

**National college of Ireland**

**Higher Diploma in Science Cloud Computing**

**2015**

**Jennifer Mwangi**

**X13121529**

[**jenninjenga@gmail.com**](mailto:jenninjenga@gmail.com)

**ShopAround App**

**Technical Report**

**Declaration Cover Sheet for Project Submission**

**SECTION 1** *Student to complete*

|  |
| --- |
| **Name:**  **Jennifer Mwangi** |
| **Student ID:**  **X13121529** |
| **Supervisor:**  **Dr. Arghir-Nicolae Moldovan** |

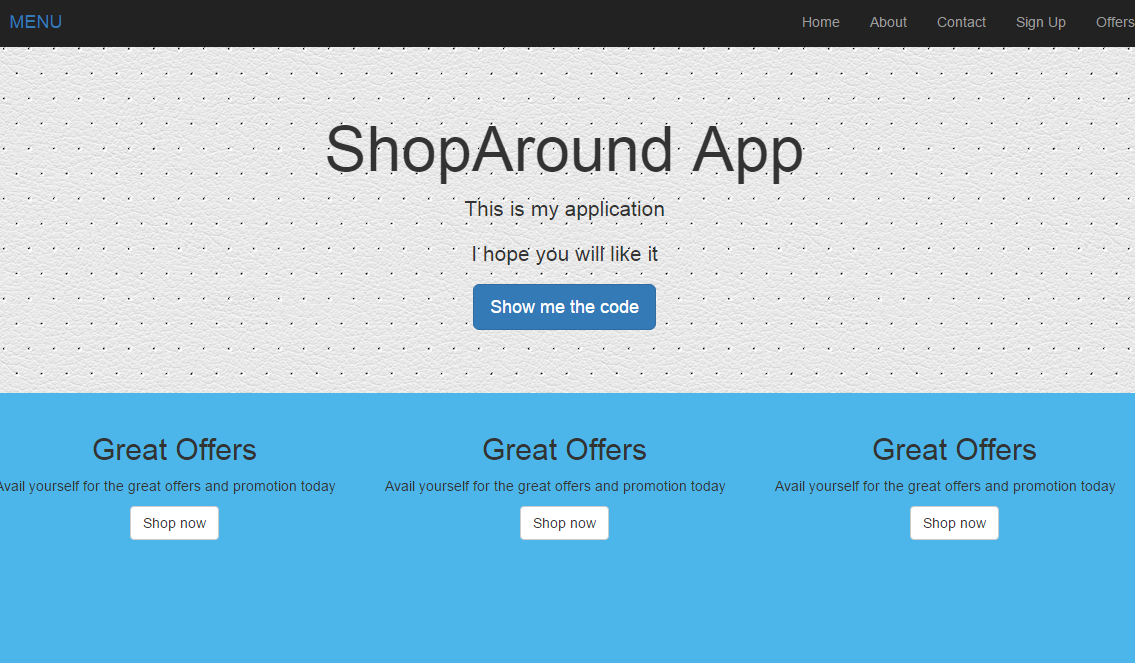
**SECTION 2 Confirmation of Authorship**

*The acceptance of your work is subject to your signature on the following declaration:*

I confirm that I have read the College statement on plagiarism (summarised overleaf and printed in full in the Student Handbook) and that the work I have submitted for assessment is entirely my own work.

Signature:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_jennifer mwangi\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_06/12/15\_\_\_\_\_\_\_

NB. If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the College’s Disciplinary Committee. Should the Committee be satisfied that plagiarism has occurred this is likely to lead to your failing the module and possibly to your being suspended or expelled from college.



***ShopAround App***

**Table of contents**

**Executive Summary**

**1 Introduction**

* 1. **Aims**
  2. **Technologies**

1. **System** 
   1. **Requirements**
      1. **User Requirements**
      2. **Functional Requirements**
      3. **Use case diagram**
      4. **Non-functional Requirements**
         1. **Performance/response time**
         2. **Availability**
         3. **Robustness**
         4. **Security**
         5. **Reliability**
         6. **Maintainability**
         7. **Reusability**
         8. **Resource utilization**
   2. **Design and Architecture**
      1. **Client/Server Architecture**
      2. **MVC Pattern**

**2.3 Implementation**

**2.3.1 MySQL Database**

**2.3.2 Social Media**

**2.3.3 The APIs**

**2.4 System Architecture**

**2.5Testing**

**2.6 GUI (Graphical User Interface)**

**2.7 Evaluation**

**3 System Evolution**

**4 Conclusion**

**5 References**

**6 Appendixes**

**6.1 Project proposal**

**6.2 Monthly Journals**

**6.2.1 Reflective Journal 1**

**6.2.2 Reflective Journal 2**

**6.2.3 Reflective Journal 3**

**Executive Summary:**

The scope of this project is to design and develop a web application which will allow the users to find products which are on sale or on promotions across different stores. The application will make use of APIs which are emerging as a game changer in digital marketing.

The Application can also help companies to generate revenue by allowing other application to view their products and services by exposing their APIs.

According to programmable web open APIs have become an integral component for tech business strategy. Enterprises now see the value in joining the API world because it allows 3rd party developers to create new services that extend a company's brand to entirely new ecosystems. Smaller organizations or individual developers are agile, allowing creativity and new user experiences involving a company's brand to flourish. (Doerrfeld, 2014)

The application which is a web based application will be available in mobile phone since the framework used is responsive making it convertible to any kind of device, only thing required is internet connection and modern browser which adhere to HTML5 version.

1. **Introduction:**

The idea of developing a mobile app that makes it easier to connect customers with offers available from different stores came to my mind when I was browsing numerous number of the websites to check which store is offering good deals. Online shopping has taken a huge lift from people daily lives, we don’t need necessary to visit the stores in order to have our shopping done, thanks to the advancement technology of the websites. Digital marketing has also stepped up to a new level since almost all the businesses wants to be upfront where marketing is concerned. Keeping customers happy and satisfied comes with a huge task of marketing and adverting the value of the products they are spending on.

Most shoppers can’t resist the temptation of a good deal or a great discount, especially when there is one staring at their face. And this is exactly what you can do with mobile application. Creating an offer that is simply too irresistible and notify about it through the app is an easy way to promote businesses.

The true business imperative of APIs is how an Internet of Customers will drive the business models of tomorrow’s global brands. Without APIs to create this customer connection as that fundamental fossil fuel of business growth, organizations will fundamentally fail to adapt to the world of tomorrow.

The purpose of this document is to set out the requirements for the development of a mobile Application that can offer the user a hassle free experience while they are enjoying the services of the App. Following the outline plan and the requirements guidelines it will be possible to deliver the expectation of the project.

The intended customers are more diverse though they must have a bit knowledge of using smartphone, taking into account of how many people shop online daily ranging from teens to middle age group, the app will cater more or less to this category.

**1.1 Aims**

The main aim of this application is to offer the user a chance to search for sales, offers and promotions available across different stores. The users will also get notifications via email when they sign up.

