Umesh Raj Satyal

A Website Design with the Mobile-first Approach

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Thesis

18th March 2016

Abstract

Author(s) Title Number of Pages Date	Umesh Raj Satyal A website design with the mobile first approach 30 pages March 18, 2016
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Software Engineering
Instructor(s)	Olli Alm, Senior Lecturer

This thesis aims to explain the mobile first design approach in web design and development and its advantages over other prevalent approaches.

In the study, the mobile web, its trends and different approaches for design and developing a mobile first web application are described and a comparison between mobile sites and mobile first sites is made. The concept of responsive web design and different grid systems are discussed as they are essential factors for responsive web design.

During the project, a responsive, mobile-first sports and activity portal was built. In the portal users can browse among different sports and recreational activities and choose the ones that interest them most. Users have a choice to choose activities with respect to price, place and activity provider. A map is also embedded with the search result interface so that users can easily get into the venue.

The portal can be used for the effective planning of vacation and leisure time. For service providers, it makes them easy to advertise their services. For those service providers which do not have a sufficient budget for creating and maintaining an online booking system, this portal can also be their official booking portal. Preliminarily, it is in the English language but it aims to be a multi-lingual website.

Keywords	Mobile-first design, Responsive Web Design, Bootstrap Front-end Framework

Contents

1 Introduction

During the last 10 years, mobile phone industries have been growing rapidly and smart phones and tablets are replacing the traditional PC-based web surfing. Since 2014, the first time in the history of the internet mobile usage has exceeded the traditional PC-based web surfing. Here the mobile usage means the combined uses of mobile applications and mobile-based web surfing. [1]

Different mobile operating systems need different tools for application development such as Android, iOS, and Windows. The applications made in those platforms should be installed on the phone to use them. Despite the fact that there are mobile applications, most of the web traffic is happening through mobile web browser. [1]

We are having several varieties of different screen sizes and we cannot adapt our websites to specific device types. So we have to check the device characteristics and scale the screen and other characteristics to find out if it fits and behaves well in all screen sizes. This is the responsive approach of web design. These days browsers are on mobile phones, desktops, televisions, tablets and even in watches.

The current mission of W3C includes ensuring the Web to be available on as many kind of devices as possible that are available in the market. With the development of small screen handheld devices in recent 5-7 years, the role of the Web as a platform for content, applications and services on these devices is increasingly important. [17]

The project done aimed to develop an application in mobile-first approach which is equally responsive to small screen, medium screen and large-screen devices. The application is a booking portal and the targeted customers are the ones making plans for their leisure time and holidays, so they can be tourists as well. The development team consisted of a back-end developer, a front-end developer and a lead developer who was responsible for bridging back-end and front end. I was responsible for all the front end design and development of the portal. The aim of the thesis is to describe all the design ideas and components of the application more specifically about the front end layout and responsiveness of the application.

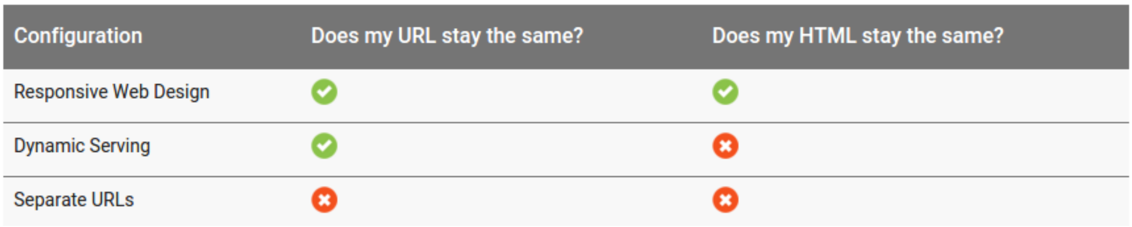
2 Mobile-first Web Design Approach

Mobile first web design is the concept of designing a website for mobile which is also adaptable in tablets and desktops [4, 111-112]. It is closely related with the content first design approach. The content first design approach is such a type of approach in which content is taken as the main consideration in designing [16].

Google characterizes three common ways to implement a mobile site:

- Responsive web design: Using responsive web design serves the same HTML scripts regardless of the users' device but it can render a different display based on the device screen size.
- Dynamic serving: It is a setup where the server asks a browser for user agent and responds different HTML and CSS depending upon that.
- Separate URL (standalone site): It serves different code to each device with different URLs. The configuration made for this method tries to detect the user's device and then redirects the suitable page for the browser using HTTP redirects. An HTTP redirect is an URL redirection technique. [2]

The information above is illustrated in figure 1.



Configuration	Does my URL stay the same?	Does my HTML stay the same?
Responsive Web Design	✓	✓
Dynamic Serving	✓	✗
Separate URLs	✗	✗

Figure 1. Mobile web, reprinted from google developers. [2]

Since the start of the 21st century, the web has become an integrated part in the human life and there have been significant changes in the way people use it and access it. It has brought new challenges for the business using the website as a tool. One could just concentrate on the PC while designing a website 5-7 years ago but then bunches of small screen and handheld devices were invented that could access the Internet and provide the facility of web surfing and people started using it heavily as shown in figure 1. To address it, many websites just shrank their regular desktop based website and made it to fit the smaller screen, which caused a difficulty to view the websites then there came the approach of designing separate websites for desktops and mobile, which was costly. Then the concept of responsive web design is brought by Ethan Marcotte on 2010. It will be discussed in more detail in 2.1. [4]

Mobile websites are usually basic and simple to start with the most important information and features. Then more detailed contents, widgets and features can be added when the screen is stretched to bigger. Thus, mobile first approach is easy and fast to start with and getting popular too and it is most suitable option for the start-ups. However to customize the running desktop website, it is easier to make mobile optimization and tablet optimization, so desktop first approach may be more useful in this case. So, reason for choosing mobile first and desktop may vary. It also depends upon the business and target customers because the goal is to provide the best user-experience to users.

2.1 Responsive Web Design

The term Responsive Web Design (RWD) was coined by Ethan Marcotte in his article "Responsive Web Design" published in 2010. According to Marcotte, responsive website uses fluid grids, flexible images and media queries to scale seamlessly according to the viewport size. It is a modern approach of web design. As the number and type of web surfing devices noticeably increase, the size and resolution of the screen also vary. RWD aims to address the different screen sizes providing maximum viewing experiences. A website designed with this method is called the responsive website. [3]

It has become immensely popular and is being widely used on the web these days because of its objective of providing better user experiences for users of all devices by allowing the website to adapt to their size, creating support and flexibility. It is a client-side concept and often used together with the term fluid-design because it transforms its layout automatically as we start decreasing the browser's width, without refreshing the page. Marcotte (2010) states in his book that a responsive design is composed of three distinct parts:

- A flexible, grid-based layout (fluid grid)
- Flexible images and media
- Media queries, a module from the CSS3 specification [4, 9]

2.2 Fluid Grid

Fluid grids are the key requirements for RWD. The components in fluid design flow and adapt to the user's environment on any kind of devices and screen sizes. Pixel-based dimensions are defined in adaptive grids, which leads to manual adjustment of height and width on certain devices but the fluid grid flows naturally within the dimension of the parent container, so that only limited adjustment works for various screen sizes and devices. With the rapid and competitive development in technologies, device manufacturers are trying to make the devices more user-friendly. It is leading the decrement in the screen sizes of mobile devices and on the other hand desktop monitors are getting wider with high resolutions which is making it difficult to plan responsive design for smaller devices. In the fluid grid system it is possible to adjust the maximum width for mobile devices. The system still works on large screen monitors because it uses percentage based calculations. [4]

In fluid grids, the maximum layout size is defined and then the grid is divided into a specific number of columns which keeps the layout clean and easy to handle. The elements are designed in such a proportional way that it will adjust when the screen size is changed.

