ID', and 'NAME'. To the right of these fields is a label 'Json Response:'. Below the input fields is a blue 'SEND' button. In the bottom right corner of the form box, the word 'domatoca' is visible.

Figure 53. Add gToken

8.4.2 Remove Tokens

The user can remove the gTokens (gateway Tokens) by providing Token Id and vNode Id. It also involves User authentication.

The screenshot shows the 'Remove gToken' page in the Quantum Portal. The header is blue with the 'Quantum Portal' logo on the left and navigation icons on the right. Below the header, there are tabs for 'MAP' and 'LIST'. The main content area has a light gray background. At the top right of this area is a 'CLAIM' button. Below it, there are two input fields: 'TOKEN' and 'VNODE ID'. A 'SEND' button is positioned below the 'VNODE ID' field. In the top right corner of the main area, the text 'Json Response:' is visible. The 'domatiga' logo is in the bottom right corner.

8.4.3 List Tokens

The user can list the gTokens by providing User Id and vNodeId. In response, the list of gTokens is displayed.

The screenshot shows the 'List gToken' page in the Quantum Portal. The header is blue with the 'Quantum Portal' logo on the left and navigation icons on the right. Below the header, there are tabs for 'MAP' and 'LIST'. The main content area has a light gray background. At the top right of this area is a 'CLAIM' button. Below it, there are two input fields: 'VNODE ID' and 'USER ID'. A 'SEND' button is positioned below the 'USER ID' field. In the top center of the main area, the text 'Json Response:' is visible. The 'domatiga' logo is in the bottom right corner.

Figure 54. List gTokens

9. Tests

9.1 Authentication testing

Authentication ensures that only the registered users can login to the Quantum Portal. To check if the User is a Registered User or not, Laravel has a method that takes User data as arguments and compares the data with the data in the Database. And this way it ensures the authentication.

Another requirement is to protect the pages not to be able to accessed without authentication. In order to fulfil this requirement, middleware is created that protects from accessing the Quantum Portal pages without being logged in. For this purpose, 'auth' middleware is created and added to all the pages routes. The tests are made and as the result It is not possible to access the pages without the user being logged in.

9.2 Security testing

Security is an important aspect when it comes access the data. The data can be User data or Entity Data.

Laravel provides various mechanisms to ensure the security of a website. The Data is passed as arguments to the `hash()` function, and returns hashed value. This way the Data is protected.

Laravel provides CSRF Protection/Cross-site request forgery (XSS). These attacks occur when the attackers are skilled to place the client-side JavaScript code in the pages viewed by the Users. In order to avoid such attacks, Laravel provides a double brace syntax `{{ $value }}` in Blade template where this syntax protects the Raw data displayed.

The Quantum Portal is developed locally and make use of MySQL database.

Showing rows 0 - 1 (2 total, Query took 0.0020 seconds.)

SELECT * FROM `users`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

	id	firstName	lastName	telephone	email	password	remember_to
<input type="checkbox"/>	1	natasha	lopez	244828282	natasha.lopez@domatica.pt	\$2y\$10\$1qJg5PITQI7Mt2v3NFPyp.18zm9q07SXdGwPlk/10/0...	NULL
<input type="checkbox"/>	2	sujane	lopez	244828282	natashalopez6444@gmail.com	\$2y\$10\$ezCCAYO3/G2aqyEEvDyMBOf5ZSysJvro0AIBTbYZZgP...	NULL

Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Query results operations

Print Copy to clipboard Export Display chart Create view

Figure 55. Database User data

The above figure shows the MySQL database with registered user data, and it shows how the Passwords are encrypted.

