# CHAPTER 1

# INTRODUCTION

**1.1 Introduction to the project:**

Today the trend is to visit the shopping malls for shopping. Each shopping mall will have number of shops pertaining to same items. Usually it is not possible for an individual to visit each and every shop before purchasing the product, hence there may be chances that he might be missing a good discount sale in the shop he didn’t visit. Further today data has more value than money, so collecting shopping habits of a user is essential for the various brands to improve their sale.

We often see many shopping apps which will display all the offers available for the products, in the android application that we have developed and prioritizes the customer requirements and displays offers of the products for which user is interested in. Our android application analyzes user’s click pattern and tries to suggest the user only those products, which matches the user’s interest instead of giving him a random advertisement suggestion. In this project we have developed an android app by name "Smart Shopping Assister", in which app displays offers available for various products in shopping malls located at different shopping centres. When app user clicks on any offer displayed, the detailed description of the product appears along with the location indicating where the product is available with that particular offer. Our android app also tracks users location so that whenever they are nearer to shopping malls, they will be pushed with various information about the events occurring at the mall, updated list of offers which are available in that mall, so he gets mood to visit the mall and purchase some products and thereby help in increasing the sales of products in shops in that mall.

**The app can be viewed from two viewpoints:**

* From user point of view, the app is very helpful for user to get customized offers and further get info about what products are available in various shops and what offers are offered by each shop. With this valuable information customer can save time and money, without compromising on his interests.
* From shops and brands point of view, the shops can push customized offers to the users of the app to attract them to their shops which will increase their sales and brands can get valuable data of what type of products are in most demand and what type of products are preferred by most buyers.

**1.2 Problem Statement**

To develop an android application through which users can register and view various offers that are available in various shopping centres .The offers are displayed based on his/her interest pattern and also on the basis of nearest mall. To track user’s location and to inform about the various offers in nearest shopping centres. The project is also aimed at developing a web application which will be deployed on cloud, through which various registered shop owners can upload the offers and also view how their offered offers are being accepted.

**1.3 Objective of the Project**

* The main objective of our app is to guide the user in purchasing various products in a shopping mall by providing valuable information related to various offers in various shops.
* Here it provides two types of offers. They may be generalized offer and customized offer. Generalized offer will be given to all the users who use our app and the customized offer will be given for those users who purchase products in that shop by using our app.
* App also tracks location of user.
* App also use app user's click pattern for analysis.
* The app also captures the shopping habits of the user.
* Allow the shop owner to view graphs depicting previous sales.
* Allow the shop owner to view different configuration options and update the offers.

**1.4 Scope of the Project**

The scope of this project is to design and develop an android application in which user can view offers on various products at various shopping malls. Hence there may be chances that he might be getting a good discount sale in the shop. Our android app also tracks users location so that whenever they are nearer to shopping malls, they will be pushed with various information about the events occurring at the mall, updated list of offers which are available in that mall. So he gets interest to visit the mall and purchase some products.

**1.5 Deployment models on cloud**

**Cloud Computing:** Cloud computing is the delivery of computing as a service rather than a product , whereby shared resources, software, and information are provided to computers and other devices as a metered service over a network (typically the Internet). Cloud computing is mainly classified into three types based on the deployment model; Public cloud, Private cloud and Hybrid cloud.

* Private cloud: Private cloud is infrastructure operated solely for a single organization, whether managed internally or by a third-party and hosted internally or externally. They have attracted criticism because users “still have to buy, build and manage them” and thus do not benefit from less hands-on management, essentially “lacking the economic model that makes cloud computing such as intriguing concept”
* Public cloud: A public cloud is based on the standard cloud computing model, in which a service provider makes resources, such as applications and storage, available to the general public over the Internet. Public cloud services may be free or offered on pay-per-usage model.
* Hybrid cloud: Hybrid cloud is a composition of two or more clouds (private community, or public) that remain unique entities but are bound together, offering the benefits of multiple deployment models. It can also be defined as multiple cloud systems that are connected in a way that allows programs and data to be moved easily from one deployment system to another.

**1.6 Software-as-a-service (SaaS)**

* The service must be available to the customer on-demand, without any requirement for the installation and configuration of any hardware or software up-front.
* The service must be paid for by the customer on an pay-per-use basis, without any requirement for a long-term contract or up-front payment. Examples of this might be monthly subscriptions per user, CPU usage, transactions, or storage.
* The provider is responsible for managing the service, without any requirement for the customer to maintain, upgrade, or otherwise administer their database.

**1.7 OpenShift**

OpenShift is a platform as a service product from Red Hat. It is also an Infrastructure as a Service (IaaS), comparable to Google Storage and Amazon S3 online storage services.

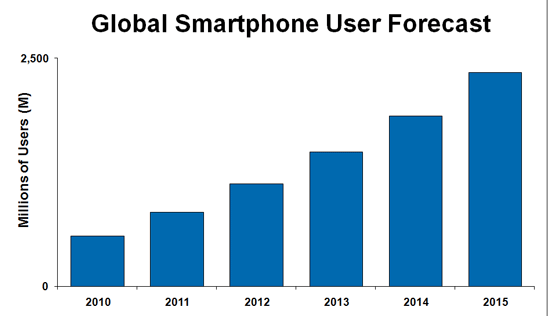
The software that runs the service is open-sourced under the name OpenShift Origin, and is available on GitHub. Developers can use Git to deploy web applications in different languages on the platform. A version for cloud computing is named OpenShift Enterprise.

OpenShift also supports binary programs that are web applications, so long as they can run on Red Hat Enterprise Linux. This allows the use of arbitrary languages and frameworks. OpenShift takes care of maintaining the services underlying the application and scaling the application as needed.

# CHAPTER 2

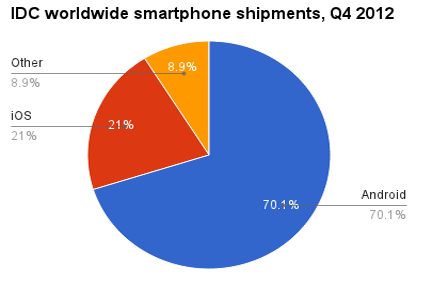
# LITERATURE SURVEY

# 2.1 Statistics of Android Usage:

****

**Fig 2.1.1 Bar graph for number of Smartphone users**

Fig 2.1.1 indicates that number of smart phone users are increasing year by year drastically. According to graph in 2015 the number of smart phone users are over 2000 million, hence providing solution to the previously mentioned problem through smart phones is one of the most apt and effective ways.



**Fig 2.1.2 Pie chart for number of android users v/s other Mobile OS**

Fig 2.1.2 illustrates that number of android users are more than triple with respect to any other Mobile OS. Hence Android is chosen as the platform for developing the solution to the problem statement.

**1) Shopping mall directory based on Wi-Fi**

In 2013, P VijayaPrasad et al., describes the development of mobile application for shopping mall directory based on Wi-Fi. It also provides information for customer to enhance their experience in the shopping mall. Shop information like Shop names, Categories, locations, description and Floor Layout are provided in this mobile application. Therefore, with this technology, product recommendations and customized marketing messages (i.e., advertisements and e-coupons for the recommended products) can be delivered to customers smart phones as soon as the customers come into a shop.

**2) Fast mall**

**Fast Mall** is an American [application software](http://en.wikipedia.org/wiki/Application_software) for [mobile devices](http://en.wikipedia.org/wiki/Mobile_device) that provides [interactive](http://en.wikipedia.org/wiki/Interactive) [shopping-mall](http://en.wikipedia.org/wiki/Shopping_mall) maps that guide users with [turn-by-turn directions](http://en.wikipedia.org/wiki/Turn-by-turn_navigation) for various places in shopping mall.

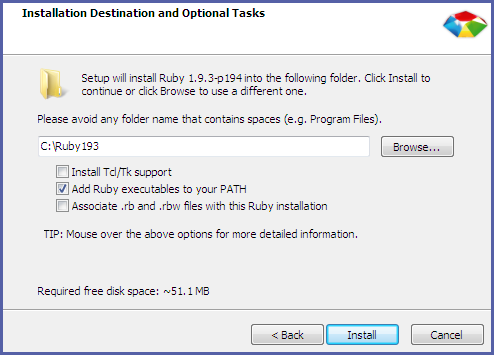
* Drawbacks in fast mall application: App does not display any specific offers available for various products in shopping mall.
* App does not use app user’s click pattern for analysis.
* App does not track location of user.

**2.2 Open Shift**

**2.2.1 Open Shift Client tools**

**Step 1: Install Ruby with RubyInstaller**

RubyInstaller provides the best experience for installing Ruby on Windows. Download the newest version 1.9.3 and launch the installer. During the installation you can accept all of the defaults, but it is mandatory that you select the Add Ruby executable to your PATH check box in order to run Ruby from the command line (see image below).



After the installation is completed, to verify that the installation is working open a Command Prompt and run:



You should get a response with your installed Ruby version:



If the Ruby version message does not display, the Ruby executable may not have been added to the path. Restart the installation process and ensure the Add Ruby executable to your PATH check box is selected as in the image above.

**Step 2: Install Git version control**

The next step is to install Git for Windows so that you can synchronize your local application source and your OpenShift application. Using Git for Windows will greatly simplify the setup and management of SSH keys using basic rhc commands. It will also allow you to SSH into your applications without the need for additional tools.

Download and install the latest version of Git for Windows. During the installation process, select the Run Git from the Windows Command Link Prompt checkbox so that Git can be run from the command line. Also, select Checkout Windows-style, commit Unix-style line endings, which is the recommended setting.

After the installation is completed, to verify that Git is correctly configured run:



If Git was installed correctly, you should see the installed version number:



**Step 3: Install the rhc Ruby gem**

After Ruby and Git are correctly installed, use the RubyGems package manager (included in Ruby) to install the OpenShift client tools:



RubyGems downloads and installs the **rhc** gem from[www.rubygems.org/gems/rhc](http://www.rubygems.org/gems/rhc).