**1.2 Technologies**

* + **API**

What is API? : API basically stand for Application Programming Interface. These are business or organisation capabilities which are exposed over the internet for other application to use them. Basically it’s a programming interface that companies exposes on the internet hence allowing applications to communicate with their backend. APIS are considered as game changer for most of the companies and business. Their benefit range from making money for instance Ebay.com, 60% of their listing are added through Ebay API. Using APIs also help organisation to save money, for instance using Amazon services.

* + **TWITTER BOOTSTRAP:**

The application was built using bootstrap frame work which is a combination of HTML and advance CSS5. Bootstrap if a font-end frame which make web application more dynamic, easier to navigate and provide more interface components on the website. The framework is free and open source. What makes the framework even more dynamic is that it’s responsive, so easily adaptable to any device.

* + **PHP/ SQL:**

While developing the application, Apache server was used for running and testing the application. PHP language was used for database together with MySQL for storing user contact details and also for grapping and outputting APIs which was in form of **Json and xml.**

* + **JavaScript:**

The application used JavaScript as well for fetching and decoding some of the APIs. The language was used again in Twitter bootstrap for smooth scrolling of the application since it has only one page.

* + **JQUERY:**

JQuery and JavaScript was used together when calling the APIs in application.

**2 System**

**2.1 Requirements**

# 2.1.1 User Requirements Definition

The Application should be user friendly, easier navigation hence unique user experience at most. The user should be able to:

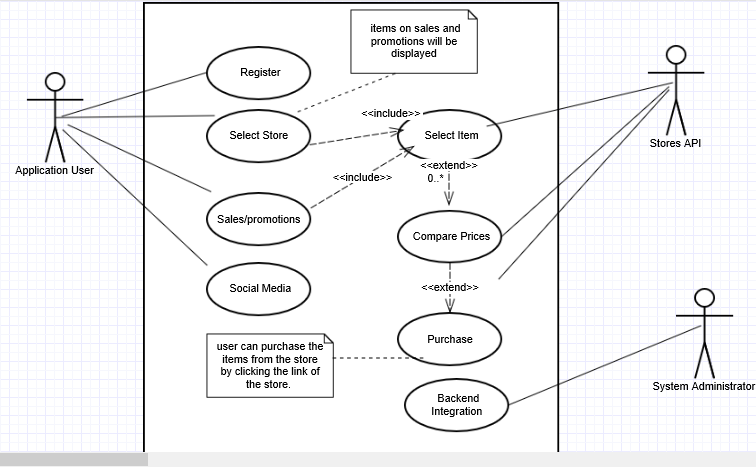
* Register and sign up for the notification of the new deals available.
* Access the required stores and get the items and the prices as desired.
* Compare the prices of the different items in different stores.
* Access other items which are not on sale.
* The user should be able to link to other website where they can proceed and purchase the items online.
* The user will be allowed to connect to different social media and share their opinion.
* Give feedback and their experience through contact details supplied.

## 2.1.2 Functional requirements

According to the systems engineering function requirements mainly defines the functionality of the software system or its components. In this report the functions will capture the behaviour of the application and this will express the services derived from the input and the processing of the particular use case.

### 2.1.3 Use Case Diagram

**The Use Case Diagram provides an overview of all functional requirements.**



### Requirement 1: user registration

The user will have to register first if they want to receive any notifications though this is not mandatory they can opt not to sign up hence no need for registration.

#### Description & Priority

This use case is not of high priority, the user can still use the Application without registering. It’s only a requirement if the user need to receive notifications.

#### Use Case

**Register**

**Scope**

The scope of this use case is to enable the user to register so that they can receive notification automatically.

**Description**

This use case describes the steps needed to register and information required.

**Flow Description**

**Precondition**

The system is in initialisation mode……..

**Activation**

This use case starts when the user click on sign up interface or button.

**Main flow**

1. The system identifies the requirement and generate a form for the user to fill up the required details.
2. The user fill up the form with the collect information.
3. The system process the details and stores the information in the user database.
4. The user receive a notification “Thanks you have successfully sign up”

**Termination**

The system clear off the registration form and gives the user opportunity to continue with the navigation.

**Post condition**

The system goes into a wait state

### Requirement 2: select store

### Description & Priority

This requirement will provide the interface for the user to select different stores available in the app.

The user will need to choose the store which will provide the deal and the promotions available. This requirement is the main use case since it will enable the user to interact and navigate through other use case. It’s also hold the main functionality of the app because without this use case there navigation will be impossible.

**Priority: 1**

#### Use Case

**Select store**

**Scope**

The scope of this use case is to allow the user to get access to the available stores, it’s also allow the user to narrow the search using various options.

**Description**

This use case describes the flow of the event inside the application. User can opt to widen the search by comparing the items from other store which will be possible by using different APIs.

**Flow Description**

**Precondition**

The system is in initialisation mode waiting for instruction from the user.

**Activation**

This use case starts when the user select the store to visit.

**Main flow**

1. The system identifies the store

1. The user select the item.
2. The system display the price
3. The user can compare the price if wishes to.

**Alternate flow**

**Purchase**

1. The system connect the user with the store website.
2. The user can proceed and buy items online.
3. The use case continues at position 3 of the main flow

**Exceptional flow**

**API**

1. The system doesn’t locate the API
2. The user opt for another store.
3. The use case continues at position 4 of the main flow

**Termination**

The system terminate.

**Post condition**

The system goes into a wait state

**Requirement 3: Social media**

**Description & Priority**

This requirement require the user to register first so that they can connect with their friends on Facebook or twitter.

* + - 1. **Use Case**

**Scope**

The scope of this use case is to allow the user to interact with other user or friends on Facebook or Twitter.

**Description**

This use case describes the process involved and how the user can choose different social media and connect.

**Flow Description**

**Precondition**

The system is in initialisation mode waiting for instruction from the user.

**Activation**

This use case starts when the user click on the social media interface.

**Main flow**

1. The system identifies the requirement and load the required page.
2. The user enter the credentials and wait for authorisation
3. The system connect the user to a particular Social media.
4. The user can interact with other user and share their views and opinions.

**Alternate flow**

The use case continues at position 3 of the main flow

**Termination**

The system presents the user with the connection to the required social media.

**Post condition**

The system goes into a wait state

**Requirement 4: Compare prices**

**Description & Priority**

The user will need to choose the store which will provide the deal and the promotions available. This requirement is in the main use case since it will enable the user to interact and navigate through other use case.

Second the user will select the item which will be displayed with the price, the user can then compare prices from the other stores by selecting a particular store.

**Use Case**

**Compare prices**

**Scope**

The scope of this use case istohelp the user to compare the prices of the items they want to buy with other stores.

**Description**

This use case describes thesteps needed to be followed in order to get the best value of the items.

**Flow Description**

**Precondition**

The system is in initialisation mode waiting for instruction from the user.

**Activation**

This use case starts when the user select the item from a particular store.

**Main flow**

1. The system identifies the use case in use.

1. The user select the stores they want to compare the price with.
2. The system process the information and return the results.
3. The user can choose to continue with navigation.

**Alternate flow**

**Item out of stock**

1. The system display the message “item not available”
2. The user can select another store.
3. The use case continues at position 3 of the main flow

**Exceptional flow**

**Item unknown**

The system fails to recognise the item

The user enter the correct item.

