

Bello Idowu

# Front End Programming in Designing Responsive Web Application for Imagine Cup



Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Information Engineering

Thesis

29 May 2015

Author(s)	Bello Idowu
Title	Front end programming in designing responsive web application for imagine cup competition
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	. NET, JavaScript and PHP
Instructor(s)	Kari Aaltonen, Principal Lecturer
<p>The purpose of this thesis work was to study and explore the creation process of building responsive web application to the end users. The specific goal was to develop a responsive Web application where professional can use their professional skills to those who are in need of it.</p> <p>The application was built, designed and styled on the Windows 8 platform and using new programming skills technology for making the web application easy to view on various devices. The core of the application was built in C# using Visual Studio as IDE and styled with CSS, HTML and JavaScript. The application is very complex with simple functionalities and UI. It is responsive and clean, which work accurately on any desktop browser or mobile browsers.</p> <p>The idea of the project was to raise professionals all across the world to get involved in helping people who are in need, who do not have sufficient funds to sponsor themselves in order to achieve their dreams and goals. It is hoped that this project will go live after the final stages are completed and the world would be a better place. This project will also provide great opportunities for people in need to find help in everyday tasks and specific professional advice and assistance.</p> <p>The project was built in a way that the functionalities and code behind is written so that the application can be expanded and new ideas can be implemented and added to current application build.</p>	
Keywords	ASP.NET, CSS3, HTML5, JavaScript, C#, web application

## Contents

1	Introduction	1
2	Usability and Responsiveness	2
2.1	Usability Overview	3
2.2	Responsive Web Usability	5
2.3	Design Principle of Usability	6
2.4	Components of Usability	9
3	Design Procedures for Responsive Application.	10
3.1	Twitter Bootstrap 3	22
3.2	Bootstrap CSS	27
3.3	Bootstrap JavaScript	29
4	Tools Used for the Development	30
4.1	Integrated Development Environment	33
4.2	ASP.NET Web Application Environments	36
4.3	Window Azure	42
5	Kind Hearted Web Application	43
5.1	Responsive Web Application	46
5.2	Problems	47
5.3	Solutions	47
6	Results	48
7	Conclusion	50
	References	

## Terms and Abbreviations

App	Application
col-xs	Column used for Extra-Small Device like Mobiles
col-sm	Column used for Small Device like Tablets
col-md	Column Used for Medium Device like Laptops
col-lg	Column used for Large Device like Desktops
col-md-offset	Column used for Pushing Column Medium right in the Layout
CPU	Central Processing Unit
CSS	Cascading Style Sheet
GUI	Graphical User Interface
HTML	Hypertext Markup language
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
IT	Information Technology
Imagine Cup	Microsoft Annual Best Application Competition for Technical Students
JS	JavaScript
LOB	Line of Business
PaaS	Platform as a Service
RWD	Responsive Web Design
REST	Representational State Transfer
SaaS	Software as Service
SQL	Structured Query Language
UCD	User Centered Design
UI	User Interface

VHD Virtual Hard Disk

VM Virtual Machine

WCF Window Communication Foundation

## 1 Introduction

The goal of the thesis was to develop the response web application for the Imagine Cup Competition in the world citizenship category.

The purpose of the application was to develop a responsive Web application that would enable the professionals anywhere in the world willing to use their professional skills for individuals or groups of individuals who are genuinely in need of professional skills but could not afford them. The application should also make it possible for a group of professionals to carry out voluntary work of interest. Underprivileged people who are genuinely in need of help every part of the world; for instance an orphaned child, a low income earner in a city, or a peasant farmer in a village. This application provides a simple platform to contribute to needy people in the communities. Any of these individuals with life-threatening diseases could be linked with medical professionals who are willing to offer free services to the individuals.

Every day the world is full of news about, loss of life; students deprived of education; people dying of hunger; health problem issue; children suffering; and depressing situations. These numbers are increasing each day and getting worse. Even as a privileged person, it is difficult get access to help in these kinds of situation.

The application aims to provide the platform for a categories mentioned above to be achieved. There are two categories of users in the application: professionals and beneficiaries. The professional represents users who would be willing to use their professional skills or tools to assist the other category of users, beneficiaries, who are truly in need of a particular service, but are incapable of affording that service. In order to achieve this both users need to register. Professionals, or Helper, register by filling the Helper Form on the website to render assistance, and beneficiaries or Needy register as well by filling in the Needy Form on the site before they can be given assistance. A Professional has an opportunity to browse through different categories of Needy profiles then read their profile through and if they show interest in meeting any of the beneficiaries then they would notify the website administrators which are comprised of the program developers. The team member would connect the professional and the beneficiary they have chosen to render assistance after making necessary investigations. The beneficiary does not have direct access to the professionals. The beneficiary cannot browse through the list of professionals or contact them. My role, as a web developer, is to develop a user-friendly, front-end application that both skilled and unskilled users can use with little or no supervision.

## **2. Usability and Responsiveness**

### **2.1 Usability Overview**

Nowadays, people, or users, are impatient to get information on web. What they do most of the time is quickly scan through website in order to get what they want. In the process of building a website, it is important to take user experience into consideration. This may enhance frequent visits to the website by the users. It is also important to know what is meant by usability. Usability is the ability to make content or products and systems easier to use, and match them more closely with users needs and requirements. It also explains the extent to which a product can be used for specific users to reach also specific goals with effectiveness, efficiency and satisfaction. Taking usability into consideration has to do with user effectiveness on the web. It explains how users complete tasks and achieve goals while working on the web interface. Also, usability is efficiency: how much effort users need in order to achieve web transaction, often measured according to time. Usability evaluation focuses on how users can learn and use a product to achieve their goal. It also refers to how satisfied users are with the process. Usability is the quality of the users' experience with products or systems, along with websites, software, devices, or applications. [13: 30]

### **Usability on Page Design**

Page design is the most visible section of the website design. Users glancing at each page at time, perhaps two or three pages if they have large screen with various windows open, using the current browser technology.

Clarity should be the goal of the page design. Users are rarely on the site for the sake of design enjoyment; rather, they prefer to focus on the content. It is imperative to make sure that page design functions across a wide range of platforms and are accessible by people who still make use of old technology; for example, two-year old browsers and two year old versions of all plug-ins and other software. It is also good to ensure that page design works on small devices and has an acceptable response any time analog modems is being used. [12: 27]

### **Usability on Content Design**

In general, users make a stopover on a website for its content; everything else is scenery. The design helps people to get access to the Web content. Web users' attention is focused towards content. It is the reason they visit online and it is first glanced at immediately after the new page load. Web usability is determined by the quality of the content which users pay close attention to; the other being if the user can find the page they want to access.

Quality content on the web can be equivalent to something different than in traditional media. Quality of production is less essential; beautiful pictures and well crafted writing are absolutely welcome, but they are no longer the describing distinctive features of quality. This is because web users are so goal-driven and very impatient, content needs to be more adapted towards contributing to rapid solutions and being useful to the user. [12:146]

## **Site Design**

Sometimes page design is the most acclaimed aspect of a website. Ultimately, along current browsers, you see only one page at a time. The site itself is never defined explicitly on the screen, but site design is more demanding and often more crucial than page design, from a usability perspective. A website will not be considered perfect unless the designers and writers concur to work towards the needs of the customer. Site designers still need to focus their information structure on the needs of customers instead of the company's own internal thinking. [12: 214]

## 2.2 Responsive Web Usability

Responsive web design (RWD) is an approach in web development that creates dynamic changes to the website appearance, depending on the screen size and orientation of the device being used to view it. RWD is one of approach to the problem of designing for the multitude of devices available to the customers, ranging from tiny mobile phones to huge desktop monitors. Google also gives recommendations to responsive web design as an industrial best practice. RWD means making the content enjoyable to interact with on the screen. Responsive web design uses so-called breakpoints to determine how the layout of the site will appear. Breakpoints are commonly based on the width of the browser.

HTML is served to all devices, using Cascading Style sheet, CSS, which determines the Web page layout, to change the appearance of the page. Instead of creating a separate site and relating codebase for wide-screen monitors, desktops, laptops, tablets and various sizes of phones, a single codebase can support users with differently sized viewports. [3: 14]



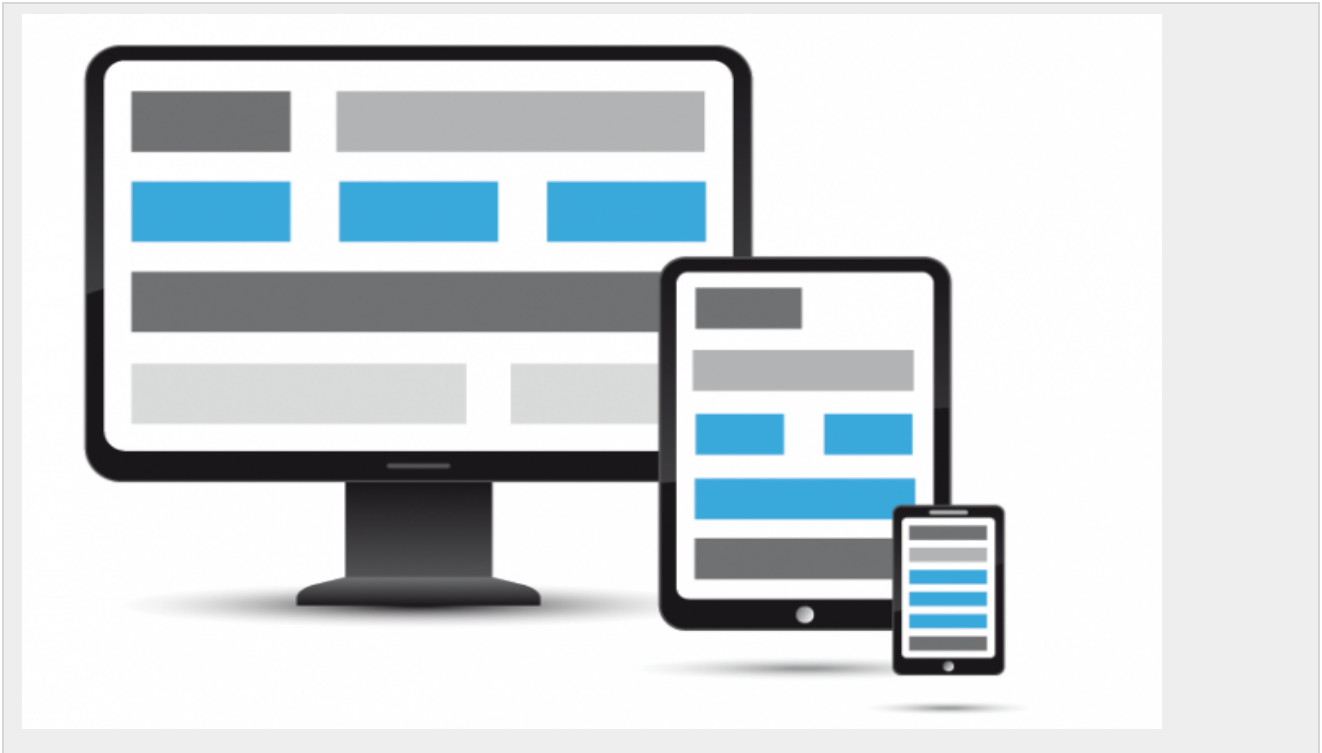


Figure 1. Responsive design interface reposition appearance on three devices [19: 1]

In responsive web design, page elements reposition as the viewport expands and shrinks. A three-column desktop design may reposition to two columns for a tablet and to a single column for a smartphone as it is shown in Figure 1 above. RWD depends on the proportional based grids to rearrange content and design elements. Responsive web design was launched in order to give equal access to information available to the user, no matter the device with which they are viewing it. It also makes it possible to hide certain items such as a navigation bar using colliding method. The RWD also has strong advantages compared to developing a separate site for different device types. The Development processes of a single codebase can be rapid compare to developing 3 or 4 specific sites. In terms of maintainability, a single codebase would be easier to update with its contents over time contrast with updating a 3 or 4 distinct codes for specific sites.

It is relatively futureproof that RWD is compliant, in the sense that it can support new breaking points needed at any time. Regardless of the size of the devices, RWD is designed so that it can support new devices. Moreover since the responsive design relies on shuffling elements around the page, design and development are required to work closely together to secure a usable experience across devices. The users' experiences can be completely different from one view of the site to the next as the elements move around the page. Therefore, it is essential that the developers work in accord with each other not only to focus on how the contents look when browsing the site but also on imagining what the end result would look like and how it would affect the users' experiences.

Responsive Web design is easy to implement using responsive-design frameworks such as Foundations and Bootstraps. It is good to keep in mind how the framework will work with the content and functionality of the site, rather than how it works generally. It is uncertain to design a website that is usable on a desktop, and even more difficult to develop web on various devices such as laptops, tablets or smartphone that may not function on desktop at the end. Therefore, it is advisable to test the web application across platforms to be certain of its functionality. [1: 3]

## **2.3 Design Principles of Usability**

### User Centered Design Process

It is quite obvious that systems are designed with the targets of a business goal, eye-catching features and the technological ability of hardware and software tools. All of targets toward system design have left out the most crucial part of the process called 'the end user'. The User-centered design, UCD, process is an approach that supports all stages of the Web development process with user-centered activities, in order to build applications that are easy to use and add value to the intended users. Rather than demanding the adaptation of users' attitudes and behaviours to learn and use a system, a system could be designed so that it supports users' intended attitudes, existing beliefs and behaviours as they interact with the tasks that the system is being designed to support. According to an industry survey, the majority of failed projects can be attributed to incomplete, inaccurate requirement. The biggest cost benefit that UCD can provide is to define requirements more accurately. [2: 1]

Change in the design process at a late hour can cause problem even cost ten times more than if identified during the requirements. Therefore user-center design should be integrated with other development activities, before embarking on web development. They should be planned and managed by a development team. Over time, UCD events would become mutual practice. In the process of working towards user-centered designs there are four important UCD principles that should be kept in mind: a clear understanding of user and task requirements; integrating user feedback to improve requirements and design; active involvement of the user to evaluate designs; and incorporating user centered design with other development activities. Calculating the previously mentioned principles would have maximum impact on the four project phases within the development process: [4: 15]

## **Planning Process**

For a developer to design an application that meets the user's specifications, the process must be thoroughly planned. However, simply ensuring that the product meets the design specification, succumbs to the users and is of the required quality is not enough. The design of the application must be cost-effective, which means, if maximizes the added value and meet the agreed deadline. Therefore, through process planning, the designer is responsible for ensuring that the product is designed to the correct specification, at the lowest possible cost and completed on time. The Planning process is the sound approach towards planning. It explains how much one can do or extent to which one can go in application development. It ensures that the proposal can be supported by the planning policy and meet the submission requirements set out by the local and national government.