**2.2.2 Setting up Your Machine**

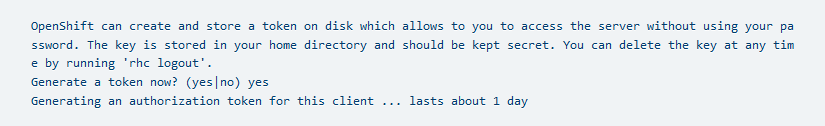
After installation is complete, open a Terminal window and run:



The OpenShift interactive setup wizard displays and prompts you to complete the rest of the process. You will be prompted for your OpenShift username and password:



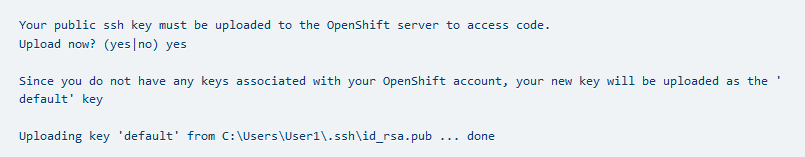
You are then prompted to generate an authorization token. Answering yes stores a token in your home directory to be used on subsequent requests. When it expires, you are prompted for your password again.



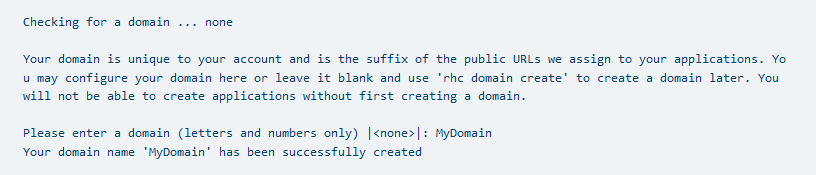
After creating a configuration file, setup will configure SSH keys so that your system can remotely connect to your applications, including deploying your applications using Git:



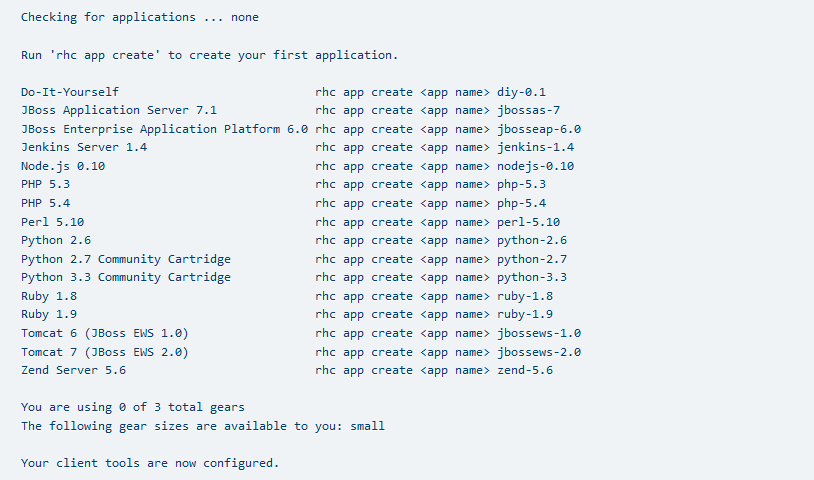
After the new SSH keys are generated, the public key, id\_rsa.pub, must be uploaded to the OpenShift server to authenticate your system to the remote server. Enter a name to use for your key, or leave it blank to use the default name. In the following example the default name is used.



After verifying that Git is installed, you will be asked to set up your domain if you don’t already have one:



Finally, the setup wizard verifies whether any applications exist under your domain. In the example below, no applications have been created. In this case the setup wizard shows the types of applications that can be created with the associated commands. The setup wizard then completes by displaying the current gear consumption along with the gear sizes available to the given user.



You are now ready to create and manage an application using OpenShift.