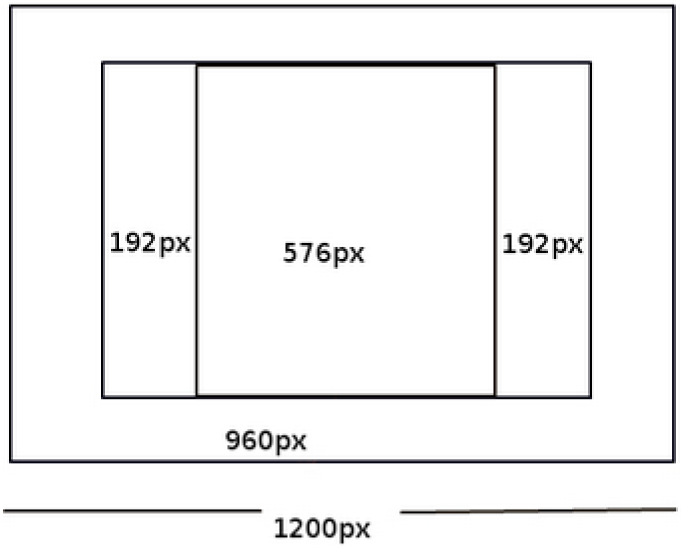


Figure 2. Grid layout with fixed unit

In figure 2 the screen resolution is 1200 px and the main container takes 960 px of area. It is further divided into three containers with the width 192 px, 576 px and 192 px which is an example of a traditional typographic grid layout with a fixed width.

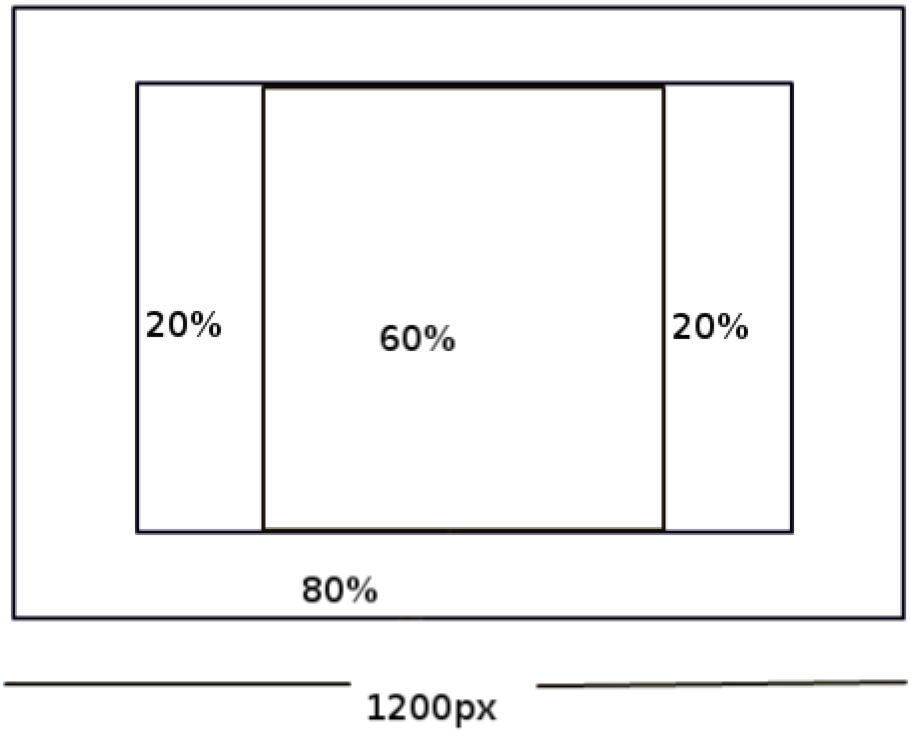


Figure 3. Flexible grid layout with percentage unit

The fixed pixel units can be converted using the relative percentage unit by dividing the target with the context and multiplying it with 100. The context represents the parent container and the target represents the child. [4, 19-23]

Target/Context * 100 % = Result
192/960 * 100 % = 20 %
576/960 * 100 % = 60 %
192/960 * 100 % = 20 %

Figure 3 shows the relative percentage distribution of the same layout that is shown in figure 2.

Fluid grids are not a definitive feature for having a responsive website. Although fluid grid adapts to the change in the size of windows or devices, using it still has a limitation of the contents to look relatively small in small screen devices. This decrease in size of the contents following a decrease in screen size demands for a careful planning of the design.

2.3 Flexible Image and Media

Traditionally images and videos has not been responsive in web sites: they are rendered once and their viewport size remains fixed once loaded.

The designer Richard Rutter discovered a rule for changing fixed images into fluid images. He used the maximum width of the image to 100%. This fluids the image with the viewport regardless of the size of the viewport. [5, 45]

```
img,
embed,
object,
video {
 max-width: 100%;
}
```

It provides the immediate solution to all the media element of any size. It renders at whatever size it wants until it reaches the size of the parent container. When it exceeds the parent container then the max-width directive forces the media to match the width of it. All modern browsers support this property of making images fluid.

2.4 Media Queries

Media queries became the W3C recommended standard in 2012. Media query was initially introduced by a Norwegian web pioneer Håkon Wium Lie in 1994. It is a CSS3 rule which allows the content to render with respect to the media screen sizes. A website asks the browser which kind of device or screen size the current user is having and delivers accordingly, so that different screen sizes can have different style with the help of media queries.[6]

There are three ways of including media queries in an application.

- It can be inserted in to the head section inside a link tag:

```
<link rel="stylesheet" type="text/css"
 media="screen and (max-device-width: 480px) and (resolution: 163
dpi)"
 href="shetland.css" />
```

- It can also be placed inside CSS file with @media rule :

```
@media screen and (max-device-width: 480px) {
 .column {
   float: none;
 }
}
```

- It can also be part of an @import directive:

```
@import url("shetland.css") screen and (max-device-width: 480px);
```

It checks the width of the device. If the given statement is false, it ignores the style but if it is true it uses the style.

```
@media screen and (max-width: 300px) {
   body {
       background-color: blue;
   }
}

@media screen and (min-width: 300px) {
   body {
       background-color: red;
   }
}
```

Let us take an example of a web page which contains the media queries mentioned above. When it is full screen, its body will have a background colour red and when we reduce the size of the browser and make it to 300px, the background colour will change in to blue. In this case, 300 px is the break point for the browser.

Different break points can be set for desktops, tablets and mobiles and give style to them on a single web page, so that it will behave well and look nice on all the devices. This is the goal of responsive web design.