Process planning comprises the selection and sequencing of processes and operations to transform chosen raw materials into a finished component. Also it is an act of preparing detailed work instructions to produce a component, which includes selection of manufacturing processes and operations, production equipment and tooling. A web planner needs to anticipate the skills and resources needed for developing, designing, deploying and operating the web. A Good websites do not always happen accidentally. As a web developer, spending time to think about a web page before it is implemented, why one want to build and who is your target, obtaining the required resources and skills needed for web development, formulating policies about information development, deployment and presentation drives the developer and users forward. [4]

## **Requirements Analysis**

Sometimes, considering developing a web application can be exciting; perhaps wanting to start laying out a home page, choosing font and colors. At this point, it is good to reconsider and go through a thorough process of requirements analysis. Requirements analysis is the process of determining user expectations for a new or modified product. Understanding the requirements in advance helps to ensure the application development goal and user needs, that the solution achieves what it is was designed to carry out. In a website redesign project, the requirements should prevent any frustrations that users may experience with the current site. Understanding the content on the web interface and, using plain language allows users to find what they need, understand what they have found, and then use it to meet their needs. It should also be actionable, findable, and shareable. [4: 17]

## 2.4 The Components of Usability

### **Learnability**

Learnability is the cost that it takes for the user to reach the stage of performance with the task, ignoring the peculiar struggles associated in finishing the task for the first time. If the approach of achieving the first task proved easily unforgettable after the first completion then the product would be learnable for the task. Learnability could be an essential part in a situation where the user is self-taught with the product. The ability of users to recognize and operate web interface functionalities without having to retain detailed information in their minds makes web interfaces more learnable. The effectiveness of the website could be linked to learnability.

A website is ultimately designed with learnability and navigation in mind to make sure users able to determine right away; what the company does, what information is displayed on the website and how to contact the company. Most users like to visit a website and immediately be able to determine how to do what they intend to. A website design that pays attention to learnability would be more successful than one that does not. Finally, Learnability is the effectiveness, efficiency and fulfillment with which stated users accomplish a capable stage of performance on specified tasks with a website, having finished those tasks already. [5] [13:12]

### **Guessability**

Guessability is the measure of the cost to the user of a product for the first time to the execution of a new task. The lower the cost, for instance in terms of the time on the task or errors made, the higher the guessability. This is likely to be the case for a website that has high proportion of one-out-users. Lack of guessability can cause serious commercial implications in the sense that it may take away users' interest in revisiting the site more often. Guessability is the effectiveness, efficiency and fulfillment with which stated users could complete specific tasks with a particular product for the first time. [13:11]

### **Measures of Usability**

Measuring usability could be advantageous in web development and for users. It keeps the developer on track toward designing website. Web usability could be measured in three separate aspects: Effectiveness, efficiency and satisfaction. [13:18]

## Effectiveness

Effectiveness is the extent to which a goal or task, for example Web development, could be achieved. The determinant of whether or not a product or design is effective for a specific task is whether or not the user could finish that task along with the product. For example, if the developer were to create a website and the task is for the people to buy and sell from the site, then the website could be regarded to be effective for the task if the user could buy and sell.

Furthermore, the quality of output also determines if the product is effective or not. It is possible that a user is able to finish a task with a product but the outcome resulting from the completion of a task is fluctuating in quality. For example, a user searches for a particular product on a website but the image of the product is not displaying accurately. Therefore, the image display could be used as a measure of the effectiveness. [13: 19]

## Efficiency

Efficiency is the amount of effort needed to complete a goal. It could also be seen in terms of *deviations from the critical path*. For most tasks given, there is a critical path for the task performance. On the other hand the least effort is needed for the method of approaching the task. If a user departs from the critical path, then this is negative in terms of efficiency. A User deviating from the critical path would waste time and effort. In addition, *Error rate* is one of the most frequent measures of efficiency. If a user can accomplish the task without committing an error along the way, then the task may require less effort than if the error were committed and need to be corrected.

An error could be categorized into various types according to why they occur and their severity. A basic difference between a 'slip' and mistake is that a *slip* occurs when a user knows a way towards achieving a particular task but unintentionally does something wrong in the process. An example would be making a typing error when writing an email. A *mistake* takes place when a user has an erroneous model of how the product functions. An example of this could be the user of a computer-based drawing package. For instance various developing tools for drawing on the Web are available and a user must first choose from the icons represented in order to create a drawing. If a user were to enter the package and later attempt and create a drawing commonly by moving the cursor around the screen, it would indicate that the user had an incorrect model of how the system product functioned. It would then, be a '*mistake*'.

Whether the error is a *slip* or a *mistake*, there could be repercussions for analyzing usability problems and specifying the solutions applicable to the error. Often, a slips error takes place because of an inappropriate designed of the product interface, or it may be due to a particular button being inadequately positioned, or controls placed too close to each other. On the other hand *mistakes* are more likely to happen because the basic rules of how a product functions are not understood. [5] [13:19; 21]

### **Satisfaction**

Satisfaction is the comfort level that the user experiences when utilizing the web or product and how users accept the product as a means of reaching their goal. Probably, 'Qualitative attitude analysis' could be one of the ways of knowing if users are satisfied with using the website or not. This is done by asking users to feel in a questionnaire, or by interviewing them, or perhaps users' comments about the website can be noted. These comments can be diagnosed in order to gain indication as to the level of satisfaction users experience utilizing the Web.

Furthermore, quantitative analysis could usually be another measurement of knowing if users are satisfied with a product. For example, if a researcher preferred to compare two different products in terms of users' behaviour towards them or wished to see whether the product met some 'standard' level of satisfaction, then this could help to measure the usefulness of satisfaction. [5] [13: 23]

## **3 Design Procedures for Responsive Application.**

### **3.1 Twitter Bootstrap 3**

Bootstrap is one of the most popular front-end frameworks to develop web applications and sites fast and is currently used by developers and designers. It is a sleek, intuitive and powerful mobile first front-end for faster and easier web development. In modern web development there are several components that are needed in almost all projects. Basic modules such as Grid, Typography, Tables Buttons, Forms and Responsiveness are provided in Bootstrap apart from the fact that there are several other useful front–end components like Navigation, Dropdowns, Modal type-head, Pagination Carousel, Breadcrumb, Tab, Thumbnails, and Headers. These are adequate enough to start Web project quickly and easily.

Moreover, the downloaded Bootstrap uses HTML, CSS and JavaScript. The downloaded Bootstrap file needs to be unzipped or extracted then folder called “Bootstrap” appears which contains a sub-folder below.

Bootstraps/

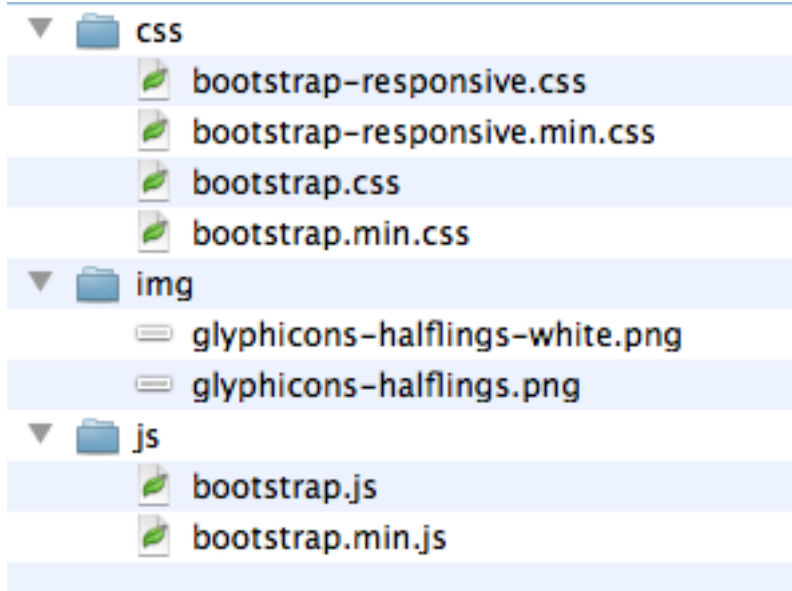


Figure 2. Bootstrap File Structure

The bootstrap-3.0.0 download includes three folders: CSS, Js, and img. Once the folder is unzipped, then added to the root the project. A minified version of the CSS and JavaScript are also included in the folder. Both uncompressed and minified version is not compulsory to be added but should be for the sake of removing all unnecessary characters from the source code without changing its functionality. The main CSS files – bootstrap.css and the minified version of it, bootstrap-min.css, are present within the 'css' folder, which is placed within the 'dist' folder under bootstrap-3.0.0.

The 'js' folder is also within 'dist' folder, which consists of the JavaScript file, bootstrap.js, and a minified version of it. There is another 'js' folder located within the 'asset' folder within the root folder. This holds html5shiv.js which HTML5 shim, used for IE8 support and respond.min.js file, used for supporting media queries in IE8. This folder also contains jquery.js on which Bootstrap's js plug-in relies on. For the sake of mobile devices, there is an 'ico' folder within the same containing icon for favicon and icons. I do use of the uncompressed version, which is a developed version because of the readability with an abundance of whitespace and a number of comments to help understand the code. [6: 1; 2] [7:3]

## Basic HTML Template

Normally, a basic HTML template is like the structure below:

```
<!DOCTYPE html>
<html>
  <head>
    <title> Bootstrap Caring Heart Template </title>
  </head>
  <body>
    <h1>Hello, Kind Hearted!</h1>
  </body>
</html>
```

Listing 1. Basic HTML Page

With the Bootstrap Web project, the link to the CSS and the JavaScript was included:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Bootstrap Caring Heart Template</title>
    <link href="css/bootstrap.min.css" rel="stylesheet">
  </head>
  <body>
    <h1>Hello, Kind Hearted!</h1>
    <script src="js/bootstrap.min.js"></script>
  </body>
</html>
```

Listing 2. Basic HTML with Bootstrap CSS and JavaScript.

## Global styles

It is imperative to know that with Bootstrap, a number of items come prebuilt. Instead of making use of the old reset block, which was part of Bootstrap 1.0 tree, Bootstrap 3.0 uses Normalize.css, which is included in the bootstrap.css file. Bootstrap is using the @baseFontFamily, @baseFontSize, and @baseLineHeight attributes for typographic base. This allows the height of the heading and other content around the site to maintain a similar line height. The global link colour of the bootstrap is set through @linkColour and applies link underline only on: hover. [6; 3]



## Default Grid System

Bootstrap uses a 12- columns grid system as the default, its span value runs from 1 to 12 columns as the device or viewport size increases. A 12- columns grid is used for a 940 wide container without responsive features enabled. Adding the responsive CSS file, the grid adapts to be 724px or 1170px wide depending on the viewport. Below the 768px viewports, such as extra small devices, for example mobile phones, the columns become fluid and stack vertically at all times. Above the 768px viewport, the grid system is suitable for small devices such as a tablet. Moreover, medium devices such as desktops could have their viewport greater than or equal to 992px, as well as for large desktops devices, which the viewport could be greater or equal to 1200px.

In order for devices to respond to the size of the pixels stated above, there would be a need to use a single set of `.col-md-*` grid classes which would be used to target different devices such as mobile phone and tablet devices (from the extra small to small range) before becoming horizontal on the desktop, medium, devices. In Bootstrap, the grid column is placed within the row class (`.row`). The classes have already been predefined for easy layout options. The Table 1 below demonstrates a predefined grid layout system.

Table 1. Summary of how Bootstrap Grid System Works across the difference Devices.

	<b>Extra small devices Phones (<math>&lt;768\text{px}</math>)</b>	<b>Small devices Tablets (<math>\geq 768\text{px}</math>)</b>	<b>Medium devices Desktop (<math>\geq 992\text{px}</math>)</b>	<b>Large devices Desktops (<math>\geq 992\text{px}</math>)</b>
Grid behavior	Horizontal at all times	Collapsed to start, horizontal above breakpoints	Collapsed to start, horizontal above breakpoints	Collapsed to start, horizontal above breakpoints
Max container width	None (auto)	750px	970px	1170px
Class prefix	<b>.col-xs-</b>	<b>.col-sm-</b>	<b>.col-md-</b>	<b>.col-lg-</b>
# of columns	12	12	12	12
Max column width	Auto	60px	78px	95px
Gutter width	30px (15px on each side of a column)	30px (15px on each side of a column)	30px (15px on each side of a column)	30px (15px on each side of a column)
Nestable	Yes	Yes	Yes	Yes
Offsets	Yes	Yes	Yes	Yes
Column ordering	Yes	Yes	Yes	Yes

Table 1 illustrates screen behaviour on the different devices at particular pixels, regard less of the minimum width on all devices: @screen-sm-min if the width of the screen gets smaller than @screen-sm-max, then the screen shrinks. [6: 3] [7: 18]

### Bootstrap Basics Grid HTML Structure

The basic grid in Bootstrap is used to create a simple layout in Web application. In order to create a simple layout, the container with a <div> that has a class of .row with the appropriate amount of .col-xs-\*, .col-sm-\*, .col-md-\* and .col-lg-\* columns equal to 12 has to be created. The layout structure could be [col-lg-12], [col-sm-4 col-md-4 col-md-4], [col-md-2 col-md-8 col-md-2], [col-md-4-col-md-8], among other combinations, depending on the individuals desire for the layout.

```
<div class="container">
  <div class="row">
    <div class="col-*-*"></div>
    <div class="col-*-*"></div>
  </div>
  <div class="row">...</div>
</div>

<div class="container">....
```

Listing 3. Bootstrap Grid System

In Bootstrap, the page layout is wrapped inside the container class in order to make web pages responsive to the target devices like mobile phones, tablets, laptops or desktops. Therefore, the following rules should be utilized for the bootstrap grid system to function appropriately:

- Row must be placed within a **.container** class for proper alignment and padding.
- Create horizontal groups of columns using rows
- Content should be placed within columns, and only columns may be immediate children of rows.
- Predefined grid classes like **.row** and **.col-xs-4** are available for quickly making grid layout.
- Columns create gaps between columns content through padding. The padding is offset in rows for the first and last column through the negative margin on **.rows**.
- Specifying the number of twelve available columns desired to span, created by grid columns. For example, three columns would use three **.col-xs-4**. [6: 4] [7: 18]

## Bootstrap Grid Behavior on the Layout

Bootstrap grid is design in the HTML body and the grid is put inside the fluid container, which in turn remove the border and spreads everything out to a full screen, as it is demonstrated below.