The use case continues at position 4 of the main flow

**Termination**

The system presents the nextrequired information.

**Post condition**

The system goes into a wait state.

**Requirement: Back-end integration**

**Description & Priority**

This use case requirement will be used by the system administrator in order to upgrade the system, provide the APIs needed so that the user can get the information they want. Maintenance of the system will also be carried out through this use case.

The use case is of high priority.

* + - 1. **Use Case**

**Back-end integration**

**Scope**

The scope of this use case is to enable the integration at the backend through use of different APIs and refining the required data for the user.

**Description**

This use case describes the process that will take place inside the Application, it will also include the maintenance, and upkeep of the App.

**Flow Description**

**Precondition**

The system is in initialisation mode waiting for instruction from the user.

**Activation**

This use case starts when the user activate the interface.

**Main flow**

1. The system respond to the administrator.
2. The user does the integration required
3. The system is updated
4. The user upgrade the system and its ready to be used again

**Termination**

The system goes back to normal.

**Post condition**

The system goes into a wait state

**2.1.4 Non-Functional Requirements**

In systems engineering and requirement engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. Non-functional requirements define how a system is supposed to be and the qualities of the system

**2.1.4.1 Performance/Response time requirement**

The Application performance should be of high level and the response time should not take more than 25’sec for most of the interface. Since the system relies heavily on Foreign APIs the response time of the some of the interface might be different from others.

However the general performance is expected to be of high level.

**2.1.4.2 Availability requirement**

The Application should be convenient and reliable to the intended users so that they will be able to access all available services.

**2.1.4.3 Robustness requirement**

The Application will require few APIs to begin with, however if the system proves to be successful then more APIs can be added in future. This will not affect the performance of the App and the whole architecture will not be compromised with. The design component should be scalable.

**2.1.4.4 Security requirement**

The system should provide measure to ensure security between end user devices and infrastructures is enhanced.

The system should protect personal information from unauthorised people.

**2.1.4.5 Reliability requirement**

The system should provide the services that is intended to the user without errors or any problems.

**2.1.4.6 Maintainability requirement**

There will be maintenance involved to support the smooth performance of the Application. There will be APIs work required since the system will be using different APIs. This should not affect the performance of the Application.

**2.1.4.7 Reusability requirement**

The Shop Around App will be easy to use, user who are used to other application they should find it easier to use the Application.

The navigation and the interfaces will be very general and self- explanatory, it will be user friendly.

**2.1.4.8 Resource utilization requirement**

The Application will make use of APIs most of the time since the whole Application is based on that. The APIs will be required to give back the information that is needed only therefore minimising unnecessary manipulation of data.

**2.2 Design and Architecture:**

**2.2.1 Client/Server Architecture**

The application is based and developed on client/server architecture. The client can use application in different devices as long as they have access to the internet.

Based on client side the application will be able to:

Register the user’s contact details

Allow the user to share their comments on social media

Users will be able to view the offers available in different stores and if they decide to buy they can proceed and the link will direct them to the stores website.

The application will offer the users a chance to see which store has better deal before they select the store of their own choice.

The application based on server side will be able to fulfil the following request:

Process the user contact detail and send them to database using MySQL server

Using web services the server will be able to send the GET request for a particular API and return the data in JSON format. The data will be decoded using the following languages PHP, JavaScript, Ajax and JQuery. The user doesn’t have to know how the request is being processed as long as they get their request fulfilled.

**2.2.2 MVC Design:**

The Shop Around App is built with MVC pattern, MVC stands for Model View Controller which is a software design pattern used to develop most of the software application. The benefit of using MVC pattern is that it promote code usability as well as implement separation of concerns.

I will explain how the three components were designed in my application.

The View: Consists of the layout of the application based on Twitter bootstrap and different layout of different stores depending on the user’s request.

The Model: This includes, application database from the user and admin, the store APIs

The Controller: Will handle the communication between the model and the view components and subsequently responds to the users and the admin through view component.

**2.3 Implementation:**

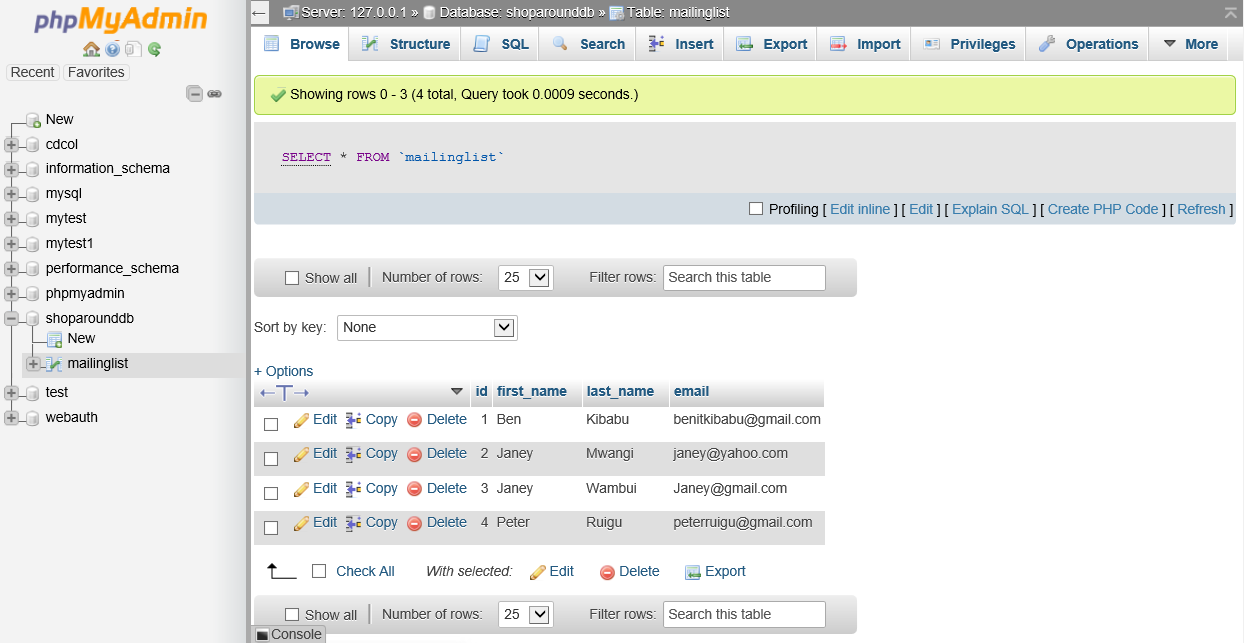
The application has three different implementation, this includes:

* + **MySQL database**
  + **Social Media**
  + **The APIs**

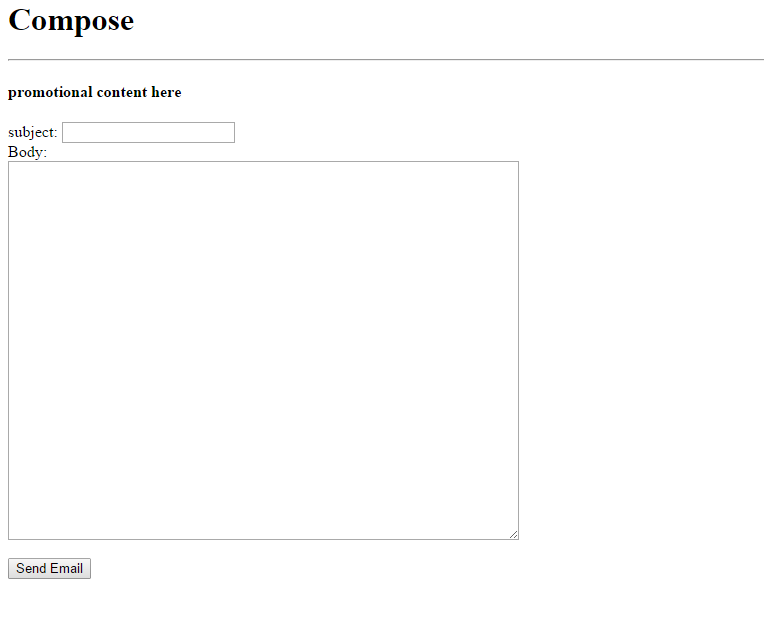
**2.3.1 MySQL database:**

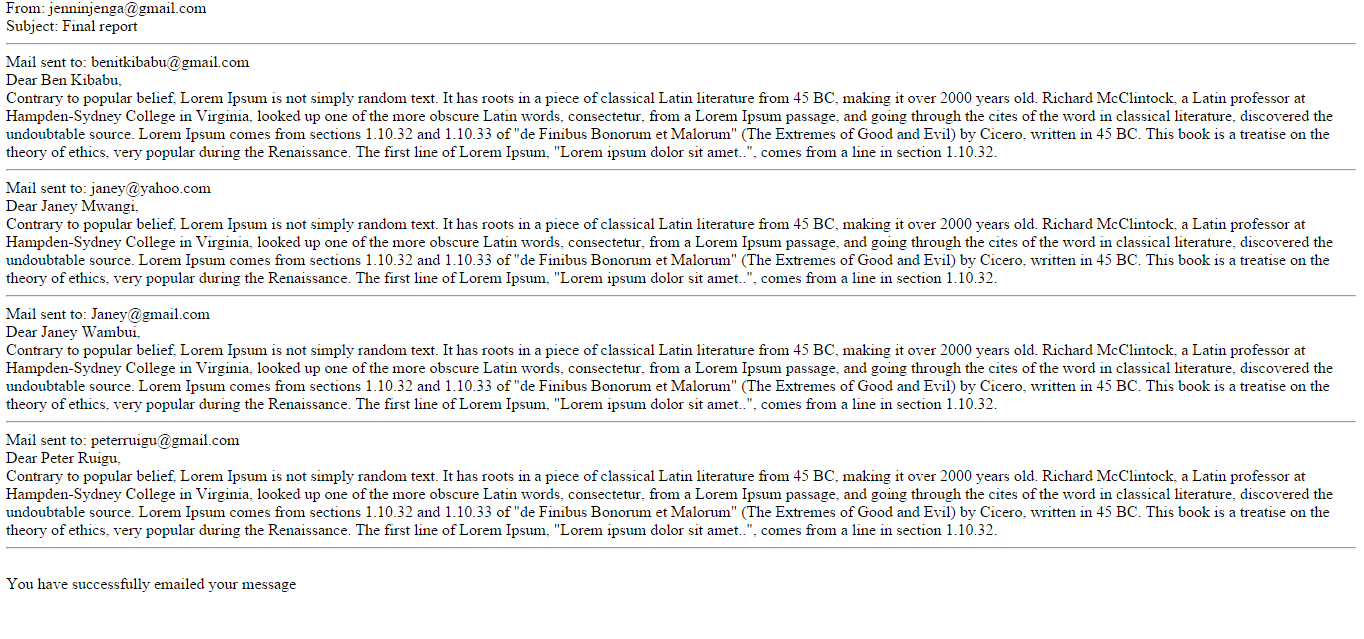
The application takes the users contact details and send them to MySQL database using phpMYadmin. Whenever the admin wants to send the notification to the user, the admin can query the database for the list of contacts available and use them to send the notification to the users. The admin then get the feedback from the browser showing the list of the contacts where the message was sent and the message itself.

**Below is a sample of the users contacts received inside the database:**



**The following two diagrams illustrate how the admin send the notifications to the users and the confirmation after the message is sent:**





**2.3.2 Social Media:**

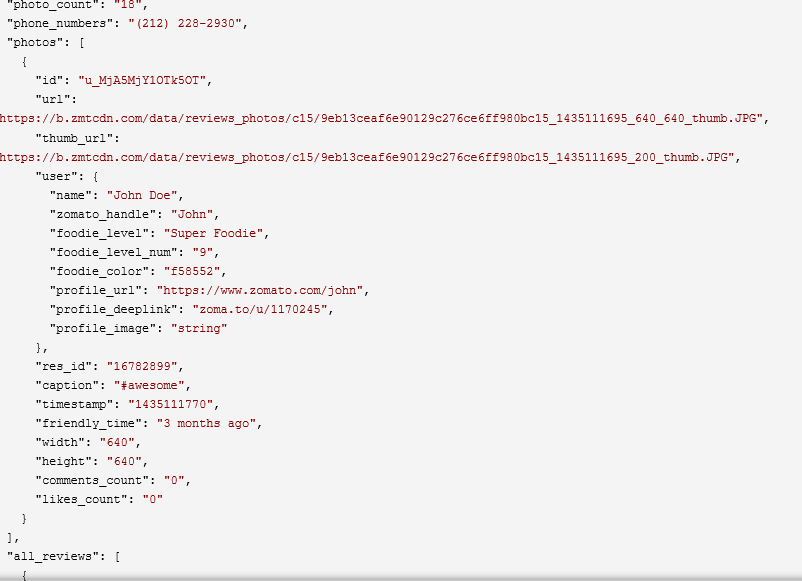
The application will enable the users to share their comments using one of the popular functionality provided by the Facebook.

**2.3.3 The APIs:**

The application is based on APIs of different stores which provide data upon the request using different languages which I have outlined earlier. These includes, PHP, JQuery, Ajax and JavaScript. Basically the controller take the request from the view and sends the request to the business layer. In this case a particular API get a request using one of the HTTP VERB, GET and sends the request to the server. The server process the request upon getting the authorization.

The data is usually inform of JSON format which makes it human readable, since this is not what we want for the user to see, the JSON is then decoded using the above languages, which gives us the actual information requested by the user.