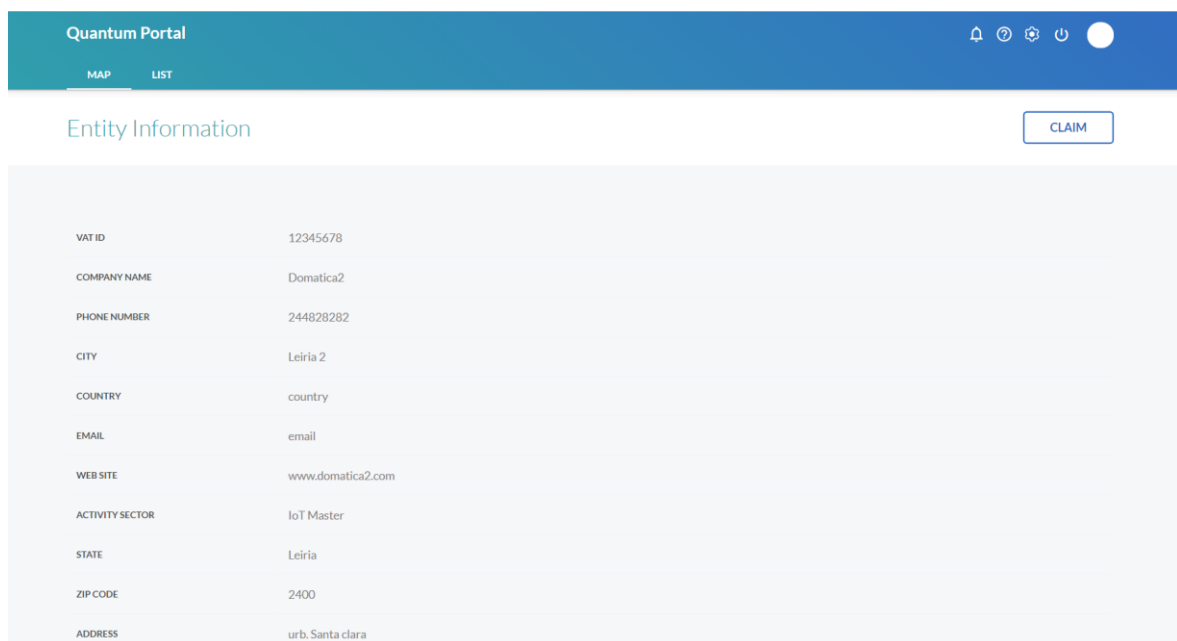
9.3 Application testing

Laravel has provided a TEST directory. When a new test case is created using artisan command line, the file is created in this directory. Tests are conducted to check the links and forms.

10. Results

10.1 REST API responses:

The data is passed using POST method from the view pages (blade files) to the Main Controller Personal Home Page (PHP) file methods. PHP Curl is used to send the data to the cloud database using the service URL. The data is passed in a JSON format. After the data is verified, response message is returned.



The screenshot shows the Quantum Portal interface. At the top, there is a blue header with the text "Quantum Portal" and navigation links "MAP" and "LIST". On the right side of the header, there are icons for a bell, a circular arrow, a gear, a power button, and a profile picture. Below the header, the main content area is titled "Entity Information" and includes a "CLAIM" button. The data is presented in a table format with the following fields and values:

VAT ID	12345678
COMPANY NAME	Domatiga2
PHONE NUMBER	244828282
CITY	Leiria 2
COUNTRY	country
EMAIL	email
WEB SITE	www.domatiga2.com
ACTIVITY SECTOR	IoT Master
STATE	Leiria
ZIP CODE	2400
ADDRESS	urb. Santa clara

Figure 56. Entity Information Response

The above figure shows one of the JSON response returned from the service URL. Given Entity Id and User authentication, JSON response Entity Information is returned. Similarly, all the other functionalities of the Quantum Portal work.

11. Conclusion

With the ever-growing technology, the need to provide a better service to the clients has become a priority for the companies. The present work aimed to aid in this direction, giving a better way to the clients to access. The Quantum Portal aims for Device management and Information management. The clients monitor and access their devices using this Portal. Clients can also request the necessary information from the Portal.

Developing the Quantum Portal using Laravel framework ensured security aspects. And also made it easy for us to develop, as it has lot many features. The development included several managements such as Accounts, Users, Nodes, Entity and Tokens. Also included Operations such as deploying system program and user program, and requesting the log files, system program and user program files. My part of work was Accounts, Nodes, Entity and Tokens management developing and designing. Testing the application locally.

As a person with no prior experience in developing a Web application, the time spent on learning the languages, tools required was worth. Thus, contributed to develop a Web application. Needless to mention, the development so far is not perfect and could be improved if enough time is provided.