```

<body>
<div class="container">
  <div class="row">
    <div class="col-xs-12 col-sm-6 col-md-3 col-lg-1 "></div> <!-- let
use colour red. -->
    <div class="col-xs-12 col-sm-6 col-md-3 col-lg-1 "></div> <!-- let
use colour blue -->
    <div class="col-xs-12 col-sm-6 col-md-3 col-lg-1 "></div> <!--
let use colour green. -->
    <div class="col-xs-12 col-sm-6 col-md-3 col-lg-1"></div> <!-- let
use colour yellow. -->
    <div class="col-xs-12 col-sm-6 col-md-3 col-lg-1"></div> <!-- let
use colour brown. -->
    <div class="col-xs-12 col-sm-6 col-md-3 col-lg-1 "></div> <!--
let use colour purple. -->
    <div class="col-xs-12 col-sm-6 col-md-3 col-lg-1"></div> <!-- let
use colour red. -->
    <div class="col-xs-12 col-sm-6 col-md-3 col-lg-1"></div> <!-- let
use colour blue -->
    <div class="col-xs-12 col-sm-6 col-md-3 col-lg-1"></div> <!-- let
use colour green. -->
    <div class="col-xs-12 col-sm-6 col-md-3 col-lg-1"></div> <!-- let
use colour yellow. -->
    <div class="col-xs-12 col-sm-6 col-md-3 col-lg-1 "></div> <!--
let use colour brown. -->
    <div class="col-xs-12 col-sm-6 col-md-3 col-lg-1"></div> <!-- let
use colour purple. -->

  </div>

</div>
</body>

```

Listing 4. Grid System in HTML.

In Listing 4 above, the `.row` class has a 6-row grid system. On the extra small screen the grid would use the entire row i.e. each one of (`col-xs-12`) is going to use a full screen of the full width screen. On the small screen (`col-sm-6`), there would be 2 blocks. There would be 4 per row on the medium screen i.e. there would be 4 of (`col-md-3`) making 12 grids and finally on the large screen (`col-xs-12`) would take 1 span on the grid i.e. 1 row of 12 entries (`col-xs-12`).

### Layout of the above Grid System

The above bootstrap grid system will then look like the layout below in practice, when it is opened on the screen.

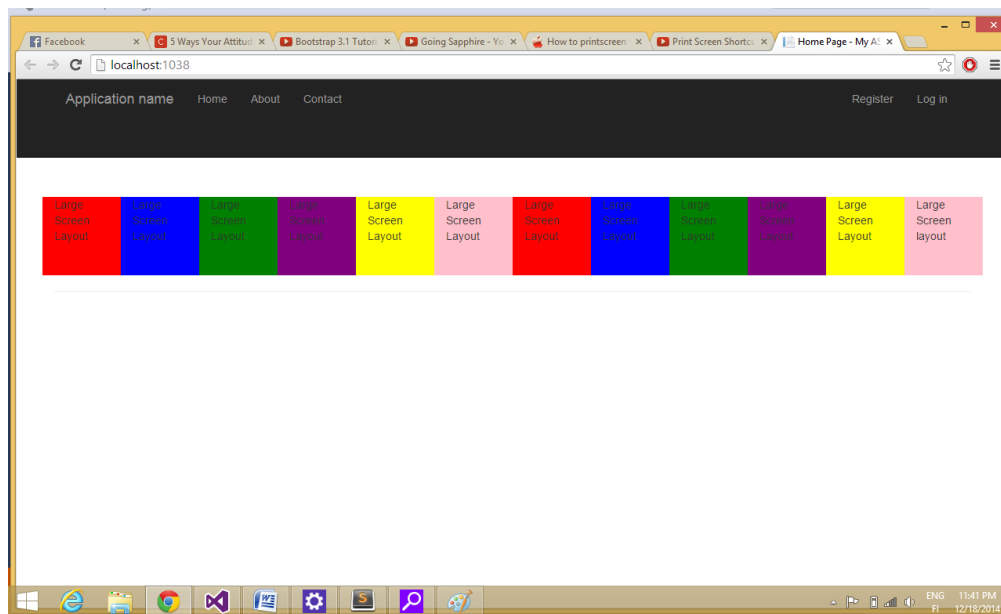


Figure 3. Large Screen Layout

When the large grid system is viewed on a large screen, or it is expanded manually on the smaller screen to the level that is equivalent to the desktop screen. There would then be one row layout that would take up to 12 entries.

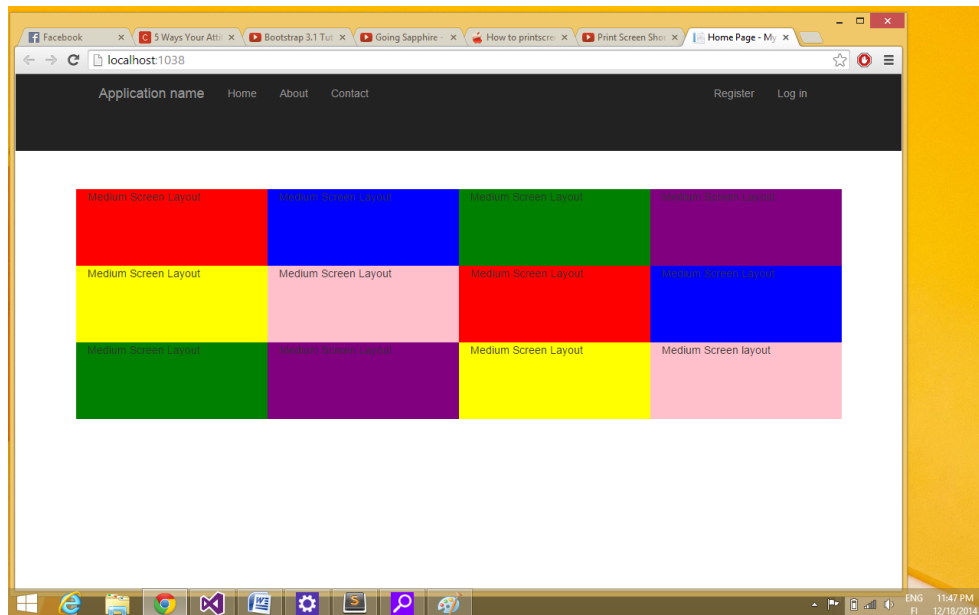


Figure 4. Medium Screen Layout.

When the grid system view on a laptop device or shrink to laptop resolution, then there would be three layouts that take up to four spaces on the grid system and this is equal to the full row i.e. there would be 4 grids per each row.

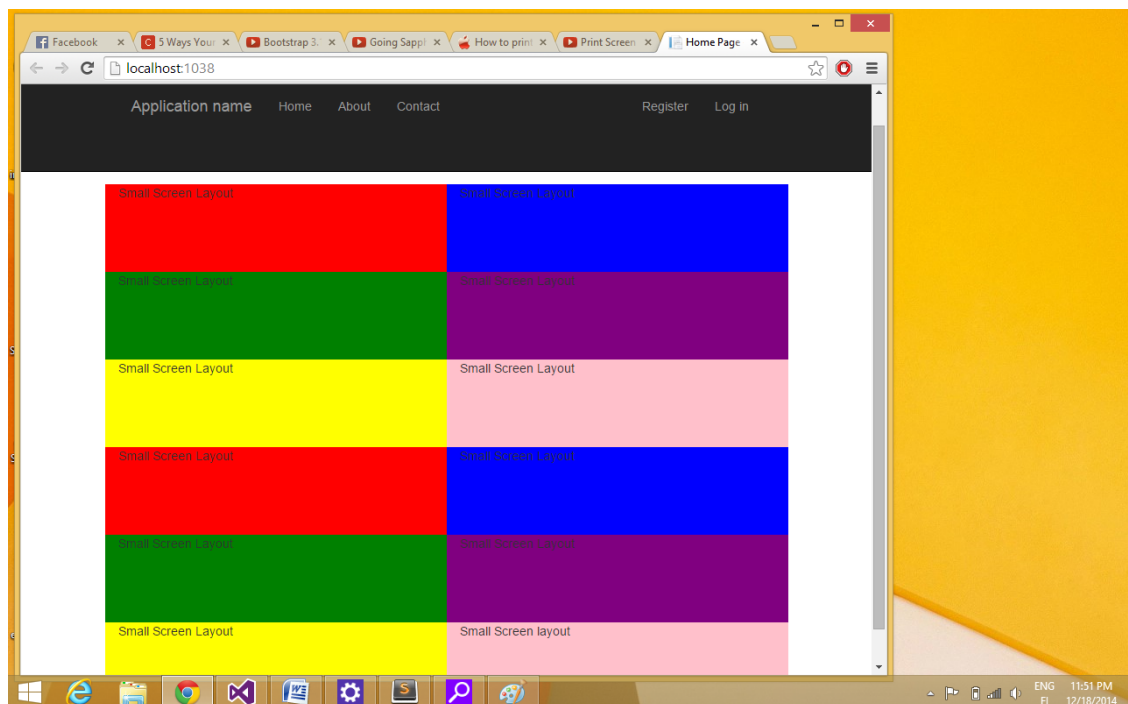


Figure 5. Small Screen Layout

When the grid system layout is viewed on a small screen, or shrink to the tablet screen resolution it will eventually reach a threshold for a small screen. Therefore, the two layouts would take up 6 grids spaces on the grid system and that is equivalent to full row. For that reason there are two per row.

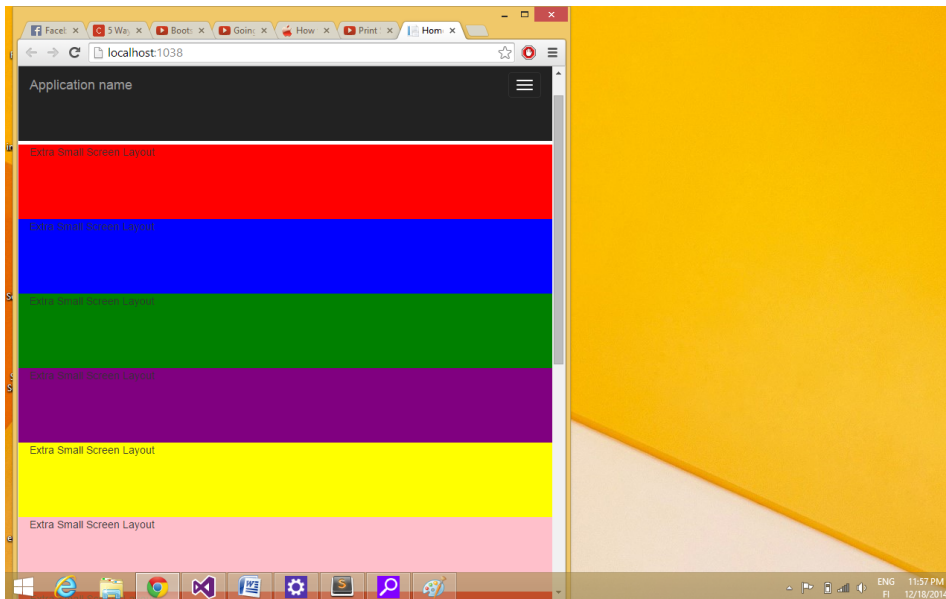


Figure 6. Extra Small Screen Layout

Viewing the grid layout on the mobile phone screen or resizing it to the extra small screen would finally reach a threshold where 12 layouts would take up to 12 grid spaces which is equivalent to full row. For that reason there is one per row.

### Offsetting Column

The offsetting column is used to move columns to the right using the `col-md-offset-*` classes. Each class moves the column over the width. Offset features are valuable for more specialized layouts. Columns are push over to allow for more spacing, for example, The `.col-xs=*` classes do not support offsets, but they are easily replicated with the use of an empty cell. To use offsets on a large screen resolution, use the `.col-md-offset-*` classes. These classes increase the left margin of a column by \* columns where \* range is from 1 to 11. Therefore offsetting classes below could be explained by offsetting column below. [7; 22]

```
<div class="row">
  <div class=" col-lg-offset-8 col-lg-1 green">
<p>col-lg-offset-8 col-lg-1 green</p>
  </div>
  <div class="col-md-4 col-md-offset-8 blue">
<p>col-md-4 col-md-offset-8 </p>
  </div>
  <div class="col-sm-6 col-sm-offset-6 yellow">
<p>col-sm-6 col-sm-offset-6 </p>
  </div>
  <div class="col-xs-offset-11 col-xs-1 red">
<p>col-xs-offset-11 col-xs-1 </p>
  </div>
```

Listing 5. Offset Grid System

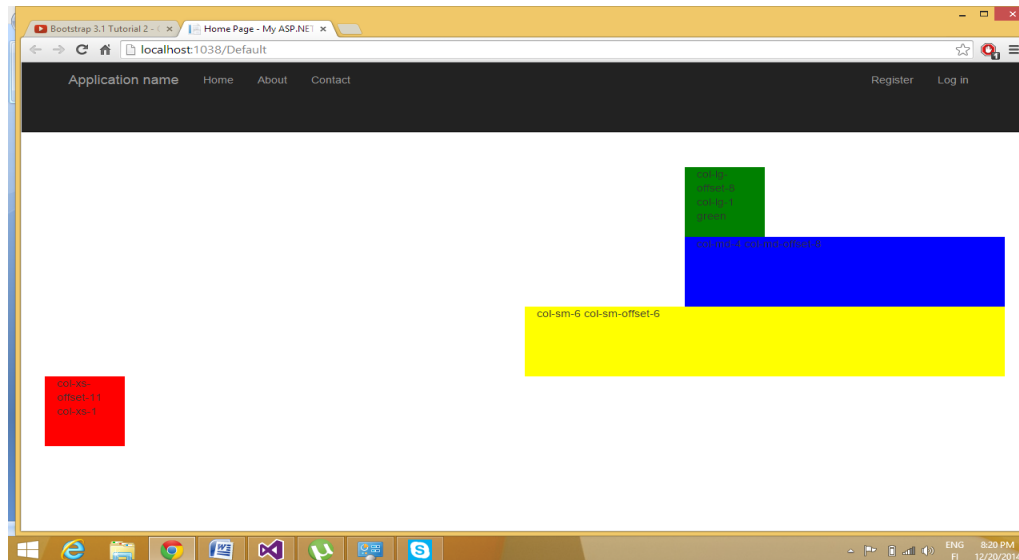


Figure 6. Offset Layout Behaviour

The offset works in such a way that it pushes the column to the right depending on the number of offset columns. For example, if there is offset-8 for the large screen layout shown in Figure 6, the grid column system (col-lg-1) is moved to the grid 8 using col-lg-offset-8. It is illustrated by colour the green.

At the medium resolution, the column (col-md-4) is pushed to the grid column 8 at the same level where large screen moved towards using offset of col-md-offset-8. Also, for the small screen, it is offset to the one-half of the grid system on the small screen resolution by using an offset of col-sm-offset-6; using the offset on the small screen does not have a distinct effect on the its grid system as seen in the Figure 6 above.

### Nested Column

A nested column occurs when a column is inserted within the existing column. In order to nest the content with the default grid system, One need to add a new **.row** and a set of columns e.g. col-md-\* within an existing .col-md-\*. Nested rows should also include a set of columns that add up to 12.