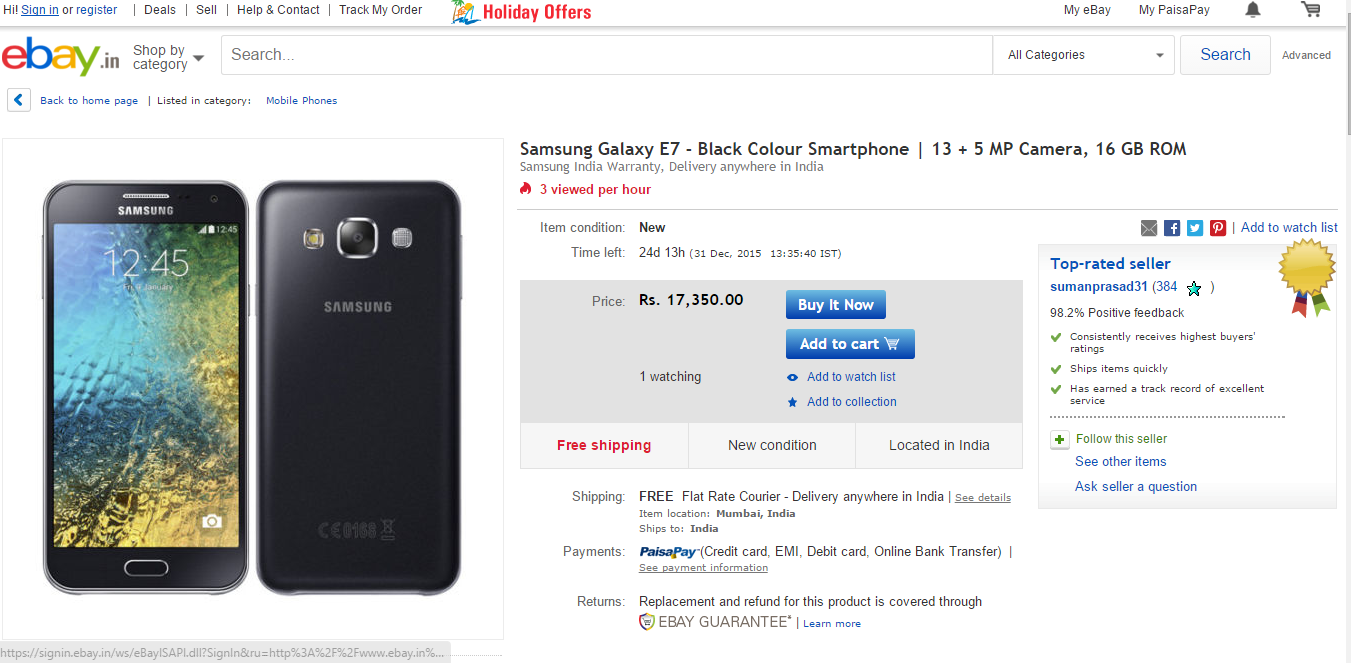
**Below is a snippet of the JSON code format:**

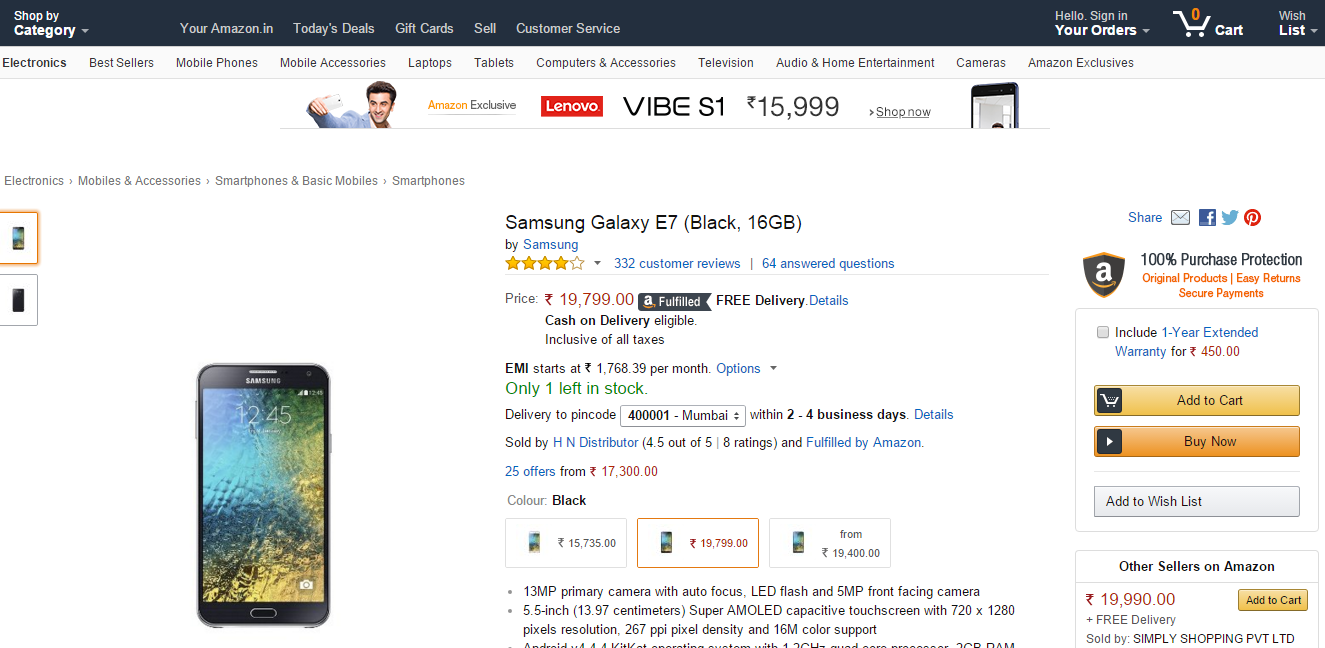


**The actual information received by the user from the browser:**



**These are some of the store selected by the user:**

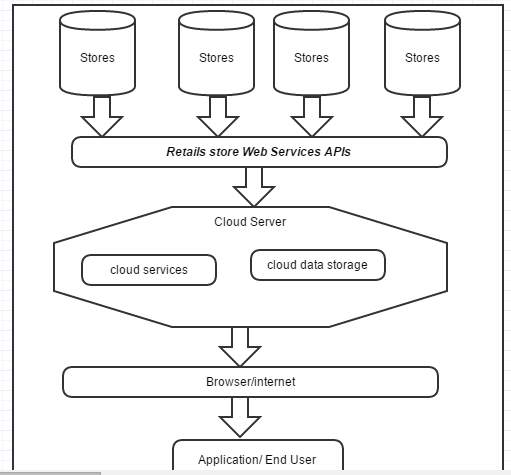




**2.4 System Architecture**

The APIs Architecture will provides the model of the whole system. The following diagram will demonstrate how different APIs will be integrated to make up the whole Application.

API’s are a quick way to get and manipulate data from other web services for your app or site. It’s also common to create “mashups” using the overlapping data from multiple APIs (such as geospatial data) to create new functionality. (Zone, 2014)



**2.5 Testing:**

Apache server was used throughout the development of this application. XAMMP which is an open source server and support multiple platform, was used for testing the application all the time. The server was used for MySQL and PHP files which requires the server to execute, HTML files, Ajax and JavaScript needed only the internet and the browser.