**2.2.3 Creating an Application**

**2.2.3.1 Using the Web Console**

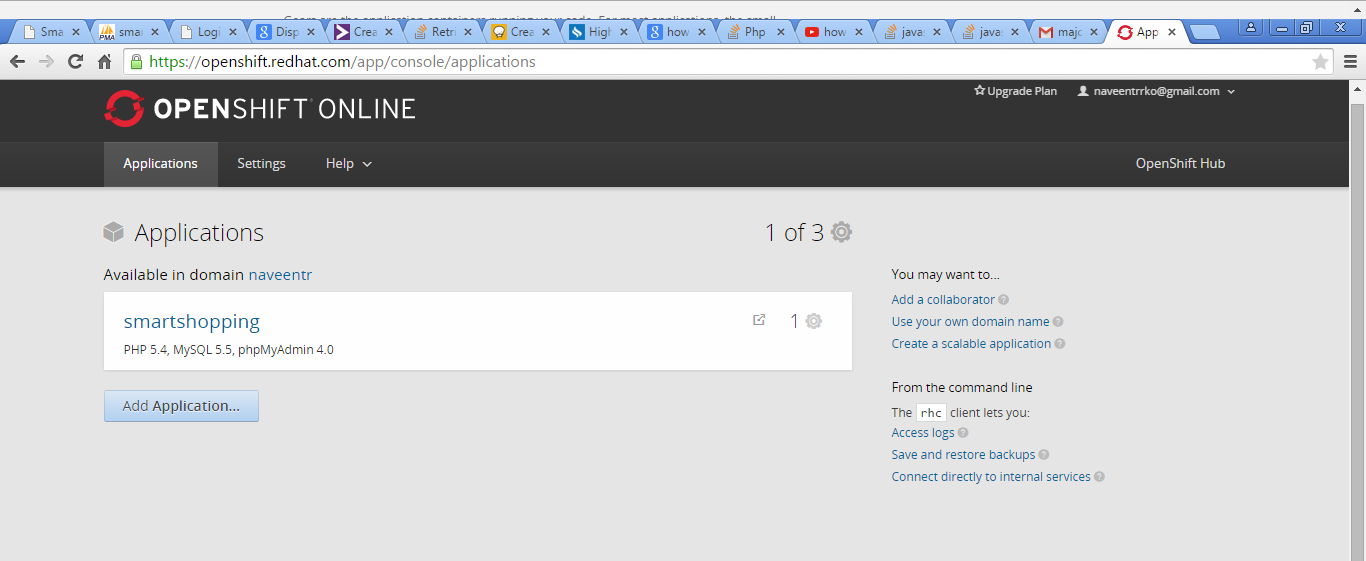


Fig 2.2 OpenShift Home page

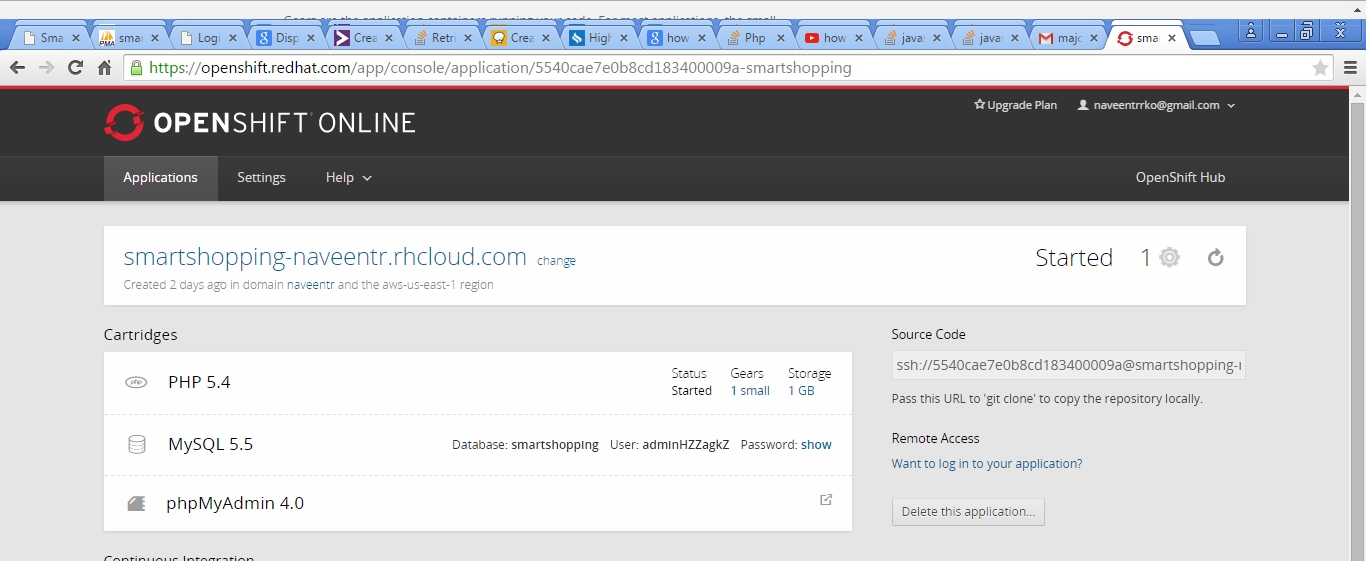


Fig 2.3 Applications deployed on cloud

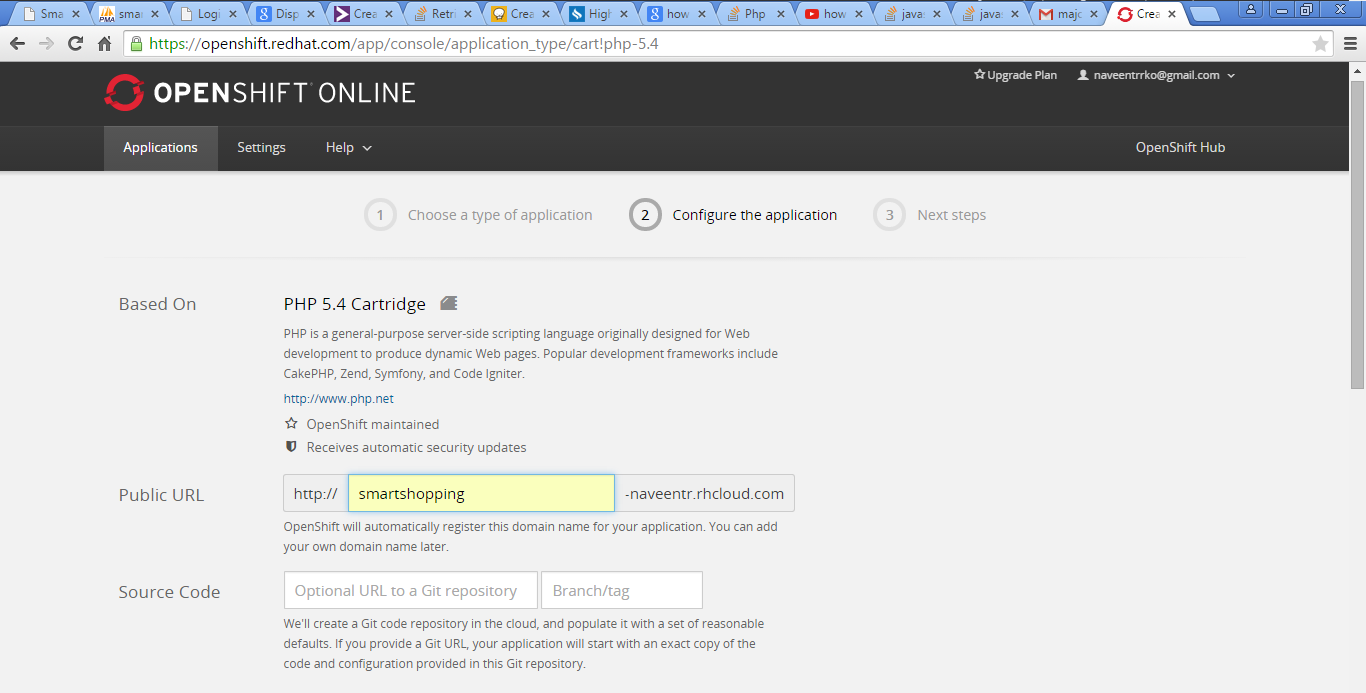


Fig 2.4 Specifying URL to access our application

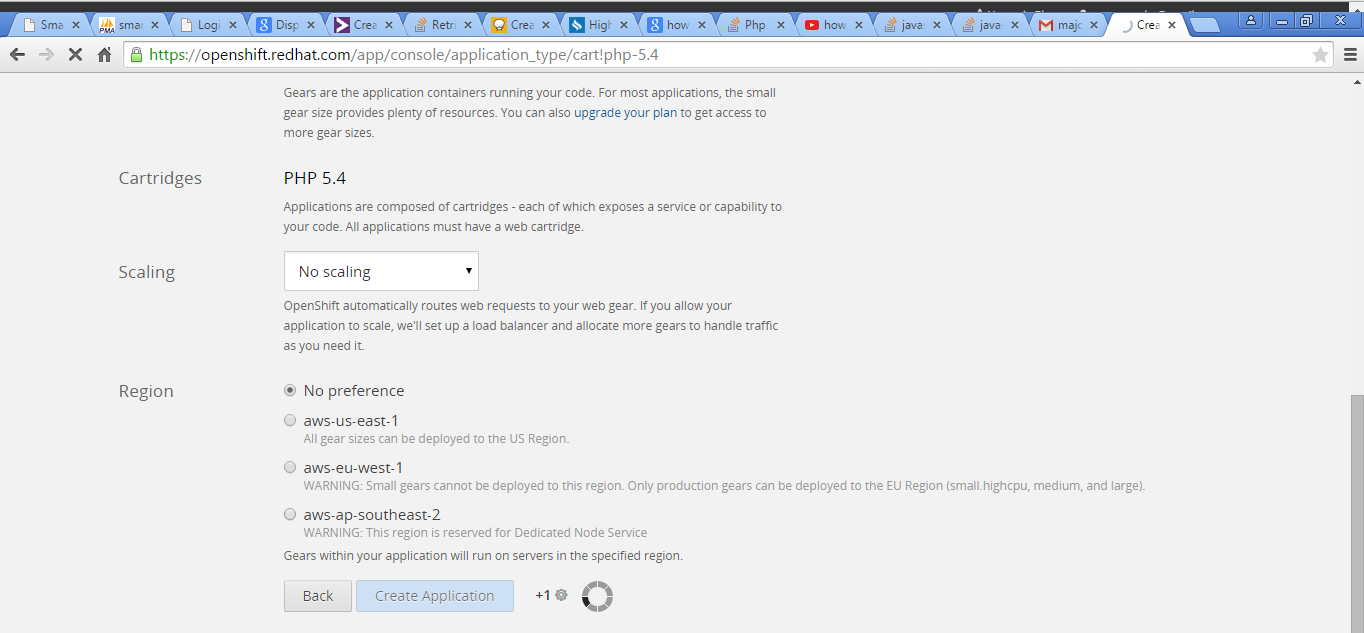


Fig 2.5 Settings required for OpenShift

**2.2.3.2 Making your first change**

OpenShift uses Git to manage application deployments on OpenShift. You make code changes on your local machine, check those changes in locally, and then push those changes to OpenShift.

**2.2.3.3 Making your first change**

Every OpenShift application you create has its own Git repository that only you can access.

**2.2.3.4 Making your first change**

If you create your application from the command line, rhc will automatically download a copy of that repository (Git calls this 'cloning') to your local system.

**2.2.4 Web Console**

If you create an application from the web console, you’ll need to tell Git to clone the repository. Find the Git URL from the application page, and then run:



**2.2.4.1 Remote Access**

OpenShift allows you to remotely access your application’s gears in order to perform various tasks such as troubleshooting, working with log files, and managing environment variables. Once you have created an application and set up your SSH keys (either by rhc setup or manual upload), you can SSH into the remote server using the rhcssh command.

To SSH into a specific application:



You should then connect to your application’s primary gear:



**2.3 HTML**

HTML or Hyper Text Mark-up Language is the standard mark-up language used to create web pages. HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>). HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent empty elements and so are unpaired, for example <img>. The first tag in a pair is the start tag, and the second tag is the end tag (they are also called opening tags and closing tags).

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a mark-up language rather than a programming language.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behaviour of HTML web pages.

Web browsers can also refer to Cascading Style Sheets (CSS) to define the look and layout of text and other material. The W3C, maintainer of both the HTML and the CSS standards, encourages the use of CSS over explicit presentational HTML.

**2.4 CSS**

Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is a cornerstone specification of the web and almost all web pages use CSS style sheets to describe their presentation. CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colours, and fonts.[1] This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for table less web design).

CSS can also allow the same mark-up page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified. However if the author or the reader did not link the document to a specific style sheet the default style of the browser will be applied.

CSS specifies a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities or weights are calculated and assigned to rules, so that the results are predictable.

CSS has a simple syntax and uses a number of English keywords to specify the names of various style properties.

**2.5 JavaScript**

JavaScript (JS) is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document that is displayed. It is also being used in server-side programming, game development and the creation of desktop and mobile applications.

JavaScript is a prototype-based scripting language with dynamic typing and has first-class functions. Its syntax was influenced by C. JavaScript copies many names and naming conventions from Java, but the two languages are otherwise unrelated and have very different semantics.

The application of JavaScript in use outside of web pages—for example, in PDF documents, site-specific browsers, and desktop widgets—is also significant. Newer and faster JavaScript VMs and platforms built upon them (notably Node.js) have also increased the popularity of JavaScript for server-side web applications. On the client side, JavaScript was traditionally implemented as an interpreted language but just-in-time compilation is now performed by recent (post-2012) browsers.

**2.6 MySQL**

The world's second most widely used open-source relational database management system (RDBMS). It is named after co-founder Michael Widenius's daughter, my. The SQL phrase stands for Structured Query Language.

The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other 'AMP' stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL.

MySQL is a relational database management system (RDBMS), and ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records. The official set of MySQL front-end tools, MySQL Workbench is actively developed by Oracle, and is freely available for use.

## 

## CHAPTER 3

**SYSTEM REQUIREMENT SPECIFICATIONS**

## 3.1 Requirements Overview:

## 3.1.1 Application Perspective

## The android application has been designed and developedin such a way that user can view offers on various products at various shopping malls. Hence there may be chances that he might be getting a good discount sale in the shop. Our android app also tracks users location so that whenever they are nearer to shopping malls, they will be pushed with various information about the events occurring at the mall, updated list of offers which are available in that mall. So he gets mood to visit the mall and purchase some products.

## 3.1.2 Application Features

This application includes the following main features:

## User can view offers on various products at various shopping malls.

## User can view various special events on the shopping mall.

## This app tracks user’s location and suggests nearest shopping mall.

## This app can navigate you to the searched shopping mall.

## Search for offers based on shopping mall or shops or products.

## Allows the shop owner to view graphs depicting previous acceptance of his offer.

## Allows the shop owner to view, add or remove offers.

## Allows the shop owner to provide generalised or customized.

## Further the app also captures the shopping habits of a user.

## 3.1.3 User Classes and Characteristics

## A variety of users may use this application for different purposes. An administrator can view all the users who are registered and he also had a privilege to remove a user. As a shop owner can upload the offers that may be either generalized or customized offer pertaining to the particular product.

|  |  |
| --- | --- |
| **USER CLASS** | **CHARACTERISTICS** |
| ADMIN | * View all the registered users. * Remove all the users. * Captures shopping habits of the user. * Can track the user. |
| SHOP OWNER | * Upload generalized offers. * Upload customized offers. * Allows to view graphs. * View uploaded offers. * Allows him to delete offers not needed anymore. |
| USER | * Can view the offers. * Can view the events nearby them. * Provide Feedback |

## Table 3.1 Table Describing User Classes and their Characteristics

## 3.1.4 Operating Environment

## The web application part will be deployed on a cloud platform called OpenShift and the web pages will be developed using HTML/PHP, JavaScript for validation and the Database will be maintained using MySQL and the server will be Apache Tomcat. We salso need an Internet Connection for the successful execution of this application.

## The android version of the phone should be 3.1 or higher along with internet connection via data connection or Wi-Fi. Google maps app to display navigation.

## 3.1.5 Design and Implementation Constraint

## The following are the design and implementation constraints:

## The user must not be in an assumption that he will always get the customized offer, after once the product purchase is made by using our app.

## The overall analysis of the user’s mood may not be 100% correct and at times may lead to ambiguous results in some cases.

## Since the project is currently deployed on a free cloud service provider, therefore the memory space provided is very limited. So the items that are present in the database are limited to a certain number.