```

<div class="container">
  <h1>Nested Column Layout!</h1>
  <div class="row">
    <div class="col-md-3" style="background-color: #dedef8;box-
shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
      <h4>First Column</h4>
      <p>I am the first column layout with the FIRST (.row)
MOTHER, I occupied my distinct place.</p>
    </div>
    <div class="col-md-9" style="background-color: #dedef8;box-
shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
      <h4>Second Column- Split into 4 boxes</h4>
      <div class="row">
        <div class="col-md-6" style="background-color: #B18904;
box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
          <p>I am the first existing column with SECOND (.row)
MOTHER and first box.</p>
        </div>
        <div class="col-md-6" style="background-color: #B18904;
box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
          <p> I am the second existing column with SECOND
(.row) MOTHER and I occupied the second boxes in the split column. </p>
        </div>
      </div>
      <div class="row">
        <div class="col-md-6" style="background-color: #B18904;
box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
          <p> I am the third existing column with THIRD (.row)
MOTHER and I occupied the third boxes in the split boxes. </p>
        </div>
        <div class="col-md-6" style="background-color: #B18904;
box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
          <p>I am the fourth existing column with THIRD (.row)
MOTHER and I occupied the fourth boxes in the split boxes. </p>
        </div>
      </div>
    </div>
  </div>
</div>

```

Listing 6. Bootstrap nested grid system

The nested content has two columns, first and second column, the second being split into four boxes over two rows. The bootstrap nested column structure above is demonstrated in the Figure 7 below. [7: 21]

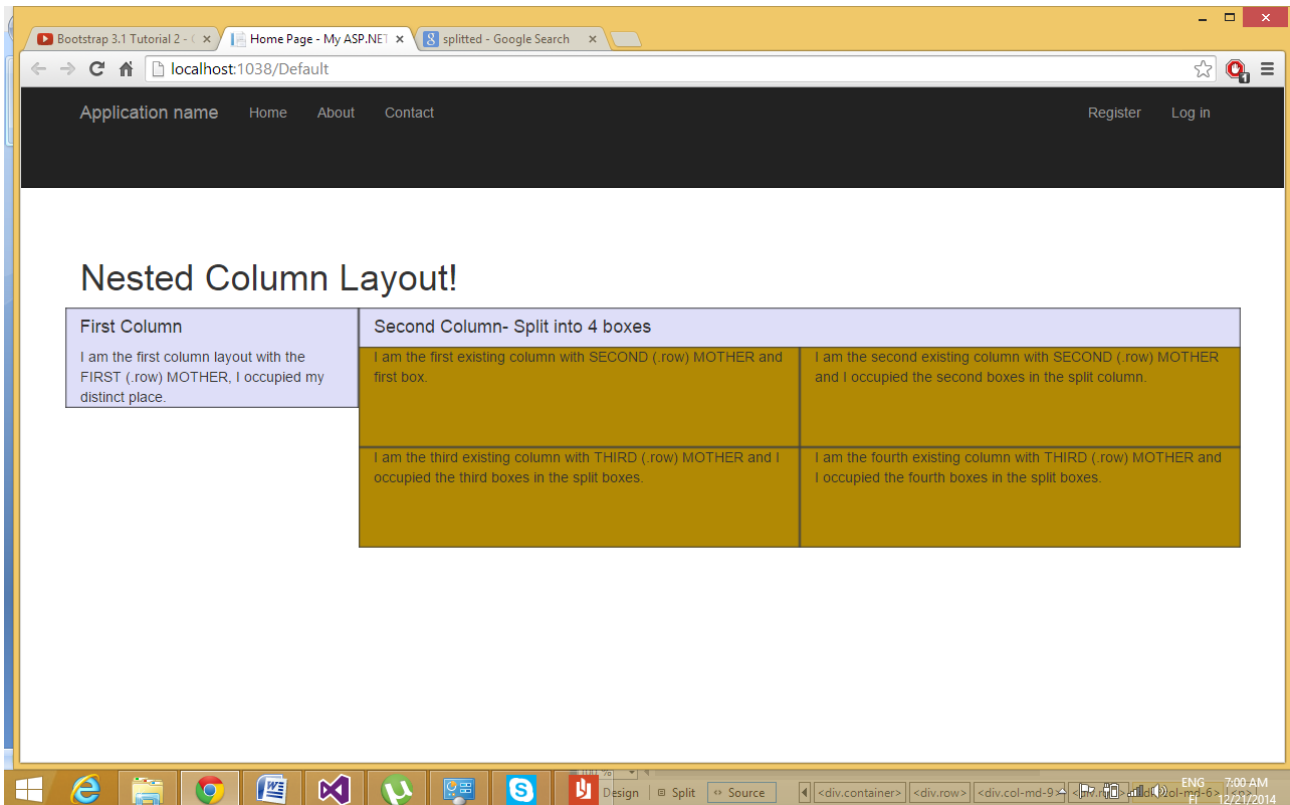


Figure 7. Bootstrap Nested Column Layout.

Figure 7 is the medium screen layout, which is the outcome of the grid system in the nested column in Listing 6. The first **.row** of the medium layout was split into a two-grid system. The first column grid is **col-md-3** and the second grid, **col-md-9**, which was split into four boxes to nest two **.rows** columns. The first **.row** splits into a two-grid system (**col-md-6 col-md-6**), which adds up to a 12 grids system; each column occupies half of the grid system. The second nested **.row** column occupied the second row and was also split into half of the grid system. [7: 24]

## Column Ordering

Column ordering is another feature of the Bootstrap grid system. It works such that one could write a column in an orderly fashion and present the same order in another way.

```

<div class="container">
  <h1>COLUMN ORDERING!</h1>
  <div class="row">
    <p>Before Ordering</p>
    <div class="col-md-4" style="background-color: #dedef8;
      box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
      I am on left
    </div>
    <div class="col-md-8" style="background-color: #dedef8; box-shadow:
      inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
      I am on right
    </div>
  </div>
  <br>
  <div class="row">
    <p>After Ordering</p>
    <div class="col-md-4 col-md-push-8" style="background-color: #dedef8;
      box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px
#444;">
      I was on left
    </div>
    <div class="col-md-8 col-md-pull-4" style="background-color: #dedef8;
      box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px
#444;">
      I was on right
    </div>
  </div>
  </div>
  </div>
  </div>

```

Listing 7. Column Ordering Grid System Structure

The order could be easily changed by a built-in grid column with **col-md-push-\*** and the **col-md-pull-\*** modifier classes where \* range from 1 - 11. This is illustrated using the layout below:

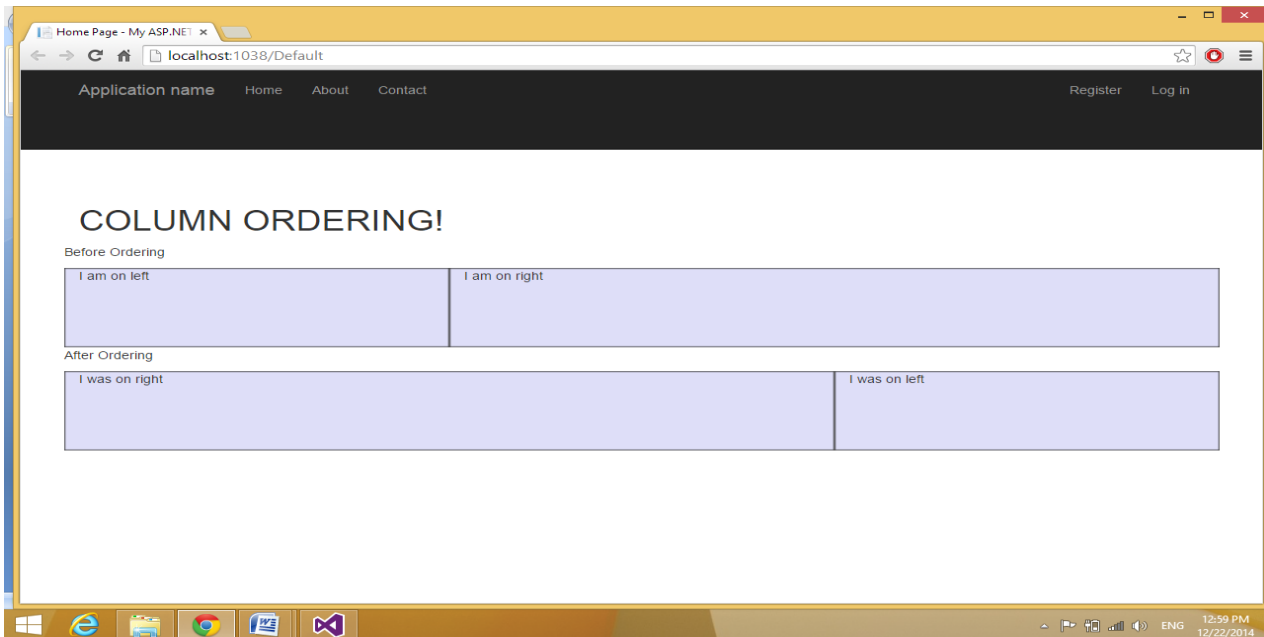


Figure 8. Column rendering layout

The layout shows the result of the grid system shown in Figure 8. In this layout there are two columns represented, the left column being the smallest and behaving as a sidebar. The column grid switches the columns layout using `.col-md-push` and `col-md-pull` classes.

### 3.2 Bootstrap CSS

CSS is the one of the most powerful tools of the Web technology that adds the look and feel to the representation of an HTML page display. The CSS settings in Bootstrap 3 theme are used for setting up HTML elements styles such as layout, colour, and user interface. The CSS bootstrap setting is also applied in custom styles by adding and overwriting extensible classes. [7: 26]

#### **Bootstrap CSS Overview**

There are key components of Bootstrap infrastructure, which make the Bootstrap approach a better, faster, and stronger Web development. Therefore, Bootstrap infrastructure will be pounded upon.

#### **HTML5 Doctype**

Because of the use of the HTML elements and CSS properties, Bootstraps requires the use of the HTML Doctype. This is included at the beginning of the master page. Using visual studio as an IDE, the other pages could inherit the HTML page from the master page.

```

<!DOCTYPE html>
<html>
.....
</html>

```

Listing 8. HTML Doctype.

## Mobile First

Bootstrap has generally become mobile first which means mobile first styles could be found throughout the library instead of separate files. In order for this to take place a **viewport meta tag** needs to be added to the **<head>** element, for proper rendering and touch zooming of mobile devices.

```

<meta name="viewport" content="width=device-width, initial-
scale=1.0">

```

Figure 9. Mobile First Style on Bootstrap 3

- Controlling of the width of the device is dependent on *width* property. Setting it to *device-width* ensures that the width is rendered across various devices (mobile, desktop, tablet, etc).
- *initial-scale=1.0* tells the browser that the Web page should be rendered at a 1:1 scale, no zooming will be applied beyond the specified range.

Mobile device users are given permission to be able to scroll and not being able to zoom using **user-scalable=no**. This is put into action by adding it to the **content** attribute to disable zooming capacity on the mobile devices as shown in Figure 10.

```

<meta name="viewport" content="width=device width, initial scale=1.0,
maximum-scale=1.0, user-scalable=no">

```

Figure 10. Disabling Zooming Capabilities on Mobile Devices.

*maximum-scale=1.0* is commonly used together with *user-scalable=no*. As it earlier in Figure 10, *user-scalable=no* may provide a user with an experience more like a native application; whereas, it is not suggested using Bootstrap. [7: 26]

## Responsive images

Images in Bootstrap 3 tend to be responsive by adding a class **.img-responsive** to the `<img>` tag. This class uses **max-width: 100%**; and **height: auto**; to the image so that it is well scaled to the parent element as mentioned in Figure 11.