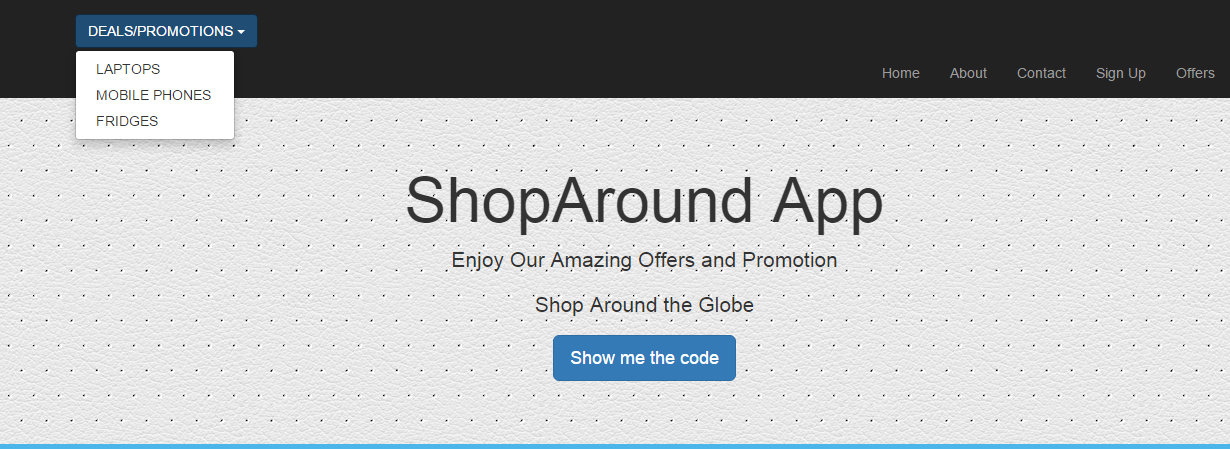
**Mashape** is a developer’s platform where different APIs and applications are added, tested and integrated based on user’s interest and purposes. I was able to use this platform to manipulate my APIs in different ways. Great tool for developers, though the platform still under development.

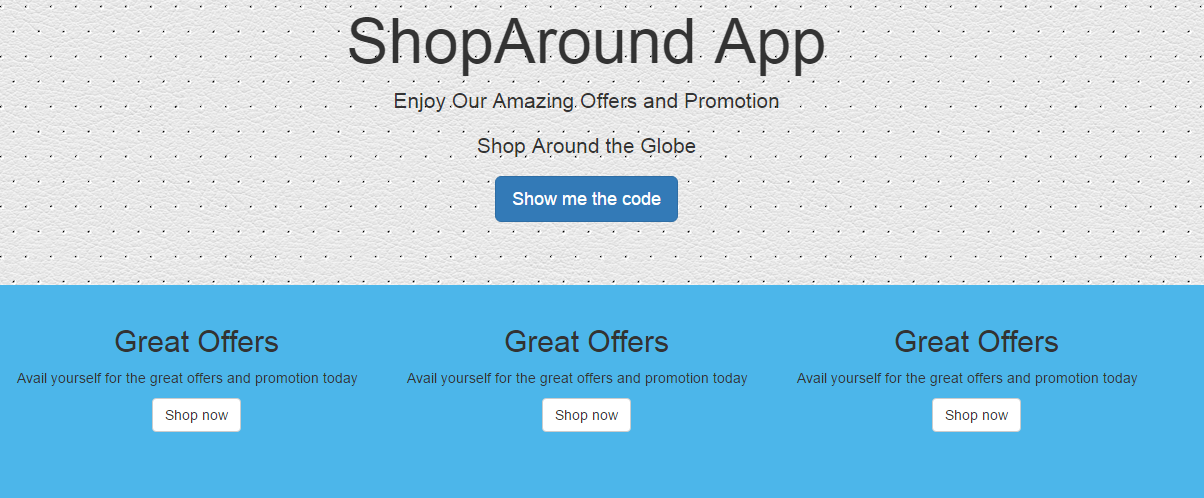
According to Mashape Team, Mashape Marketplace is building the largest API hub and management platform for cloud APIs, driven by a passionate community of developers from all over the world. (Mashape, 2015)

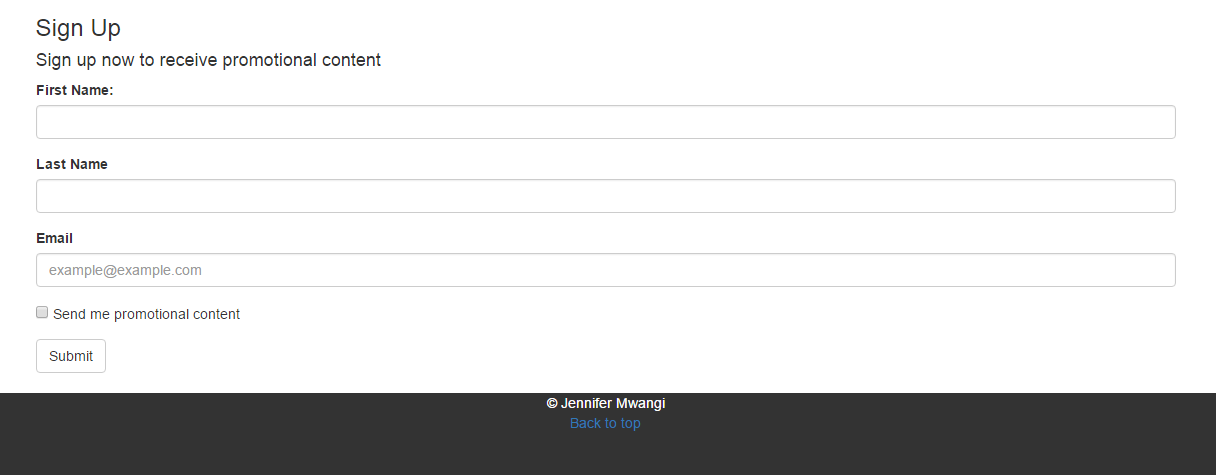
After the completion of the project, all functionalities were tested on a different devices and different browser.

**2.6 Graphical user interface (GUI):**

**The following screenshot illustrate some of the main interface in the application.**







**2.7 Evaluation:**

Due to unreliability of some of APIs, I was forced to work with only one API which was more reliable and available all the time. The API has different END point and I managed to use for different products and different stores. The API was used in different languages but worked well with JavaScript and Ajax.

Some of the limitation of using APIs is that you can only work with what is offered on a particular API else it had to get specific data that you needed.

The following developer’s dashboards were used while testing the APIs:

Programmable web

**3 System Evolution**

The system will scale up in future, maybe add more functionalities which may have been left out during the design, or more useful requirements which can be realized after the development of the Application. The scope of the project can be widen if need be to cover more diverse usage

**4 Conclusion:**

The main purpose of the project was to show how the APIs works and how they can be used to generate revenue to different organisation through sales. As I mentioned earlier one of the main benefit of exposing API was to generate sales. APIs are transforming Companies to compete into digital marketing as well as meet rising customer’s expectations across multiple channels and platform. In other word the APIs are benefitting organisations and the consumer in general. While I was doing research and trying to get the APIs I found that not all organisation are open to the idea of exposing their data. In essence some of the organisation were against the idea of using their API to compare the prices with other stores, and that was the major challenge for me.