## 3.2 System Features

## 3.2.1 Uploading Offers for a product

## 3.2.1.1 Description and Priority

## This system feature should be of high priority. As shop owner uploads offers that may be generalized offer and customized offer. Generalized offer will be given to all the users who use our app and the customized offer will be given for those users who purchase products in that shop by using our app.

## 3.2.1.2 Stimulus and Response

## If shop owner uploads offer on any products then that offer needs to be displayed immediately in the android application.

## 3.2.1.3 Functional Requirements

## If the user is attracted by an offer on the desired product that he was choosing for, then he will definitely buy that product.

## 3.2.2 Searching for a product

## 3.2.2.1 Description and Priority

## This system feature should be of high priority. A user should be able to enter a product name and click on the Search Button. The system analyses the user’s search query and matches it against the available products in the database. In case a corresponding match is found, the system displays the product results that best suit the user. Otherwise if no match is found then, the user is redirected to the no results page where all the available items in the database are also displayed.

## 3.2.2.2 Stimulus and Response The user enters a search keyword and clicks on the product search button. The system checks if the keyword that the user entered exists in the database or not. When a match is found, the system retrieves all products matching the keyword and displays it to the user. The displayed results keep changing every time the user searches for a keyword. In case if the system does not find a match to the entered keyword, it directs the user to the no result page that contains a list of all available products in our database.

## 3.2.2.3 Functional Requirements

## The user should be able to view results of searches of a particular keyword by entering the text and clicking on the search button otherwise an appropriate message should be notified to the user.

## 3.2.3Analysing the Input Query

## 3.2.3.1 Description and Priority

## This system feature should be of high priority and the software should analyze the keyword that the user provides against existing products in the database. When an occurrence is found, the system retrieves all the information it has about the products and displays it to the user. If not, then the system displays an appropriate message to the user.

## 3.2.3.2 Stimulus and Response

## The system should read the input provided by the user and match the input with the existing database products. If a match is found it should display the corresponding information related to the match otherwise display the error message and the available items in the database so that the user can choose something from the database if he/she wants to do so.

## 3.2.3.3 Functional Requirements

## The system should analyse the text entered by the user, find a corresponding match for it in the database and in case a match is found, display the results. In case the match is not found, the system should display appropriate message to the user.

## 3.3 Non-Functional Requirements

## These are requirements that are not functional in nature, that is, these are constraints within which the system must work.

## 3.3.1 Performance Requirements

## The Overall performance should be reliable and should enable the searches to work efficiently.

## User interface created will have a direct impact on performance, a simple and an easy to use interface will result in a better performance.

## User must be able to make the searches by a quick and efficient manner. Also database server access should be quick enough for update and retrieval processes.

## 3.3.2 Safety Requirements

## Since the database is shared among different administrators, the access to the database of the software should be secure. We can maintain a back-up of the database so that in case of any crash, the back-up copy can be used.

## 3.3.3 Security Requirements

## Only authenticated user is allowed to login and make changes to the database.

## There will be proper security regarding the access of data.

## Database security is the most important security requirement. Any unauthorized access to the database may corrupt the database and may lead to irrevocable loss of data

## 3.3.4 Software Quality Attributes

* **Capacity, Scalability and availability:**
* The System shall achieve high availability at all times.
* The system shall be scalable to support additional clients.
* **Maintainability.**

The system should be optimized for supportability, or ease of maintenance as far as possible. This may be achieved through the use documentation of coding standards, naming conventions, class libraries and abstraction.

## 3.4 User Requirements

## The admin should ensure proper registration process by authenticating himself.

## The software results page access should be instantaneous for the active users.

## 3.5 Software Requirements

## 3.5.1 Software Requirements

* Cloud: Open Shift Cloud Platform.
* Database: MySQL.
* Server: Apache Tomcat.
* Browsers: Web Browsers such as Safari, Mozilla Firefox, Google Chrome etc.
* Android version: 3.1 or higher.

**3.5.2 Hardware Requirements**

* Processor: Intel Core i3
* RAM: 1GB
* Hard Disk: 20GB
* Monitor : 15 VGA Colour
* Mouse : Logitech
* Mobile phone : Android device( version of 4.4.2v)

**CHAPTER 4**

**SYSTEM ANALYSIS**

**4.1 Existing System**

**Fast Mall** is an American [application software](http://en.wikipedia.org/wiki/Application_software) for [mobile devices](http://en.wikipedia.org/wiki/Mobile_device) that provides [interactive](http://en.wikipedia.org/wiki/Interactive) [shopping-mall](http://en.wikipedia.org/wiki/Shopping_mall) maps that guide users with [turn-by-turn directions](http://en.wikipedia.org/wiki/Turn-by-turn_navigation) for various places in shopping mall.

Drawbacks in fast mall application:

* App does not display any specific offers available for various products in shopping mall.
* App does not use app user’s click pattern for analysis.
* App does not track location of user.

**4.2 Proposed System**

Through this project we are developing an android application named “Smart Shopping Assister”. The main objective of app is to guide the user in purchasing various products in a shopping mall by providing valuable information related to various offers in various shops. Further the app also captures the shopping habits of the user.

The app detects whether the customer is near a shopping mall by comparing the GPS coordinates with the database. This mechanism is used so that we do not flood customer’s phone with data at unnecessary time. When GPS coordinates matches, based on customer interests, which are previously captured, offers in various shops of the corresponding shopping mall will be pushed to the app by the server.

Server is the one that has logic for performing data analytics on captured data. Server performs these analytics on periodic basis.

One important aspect that needs to be focused on is how to capture data? To capture data the app uses users search pattern and also the app generates various periodic survey forms and ads. To ensure that he attends these surveys, various offers are provided in the form of promo code, which can be used to claim offer at shops. Not only these, but other mechanisms are used to capture user’s habits and interests without causing any inconvenience to the user.

**4.3 Feasibility Analysis**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimate. During system analysis the feasibility study of the proposed system is to be carried out. That is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirement for the system is essential.

Three key consideration involved in the feasibility analysis are

* Economic feasibility
* Technical feasibility
* Operational feasibility

**4.3.1 Economic feasibility**

The study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system is well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

**4.3.2 Technical Feasibility**

The study is carried out to check the technical feasibility, that is, the technical requirements of the system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resource. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**4.3.3 Operational Feasibility**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and make the user familiar with it.

**CHAPTER 5**

**SYSTEM DESIGN**

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design can be defined as the application of systems theory to product development.

**5.1 Design Specific Activities**

Design is the creation of a plan or convention for the construction of an object or a system. Design is a roadmap or a strategic approach for someone to achieve a unique expectation. It defines the specifications, plans, parameters, costs, activities, processes and how and what to do within legal, political, social, environmental, safety and economic constraints in achieving that objective.

**5.2 Activity Diagrams**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows). Activity diagrams show the overall flow of control.

Activity diagrams are constructed from a limited number of shapes, connected with arrows. The most important shape types:

* Rounded rectangles represent actions.
* Diamonds represent decisions.
* Bars represent the start (split) or end (join) of concurrent activities.
* A black circle represents the start (initial state) of the workflow.
* An encircled black circle represents the end (final state).

**5.2.1 Login**

|  |  |
| --- | --- |
| **ADMIN** | **SYSTEM** |
| Click on admin login button  Enter username and password | Display login page  Redirect to admin home page  Authenticated  Verify Username and Password  No  YES |

**Fig 5.1: Login Activity Diagram**

The Fig 5.1 refers to the activity diagram that describes when the administrator logs into the system using their login credentials. When user clicks on the login button a page will be displayed in which user can enter his login credentials like user name and password, which will be verified and authenticated. After authenticating user will be redirected to the admin home page.

**5.2.2 Offers provided by shop owner**

|  |  |
| --- | --- |
| SHOP OWNER | SYSTEM |
| Select type and enter username and password | Display the home page containing generalized and customized tab    Generalized offer    Display generalized offer tab  Display Customized offer tab  **Yes**  **No** |

**Fig 5.2: Offers given for product**

The Fig 5.2 refers to the activity diagram that describes as and when the shop owner logins and he can be able to upload the offers that may be either generalized offers and customized offers for the products.

**5.2.3 Search for a product**

|  |  |
| --- | --- |
| USER | SYSTEM |
| Enter product name and click on search button | Display not found result page  Display result page  Product found?  Search for a product in a database  No  Yes |

**Fig 5.3: Search for a product**

The Fig 5.3 refers to the activity diagram that describes when the user searches for a product. If the product is found in the database then the user is redirected to the corresponding results page otherwise the user is redirected to a page in which all the products that are listed in the database are listed.

**5.2.4 Analyzing the habit of user**

|  |  |
| --- | --- |
| USER | SYSTEM |
| Enter the product to be searched | Display habit page  Calculate overall habit based on previous and current search  Display the habit associated with product  Is this the user's first search?  Product found  Display no result page  Search for a product in the database  No  Yes  Yes  No |

**Fig 5.4: Analyzing the User's habit Activity Diagram**

The Fig 5.4 refers to the activity diagram that describes the sequence of steps involved in analyzing the habit of the user. The system takes a cumulative average of all the previous habits of the user to determine his current habit.

**5.2.5 Adding new offer**

|  |  |
| --- | --- |
| **SHOP OWNER** | **SYSTEM** |
| Enter product and offers details and click on submit button  Click on Add new offer button | Display successful message  Verify product and offer details and add to database  Redirect to Add new offer page |

**Fig 5.5: Adding new offer to Database Activity Diagram**

The Fig 5.5 refers to the activity diagram that describes the sequence of steps involved in adding a offer into the database. When shop owner clicks on the add new offer button, he will be redirected to the respective page in which he can enter product details which will be stored in our database.

**5.2.6 Tracking user location**

|  |  |
| --- | --- |
| while app is running in background | While app is running in foreground |
| Near to any mall  Location changed  Notify users  Listen for location update  Service  No  No  Yes  Yes | Location changed  Notify users  Near to any mall  Listen for location update  No  Yes No  Yes |

Fig 5.6: Tracking user location

The Fig 5.6 refers to the activity diagram that shows the sequence of steps in tracking user location in both the situations i.e., while app is not running in background and while app is running in background.