```

```

Figure 11. Making Images Responsive on Bootstrap 3.

## Topography

Bootstrap topography uses Helvetica Neue, Arial, Helvetica and Sans-serif in its default font stack. These are the standard and default fonts that are added to all major computers. If for any reasons these fonts do not exist, they would retreat to Helvetica, and then inform the browser to use the default font meant for the browser. All body copy has the font-size set at 14 pixels coupled with the line height set at 20 pixels. The paragraph `<p>` tag possesses a margin-bottom of 10 pixels, or half the line-height. Bootstrap topography features can be used to create headings, paragraphs, lists and other inline elements. [7: 25]

## Headings

In Bootstrap, all the six standard heading levels, h1-h6, been styled with `<h1>` at 36 pixels tall, and the `<h6>` at 12 pixels as mentioned in Figure 12.

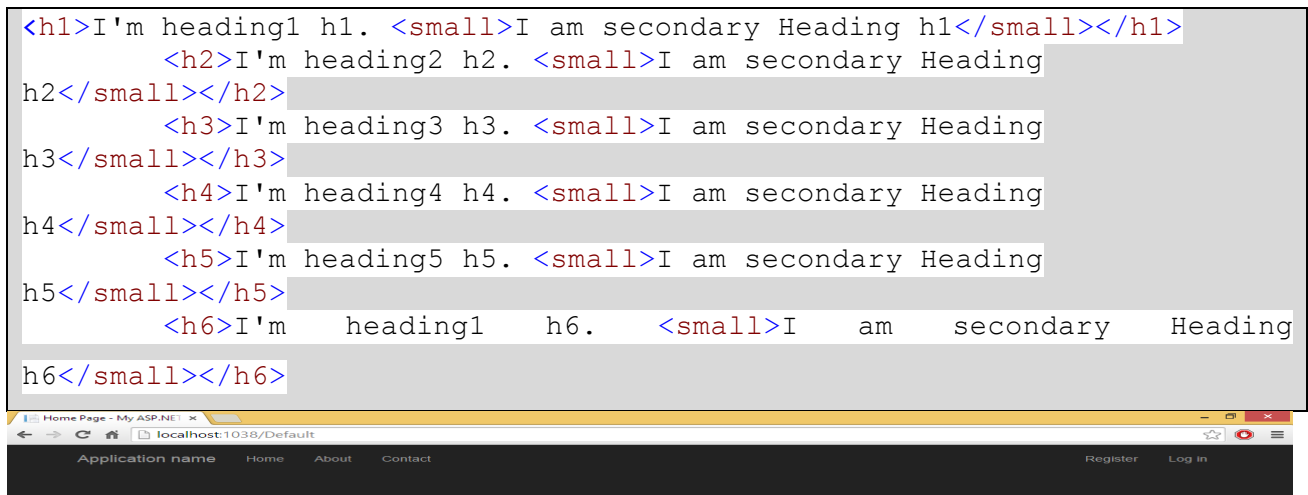


Figure 12. Headings and Inline Subheading in Bootstrap

Furthermore, an inline subheading could be added to any of the headings by adding `<small>` around any of the elements. This would result in smaller text in lighter colour as it is show in Figure 12.

## Lead Body Copy

A lead body is implemented when placing emphasis on a particular paragraph, which also makes the text font size larger, with lighter weight and a taller line height. In order for the lead body to be effective, a `class="lead"` needs to be included in the element, for example `<div>`, `<p>`. `Class="lead"` can be used anywhere in the paragraph but is primarily used for the first few paragraphs in a section.

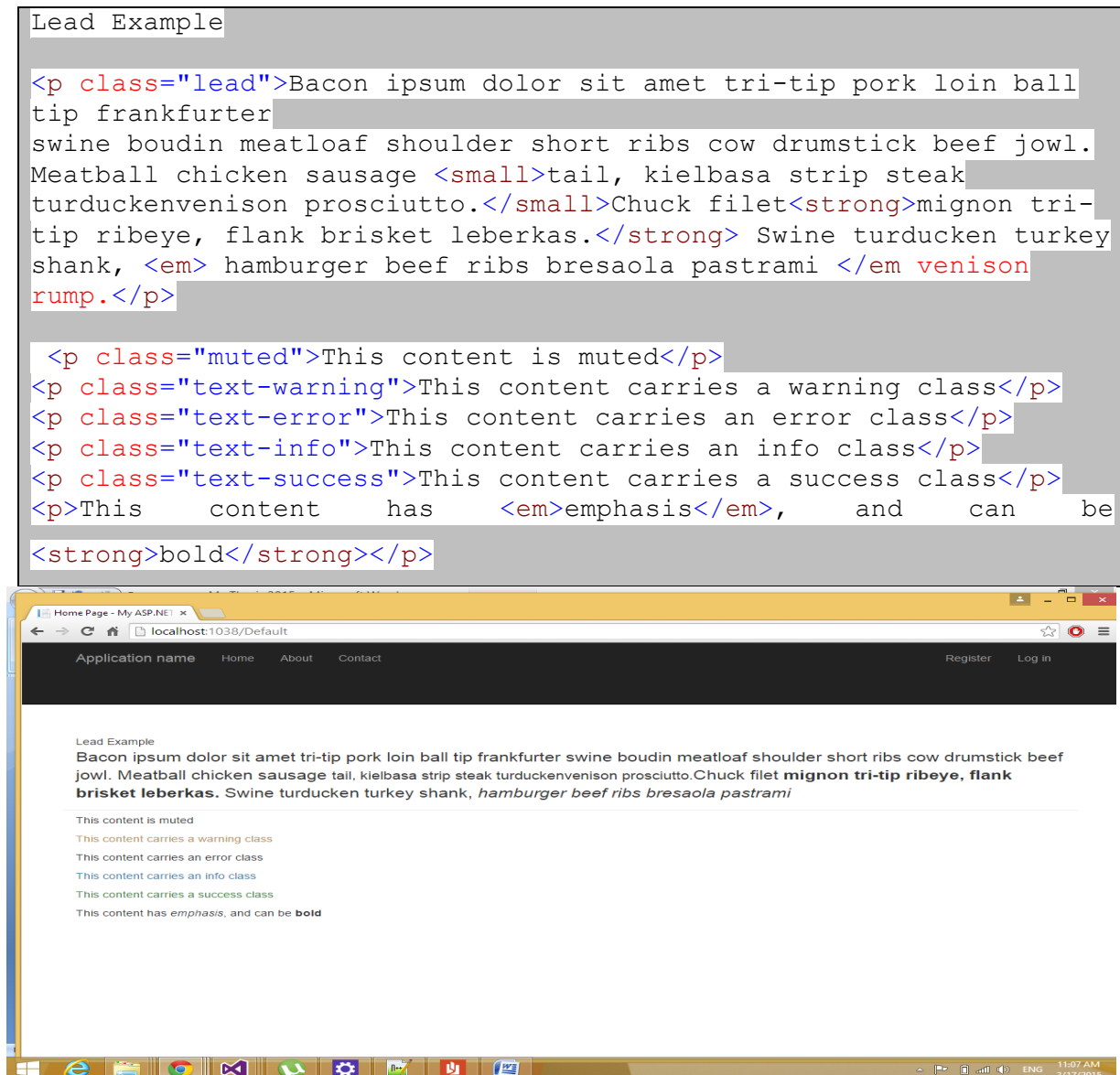


Figure 13. Lead, Bold and Italics CSS bootstrap class tags Behaviour

Along with lead, there are some of the classes that go along with the lead body for example by using a **small tag, bold and italics tags** within a heading by using the class of each tag to display the effect in the text. The examples are shown in Figure 13. [7: 10; 11]

### 3.3 Bootstrap JavaScript

Components of JavaScript come along with Bootstrap in the package of JQuery plugins. Bootstrap comes along with 13 JQuery plugins that widen the features and enrich the interaction to the website. Being an advanced JavaScript developer is not a criterion for understanding Bootstrap



JavaScript plugins. Bootstrap becomes effective in the code when the file is downloaded and installed into the application. The file is then referenced on the head of the HTML and a JavaScript code can be written at the bottom of the HTML body, inside opening and closing of the script tag **<script> </script>**.

It is important to include script at the bottom of the closing body tag **</body>** to avoid inadequacy when loading the HTML page. Placing JavaScript in the head tag may allow some elements to be missed at the end, when the page loading is completed. [7: 73]

## **Bootstrap JavaScript Plugins Functionality**

### **Modal**

A Modal is a child window that is spread on the surface of its parent window. It is used to show content from the separate source and is capable of establishing communication but, does not deviate from the parent window. A modal could function individually on its own as a plugin, by including **modal.js** or else it functions in Bootstrap plugin using the minified bootstrap file **bootstrap.min.js**. Information and communication is provided through a child window and it is primarily used to hold slideshows and login information during registration. A modal plugin is perhaps one of the special preferences of Bootstrap features.

The modal plugin's hidden content can be toggled through **data attributes**: Set attribute **data-toggle="modal"** on a controller element, such as a button or link, alongside, a **data-target="#identifier" or href="#identifier"** to aim at specific modal, using the **id="identifier"**, to toggle. Also, through JavaScript, the modal plugin can toggle by calling the modal with **id="identifier"** with one line of JavaScript code: **\$('#identifier').modal (options)**. This is demonstrated in Figure 14.

```

<h2>Example of creating Modals with Twitter Bootstrap 3</h2>
  <!-- Button trigger modal -->
  <button class="btn btn-primary btn-lg" data-toggle="modal" data-
target="#myModal">
  Launch demo modal </button>
  <!-- Modal -->
  <div class="modal fade" id="myModal" tabindex="-1" role="dialog"
aria-
  labelledby="myModalLabel" aria-hidden="true">
  <div class="modal-dialog"> <div class="modal-content">
  <div class="modal-header"> <button type="button" class="close"
data-dismiss="modal"
  aria-hidden="true"> &times;
  </button> <h4 class="modal-title" id="myModalLabel"><strong
style="color:black;">
  This Modal title CHAPTER </strong></h4> </div>
  <div class="modal-body"><h4 style="color:black;"> Add some text
here </h4></div>
  <div class="modal-footer"> <button type="button" class="btn btn-
default" data-
  dismiss="modal">Close </button>
  <button type="button" class="btn btn-primary"> Submit changes
</button></div>
  </div><!-- /.modal-content -->
  </div><!-- /.modal -->
</div>

```

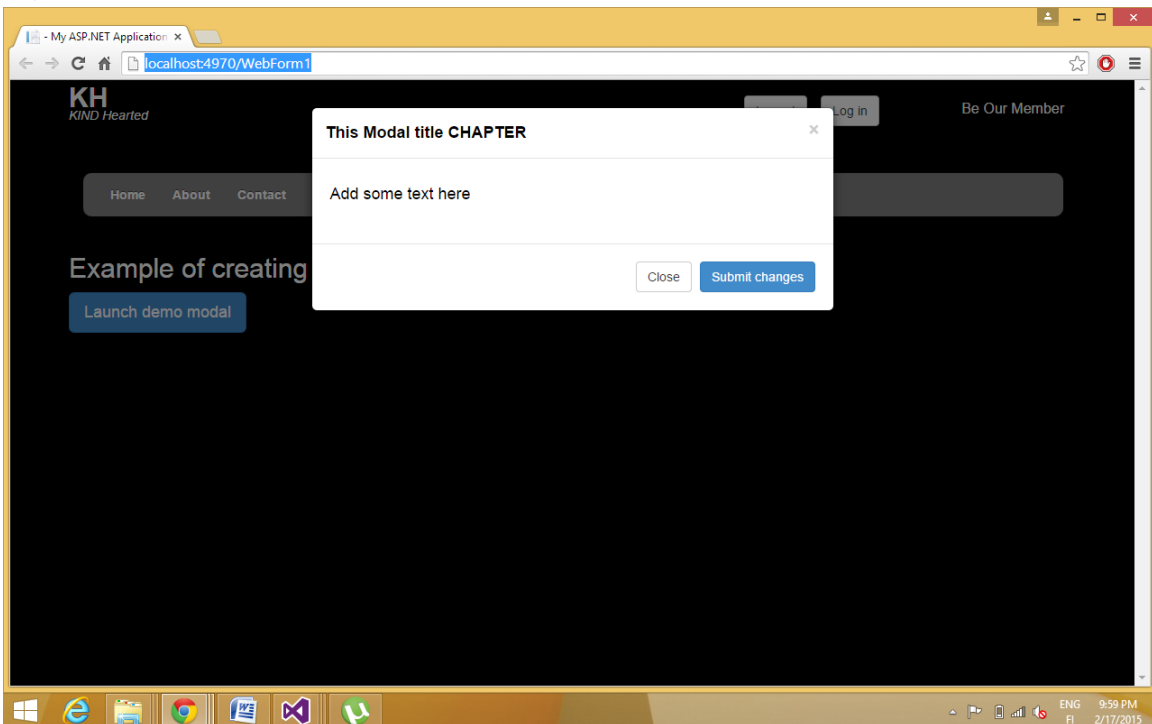


Figure 14. Modal Window Demo

The modal window is triggered by using a button tag `<button>`; the **data target="#myModal"** is the target of the modal that was loaded on the page. There are two classes to be noted when using a node. The first is **.modal** which identifies `<div>` content as a modal and the second **.fade** class that

caused content to fade in and out when a modal is toggled. There are other attributes such as **aria-labelledby="myModalLabel"** which reference the modal title. The attribute **arial-hidden="true"** keeps a modal window visible until the associated button is trigger.

- **<div class="modal-header">** which is used to define the header of the modal window style.
- **class="close"** sets a style to close button of the modal window.
- **data-dismiss="modal"** is a data attribute for customized HTML5 and is used for the closing of the modal window.
- **modal-body** class is used to style the modal window body and a **modal-footer** class is for the footer modal body style
- **data-toggle="modal"** is the customized HTML5 data attribute responsible for opening of the modal window. [7: 75]

## Tooltips

Tooltips is used when describing a link. It is updated to function well without the use of images, as well as animate using CSS animation and data-attributes for local title storage. This plugin can function individually by adding **tooltip.js**. Otherwise, one can include *bootstrap.js* or the minified *bootstrap.min.js*. Content and markup are generated on demand using the tooltip plugin. Tooltips are placed after trigger element by default. Likewise, it is passed to the project in two ways:

Firstly, Through **data attributes** called **data-toggle="tooltip"** to an anchor tag mentioned in Figure 15. The anchor title is near to the text of a tooltip. The tooltip is positioned to top through the plugin by default.

```
<a href="#" data-toggle="tooltip" title="Example tooltip">Hover over me</a>
```

Figure 15. Tooltip Data Toggle

Secondly, Through JavaScript in Figure 16: Tooltips is triggered via JavaScript.