2.5 Responsive and Adaptive Design

Before comparing responsive design with adaptive web design (AWD), let me briefly introduce the adaptive design approach. Adaptive web design was coined by Aaron Gustafson. It uses static layouts which are based on different breakpoints and do not respond initially. The common approach of designing an adaptive web page includes designing static sites for a minimum of six screen widths that are: 320 px, 480 px, 760 px, 960 px, 1200 px and 1600 px where 320 px is the width of the extra small device. It is also worth mentioning here that the responsive design approach came later than adaptive web design and solved the problems related to adaptive web design. [8]

The ultimate aim of both RWD and AWD approaches is to allow websites to be viewed with different screen sized devices. They only differ with the method of delivery. RWD is based on flexible fluid grids where there is one layout which changes with the screen sizes but AWD is based on predefined screen sizes. The choice of method depends upon the user's necessity and the target audience and budget of the company.

After defining different terms and techniques used it is much easier to define the mobile-first approach in this stage. Using media queries lets developers choose base CSS classes and ids. If base CSS classes and ids are set for mobile devices, the approach is the mobile-first design approach. Additional or different styles can be applied using the `@media` rule for other screens that are larger than that on a mobile device. `@media` is a CSS rule to include a block of CSS properties if a certain condition is true. This will help the web pages to load faster on smaller devices as it does not need to pass through different `@media` conditions.

2.6 Responsive Web Design and Standalone Mobile Site

These days the common practice includes designing a web site for all the targeted devices using RWD or designing a standalone site for mobile devices. Some of the strengths of RWD include the following:

- Single website renders in all the devices, so, the hosting cost is minimized. One set of code work for all the platforms.
- Building a single website is much cheaper than building separate sites for desktops and mobile devices. It means that the initial cost is comparatively low. Start-up, not having sufficient funding may fit in this.
- The same design works for mobile and desktop sites and they can have the same functionalities.
- It is elegant and streamlined and can have legible content and great UX experiences.
- Updating the content is straightforward as it is not necessary to update it in two places. [4][8]

There are some drawbacks of using responsive web design. Some of the common criticism include the following:

- It cannot be quickly fixed. Maintenance may take a longer time and it may cost high.
- File sizes are quite large which may cause slower viewing and downloading may take a longer time if it is not designed wisely.
- It is challenging to get it working on all platforms. Testing takes a longer time and can also be expensive. [4][8]

While going responsive the specific content for the mobile can be chosen using CSS and JavaScript and it also reformats the pages on the browsers on the client side. Most likely, responsive web design is the future of web design and it will replace separate designing for mobile sites.

2.7 Mobile Content Optimization

Content optimization is a very crucial element in the web design and development process. Businesses that use a website which renders across a wide variety of mobile devices is very critical. A smart mobile strategy extends beyond design to content development as well. [7]

The main factors leading the necessities of mobile content optimization are screen size, performance of the device and the internet connection. Images are the heaviest elements used on a site and they affect the performance. It takes a long time for images to render and fit the screen size which may increase the probability of the users to be impatient and hit the back button or quit surfing. So it is very important to provide different sizes of images for mobile sites to those on desktops which are light in size. Following the images, optimization and minification of other resources like HTML, CSS, and JavaScript also play an important role in performance.

3 Bootstrap 3 (Mobile-first Design Approach)

Bootstrap is the design framework used in this project. It has also a great role in the promotion of the mobile-first design approach. Bootstrap 3 is the latest version of Bootstrap from which it started to follow the mobile-first approach. Before that it was a desktop-first responsive web design framework. This part of the thesis contains a comprehensive description of it, the reason for choosing it, file structure, implementation of CSS, grid system, media queries, JavaScript components and browser support.[10]

Bootstrap is a front-end design framework which is written in HTML, CSS, LESS, SAAS and JavaScript. As of February 2015, it was on the top of the list on GitHub which has over 77,920 stars and 30,008 forks [9]. Aiming at consistency in web development and internal tools, Bootstrap was developed by Mark Otto and Jacob Thornton from Twitter and released in August 2010. Since then, it has been the most starred GitHub development project. The current stable release of Bootstrap is version 3.3.2 and it is distributed under MIT License (Apache License 2.0 prior to 3.1.0) [10]. It easily and effectively scales the website with a single code base from small screen devices to large screen devices with the help of CSS media queries.

The older releases of Bootstrap were targeted at the desktop clients and they are modified for tablets and mobiles but with the latest release (version 3) of Bootstrap, it is going with the mobile first approach. So, the sites designed with this version target to the mobile devices and it scales to the desktop, tablet and other larger screen devices depending upon the size of screen. For presentation and layout, it uses a grid-based system which includes four grid classes for four different types of device classes: large screen desktop, regular desktop, tablets and phones.[10]

Designers choose a framework to make similarities in the codes. It is the tool that makes development faster and smoother. There are many front-end frameworks available. The reasons for choosing Bootstrap as a framework may vary, but its main strengths could be as follows:

- It has followed the "mobile first" approach from version 3 (the current version) and it works on all modern browsers.
- It is lightweight, consistent, customizable and encourages the use of LESS CSS
- It comes with jQuery plugins and some of the jQuery plugins also offer bootstrap theming.
- It has a variety of good looking templates available to start with.

Although Bootstrap is highly customizable, many sites using it look alike, naming of the classes is mediocre semantic and the jQuery plugins used are limited.[10]

3.1 Bootstrap CSS

CSS of Bootstrap is built using LESS. LESS is a CSS pre-processor with additional functionalities such as variables, mixins and functions which after compilation changes to a normal CSS file. It makes the task easier as we do not need to change the value of each class or id like in a normal CSS file. We can just change the value of one variable and recompile it. Along with Less, Bootstrap has also an official SAAS port, which is another CSS pre-processor [10]. Some important CSS components of Bootstrap are briefly discussed below.

Bootstrap Grid System

Grid systems are used to create a page layout with rows and columns that house the contents. Bootstrap follows a mobile first, fluid grid system which divides a page up to twelve columns as the device or viewport size increases. It is in the mobile-first approach so the base styles in the stylesheet are targeted at mobile platforms and then modified when device width scales to larger screens.[10]

Container and Container-fluid

Rows must be kept inside either container or container-fluid. The container has fixed width and container-fluid has full-width. It cannot be nested. It has no width for extra smaller devices so that it fits automatically: 750 px for the small devices with the screen size more than 768 px, 970 px for the for the medium devices with the screen size more than 992 px and 1170 px for the large desktop devices with screen size more than 1200 px.[10]

Media Queries

Bootstrap has media queries inside its LESS file, which creates the key break points in the grid system. The breakpoints are set to 1200 px, 992 px and 768 px for the large screen desktop, desktop and tablets respectively. There is no media queries for mobile phones with screen sizes less than 768 px because it is the default view of the platform. For small devices like tablets which have a minimum width 768 px, the following media queries are used:

```
@media (min-width: @screen-sm-min) { ... }
```

For devices like desktops which has a minimum width 992 px, the following media queries are used:

```
@media (min-width: @screen-md-min) { ... }
```