**5.3 Architectural Design**

****

Fig 5.7 Architectural Design

**Important Entities of Our Project**

* **Cloud storage**

Since the app is expected to be used by millions of customers who visit shopping malls, the project needs a storage which is highly scalable and easily accessible from anywhere, hence we make use of cloud storage to store data pertaining to individuals and also the product details provided by brands.

* **Server**

Server is the one that contains the logic to perform data analytics on the available data. Further server is used to pull and push data from and to the user. Here data refers to interests of user and customized offers. Servers performs analytics on periodic basis.

* **Database of Shopping malls Location**

This database is used to store the GPS co-ordinates of the various shopping malls in the city, which is used to compare with users location to identify whether he is near a shopping mall or not**.**

* **Input Agents**

The main input agents for the application are brands of product, who provide details of how much quantity of products are distributed among the various shops in the shopping mall, and other info about products. The shops in the malls who uploads the bills of the users. Both above mentioned agents uses webpage developed through this project to insert their data into cloud. Social networking sites also provides valuable information about the contacts and friends list of the users.

**5.4 Design Principles**

Software design principles represent a set of guidelines that helps us to avoid having a bad

design. We have taken care that the design principles with respect to object-oriented methodology such as literate programming, stepwise refinement, encapsulation and modularization are all met in our project.

**5.4.1 High Level Design**

The high level design defines the project level architecture of the system. This process includes the decomposition of system requirements into alternative project architectures and

then the evaluation of these project architectures for optimum performance, functionality, cost, and other issues [technical and non-technical]. In HLD, we design the component level

abstractions and interactions.

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well. The following Use-Case diagram describes the high-level design concept of our project.

This application consists of three modules. Their use case diagram is as follows:

1. App user.
2. Store user.
3. Administrator.

**5.4.1.1 App user**

App user

F**ig 5.8 Use case diagram for app user**

In the first case scenario, the app user is the actor. The app user can perform actions such as login, logout, view special events, view offers, search for offers and get directions to

mall. Apart from that the App user can also search for a product and should also be able to

open and close the application.

**5.4.1.2 Store users**

Store user

**Fig 5.9 Use case diagram for store user**

In the second case, the actor is the store user. The store user is supposed to perform various functions such as login, logout, view offers, add offers and remove offers for an displayed product. Apart from that the store user can also search for a product and should also be able to open and close the browser.

**5.4.1.3 Administrator**

Admin

**Fig 5.10 Use case diagram for administrator user**

In the third case, the actor is the administrator. The administrator is supposed to perform various functions such as login, logout, view store owners, view app users, remove shop owner and remove app users. The admin also has function to add special events taking place in particular mall.

**CHAPTER 6**

**SYSTEM IMPLEMENTATION**

Implementation is the process when the theoretical design is turned into a working system. Thus it can be considered to be the most critical stage in achieving a successful new and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

**6.1 Actual Implementation**

**6.1.1 Introduction of Android:**

Android is an operating system based on the Linux kernel with a user interface based on direct manipulation, designed primarily for touch screen mobile devices such as smart phones and tablet computers, using touch inputs, that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, and a virtual keyboard. Despite being primarily designed for touch screen input, it also has been used in televisions, games consoles, digital cameras, and other electronics.



**Fig 6.1 Architecture of Android**

* Applications (written in java, executing in Dalvik)
* Framework services and libraries (written mostly in java)
* Applications and most framework code executes in a virtual machine
* Native libraries, daemons and services (written in C or C++)
* The Linux kernel, which includes drivers for hardware, networking, file system access and inter-process-communication.

**6.2 Basic Classes of Android:**

**6.2.1 Activity class:**

An activity is a single, focused thing that the user can do. Almost all activities interact with the user, so the Activity class takes care of creating a window for user in which user can place required UI using setContentView (View) method. While activities are often presented to the user as full-screen windows, they can also be used embedded inside of another activity (using ActivityGroup). There are some methods almost all subclasses of Activity will implement:

* onCreate(Bundle)and onStart()- is where you initialize your activity.
* onDestroy()and onStop()- is where you deal with the user leaving your activity.

**6.2.2 Service class:**

A Service is an application component that can perform long-running operations in the background and does not provide a user interface. Another application component can start a service and it will continue to run in the background even if the user switches to another application. Additionally, a component can bind to a service to interact with it and even perform inter process communication (IPC). For example, a service might handle network transactions, play music, perform file I/O, or interact with a content provider, all from the background. Some important methods of Service are:

* + startService()-A service is "started" when an application component (such as an activity) starts it by calling startService(). Once started, a service can run in the background indefinitely, even if the component that started it is destroyed.
  + bindService()-A service is "bound" when an application component binds to it by calling bindService(). A bound service offers a client-server interface that allows components to interact with the service, send requests, get results, and even do so across processes with interprocess communication

**6.2.3 Broadcast Receivers**

Broadcast Receivers simply respond to broadcast messages from other applications or from the system itself. These messages are sometime called events or intents. For example, applications can also initiate broadcasts to let other applications know that some data has been downloaded to the device and is available for them to use, so this is broadcast receiver who will intercept this communication and will initiate appropriate action. Method that all subclasses of Broadcast receiver will override is:

* onReceive(Context context, Intent intent)-This method is called when the Broadcast Receiver is receiving an Intent broadcast. During this time user can use the other methods on Broadcast Receiver to view/modify the current result values.

**6.2.4 SQLiteOpenHelper**

A helper class to manage database creation and version management. You create a subclass implementing onCreate(SQLiteDatabase),onUpgrade(SQLiteDatabase,int,int) and optionally onOpen(SQLiteDatabase) and this class takes care of opening the database if it exists, creating it if it does not, and upgrading it if necessary. Transactions are used to make sure the database is always in a sensible state.

**6.2.5 Content Providers**

Content providers manage access to a structured set of data. They encapsulate the data, and provide mechanisms for defining data security. Content providers are the standard interface that connects data in one process with code running in another process. When you want to access data in a content provider, you use the ContentResolver object in your application's Context to communicate with the provider as a client. The ContentResolver object communicates with the provider object, an instance of a class that implements Content Provider. The provider object receives data requests from clients, performs the requested action, and returns the results.

**6.2.6 AsyncTask**

AsyncTask enables proper and easy use of the UI thread. This class allows to perform background operations and publish results on the UI thread without having to manipulate threads and/or handlers. AsyncTasks should ideally be used for short operations. Method mentioned below is invoked by calling activity using execute().

* doInBackground(Params...)-This step is used to perform background computation that can take a long time. The parameters of the asynchronous task are passed to this step. The result of the computation must be returned by this step and will be passed back to the last step.

# CHAPTER 7

# TESTING AND RESULTS

# Software testing is a critical element of software quality assurance and represents the ultimate review of specifications design and coding .The testing phase involves the testing of system using various test data. Presentation of data plays a vital in the system testing. After preparation of the test data the system under study is tested.

# The test data and errors found are corrected by the following test steps and are recorded for future reference. Thus a series of testing is performed on the system before it is ready for implementation.

# Testing Methodologies

# The various types of testing on the system are as follow:

# Testing Objectives

# Test cases

# Unit Testing

# Integration Testing

# Acceptance Testing

# 7.1 Testing Objectives

# Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as yet undiscovered error. A successful test is one that uncovers an as yet undiscovered error.

# Testing should systematically uncover different classes of errors in a minimum amount of time and with a minimum amount of effort. A secondary benefit of testing is that it demonstrates that the software appears to be working as stated in the specifications. The data collected through testing can also provide an indication of the software's reliability and quality. But, testing cannot show the absence of defect -- it can only show that software defects are present.

# 7.2 Test Cases

# A test case in software engineering is a set of conditions or variables under which a tester will determine whether an application or software system is working correctly. The mechanism for determining whether a software program or system has passed or failed such a test is known as a test oracle. In some settings, an oracle could be a requirement or use case, while in others it could be a heuristic. It may take many test cases to determine that a software program or system is considered sufficiently scrutinized to be released. Test cases are often referred to as test scripts, particularly when written. Written test cases are usually collected into test suites.

# 7.3 Unit Testing

# Unit testing focuses on verification effort on the smallest unit of software design module. Using the unit test plan, prepares in the design phase of the system as a guide, important control path are tested to uncover errors within the boundary of the module. The interfaces of the each of the modules under considerations, boundary conditions were checked all independent paths were exercised to insure that all statements in the module are executed at least once and all error handling paths were tested. Each unit was thoroughly tested to check if it might fall in any possible situations. This testing was carried out during the programming itself. At the end this testing phase each unit was found to be working satisfactorily as regarded to expected output from the modules.

# 7.4 Integration Testing

# Data can be lost across the interface; one module can have adverse on another’s sub function. When such modules are combined, it may not produce desired major function; global data structure can also present problems. Integration testing is a symmetric technology for constructing test to uncover error associated with the interface. All modules are combined in this step. Then the entire program is tested as a whole.

# 7.5 Acceptance Testing

# Acceptance of test of a system is the factor for the system. The requirements under consideration are listed for user acceptance by keeping in constant touch with the perspective user of the system at the time of design. First each module was tested independently to find out how it works. Later integration of all modules was done and tested.