12. Future Works

The present work is proportional to the back end developed. The database for now only has User authentication, gToken authentication, which is implemented in the present work. And few of the functionalities work as the data is not completely available online. So, the future work includes other authentication methods to be implemented such as Session Id authentication method. Improvements has to be done in claiming, releasing and replacing the nodes.

At present the clients are the single users, but in future, there will be two different roles. It is going to be Admin and normal users. Admins create the users, edit the users or delete them. In addition to this, Admins can also have access to all the managements and operations of Quantum Portal. But the normal users only have access to view their profile, edit profile, monitor the nodes. This can be implemented by having two different middleware's each for admin and normal user. This way it is possible to restrict the pages for normal users.

References

- [1] Lathifah. Alfat, 'Sentinel web: Implementation of Laravel framework in web based temperature and humidity monitoring system', Indonesia, Oct 16-18th, pp. 46-50,2015.
- [2] A.Valjerevic, H.S.Venter, M.Ingles, 'Towards a Prototype for Guidance and Implementation of a Standardized Digital Forensic Investigation Process', Pretoria,South Africa, 2014.
- [3] Ram Kumar Paliwal, Upasana Paliwal, 'A review study on secure web application development using PHP with Laravel Framework', Rajasthan, pp. 206-209, International Journal of Engineering Research in Computer Science and Engineering Vol 3, April 2016.
- [4] Y.Zhang, D. Geisert, M. Yang, 'Building genomics foundation for precision medicine research: A portable multitask data management system', U.S.A.
- [5] He Ren Yu, 'Design and Implementation of web based on Laravel framework', Nankai University Binhai College, pp.301-304, Published by Atlantis Press, 2015.
- [6] A. Singh, S. Sharma, S. Singh, 'Android Application Development using Android Studio and PHP Framework'.
- [7] Majid A. Al-Tae, Suhail N. Abood, Waleed Al-Nuaimy, Ali Al-Ataby, Zahra J. Mushin, 'Mobile Health Platform for Diabetes Management Based on the Internet-of-Things', UK, IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT),2015.
- [8] H. Saeed, A. Shouman, 'Near-Field Communication Sensors and Cloud-Based Smart Restaurant Management System', PP. 686- 691, Ontario, Canada, 2016, IEEE.
- [9] Awesome Websites built with Laravel framework: <https://codecondo.com/15-websites-built-with-laravel/>
- [10] Built with Laravel: <http://builtwithlaravel.com/>
- [11] Kyle Schutt, Osman Balsi, 'Cloud Software Development Platforms: A Comparative Overview', Baltimore, USA, June 8-10,2016.
- [12] Natalya. Prokofyeva, 'Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems', Latvia, December. pp. 52-54,2016.
- [13] Symfony framework: <https://symfony.com/>
- [14] Yii framework: <http://www.yiiframework.com/>
- [15] CodeIgniter framework: <https://www.codeigniter.com/docs>

- [16] Cake PHP: <https://cakephp.org/>
- [17] Trupthi Tiwari, 'Comparative study of different frameworks of PHP', pp. 264-267, International Journal of Research in Computer and Information Technology (IJRCIT), July,2016.
- [18] Lupasc Adrian, Dumitru Eugenia Georgiana, Niculescu Ana-Maria, 'Zend Framework', Galati,Romania, pp- 115-122,2013.
- [19] Kohana framework: <https://v2docs.kohanaframework.org/documentation>
- [20] Aura framework: <http://documentation.auraframework.org/auradocs>
- [21] Slim framework: <https://www.slimframework.com/docs/>
- [22] Silex framework: <https://silex.symfony.com/doc/2.0/>
- [23] Why is Laravel the best PHP framework in 2017?
<https://www.linkedin.com/pulse/why-laravel-best-php-framework-2017-ved-raj>
- [24] Laravel Benefits to solve web development problems
<https://www.youtube.com/watch?v=nYo3ROy9mfU>
- [25] M.Mahalakshmi, Dr.M.Sundararajan , 'Tracking the Student's Performance in Web-Based Education Using Scrum Methodology', Chennai, India, pp. 379-382,2015.
- [26] Laravel tutorials: www.tutorialspoint.com/laravel/
- [27] Laravel Documentation: <https://laravel.com/docs/>