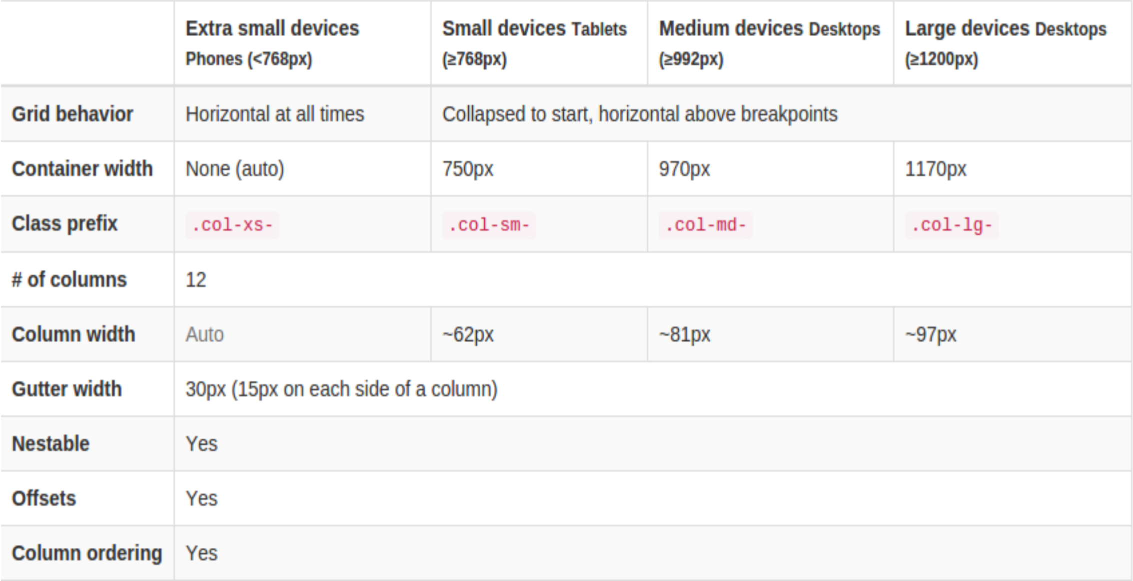
For large screen devices which has a minimum width 1200 px, the following media queries are used:

```
@media (min-width: @screen-lg-min) { ... }
```

Including max-width as a parameter limits the CSS to narrower set devices. [10]

Columns

Table 1: Bootstrap grid options



	Extra small devices Phones (<768px)	Small devices Tablets (≥768px)	Medium devices Desktops (≥992px)	Large devices Desktops (≥1200px)
Grid behavior	Horizontal at all times	Collapsed to start, horizontal above breakpoints		
Container width	None (auto)	750px	970px	1170px
Class prefix	`.col-xs-`	`.col-sm-`	`.col-md-`	`.col-lg-`
# of columns	12			
Column width	Auto	~62px	~81px	~97px
Gutter width	30px (15px on each side of a column)			
Nestable	Yes			
Offsets	Yes			
Column ordering	Yes			

A row is divided into twelve columns which are the immediate children elements of a row. The contents of the page should be kept within columns. .col-xs.*, .col.md.*, col-sm.*, col-lg-* scales the layout in the desired number of columns and col-*-offset-* can set the required margin between the columns.[10]

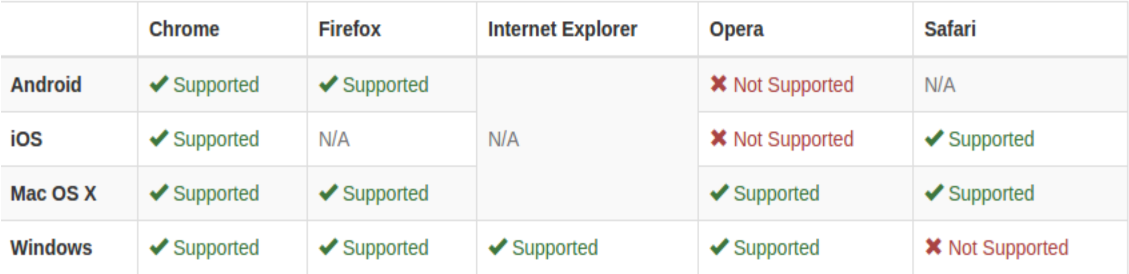
3.2 Bootstrap JavaScript

Bootstrap has several JavaScript components in the form of jQuery plugin.
It can be placed in an application individually using Bootstrap's individual *.js files as per the requirement of the application or the whole JavaScript file bootstrap.js or a minified file bootstrap.min.js. Plugins do not fall back particularly gracefully when JavaScript is disabled on the browser but user experience may fall.[10]

3.3 Browser Support

Bootstrap is specially targeted at the latest desktop and mobile browsers. Older browsers might display styles differently though it is fully functional. It supports Internet explorer 8 and above. The browser support of Bootstrap is tabulated as follows:

Table 2 Browser support of Bootstrap[10]



	Chrome	Firefox	Internet Explorer	Opera	Safari
Android	✔ Supported	✔ Supported	N/A	✖ Not Supported	N/A
iOS	✔ Supported	N/A		✖ Not Supported	✔ Supported
Mac OS X	✔ Supported	✔ Supported		✔ Supported	✔ Supported
Windows	✔ Supported	✔ Supported	✔ Supported	✔ Supported	✖ Not Supported

The table consists of three characteristics supported, not supported and N/A. N/A here means that it does not exist at all. For example N/A for Internet Explorer on Android means that we cannot have Internet Explorer on Android at all.

4 Bookndo.com

This chapter is about the design process of a mobile responsive site. It starts with a brief introduction of the project and company and consists of requirement analysis, design tools and finally it takes the reader through the design process.

The website bookndo.com is a product of Bookndo which is an auxiliary company name of Gearent Oy, a limited company registered in Finland. It is a company which works in the principle of software as a service which means it is a company that provides services. The services are related to sports and they can be activities, equipment rentals, facility rentals, lessons and the courses, guides and equipment repairs. [11]

There are different online booking systems for flights and hotels available on the market and the concept was taken from them. The booking system aims to fulfil the gap between sports activity providers and the activity users firstly on the Finnish market and then the rest of Europe and possibly the remaining parts of the world in the future.

The companies that provide facilities are service providers for Bookndo and the users can be individuals interested in sports, tourism, companies and students.

4.1 Requirement Analysis of Bookndo.com

As already mentioned in section 4, the project was carried out to bring sports activity service providers and users to bring together and make the booking process easier. The primary goal of the project was to build a mobile-friendly web based application where service providers can make their profile, fill the schedule, rate, terms and conditions, payment methods and also upload the images related to their sports activities. They should also be able to locate their location on the map. Users can book the activities easily using any types of web surfing devices such as desktop computers, tablets, smartphones and any type of browser such as Google Chrome, Safari, Firefox, Opera etc. The admin from Bookndo can monitor the activities and take action on false booking and register and also cancel and notify the users.

Analysing the requirements on the ground level, the application has three basic requirements which are listed below:

- Web interface which can be used by service providers to make their profiles and write in details about their service.
- Web interface and service which can be used by users to book and pay for the activities.
- Web interface for the admin to monitor booking.