However I learnt a lot about APIs and how powerful they can influence business into day’s digital economy. APIs are dynamic and there is so much you can do with them once you come to know how they function. They are also saving a lot of time for the developer and programmers since you don’t have to build any particular application in order to use them. They can be executed using different languages and use different platform.

Regarding my application I was unable to finish up in standard since I found out that there so much to learn from API before implementing some of the functionalities, however I was able to demonstrate how the APIs can be used by another application smoothly.

In future I would like to work with APIs and learn how to integrate the APIs professionally using different languages. So it’s my intention to carry on with this project and give it a good polish up in future.

**5 References:**

Doerrfeld, B., 2014. *Programmable Web.* [Online]   
Available at: http://www.programmableweb.com/news/how-apis-are-enabling-unique-business-partnerships/elsewhere-web/2014/12/28  
[Accessed 05 December 2015].

Mashape, 2015. *marketplace.* [Online]   
Available at: https://market.mashape.com/dashboard  
[Accessed 05 December 2015].

Zone, C. S., 2014. *Computer Science Zone.* [Online]   
Available at: http://www.computersciencezone.org/50-most-useful-apis-for-developers/  
[Accessed 08 August 2015].

**6 Appendix:**

**6.1 Project proposal:**

**Objectives**

The Shop Around App is an application which will provides different ways of shopping to the customers using various supermarket on the market.

The main aim of the App is to narrow down the offers and sales offered in different stores which mainly will be displayed or will be available on the app. If the prospective customers are happy with the offers and they wish to proceed and make a purchase, they can do so by clicking the link of the store which will take them to stores website.

The idea is to save time for the shopper who spend a lot of time looking for the offers available in different stores by visiting stores websites and sometimes the offers are not very visible unless they keep on navigating to a particular products.

The idea is to develop an app where the retails such as Tesco, Dunne stores, Lidl, Aldi and marks and spencer, will be uploading their deal of the week so that the customer will be aware of where to shop instead of browsing all the website for deals and promotional offers.

Easier way of shopping around, more like done deal or just eat.ie.

**Technical Approach**

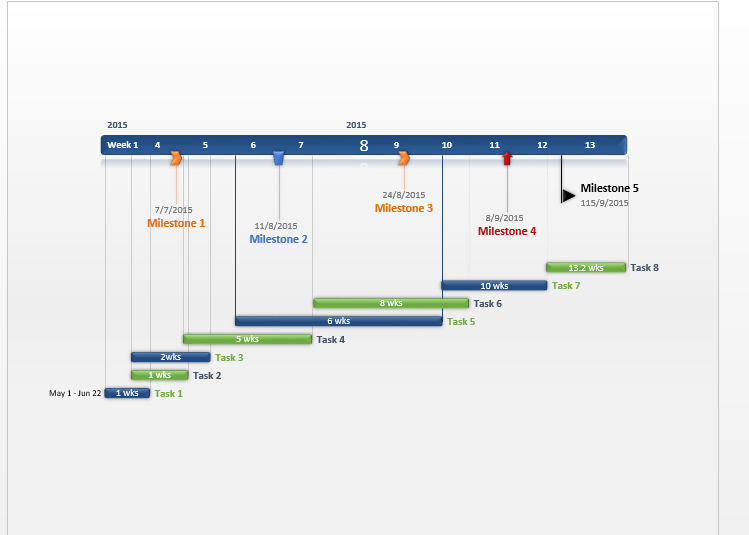
This will be done by using different API and comparing them. Stores can also upload their products using API (more research needed, working on that at the moment)

The App will be built mainly for android phones at the moment and if it’s proves to be successful, other platform will be considered.

Security issues and usability purposes will be addresses later.

**Project Plan**

**Grant chart: Project Management plan will be used throughout until the project is finalised.**



**Technical Details**

The language to be used in the implementation might be **JQuery Mobile App** or **Twitter Bootstrap**

**Python** language for the server side scripting.

**MYSQL** Database to store user details.

**Evaluation**

Validation will be done to make sure the App work well in different browser.

Shop Around App will be uploaded into cloud hopefully Amazon and will be available for download in Play store.

The feedback sheet will be used to give different views and opinion after testing the performance of the App.

**6.2 Monthly Journal**

**6.2.1 Reflective Journal 1:**

**My Achievements**

This month, I was able to outline the requirements of my project through my proposal. Am designing a mobile application which will enable customers to compare the prices and find sales, offers and promotions of different products in various stores.

My contributions to the projects included, research on the APIs of some of the stores which I require for my project. I have also tried to use some of the frame work and tried to work out which one will suit my specification as well as functionality. My mobile application will features different language and framework which I intend to work on in the next month.

**My Reflection**

I felt, it worked well to use different APIs for various stores to compare the prices and that way, it will give the users different choices for the products with the best prices available.

However, I was not successful in finding the suitable APIs which I would like to analyse on my project, am planning to enlarge my scope by adding any APIs of any store that I can get without focusing only on supermarkets.

**Intended Changes**

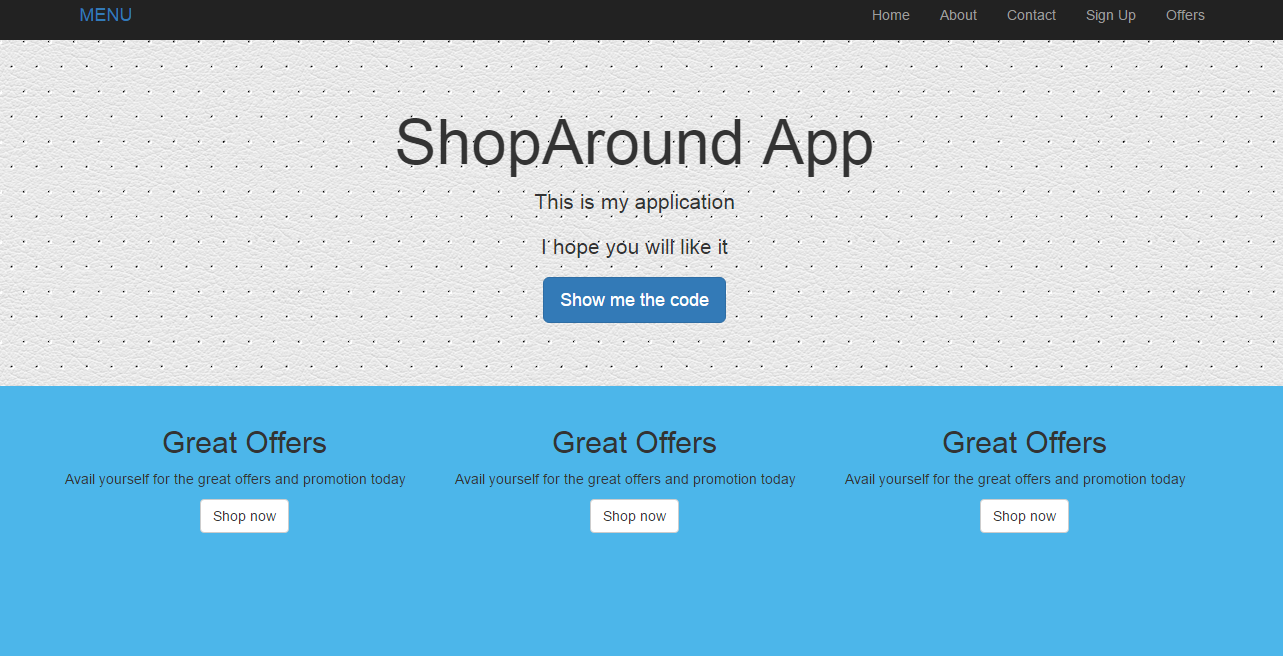
Next month, I will try to do more research on APIs, narrow down the main information (data) that I need from different APIs, see what are the limitations and the benefits and try to understand how they work.