```
$("#identifier").tooltip(options)
```

Figure 16. Tooltip triggering

A tooltips plugin example is demonstrated in Figure 17 below.

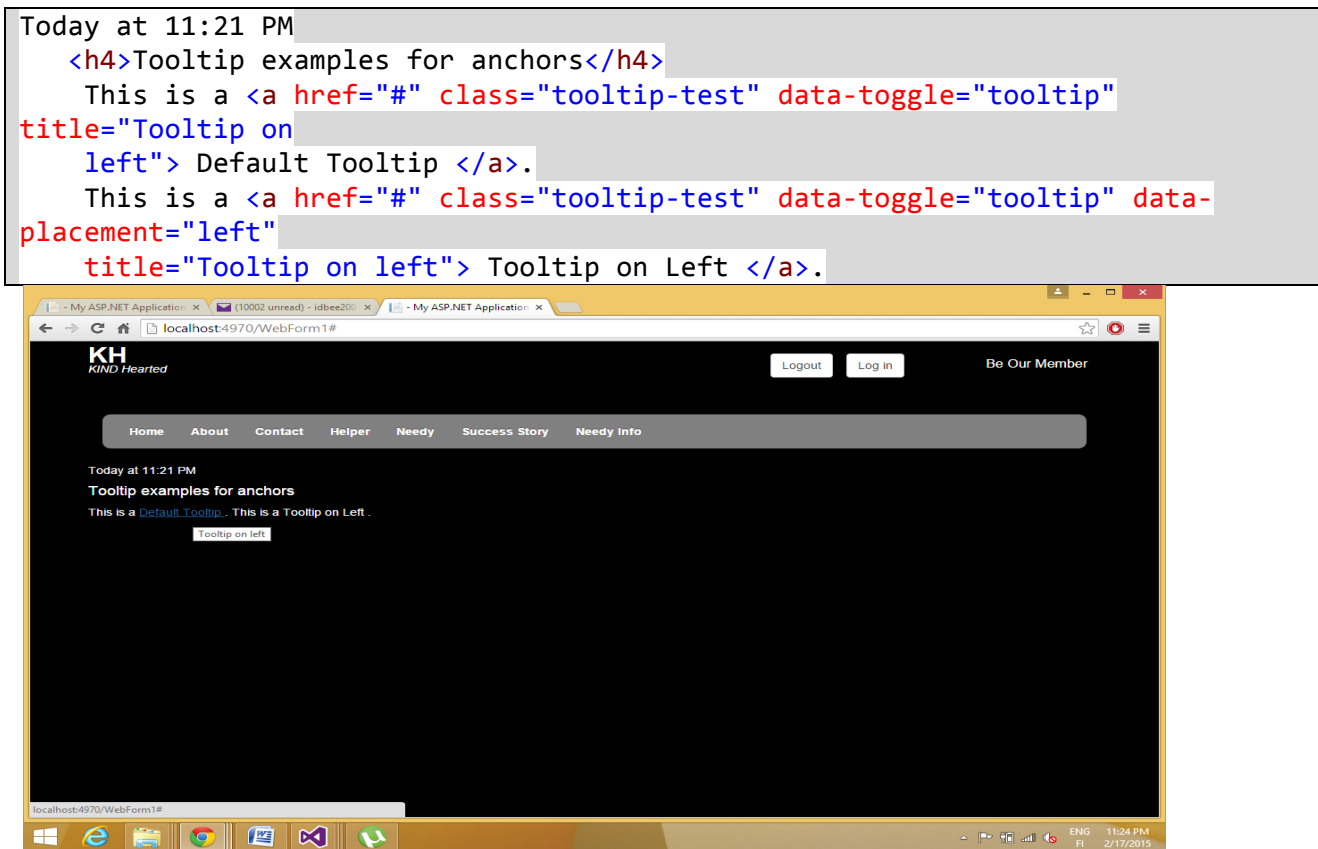


Figure 17. Tooltips Plugin Demo

The above screen application was demonstrated using the HTML in Figure 17 and the tooltips JavaScript code mentioned above. The default tooltips display the text within the anchor tag; `<a>Tooltip on left </a>`. [7: 82]

## 4 Tools Used for the Development

### 4.1 Integrated Development Environment

#### Visual Studio 2013

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop a **computer programs**, as well as a web sites, web services and web applications. Visual Studio also makes use of the Microsoft software development platform such as Window Form, Window Presentation Foundation, Window Store and Microsoft Silverlight. It can produce both native and managed codes.

Visual Studio includes code editor supporting intellisense intelligent code completion, likewise, code refactoring. The combined debugger function as both a source-level and a machine level debugger. Forms Designer is another built in tools that is included for building a graphical user

interface (GUI) application, Web designing, class designing, and database schema designing. It takes plug-ins that enhance the functionality along with adding support for **source-control** systems, such as Subversion, and adds new toolsets such as editors and visual designers for **domain-specific languages** or toolsets for other aspects of the **software development lifecycle**, for example the **Team Foundation Server** client: Team Explorer.

Microsoft Visual Studio is also supported using different programming languages or built-in languages such as C, C++,/CLI, through Visual C++), VB.NET, through Visual Basic.NET, C#, through Visual C#, etcetera. It allows the code editor and debugger to support any programming languages, provided there is an existence of the particular languages. [8] [9: 11]

### Visual Studio 2013 Overview

After Visual Studio has been installed, it is able to run for the first time. When this process is complete, Visual Studio 2013 opens in order to start working as show in Figure 18 below.

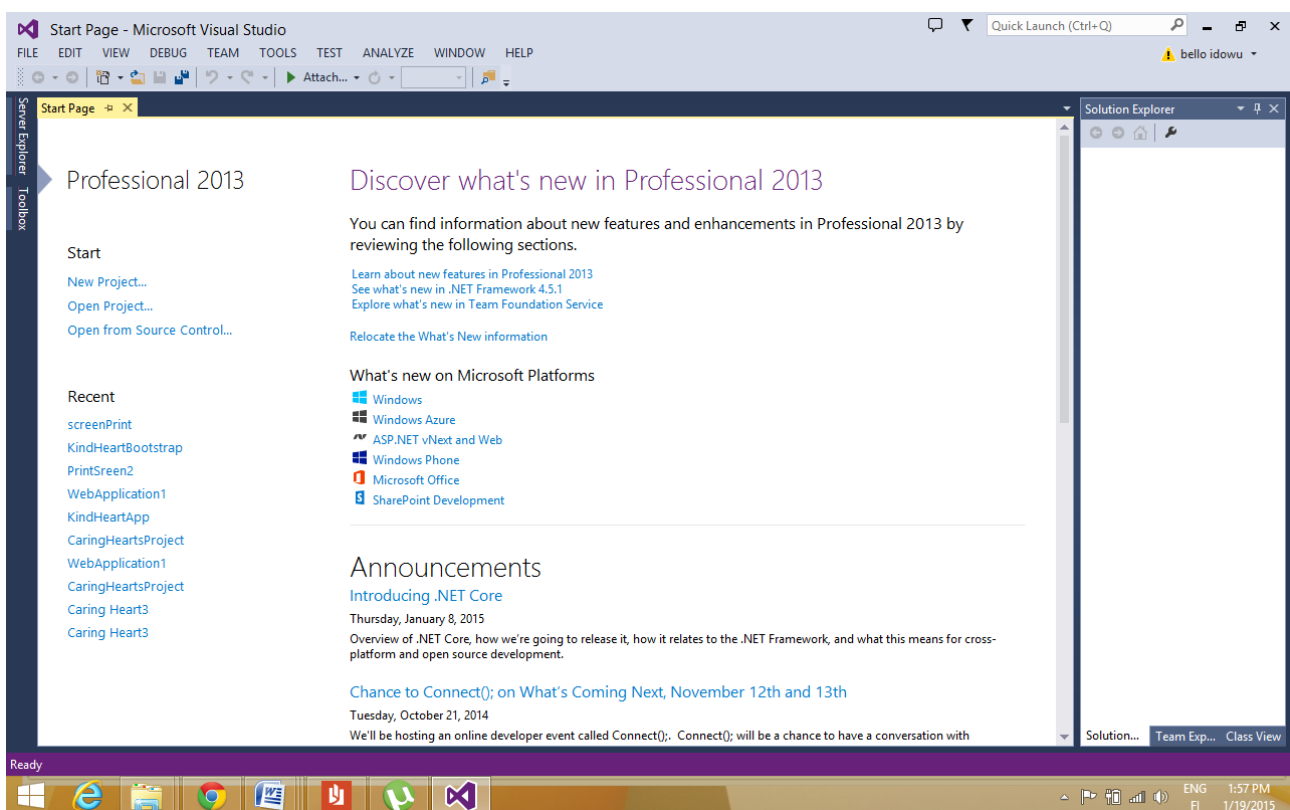


Figure 18. Visual Studio 2013 Start-up Page

The Start-up page contains links where one could find what is current in various Microsoft technologies; selection of videos speak on different features of Visual Studio and any recent information regarding Visual Studio. [10: 7]

## Visual Studio IDE Components

Along the top of the environment of the Visual Studio start page are regions where Menus and toolbars are positioned; likewise, the selection of sub windows, or panes, come out on the left, right, and bottom of the main window area. The main editor is also located at the center. While the code file is open, a form, an XML document, or other related files, appear in this space for editing. A tab is created whenever each file is open and one could easily switch between the files that are opened.

Furthermore, there is a set of tool windows on the either side of the editor space. These areas give additional contextual information and functionality. In the perspective view of the general developer settings, the default layout includes the **Solution Explorer** and **Class View** on the right, towards the bottom, and on the left are the **Server Explorer** and **Toolbox**. The tool windows on the left are in their collapsed, or unpinned, state. Therefore, when the tool window's title is clicked on, it expands and when it is no longer in focus or the cursor is moved to another area of the screen, it collapses. The three series icons are shown at the top right of the window after the tool window is expanded; it is identical to those shown in the top right corner of the left image of Figure 19.

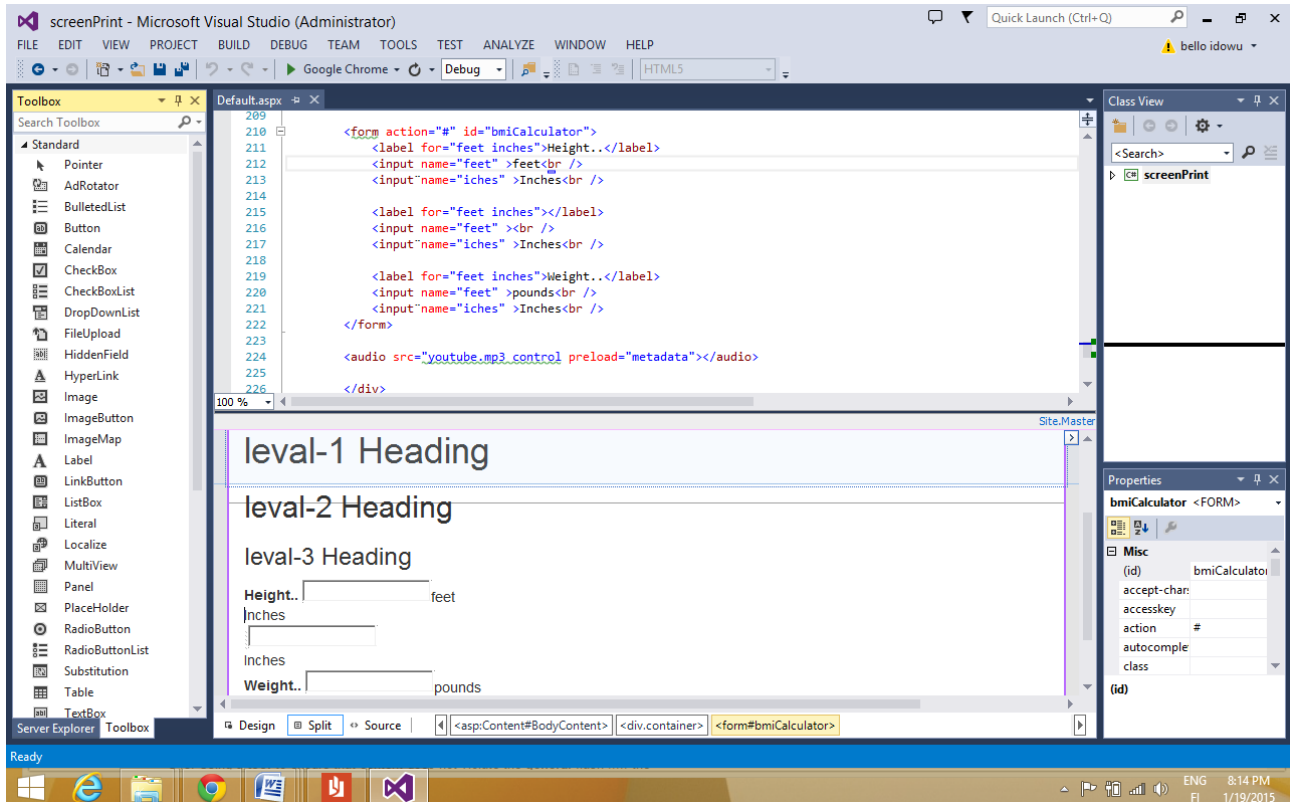


Figure 19. Tool Window Expansion Showing Default Explorer, Class View, Server Explorer and Toolbox on the left.

In order for the tool window to stay in its expanded or pinned state, it is needed to press the icon in the middle, which has a pin sign. The rotation of the pin to the angle of 90 degrees is an indication that the window is now pinned. The third icon with the symbol **X** closes the pinned window when is clicked upon. The tool window could also be re-opened again by selecting it from the view menu. Moreover, it is beneficial as well to be aware that there are some tool windows that are not accessible through the view menu, for instance those that are involved with debugging, such as thread and watch windows. Often, times these windows are accessible through an alternative menu item; for debugging windows, it could be found in the debug menu.

Figure 19 shows the content menu that appears when the first icon, with down arrow symbol is clicked. Each item in the down arrow list represents a diverse approach for arranging the tool window. Furthermore, with the float option, the tool window could be moved and placed anywhere on the screen independent of the main IDE window. This approach is efficient when having various screens, because it is possible to move multiple tool windows onto the additional screen to allow the editor to use a maximum screen real state. In order to make the tool window to have additional tab in the editor space, there is a need to select the dock as Tabbed Document option in the down arrow. [10: 8]

## 4.2 ASP.NET Web Application Environments

### Creating a Web Application Project in Visual Studio 2013

Creating a project in Visual Studio has different template of getting it done. For instance creating a website project in VS2013 is slightly different from creating Windows-type project. Each language also has its own set of project templates; there are no actual options when creating the project. Developing a Web application project has changed significantly in Visual Studio 2013 compared to the previous VS version; not necessarily increasing in number or variety of projects rather, Microsoft has taken upon themselves to render finer clarity to the dialog box and control to the developer who is creating the application. Figure 20 demonstrates how the application is developed in Visual Studio using Visual C#. The application start by selecting File -> New -> Project. Then, on the left, navigate to the web node template shown on the dialog box in Figure 20.

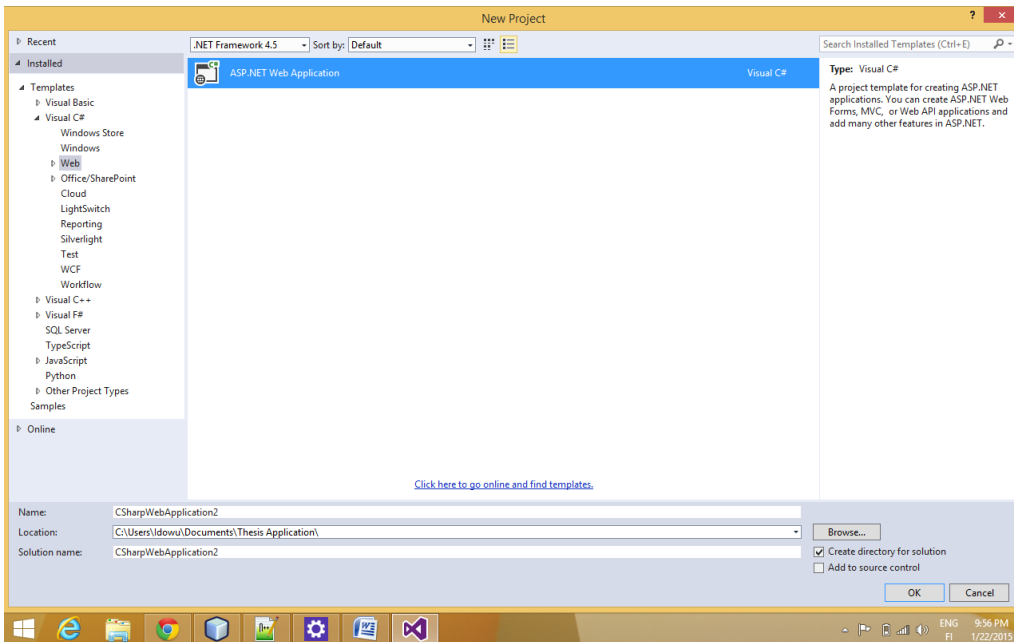


Figure 20. New Project Dialog Box

There is only one template to select on the dialog box above. Any of the Web application templates can be built from this selection. After providing the name for the project and the location to save the application then click the Ok button then open another ASP.NET dialog box with several templates from which one can choose:

- **Empty** : This provides the application with an empty application when it is launched. One has to add new items and functionality needed to build the web application from square one.
- **Web Forms** : This is used to build the traditional ASP.NET web form applications.
- **MVC** : This pattern is used to create a Model-View-Controller(MVC) application.
- **Web API** : This is used to create a REST- based application programming interface (API) that uses HTTP as the underlying protocol. The Web API and MVC are similar; the only difference between these two templates is that the Web API project presumes that there will be no user interface defined.
- **Single Page Application** : This is used to build the web page including rich functionality carried out using HTML5, CSS3 and JavaScript running on the client side, in the browser.
- **Facebook** : This template is for creating a Facebook application, it involves ability to incorporate with functionality like News Feeds and Notifications.[10: 20]



## Master Page

Master Page is the useful feature of web application development in visual studio. Master page makes it suitable for developer to create sections that can be customized. Using this action makes it possible to have single page design that consists of common elements that would be shared through the entire site, as well as define areas that can accommodate content and inherit it on every pages on the site.

Adding a master page to the Web application project can be done either from the website using the Add New Item command or from the context menu in the Solution explorer. During this process, Add menu Item dialog is displays a dialog box shown in Figure 21, which has a large number of items that can be added to the web application.

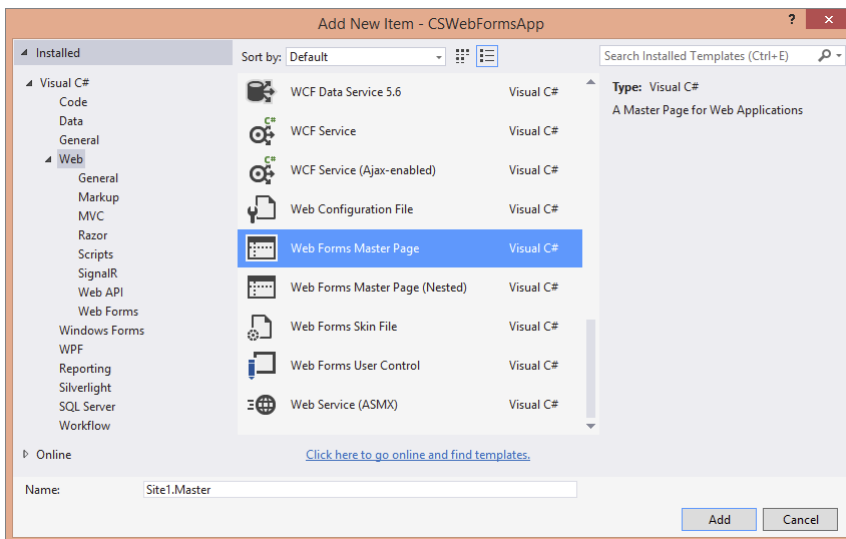


Figure 21. Adding a Web Form Master Page

A master page is a basic web page template, which contains two empty **Content Placeholder** components shown in Listing 17.