# 7.6 Test Cases and Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **TEST CASES** | **ACTIONS** | **RESULT EXPECTED** | **STATUS** |
| **1** | Admin Login | 1. Don’t enter the user name and password.  2. Click on the SUBMIT button. | System should not take any action ”please fill out this field“ should appear. | OK |
| **2** | Admin Login | 1. Don’t specify the user name and specify the password and 2. Click SUBMIT button. | System should not take any action ”please fill out this field“ should appear. | OK |
| **3** | Admin Login | 1. Enter the valid user name and password.  2. Click the SUBMIT button. | System should direct user to the page where admin can perform his actions. | OK |
| **4** | User Sign Up | 1. Enter the Sign Up details.  2. Click the SUBMIT button. | System should direct to login page. | OK |
| **5** | Add Generalized Offers | Enter the Offer details and click the SUBMIT button. | System should direct to  Generalized Offers Page. | OK |

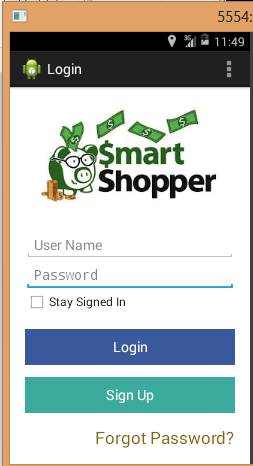
Table 7.1 Test cases and results

# CHAPTER 8

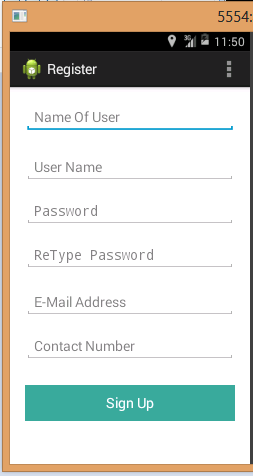
**SNAPSHOTS**

****

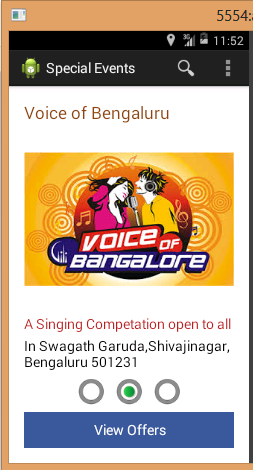
**Fig 8.1 Android home page**

****

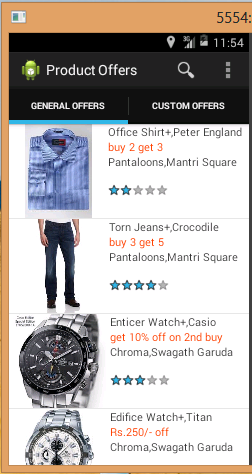
**Fig 8.2 Android login page**

****

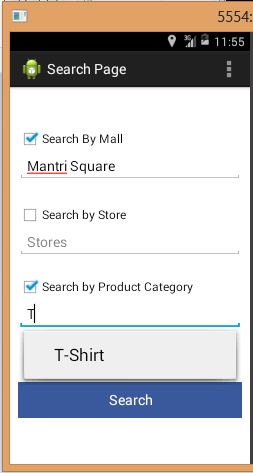
**Fig 8.3 Android register page**

****

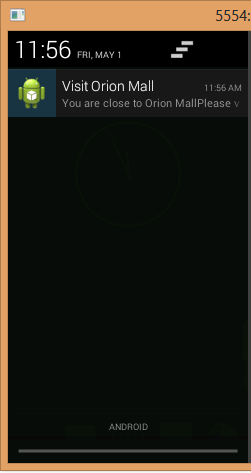
**Fig 8.4 Android special event page**

****

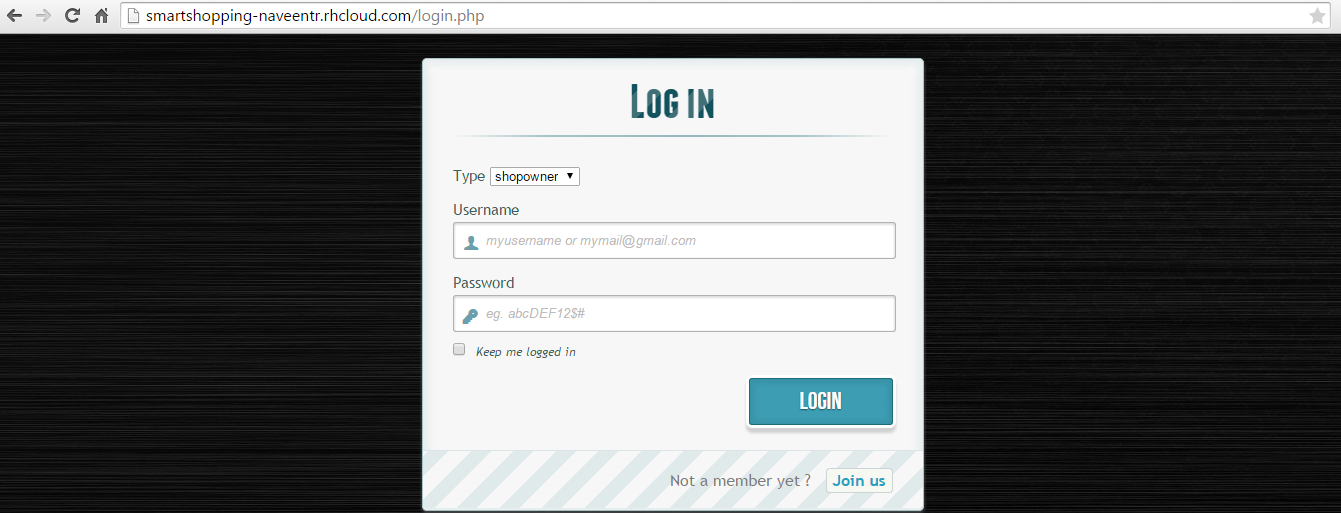
**Fig 8.5 Android product offer page**

****

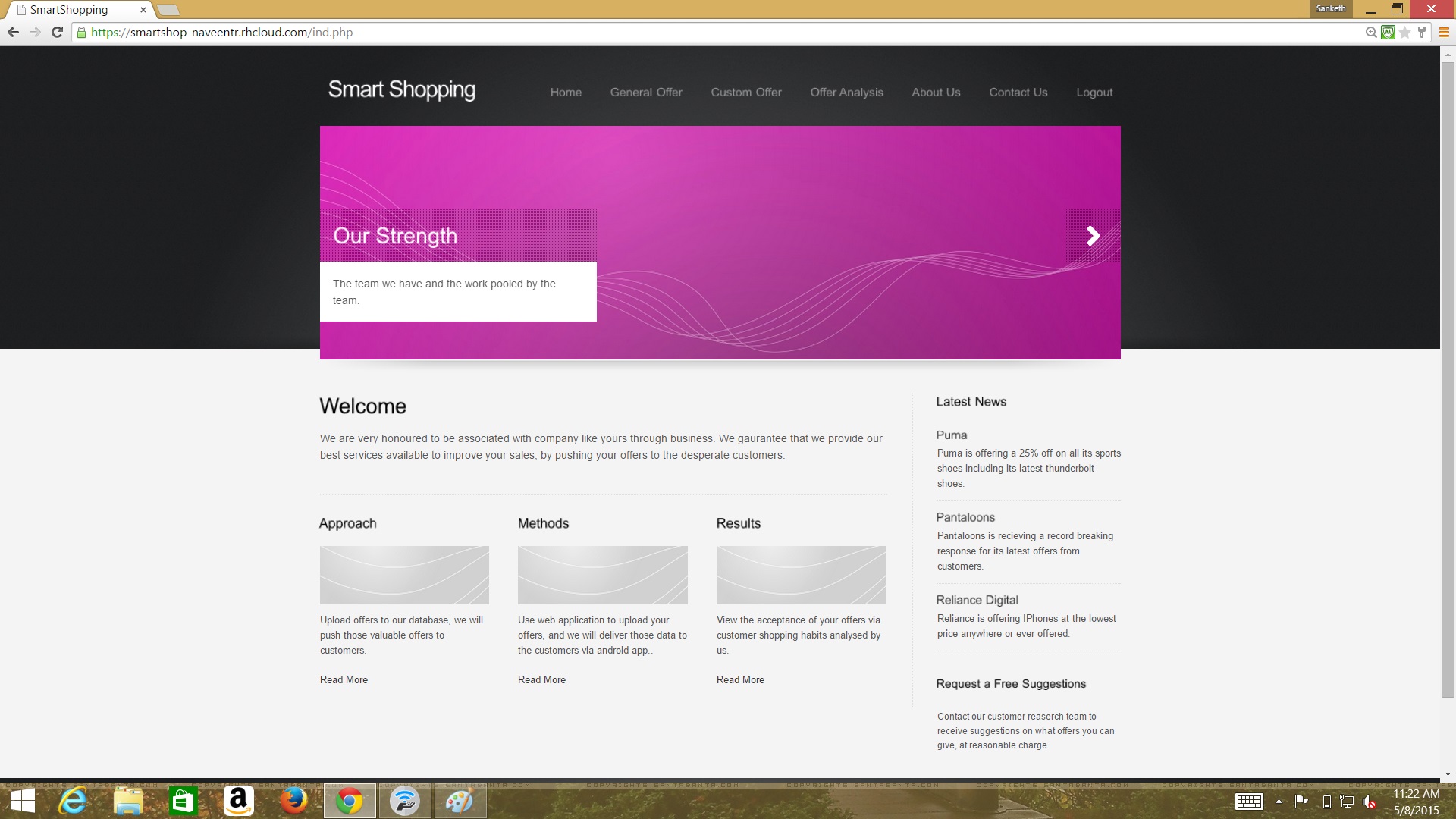
**Fig 8.6 Android search page**

****

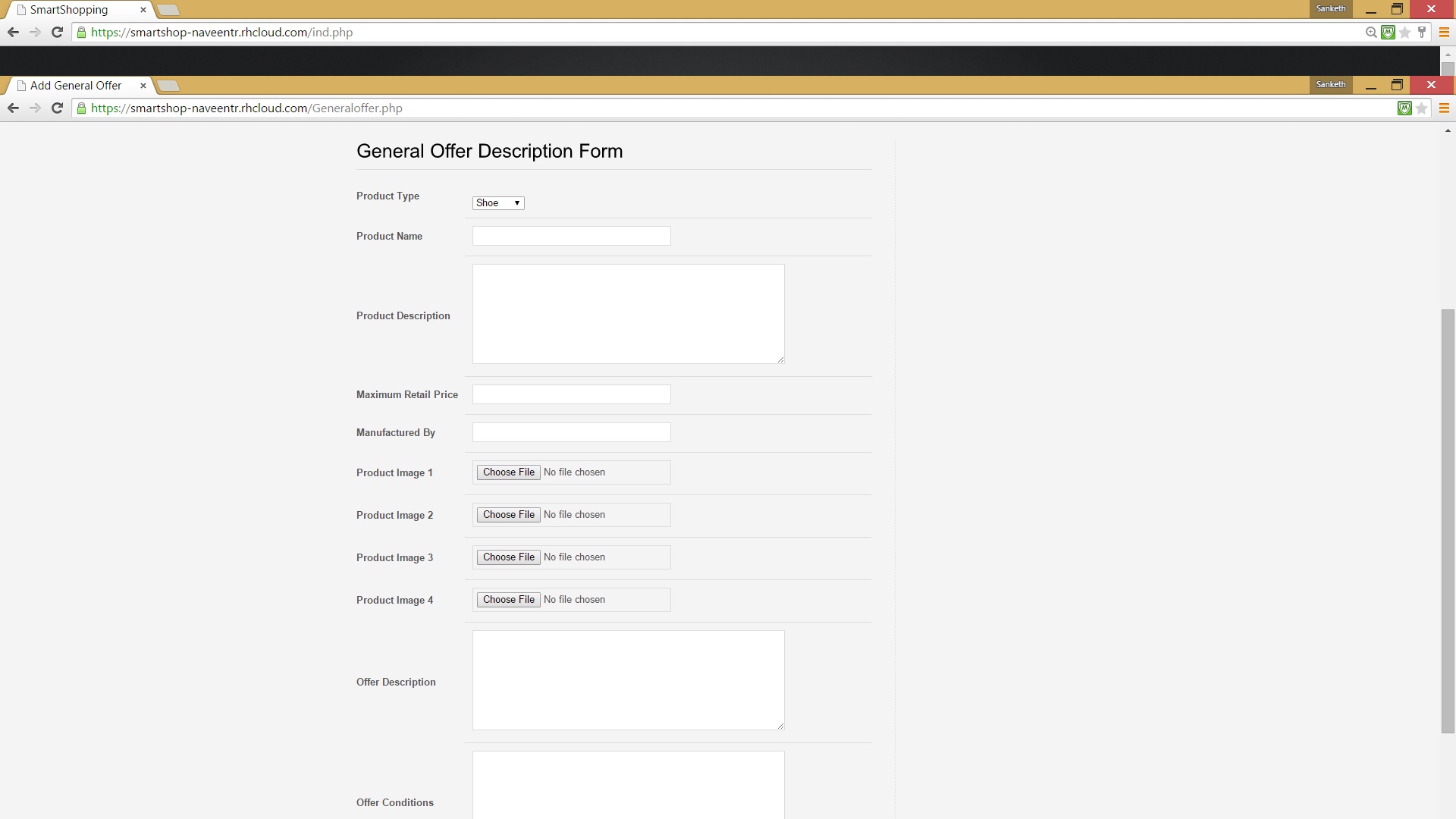
**Fig 8.7 Android message page**

****

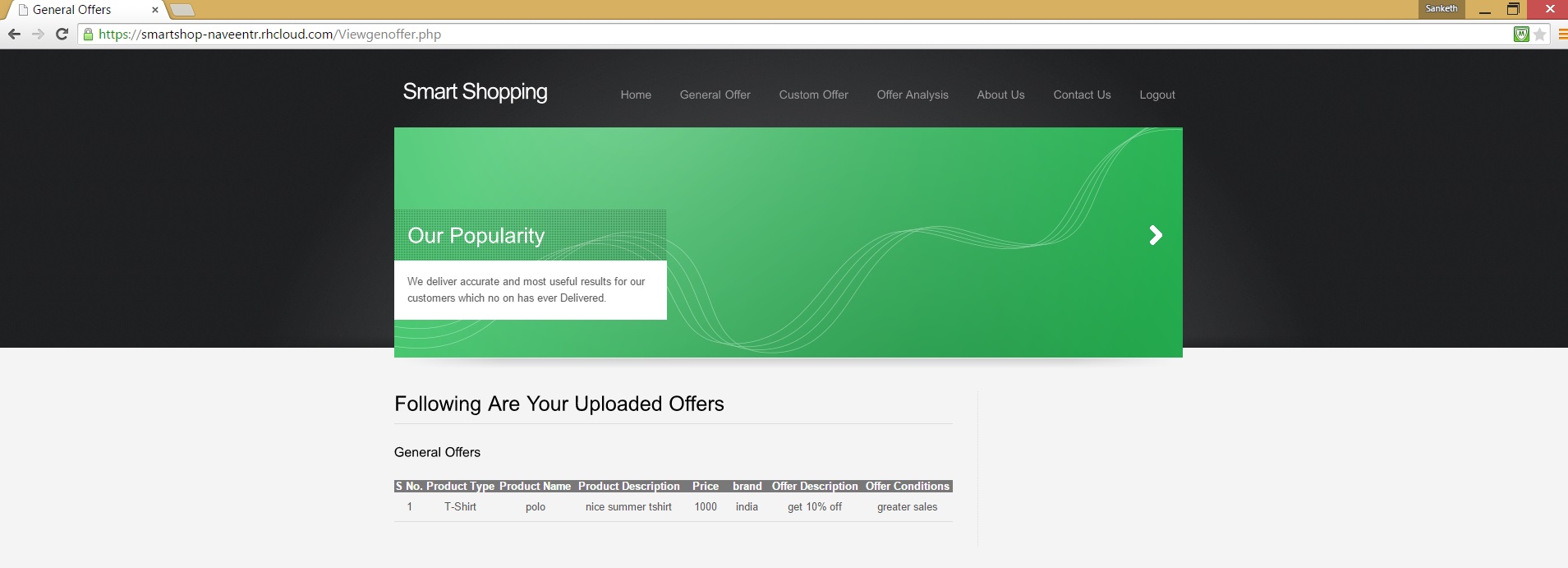
**Fig 8.8 Web part login page**

****

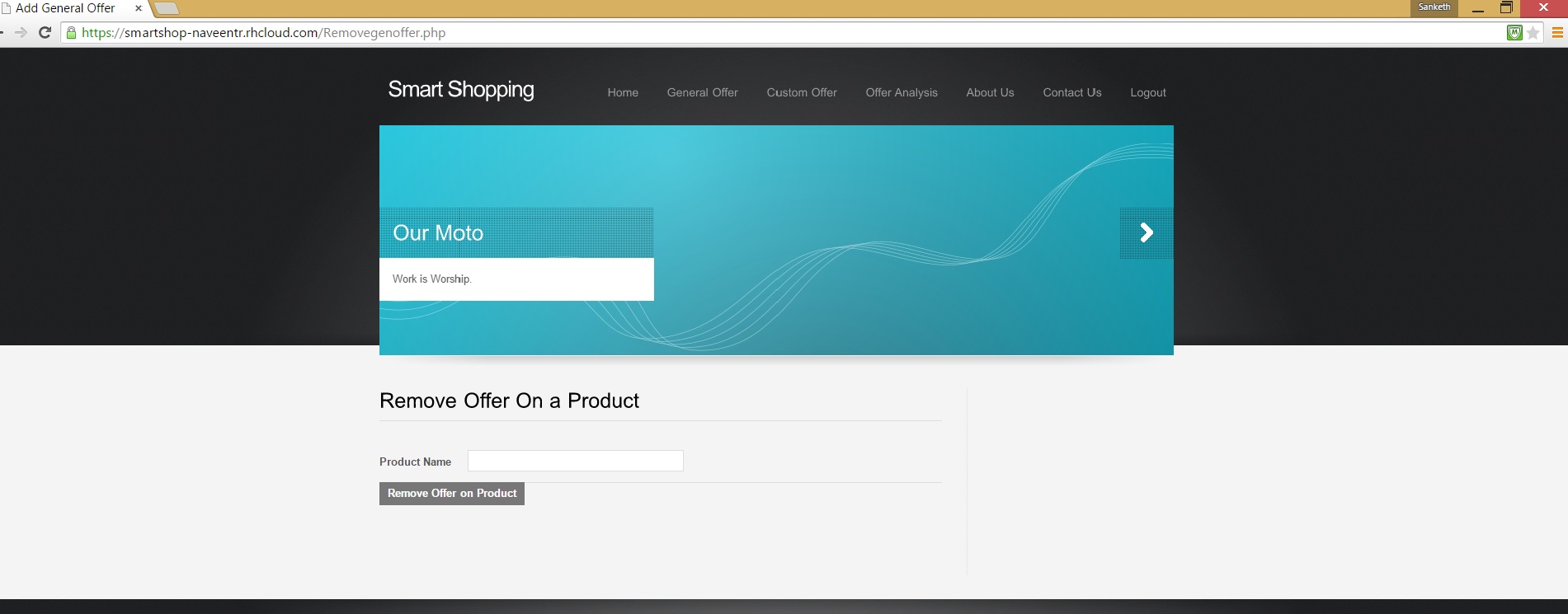
**Fig 8.9 Shop owner home page**

****

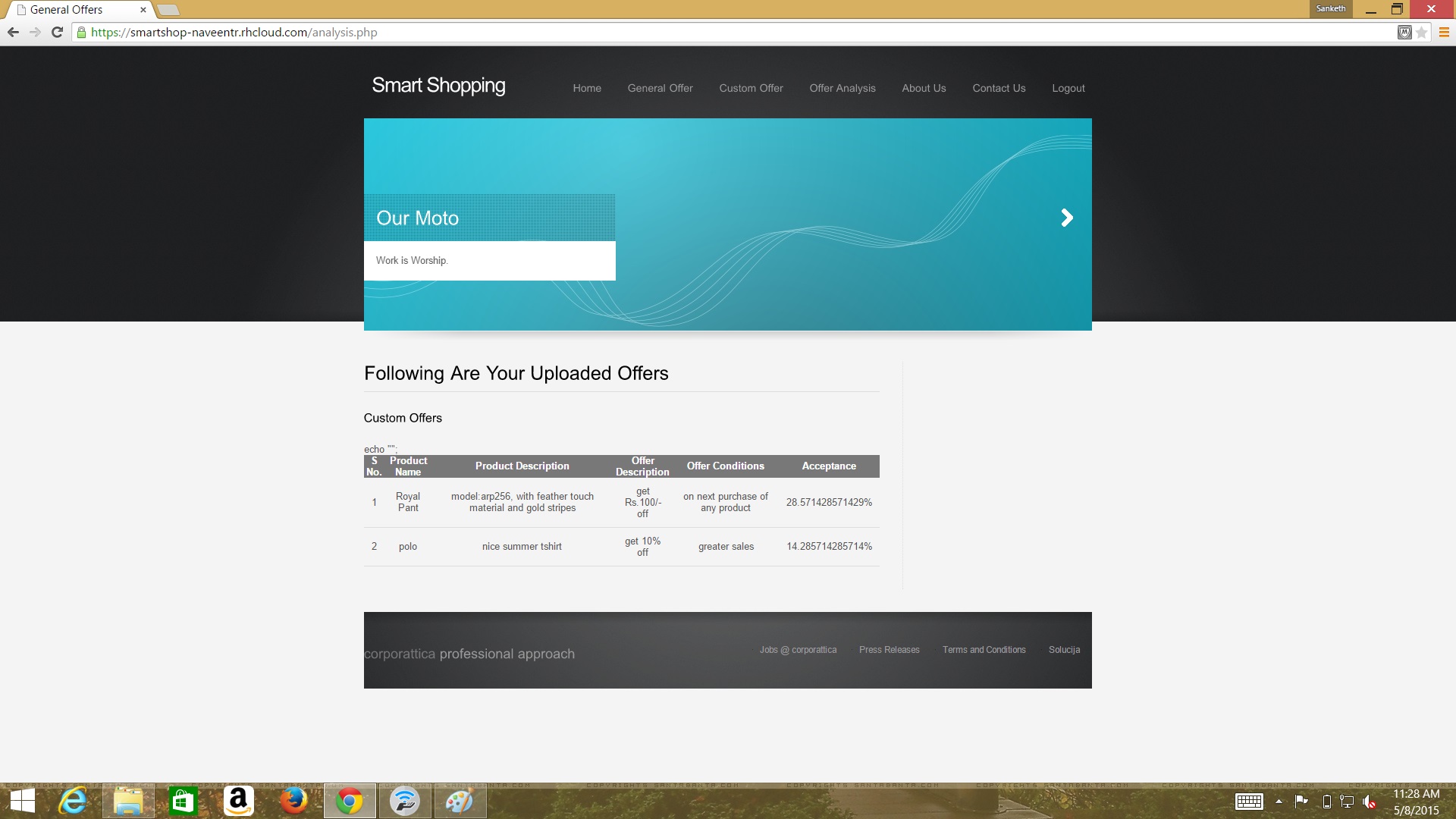