In addition to these functional requirements, the application has also some technical requirements, which are listed below:

- The application should be written in HTML5 and CSS3 should be used to style it.
- The application should be responsive and mobile first.
- Bootstrap should be used as the front-end design framework.
- The application should use PHP for the server-side scripting language and postgresql for storing data.

- Cakephp should be used as the web application framework.

This project was done to build an online booking system (portal) which should be responsive, secure, dynamic and fast-loading. HTML5 is the latest version of HTML when combined with CSS3 is being discussed as the future of web development. The option of HTML5 could be on earlier version of HTML (HTML 4.01). Some reasons for choosing HTML5 as the required scripting language in this project are listed below:

- It is a new standard and all modern browsers support it.
- The development process becomes faster so it can be cheaper also.
- It is mobile- friendly.

When discussing HTML5, it is assumed the CSS3 is included in it. It is a new release of CSS which supports all modern browsers for styling. It allows effects like shadows, animations, canvas and rounded corners with a few lines of codes, which was not possible with the previous versions of CSS.

Since there are no separate sites for mobiles and tablet, the application should work on both of them. Therefore, the Bookndo management decided that the application should be responsive. It should also work fine and provide good user experiences with all the sizes while resizing the browser. Media elements like images and videos should work fine with all types of devices. The application is targeted at the sports enthusiastics and they may find using the mobile as a web surfing tool a better option because it is hand-held and will be with the user most of the time. By choosing the mobile-first strategy we wanted to be connected to the users all the time and we also expected more traffic to the portal from mobile devices.

Bootstrap was described in details in chapter 3. The application should use it as a front-end framework because

- It uses HTML5/CSS3
- It is a responsive design framework and follows the mobile first approach.

This requirement is the result of the first and second requirements.

4.2 Development Environment Setup

After analysing the requirements and technologies, the following tools were installed on the development machine.

Sublime Text 2
It is a text editor for code, markup and prose.

XAMPP
It is a PHP development environment which provides the web server to test a website locally.

Heroku toolbelt
Heroku is a cloud-based platform which allows to easily manage the deployment of an application. Bookndo uses Heroku.The Heroku toolbelt is a command line interface which is used to create and manage Heroku applications.

Git
Git is used for version control. It allows a team of people to work together using the same files without confusion.

PHP and postgresql
PHP is a server side scripting language used in the project and postgresql is a database management system.

Slack
It is a real time messaging application. It is used as a communication tool between the development team members.

Trello
It is an application to organize the project into boards. It is used as the work division tool between the development team members.

The development team consisted of three members: A front-end developer, a back-end developer and a lead developer. I was responsible for front-end development. Front-end development is client-side development which consists of planning, designing and developing the part that is visible to the client and users viewing the web page.

Different members of the development team were using different machines as development machines, so the tools varied. Since the development process was done in a team, it was first tested with a localhost using XAMPP and then pushed to the server using the Heroku toolbelt. For the version control, Git was used.

After installing and setting up the development environment, Bootstrap and Cakephp were downloaded and installed in the project folder. Since Bootstrap works with JQuery, it was also downloaded and installed in the project folder.

4.3 The Design

In this section the design process of the mobile first site in practice is discussed keeping the user interface in centre. Designing and developing different UI components for a mobile first website and their behaviour while scaling them to a larger screen is also discussed. The design contains the latest and prevalent approaches on the market.

Since Bookndo.com is an online booking portal the main element contains the information regarding the service providers, their location in the map, and the booking system. The layout components, their nature and content and the design method are briefly described below.

4.3.1 Top Navigation

The navigation bar of the website is static which means that navigation bar is always visible on top when page is scrolled. It is an important feature of a website because it contains a lot of useful content. In this particular website, the buttons for signup and login are on the navigation bar, which makes it more meaningful to make it static.

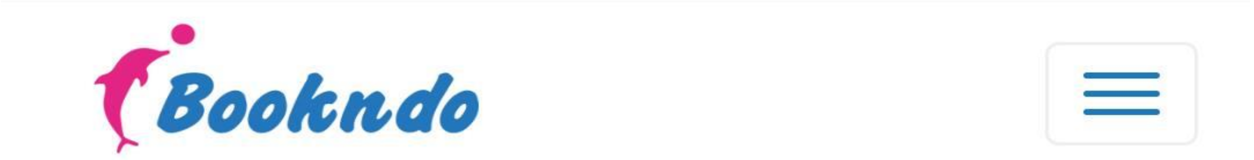


Figure 4. Top navigation of the website

Keeping the most important content on the site in the beginning and adding more content and styling them in a different way is the main designing principle of mobile-first design. Utilising the same principle, the company logo and a drop-down menu containing the navigation elements are kept in the site.

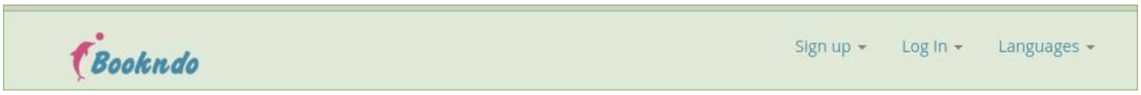


Figure 5. Top navigation of the desktop/tablet view

For the desktop and tablet views, there are more space available, so it is expanded to show all the list items as in figure 5.

4.3.2 Main Search Bar

The main search bar follows the navigation bar. Users can choose a place, specify the number of people and search for the available activities. It is a form element. Initially it is vertical, when it is stretches and more spaces will be available, it will become horizontal. It has three layers. The first layer contains two inputs and a search button, the second layer is a transparent black background and the last layer is carousel (slideshow) of images of different sports activities.

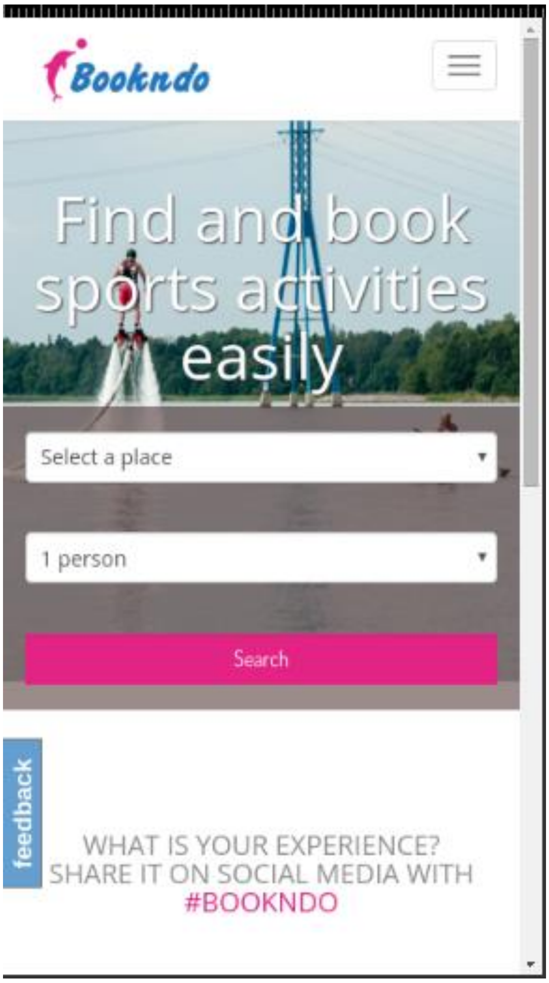

Bookndo
Find and book sports activities easily
Select a place
1 person
Search
feedback
WHAT IS YOUR EXPERIENCE? SHARE IT ON SOCIAL MEDIA WITH #BOOKNDO

Figure 6. Main search bar mobile view

The figure 6 shows how the main search bar is implemented in mobile.