```
<<asp:ContentPlaceholder ID="MainContent"
runat="server">
</asp:ContentPlaceholder>
```

Listing 17. Master Page Content Placeholder

One component is located in the body of the web page and the other in the head. This is the location where we put our information detail for each particular page is inputted. It also comes with the extension of **.master**. When master page has been finalized, one can still use it to detail pages for the new web forms in the project by adding a new **Web Forms Master page** shown in Figure 21. This displays the **Select Master Page** dialog box, which can be done also on the Web Site

project. The **Add New Item** window contains a check box titled **Select Master page**. When this is checked, then display the select a Master Page dialog, agreement to the selection dialog is applied when OK is clicked. This action adds a new web form page to the project including one or more content controls, which map to the **ContentPlaceHolder** controls on the master page. [10: 386]

#### 4.3 Window Azure

Window Azure is an integrated Microsoft operating system that is used for cloud computing to facilitate scalable Web application management over the internet. The hosting and the management environment are managed by the Microsoft data centers. Window Azure is complementary to but distinct from Windows server, which is used to design for onsite hosting and management. It can be used to create, distribute and upgrade Web applications without the need to maintain expensive, usually wasted resources onsite.

Window Azure is the most important component of the platform, which the operating system designs for the cloud. Its purpose is to bring an abstraction from the physical components and a set of services that is usable for every application. It also abstracts a set of servers, which administer a common platform for building services and applications in a completely virtual environment. The main duty in the virtual environment is to work with the servers, there is no need of installing application on server A or server B and the application deploy is not on the specific disk C or D. There is also no need of configuring the network card or the virtual directory so as to reveal one of the services.

Window Azure displays a manner to store data which is called *local storage* similar to the normal traditional operating system. Nonetheless hard disks are not included in Azure storage nor do they share traditional networks such as \\servername\sharename; rather, there is a provision of shared storage provided by Window Azure. It is not possible to see Azure processors when the physical server case is opened; despite that, Azure provides CPU power. When an application is uploaded to the environment, Windows Azure selects the best servers, disks and loads balancing strategies for a particular solution. [11: 13]

#### **Merit of Cloud-only Approach**

- There is no need of hardware configurations, drivers, system components or an operating system.
- One does not need an inbound Internet connection.

- There is no need of one's own router installation and configuration, firewall, or public IPs; therefore, one does not need network configuration expertise.
- Patches application or monitoring the system for hardware or software failures is not needed.
- There is no need to do monitoring and maintenance the system. Microsoft takes care of this 24 hours.
- The load balancer configuration is logical, not physical.
- Dimension precision is not needed for the hardware, it is purchased in advance.

Windows Azure recognises every pieces of code that can be revealed and used by the term *service*. For instance, an ASP.NET application, a *while*-block that removes some messages from the queues and a Windows Communication Foundation (WCF) service are all services. Each service is hosted in the platform on different servers. Each server, or more precisely, each node, is based on Windows Server 2008 R2, so virtually any code that run on these premises can run in the cloud as well.

The hosting environment is shared on various nodes that are received by the servers. The individual server is shared by the *server collection* that inhabits in the container. The *container* can referred to as a big shipping container that one can see on the tops of barges and freight trains, and it is placed in a data center by plugging in a huge cable to supply network connectivity and electrical power. Windows Azure manages everything from this point on. [11: 17]

### **Role of Virtual Machine**

The role of a virtual machine is the means by which the developer or IT Manager can transfer an on-premises virtual machine to the cloud. It runs a virtual hard disk (VHD) image of a Windows Server 2008 R2 virtual machine (VM). The VHD needs to be created from an on-premises Windows Server machine; whereby, the image can be uploaded to the Windows Azure at a particular point. Later, it is saved in the cloud, and on request the VHD can be loaded into VM role and executed. Customers can use Windows Services and schedule tasks for the machine by configuring and maintaining the operating system. [11: 32]

### **Cloud Computing Overview**

Today, businesses of all sizes encounter terrible costs and complications when extending and customizing their application. Despite this restriction of the economy, IT developers need to discover new progressive ways of achieving more with fewer resources, but also simultaneously

find new creative approaches to hold on with the changing needs of the business. At the moment, developers are beginning to explore newer cloud-based platforms as a means to obtain efficiency and agility in a higher level. Cloud computing benefits are impressive, because they help to focus on the business and not on the infrastructure.

However, the idea behind cloud computing may be complex for customers since the services and components used, as regards cloud applications, are inherently distributed across the boundary of the organization but the customers orientation is in the environment of having a very large base of existing heterogeneous and distributed business applications across various platforms, vendors and technologies. Finally, the challenges of customers' transitions from the existing server-based applications to building cloud applications which often involves introducing new programming models, tools and runtimes have been resolved by Microsoft by the use of **Windows Azure AppFabric**.

### **Windows Azure AppFabric**

Windows Azure AppFabric is another set of services managed on top of the operating system which gives a comprehensive middleware platform for developing, deploying and managing applications on the Windows Azure platform. It gives further development productivity by adding in higher-level Platform-as-a-Service (PaaS) capabilities on top of the familiar Windows Azure application model. Windows Azure AppFabric is used by the developers to connect application pieces together, manage identity and access control, cache remote resources, and create composite applications. It also helps to unite existing applications to the cloud using protected connectivity across network and geographical borders, giving steady development model for both Windows Azure and Windows server.

Finally, Windows Azure AppFabric enhances development productivity by providing a higher level abstraction for developing end-to-end applications, and management facilitation; while maintenance of the application takes the merit of the hardware and software infrastructure. [11: 31; 33]

### **Three Keys Components of the Windows Azure AppFabric**

#### 1. Middleware Services

Windows Azure AppFabric provides pre-built, higher-level, middleware services that boost abstraction level and reduce complexity of cloud development. These services are open and reusable across languages, for instance .Net, Java, Ruby, PHP, and give developers a powerful, pre-built "class library" for next-generation cloud applications. These services stand-alone, or combined services can be utilised by the developers for the provision of composite solution. [11: 33]

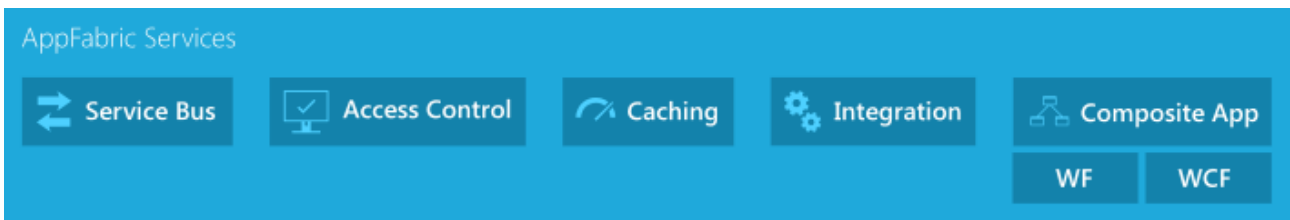


Figure 22. Middleware Services

## Service Bus

Services buses provides secure connectivity and messaging capabilities that enable building distributed and disconnected application in the cloud, as well as hybrid applications across both on-premise and the cloud. It helps in using different communication and messaging protocols and patterns, and removing doubt from the developer about the delivery assurance, reliable messaging and scale. The service bus is located in the cloud and is accessible by any application that possesses Internet connection. One needs to create a new service namespace with the help of *portal* if one wants to start a message exchange through the service bus from any application. Creating services and message exchange by service bus require Internet standards such as HTTP and REST. [11: 33]. Figure 22 illustrates a typical Service Bus message flow.

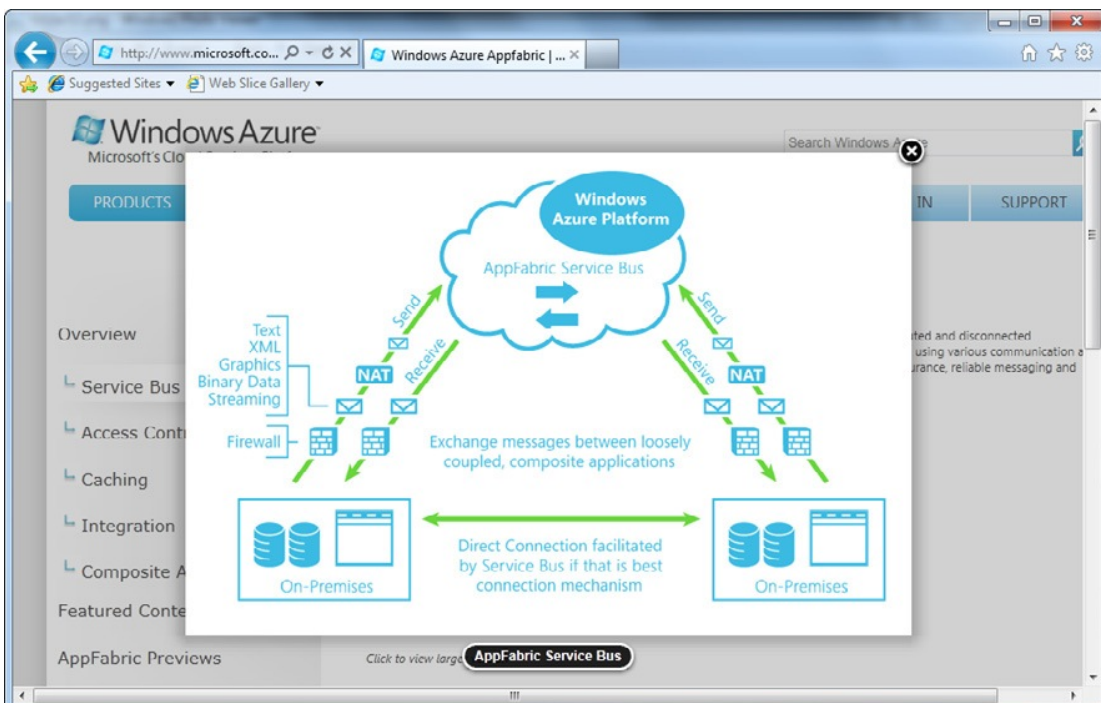


Figure 23. Service Bus Message Flow

The on-premises application situated at the back of the firewall, or NAT, registers itself on the service bus using a POST request to the service namespace URL, created in the portal. Message

sent go to the public URL after the application has been authorized, so that the Service Bus operates as a relay; on-premises applications accept the messages forwarded by Service Bus. This prevents exposure of on-premises application to the business world. It also helps new applications to become part of the message flow. In order for a new application to receive the same messages, it needs to register itself in the service namespace.

The Service Bus can broadcast messages that contain text, XML, graphics, binary data and streaming data. It unmarks classes to different programming environments using a set of SDK. Developers can still make use of REST and HTTP to communicate with Service Bus nodes if SDK is not available for a particular platform. [11: 34]

### **The Access Control Service**

The access control service is the second released component of the Windows Azure AppFabric which enables an easy way to provide identity and access control to web applications and services, while integrating with standard-based identity providers, including enterprise directories such as Active Directory, and web identities such as Windows Live ID, Google, Yahoo! and Facebook.

This service allows permission decision to be withdrawn out of the application and into a set of declarative rules that can convert incoming security request into request that is understood by the applications. Simple and familiar programming model gives distinctive definition to these rules, resulting in cleaner code. It also helps to manage users' permission, saving the effort and complexity of developing these capabilities. Finally, Application is place inside Windows Azure AppFabric Access Control after the permission rule has been removed. Authorizations can be manage by the developer using both the developer portal and the provided command-line tool as requested. [11: 34]

### **The Caching Services**

Caching increases the production of Windows Azure and SQL Azure based applications by making available a distributed, in-memory application cache, provided using the same access as other Windows Azure services, No installation or management of instances, dynamically increased or decreased cache size are needed. Pre-integration with ASP.NET facilitates simple acceleration of web applications without to modify application code. [11: 36]

## **The Integration Service**

The integration services provides common BizTalk Server integration capabilities, for instance pipeline, transforms, adapters, on Windows Azure to increase production and make development easier by using out-of-box integration patterns. The integration service also helps deliver business user enablement capabilities in higher level such as Business Activity Monitoring and Rules, likewise self-service trading partner community portal and provisioning of business-to-business pipelines.

The concept is to relieve the integration between applications that is hosted on the Window Azure platform and third-party Software as a Service (SaaS) solutions. Furthermore, The Integration Service stretches the Service Bus service to relieve integration with current Line Of Business (LOB) applications. [11: 37]

## **2. Composite Application**

As developer produced next-generation applications in the cloud, they are progressively bringing together their application as a composite from pre-built components and services; either developed in house, or consumed from third parties cloud services. Also a majority of the cloud applications require an access to critical on-premises business applications data; composition should be done easily and securely from the on-premises. In conclusion, the composite applications that is highly distributed in nature needs more sophisticated deployment and management skills for managing all of the distributed element that stretch the Web, middle-tier and database tier.

Windows Azure AppFabric cloud middleware platform is accelerated by Microsoft to provide a full composite application environment for developing, deploying and managing composite applications. The AppFabric composition environment delivers three main benefits:

### **Composition Model**

A composite model is a set of .Net Framework extensions for application composition on the Windows Azure platform. This is developed on the simple Azure Services Model ideas and adds new skills for describing and integrating the components of an application. Persistent composition model is provided for both Windows Azure and Windows Server.

## **Visual Design Experience**

A new visual studio based designer skill grants an opportunity to bring together code from the existing application components, forward with newer cloud services and link them together as a single logical entity.