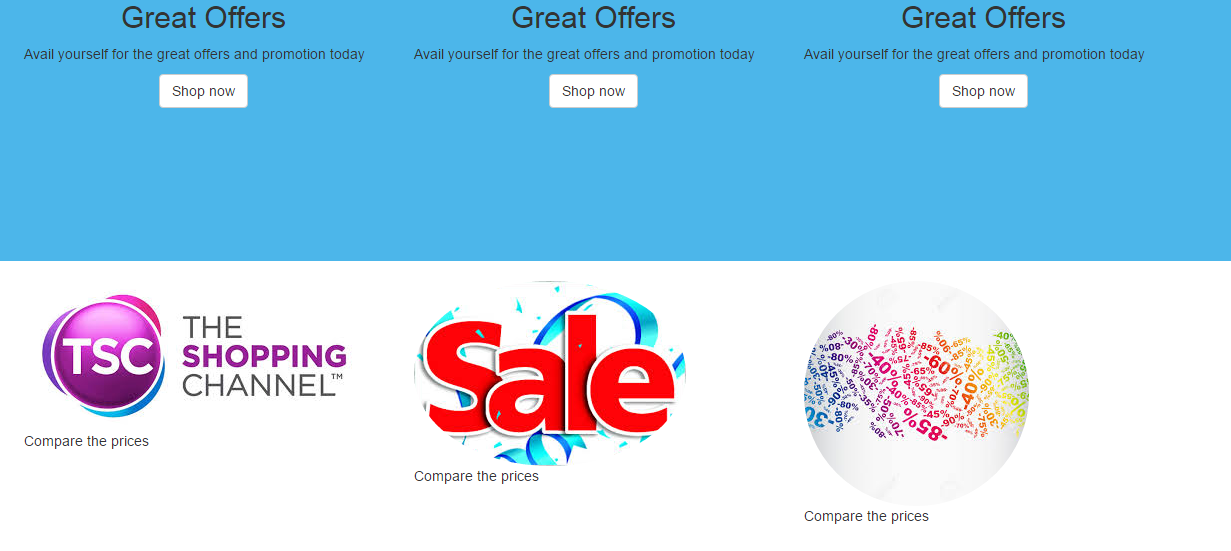
After researching on APIs, I realised that it’s not easy to get APIs, simply because most of the stores they don’t want to share their data with everyone and the only available APIs, I found that they don’t have all the information that I need, some of the data is not accessible. The next big task I have is to work on APIs since they hold the base of my project, without them my project will not function.

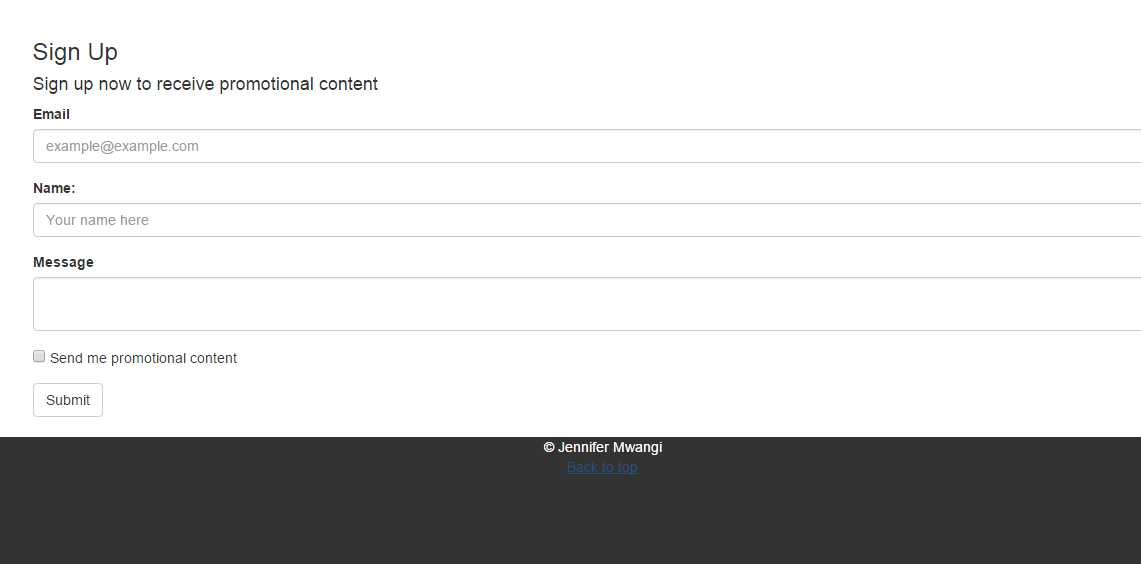
**6.2.2 Reflective Journal 2:**

**My Achievements**

This month, I was able to build my application using the latest twitter bootstrap frame work. I have implemented some of the functionalities, which includes the design of the application and its layout in general. Which looks like the diagrams below. Since my whole application involves working with APIs of different stores, I have managed to get few of the APIs though they are very restrictive in terms of data. I could only get specified data at once which is not helping with my application.







**My Reflection**

After getting the APIs I have noticed that most of them they are using RESTFUL web services so the data is supplied as in Json. I have to decode the information from Json to get the products on the Application. The easier way for me is to use python which am learning at the moment. I have the sign up form for the user to register but for now the form is not working since I have not implemented the functionalities required. Am working on PHP as well to learn how to store the information entered by the user.

**Intended Changes**

Next month, I will try to work on PHP and Python since they are the main programming languages that I need to use on my project. Am also trying to get more APIs which can meet my requirements as intended. I will also work on my layout after I get more APIs, at the end they will determine the products or the items which can be displayed on the Application.

Finally I have realised that I need to put more time at my project and work on the different languages as I am still finding it hard to understand.

**6.2.3 Reflective Journal 3:**

Due to time constrain and medical issue I couldn’t meet the deadline on my project, I decided to defer my project since I needed more time. I took the opportunity to look for more APIs since it was really hard to get the APIs and the available one’s had their own restriction on the usage. Most of the organisation were not ready to lend out their API even for educational purposes.

I decided to sign up with programmable web website where they provide the documentation of various APIs on the market. After requesting for APIs from more than 20 organisation I managed to get five of them which I used to work with during the development of the project.

In the end I was able to accomplish some of the main functionalities in my application though not up to my expectations due to APIs constrains. Overall I learnt so much from this project.