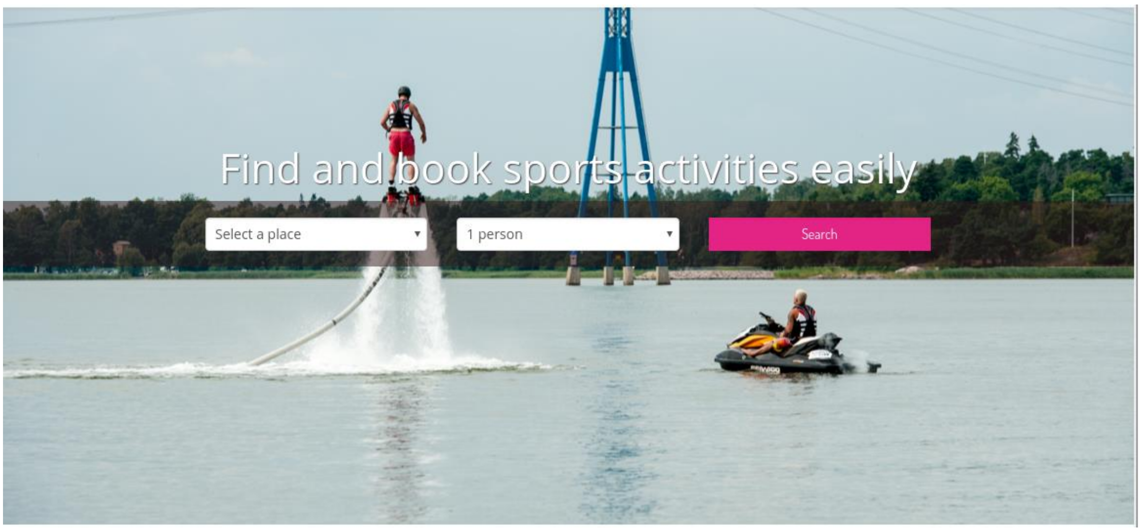


Figure 7. Main search bar desktop/tablet view

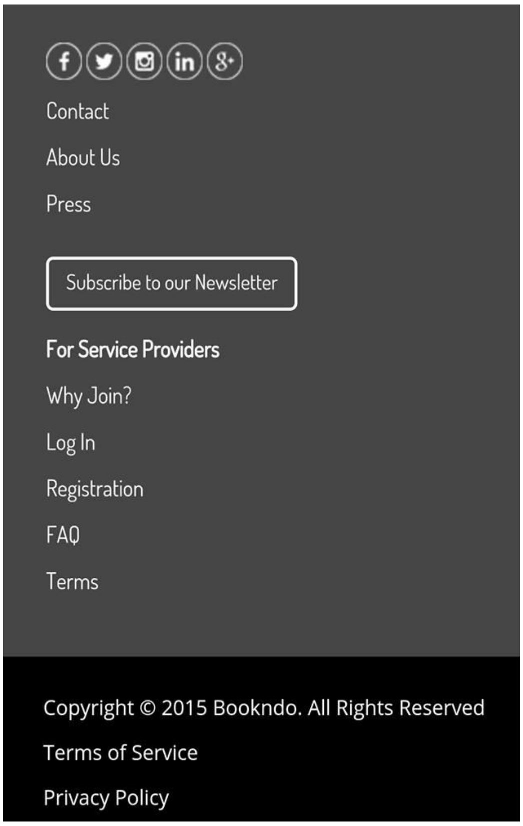
The figure 7 shows how the main search bar is implemented in the desktop.

4.3.3 Footer

The footer of a website is equally important as a header. They are the key elements containing links to the important pages which potential customer or user wants to visit before making any purchases. It can also be the last opportunity to grab the attention of customers not to leave the website.

The Bookndo footer contains two rows. The upper one is also the main footer of the website with a grey colour background which has important links to the pages. It has social media links to the Bookndo, contact information, about Bookndo, press release, a link to subscribe the newsletter. It also contains some extra information and a link to the frequently asked questions. Since it is the last opportunity to get the user's attention, it also contains links to registration and login. The lower row has a black background to make it unique and nicely visible which contains links to the terms of services and privacy policy.

The footer is divided into three equal columns using a col-md-4 class. It will appear as a single column on small screen devices and when it scales to large-screen devices and more spaces are available, it will be a three-column layout.


Contact

About Us

Press

Subscribe to our Newsletter

For Service Providers

Why Join?

Log In

Registration

FAQ

Terms

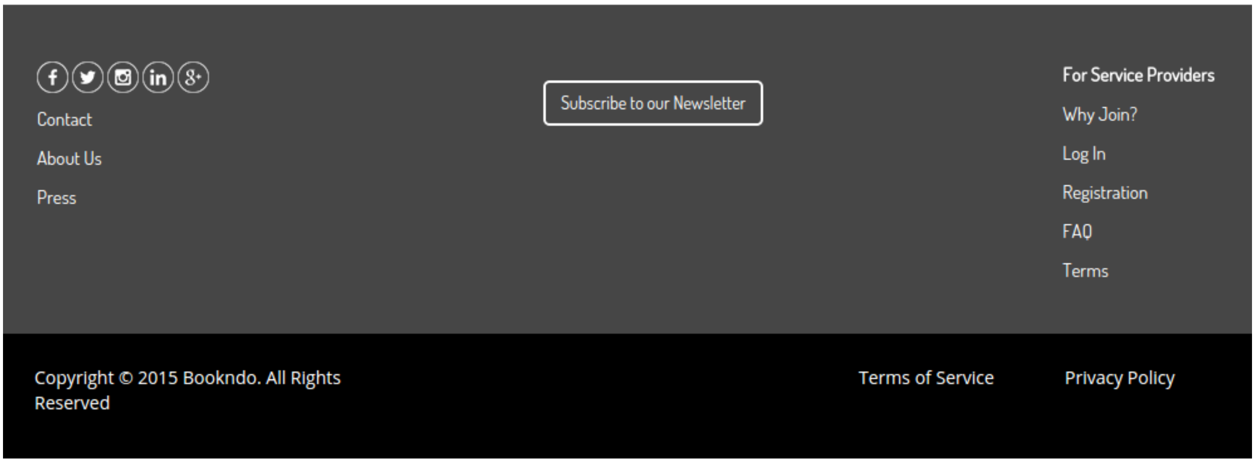
Copyright © 2015 Bookndo. All Rights Reserved

Terms of Service

Privacy Policy

Figure 8. Footer mobile view

The figure 8 shows how the footer is implemented in mobile.


Contact

About Us

Press

Subscribe to our Newsletter

For Service Providers

Why Join?

Log In

Registration

FAQ

Terms

Copyright © 2015 Bookndo. All Rights Reserved

Terms of Service

Privacy Policy

Figure 9. Footer desktop/tablet view

The figure 9 shows how the footer is implemented in mobile.