## **Manage as a Service**

The composite Application service is a multi-tenant, maintained service that consumes the definition of the Composition Model and automates the deployment and management of the end-to-end application - Developer and IT Professionals today are needed to eliminate manual steps.

## **3. Scale-out Application Infrastructure**

The AppFabric Services and the composite applications, developed with the use of Composition Model, is built upon an advanced, high performance application infrastructure that has been developed for cloud-scale services and mid-layer components. The AppFabric Container contributes Base-level infrastructure automatically provide scale out availability, multi-tenancy and sandboxing of the application components. The main capabilities given by the AppFabric are:

### **Composition Runtime**

Composition runtime takes care of the full cycle of the application component including the loading, unloading, starting, and stopping of components. It also renders assistance to the configuration such as auto-start and on-demand activation of components.

### **Sandboxing and Multi-tenancy**

High-density and multi-tenancy of the hosted components are enabled by sandboxing. The container captures and multiplies the tenant context to the entire application and middleware components.

### **State Management**

State Management produces data and persistent management for the application component that is being hosted in the container.



## **Scale-out and High Availability**

The container implements scale-out by granting the application components to be cloned and distributed automatically; for state components, the container also provides scale-out and high availability by applying partitioning and replication mechanisms. Partitioning and replication SQL Azure mechanisms are also shared by the AppFabric Container.

## **Dynamic Address Resolution and Routing**

Components can be arranged or reconfigured automatically in a fabric-based environment. The container automatically and efficiently routes requests to the services and target components.

## **SQL Azure**

Microsoft SQL Azure is an extremely accessible, and scalable cloud database service built on SQL Server technologies. As far as SQL Azure is concerned, the developer does not need to install, setup, patch or manage any software. It does not require any physical administration, built with fault tolerance and high availability. Moreover, Using T-SQL based relational model on SQL Azure, and the same powerful development and management tools used for on-premises databases make developers productive on SQL Azure rapidly. [11: 39]

## **5 Kind Hearted Web Application**

### **5.1 Responsive Web Application**

As previously mentioned, the application focus on professional using their professional skills to those who could not afford it; it was designed to be a responsive Web application using Microsoft Visual Studio as an IDE. The application is supported by all browsers on various devices.

### **Navigation Bar Pages**

Navigation bar pages is the bar comprised of the kind hearted website pages which the users can navigate through, each page in the bar serves different functions and representations. Above all, the pages are all responsive on different devices just as the goal of the project implies. Figure 31 shows how the pages are responsive on different devices such as laptops, tablets and mobile phones.





Figure 25. The Kind Hearted Application Home Page on the Tablet

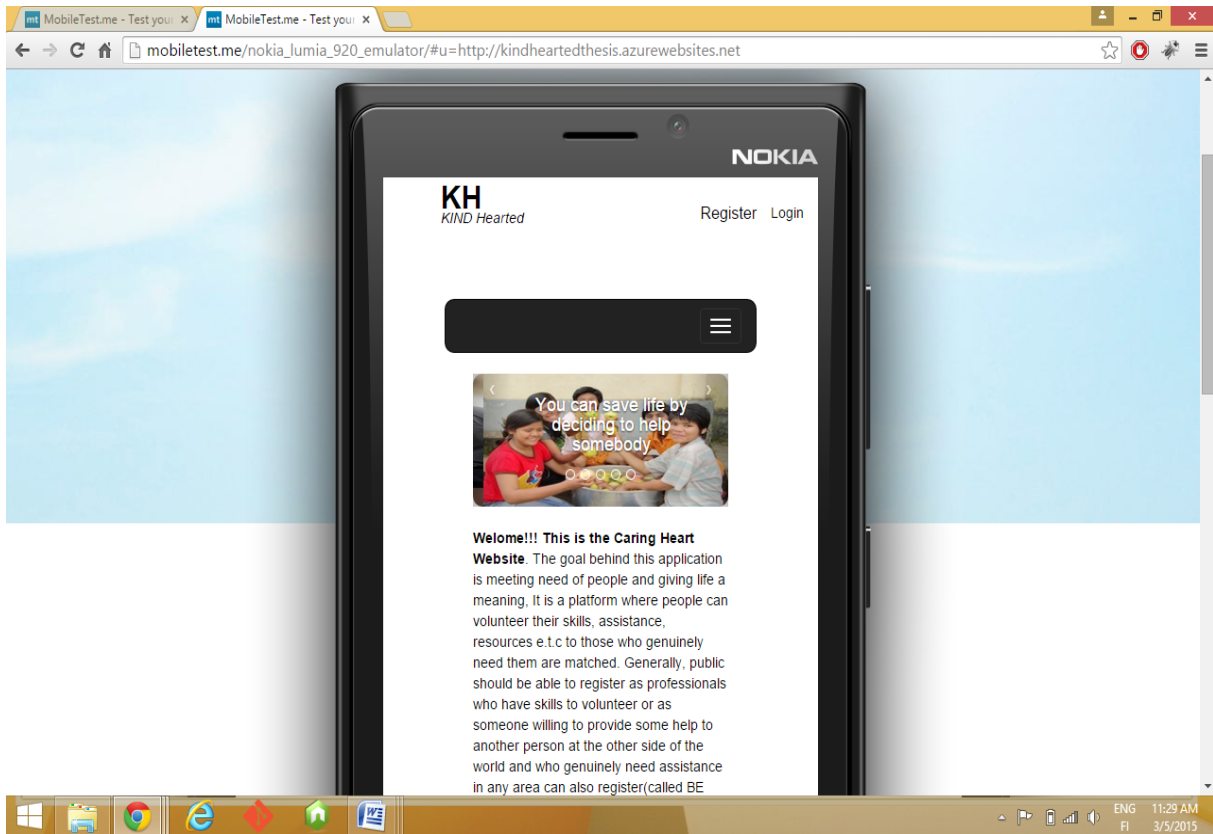


Figure 33. The Kind Hearted Application Home Page on the Mobile Phone

The home page is the default page that appears to the users when they visit the website URL address, it is an interface that users use to access all other pages on the website. It also gives the user an overview of what the site is comprised.

### Register and Login

The application registration form contains two forms which user can choose the form appropriate to them and a user profile which has a user picture. Information will be provided after the user logs in as a member.

### 5.2 Problems

The biggest challenges towards building the application were time and inexperience in building all the application features. Although more features will be added to the application later on, the demonstration which functions adequately, was presented.

A major problem was to make the text responsive when it is opened in different devices and also to make the application supported by all the browsers. Research was commenced and a series of

application testing on different screen devices was performed. Several tutorials were followed and new technologies were learned from the beginning. Furthermore, because time was an issue, the developers had to meet on Skype and shared the screen because of the limited time frame.

Meeting times and places were rescheduled many times because of the ongoing studies or work hours. The developers always encountered some kind of challenge towards moving the project forward. The ideas and name of the project were chosen at the beginning stages. The developers also faced code problems which took hours to debug.

### 5.3 Solutions

Even though the developer were not yet skilled professionals to achieve the specific task given, with the desire to learn, creative, team work and great input from the supervisor, the developers were able to learn faster. At the end, solution to the problems were found and, as a front-end developer, I realised that I needed to target different devices that want to access our website and also to put my text in the grid view container and set specified size for Desktop, laptop, tablet and mobile viewers in the HTML tags.

More so we were able to find suitable time that we could meet from time to time after checking through our schedule carefully. Dropbox and emails, helped us to exchanged files and keep the team.

## **6 Results**

The application called Kind Hearted is designed in order to participate in the Microsoft Imagine Cup World Citizen competition. The application user interface was designed by HTML 5 and Bootstrap using mobile first responsive design.

### **Responsive design**

Nowadays the web is going mobile as people access the web application through smaller electronic devices. Most of these electronics devices are of the different types, shapes and screen sizes. There is a tendency that one may lose a large percentage of users and customers if there is no provision for a mobile website. Therefore, it is purposeful to design a web application with a user interface layout adaptive to any screen size resolution. In the process of designing an adaptive Web application it is important to understand that the adaptive web is not only limited to the screen size of the mobile devices but also the text size in different devices and how the mobile devices connect to the internet and interact with the web content. For instance, recent mobile devices have a touch screen which is a new way to interact with web contents [1].

To develop a website in a mobile context and not only for different screen sizes, there are many challenges to be resolved upfront in mobile development. The need for mobile context that makes what content is crucial and how to best present the content as fast as possible is important in Kind Hearted web application and any Web application in general [1]. Below are the Kind Hearted applications features; although more features would be added in the futures to the application, if there be need.

### **Membership Page**

The membership page is the user registration page which is filled out by those who want to join the program. The pages are of two types:

#### **1 Professional (Helper)**

Professional is also called Helper in the application; they are required to give pieces of information while registering for instance the professional skill field or profession, motivations and how he or she intends to help.

## **2 Beneficiary (Needy)**

The beneficiary is called Needy in the application; they are also required to give information regarding the challenges he or she is facing during registration.

### **User Profiles**

A user profile is the profile that belongs to both Professionals and Beneficiaries. Each user would have a profile page containing user profile picture and displayed information. There would be an indication after the user has logged in giving a signal if the user has a new message. The user profile would be accessible after the user has been authenticated. Each Helper is authorized to look at the profile of the beneficiary could only be authorized to view a professional profile if that professional has agreed to offer help to that particular beneficiary.

A message feature will be added in the profile page to make users communicate together with each other after they have been linked. This means that the users would not be allowed to interact with one another unless they are connected. A professional can browse through the needy list after the authentication, he or she can visit the beneficiary's profile and read more details about that beneficiary and when the professional is satisfied, decide to render assistance to the needy. The professional contacts the administrator through the message feature. The professional and the beneficiary would be linked after the authenticities of both parties are proved beyond reasonable doubt.

### **Monitoring Aspect**

A monitoring aspect is the feature that is being controlled by the administrators, ensuring that the continuity of the linked users is monitored adequately. There are other features that is meant to alert site administrators and block the beneficiary in case the professional suspects anything fraudulent and vice versa.

## 7 Conclusions

The goal of the project was to develop responsive web application with ASP.NET in order to participate in the Microsoft Imagine Cup competition. The application has two categories of users the Professionals, called Helpers, and the Beneficiaries or those in need of professional assistance but could not afford such assistance.

Both professional and beneficiary users can register and log on securely on the website. The data inputted by the users is store to and retrieved from the Sql-server 2012 database using the 3-tier web development architecture. Though there are still some features yet to be incorporated into the application, the team will continue to work on the project until all the features determined to be included into the project from the beginning are added. Nevertheless the fundamental features required to implement in the application have been implemented.

Altogether, the project was a success, I learned and acquired more skills in the process as a front-end programmer of this ASP.NET web application which has a standard HTML front end rather than using common ASP.NET front end control available in the toolbox. When the application is developed and deployed fully, it would help bring professionals who are willing to use their professional skills to help out individuals who are in need of their professionals' assistance but cannot afford it, with distance not being a barrier.



## References

1. Brad Frost. Creating a mobile-First Responsive Web Design [online]. April 16, 2012.  
URL: <http://www.html5rocks.com/en/mobile/responsivedesign/>. Accessed November 30, 2014.
2. Christian Holst. Adaptive vs. Responsive Layouts and Optimal Form Field Labels [online].  
November 8, 2012.  
URL: <http://www.smashingmagazine.com/2012/11/08/ux-design-qa-with-christian-holst/>.  
Accessed August 15, 2014
3. Jakob Nielsen. Responsive Web Design and User Experience [online]. May 4 2014  
URL: <http://www.nngroup.com/articles/responsive-web-design-definition/>  
Accessed July 10, 2014.
4. Nigel Bevan. Usability Net [online]. June, 2003  
URL: [http://www.usabilitynet.org/management/b\\_overview.htm](http://www.usabilitynet.org/management/b_overview.htm) /Accessed May 2014.
5. Jakob Nielsen. Introduction to Usability [online]. January 4 2012.  
URL: <http://www.nngroup.com/articles/usability-101-introduction-to-usability/>  
Accessed August 10, 2014.
6. Jake Spurlock. Bootstrap Scaffolding, Bootstrap CSS in: Dave Winer editor. Bootstrap: O'Reilly;  
2013.
7. Mohtashim M . Tutorial point [online]. India, Private limited company, 2015.  
URL: <http://www.tutorialspoint.com/bootstrap/>  
Accessed January 27, 2015.
8. Microsoft visual studio [online]. United State, Wikipedia foundation, 2015  
URL:<http://www.wikipedia.org/wiki/MicrosoftVisualStudio>  
Accessed January 25, 2015.
9. Jeff martin. Discovering Visual Studio 2013 in: Janbal Dharmaraj editor. Visual Studio 2013  
cookbook: Second Edition March 2014.
10. Bruce Johnson. Integrated Development Environment, Web Application in: Daniel Scribner  
editor. Professional Visual Studio 2013.
11. Roberto Brunetti. Introduction to the Window Azure Platform, Creating a Web Role Project,  
Window Azure Storage, Window Azure Operating System Details, Accessing Azure Services

From Everywhere in: Terea E sey editor. Window Azure Step by Step 2011.

12. Jakob Nielsen. Page Design, Content Design and Site design in: Steve Weiss editor. Designing Web Usability 2000.
13. Patrick Jordan. What is Usability in Taylor Francis editor. An Introduction to Usability 200
14. Responsive Web Design- The wild west of the web [online]. July 4, 2013.  
URL: <http://www.usabilitymatters.com/responsive-web-design-a-process-primer/> Accessed February 2, 2015.
15. Usability First [online]. November 2, 2013.  
URL: <http://www.usabilityfirst.com/about-usability/introduction-to-user-centered-design/> Accessed March 10, 2015.
16. Usability.gov-improving the User Experience [online]. February 5, 2015.  
URL: <http://www.usability.gov/what-and-why/usability-evaluation.html> Accessed October 5, 2015.
17. Ayantek Requirements Analysis- a critical First Step in Web Site Design [online]. February 19, 2013.  
URL: <http://www.ayantek.com/requirements-analysis-critical-first-step-web-site-design> Accessed December 15, 2014.
18. Nitin Hayaran. Vertically Responsive Design [online]. March 8, 2013.  
URL: <http://www.nitinh.com/2013/03/vertically-responsive-design-keeping-things-above-the-fold/>  
Accessed March 29, 2015.