Appendix A

ACTIVITIES	October [21st]	November	December	January	February	March	April	May	June	July
Training	X	X								
Planning and Analysing the Requirements- Initiated the Portal by designing for the Dashboard Menu, Map view with Quantum Nodes indicating Online and Offline Status.			X	X						
Nodes Management- Authentication methods for claiming nodes, edit nodes page with map view having a draggable marker to locate the new nodes, Replace and Release nodes.					X					
Account Management- User Authentication, Sending Email from the Company to the Registered Users, Email verification process, Password Recovery emails. Tokens Management- Design and development. Entity Management- Design and development.						X	X			
Change of design of the Portal, Responsive design with big screens and laptop screens, Implementation of the developed Portal to a new design.								X		
Worked with Online API, Sending and Receiving the Data from the Database through Cloud to the functionalities in the Portal and making it a Real-time application.									X	X
Report Work			X	X	X	X	X	X	X	X

Table 2. Internship Tasks

Appendix B

Activity tracking page for Waiters

Ali ShoumanHussain SaeedJane DoeMais ElfarMostafa ShabkaWaiter Test

Current Activities for Waiter Test

Table Number	Activity	Status	Time-In
4	Order is for this table is ready to serve	Incomplete	2016-05-19 01:57:24
4	Order is for this table is ready to serve	Incomplete	2016-05-19 01:57:53
4	Table is requesting you	Incomplete	2016-05-25 15:53:43
4	Table is requesting you	Incomplete	2016-05-25 17:27:35

Figure 57. Waiter page on the Web application

Order tracking page for Chef						
Table	Orders	Number of Orders	Comments	Time in	Order Start time	Order End time
4	Pancakes	1		2016-05-25 15:46:07	Get Time	Get Time
4	Fried Mars Bar	1		2016-05-25 16:07:13	Get Time	Get Time
4	Breakfast Burrito	1		2016-05-25 17:25:01	Get Time	Get Time
2	Oatmeal	1		2016-06-07 13:55:04	Get Time	Get Time
2	Oatmeal	1		2016-06-07 13:55:04	Get Time	Get Time

Figure 58. Cook page on the Web application

Daily Accounting Information		Statistics for Resturant Performance	
Statistic	Value	Statistic	Value
Total Revenue Earned Today	\$0.00	Number of Orders Placed	0
Total Expenses Today	\$0.00	Most Ordered Dish	N-A
Total Payroll Today	\$0.00	Average Time to Start Preparing an Order	00:00:00
Net Income Earned Today	\$0.00	Average Time to Complete Preparing an Order	00:00:00
Total Number of Sales Today	0	Average Time to deliver Order to Customer	00:00:00
Average Net Income per Sale	\$0.00	Total Average Time to Complete an Order	00:00:00
Busiest Hour(s)	N-A		
Slowest Hour(s)	N-A		

Figure 59. The 'View Analytics' feature of the Manager page

Appendix C

	CakePHP2	CodeIgniter	Symfony2	Yii	Phalcon
Newest version	3.3.6	3.1.0	3.1.4	2.0.9	3.0.0
Release date	2016 Oct 14	2016 Jul 26	2016 Sep 3	2016 Jul 11	2016 Jul 29
License	MIT	BSD-style	MIT	BSD	New BSD License
Requirements	>= PHP 5.2.8	>= PHP 5.1.6	>= PHP 5.5.9	>= PHP 5.4	>= PHP 5
ORM	Custom	ActiveRecord	Doctrine 2, Propel	Database Access Objects (DAO), Active Record	Phalcon
Code generation	CLI	Combustor	CLI	Yii CLI, Gii (Web based)	Phalcon Developer Tools
Template engine	Custom, bet Smarty/Twig	PHP, Simple template parser "{var_name}"	PHP, Twig	Razor, Smarty, Twig etc.	Volt, PHP
CRUD generation	Scaffolding	Grocery CRUD	SensioGenerator Bundle	Gii	Scaffolding
Web-site	cakephp.org	codeigniter.com	symfony.com	yiiframework.com	phalconphp.com

Figure 60. PHP Frameworks Comparison