4.3.4 Image Adjustment

Bookndo is a portal related to sports activities, so it contains plenty of images of different activities. It is very important to make them clear, nicely visible, well labelled, nicely aligned and well adjusted. The images are edited using Photoshop and depending upon the position of the image, different techniques are applied to make them responsive. Bootstrap has its own class img-responsive with the properties display: block; maximum-width: 100% and height: auto; which helps images to scale nicely with parent elements. Making height automatic and full width helps images to overflow as shown in figure 10.

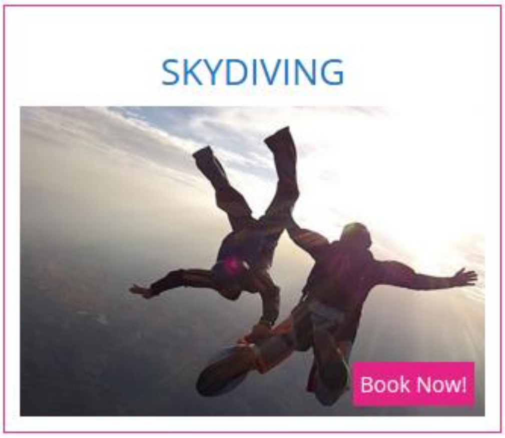


Figure 10. Image from search results

To summarize the design process, make the design interactive, user friendly and consistent, some steps taken are listed below:

- Shape, size and border radius of the button is the same throughout the website.
- The same margins and paddings are used on all the subpages and the units used are em(relative units) and percentages so that they scale with the device. It helps all the subpages look similar and consistent.
- Forms used in the design are similar and consistent. The necessary fields are marked with asterisks (*) and form handling is also applied to each form used to make them user friendly.
- Headings and subheadings are consistent. They have the same size and color throughout the website.

- The mobile view is always kept in the centre while designing. Bootstrap is heavily used.
- The codes and scripts used in the design follow the DRY (Don't repeat yourself) method. It is heavily commented to make future maintenance easy. The classes and methods used just do one thing which helps to ease the confusion in the design process.
- The codes and scripts used in the design are reusable and also can be taken as a reference. Modification and further enhancement is easier.

4.4 Mobile Content Optimization in Practice

The user's perspective and usability is always kept in high preference in the whole designing process, so that the contents and images are optimized to provide the best performance and views from mobile devices. Different images are assigned for light-weight mobile devices. Different images are treated differently by keeping the user's perspective in mind. Some images are cropped and some are resized to assure the user's experience as enjoyable and seamless which is consistent across different platforms.

Bootstrap provides classes like hidden-sm to hide elements on small-screen devices and hidden-lg, hidden-md,hidden-xs for large, medium and extra small screen respectively. This makes it easier for mobile users to hide the contents that are not necessary for them. For example, users generally do not want to fill in large forms or write an email from mobile devices to contact. Therefore they are hidden on mobile device screens and appear only on large screens.

CSS and JavaScript files are minified and compressed. Unnecessary elements and removed as the browser has less data to process and render. Also, Bootstrap provides the files in minified form like *.min.css, *.min.js etc.

4.5 Testing and Performance Result

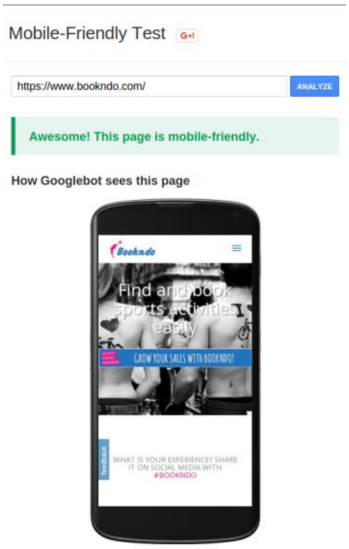
Testing of the website is done on a regular basis by developers/designers and also by someone other than them who does not have any idea about the development process. Every piece of design is tested with all the browsers to make sure that it works fine with all the modern browsers available.

The portal is tested with the available mobile devices (Samsung galaxy S4, Iphone 6 and Ipad) in working environment. Due to the lack of other actual mobile devices and devices with a medium-sized screen, the portal was tested using Google chrome's virtual devices (developer tool) which provides a fairly accurate result like real devices. The portal was also tested using different online tools available. Some tests made and their results are described below:

4.5.1 Google’s Mobile Friendly Test

Google has added a label "Mobile friendly" in the search result. According to the Google webmaster central blog, a page is eligible for the “mobile-friendly” label if it meets the following criteria as detected by Googlebot:

- The site is not using the software like Flash which is not common on mobile devices.
- The site uses the text that is readable without zooming the website.
- The content of the site fits the screen so that users do not need to scroll or zoom to see the content.
- The links that are used on the site are correctly placed with enough spaces so that they can be easily tapped. [12]



Mobile-Friendly Test

https://www.bookndo.com/

Awesome! This page is mobile-friendly.

How Googlebot sees this page

G+1
ANALYZE
Bookndo
Find and book sports activities easily
GROW YOUR SALES WITH BOOKNDO!
Feedback
WHAT IS YOUR EXPERIENCE? SHARE IT ON SOCIAL MEDIA WITH #BOOKNDO

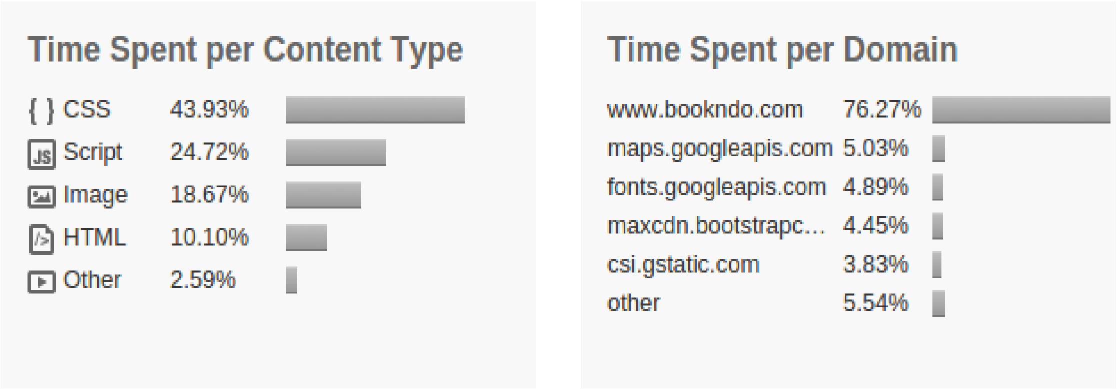
Figure 11. Google's mobile friendly test

The portal passed the Google's mobile friendly test which is shown in figure 11.

4.5.2 Pingdom Website Speed Test

The Pingdom website speed test is an online tool to test the website's speed and performance. It is a very famous and widely used tool by many providers. It examines all parts of a web page like file size, load time and details about every single element of a web page like HTML, JavaScript CSS and images. Tests are carried out using a dedicated server from a different location with a real web browser. [13]

In the Pingdom website Speed Test, the portal got the grade 82 out of 100. The page size is 618.7 kb half of which is consumed by images and load time 648 ms when tested from Amsterdam, Netherlands. The result of the page analysis by content type and domain are shown in figure 12.



Time Spent per Content Type

{ } CSS	43.93%
Script	24.72%
Image	18.67%
HTML	10.10%
Other	2.59%

Time Spent per Domain

www.bookndo.com	76.27%
maps.googleapis.com	5.03%
fonts.googleapis.com	4.89%
maxcdn.bootstrapc...	4.45%
csi.gstatic.com	3.83%
other	5.54%

Figure 12. Pingdom website Speed Test

4.5.3 GTmetrix performance Test

GTmetrix is a free online tool that grades the website’s speed, analyzs 30 different ways to improve the performance of the website and provides a report. This is also a very popular online tool for the speed and performance testing of a website [14]. In the GTmetrix performance test, the portal got grade 80 out of 100 when checked from the server region Vancouver, Canada using Firefox 41.0. It shows the page load time 3.5s and the page size 603 kb, which is similar to the Pingdom website speed test.

From the results of different performance tests, it can be summarized that the goal of the project was almost achieved. The main goal of the project was to create a mobile-friendly website. It is close to perfection as it passed the Google's mobile-friendly test.

The table 3 compares the average page weight of the year 2015 with the page weight of Bookndo.

Table 3 Average page weight. Reprinted from Httparchive report (2015) [15]

	Average page weight 2015 (kB)	Bookndo front page (kB)
HTML	66	6.3
CSS	76	60.6
JavaScript	363	186.1
Images	1443	260.8
Flash	53	0
Other	223	41.8
Total	2262	555.6

Table 3 clearly shows that Bookndo is under the average in file size compared to the websites deployed in 2015. The elements of the sites are very lightweight and they do not include any components such as flash which are not mobile-friendly.

5 Discussion

During the project several benefits as well as challenges were realized they are briefly described below.

5.1 Benefits

As discussed in chapter 5, most web designers and developers started to think about the mobile version before desktop version. Therefore, the mobile first design approach is convenient and desired. As it gives more priority to the content, it is straightforward for users to use. As a single domain work for all, users can easily remember the domain of website.

For the startups as which do not have sufficient resources for developing mobile sites, desktop sites and native applications separately, the mobile-first design approach is like a boon.

In this project Bootstrap as a front-end framework made the work easier and the workflow faster as it has many ready-to-use items and classes. As it is the most used and preferred framework, updates are frequent. Therefore also for future needs like maintenance of the website, it is easier to find the manpower working with Bootstrap.

Using Git for version control and Heroku tools made the workflow smooth and it was easier and less confusing to work in a group.

5.2 Challenges

It was challenging to design something for smartphones keeping smartphones in mind that there are various types of smartphones in the market with different sizes and capacities and the trend is increasing rapidly. There were limitations in the possibilities that worked for mobile devices. Therefore, mobile devices were always kept in core importance of the brainstorming as not all the designs worked for mobile devices. Another challenge during the project was cross browser compatibility as there were many browsers available and the portal was expected to work in all of them. Testing was another challenge in development process as the portal was required to be tested with different devices and not all the real devices that were considered during the development process were available. Using virtual devices did not always give accurate result.

6 Conclusion

The main goal of the project was to discuss about mobile first design approach for which a project was built as an online booking system (portal) which is responsive, secure, dynamic and fast-loading. The most popular and recent technologies HTML5, CSS3, jQuery and bootstrap (as a front-end framework) are used to design the front end, PHP is used as a server-side scripting language and PostgreSQL is used as database system. It is built in CakePHP framework which follows MVC pattern and written in PHP .The prototype of the application is built in Photoshop. The cloud application platform Heroku is used to build and deploy the app and git is used for version control.

The portal is supported on most of the modern browsers and devices. Images and media are scalable, responsive and easily loadable. Margins and paddings are well set to give the best result possible in cross-platforms to provide better user experience. The portal contains two different layouts for service providers and users. Most of the pages are designed in the mobile-first approach except some calendars which are too wide for mobile devices. Moreover the portal can be used by service providers to add their services and schedule, terms, conditions and photographs of the service they provide. They can also edit and update them when necessary. Users can use the portal to book the activities, trainings and courses get information about that and pay. Finally the portal is purely dynamic so it is easy to update and maintain in the future. Most of the classes used to design are reusable. Most of the html elements are semantic.

As a front-end designer writing the scripts from scratch for the design was easier for me however changing the layouts and design from the dynamic pages when the concept was changed in the middle of the project was a little bit challenging because knowledge of server-side scripting and how they are used was mandatory in that phase. Prior experience of HTML, CSS and JavaScript, support of the development team members and flexibility of the deadline made the task easier. Also different courses related to software development taken in Metropolia helped a lot during the project. In conclusion, the project was satisfactory and it met most of the goals.

References

1. The global mobile report by comScore [Online]. Gartner.
URL: http://www.statista.com/statistics/266219/global-smartphone-sales-since-1st-quarter-2009-by-operating-system/
Accessed 15 February 2015

2. Google developers [online]
URL: https://developers.google.com/webmasters/mobile-sites/
Accessed 9 February, 2016

3. Ethan Marcotte. Responsive Web Design [online]. A List Apart; 25 May 2010
URL: http://alistapart.com/article/responsive-web-design , Accessed 15 February2015

4. Ethan Marcotte. Responsive Web Design. New York, U.S.A , A Book Apart; 2011

5. Ethan Marcotte. Fluid images [online]. A List Apart; 7 June 2011
URL: http://alistapart.com/article/fluid-images , Accessed 20 February, 2015

6. Media Queries W3CRecommendation [online]
URL: https://www.w3.org/TR/css3-mediaqueries/
Accessed 25 February 2015

7. Jayson Demers. Forbes.com [online]. 7 November 2013.
URL: http://www.forbes.com/sites/jaysondemers/2013/11/07/10-steps-to-creating-a-mobile-optimized-content-marketing-strategy/#3f8fc4636e5e

8. Aaron Gustafson. Adaptive Web Design. Chattanooga,Tennessee; 2011

9. Official Github repository of Bootstrap[online]
URL: https://github.com/twbs/bootstrap
Accessed 21 February 2015

10. Official web page of Bootstrap[online]
http://getbootstrap.com/
Accessed 21 February 2015

11. Bookndo terms and conditions [online]
URL: http://www.bookndo.com/terms_and_conditions/privacy_and_cookies
Accessed 21 February 2015

12. Google webmaster central blog [online]. 18 November 2014.
URL: https://googlewebmastercentral.blogspot.fi/2014/11/helping-users-find-mobile-friendly-pages.html
Accessed 20 April 2015

13. Pingdom official website [online]
URL: https://www.pingdom.com/
Accessed 25 April 2015

14. GTmetrix official website [online]
URL: https://www.gtmetrix.com/
Accessed 25 April 2015

15. Http archive [online]
URL: http://httparchive.org/trends.php
Accessed 15 May 2015

16. Steph Hay. Content-First Design [online]. A List Apart; 29 April 2015
URL: http://alistapart.com/blog/post/content-first-design
Accessed 9 February, 2016

17. W3C official website [online]
URL: https://www.w3.org/
Accessed 21 February 2015