**Fig 8.10 Shop owner uploading offers page**

****

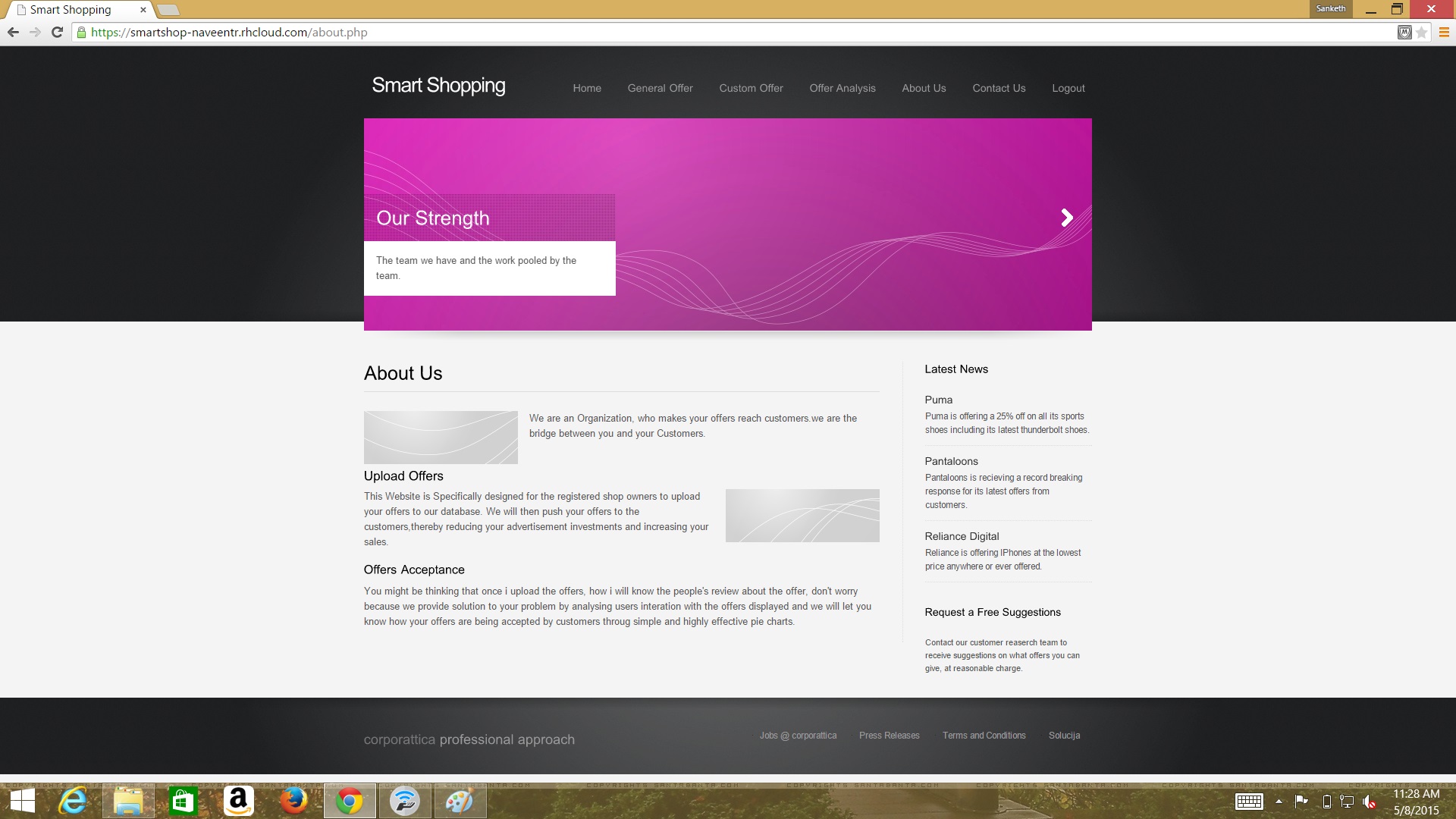
**Fig 8.11 Displaying all uploaded generalized offers page**

****

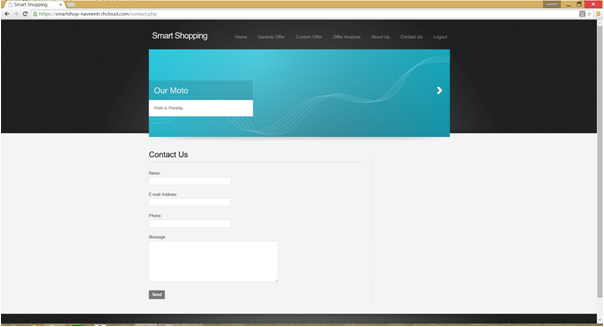
**Fig 8.12 Shop owner removing an uploaded offer page**

****

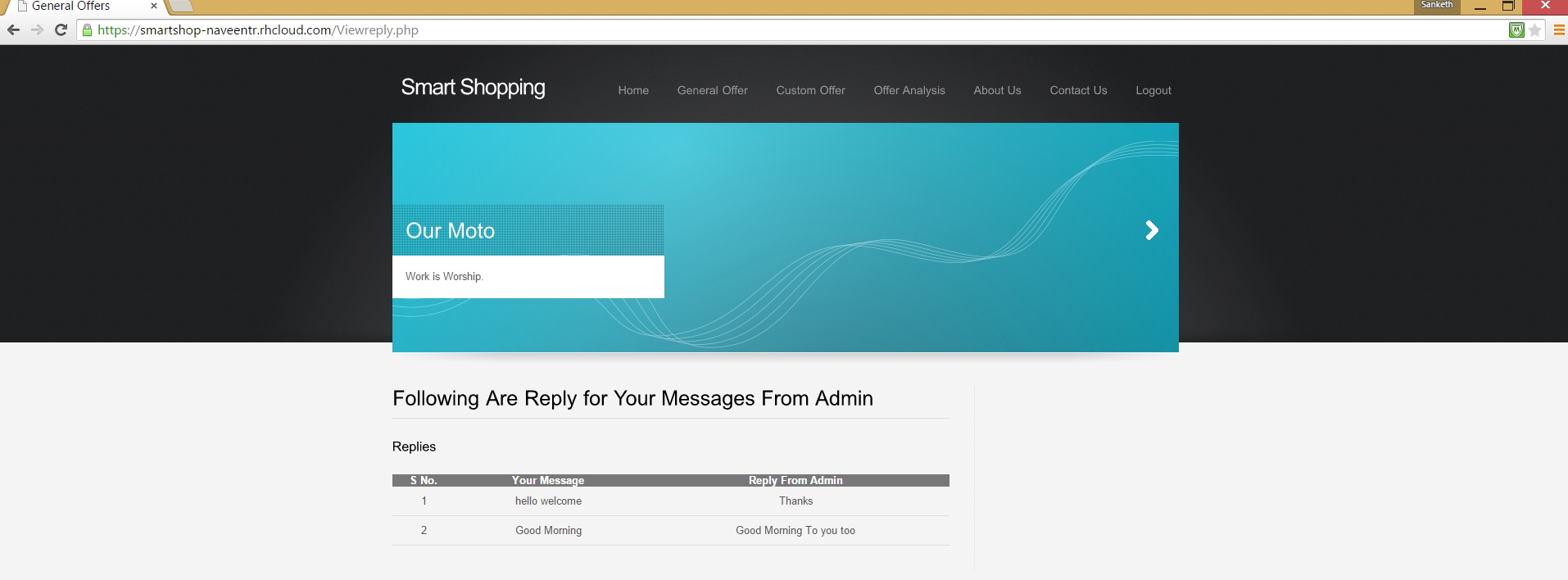
**Fig 8.13 Displaying all uploaded Customized offers page**

****

**Fig 8.14 About us page**

****

**Fig 8.15 Feedback page**

****

**Fig 8.16 Displaying all replied messages**

**Chapter 8**

**Conclusion and Future Enhancements**

**8.1 Conclusion**

Through this project we have tried to develop an android application, which is going to be beneficial for the shops in that shopping mall. This project is unique since it focuses on making the application extremely user-centric in nature. The main aim of the project is to provide various offers to the users pertaining to various shopping malls. One of the major advantage of the android app is that, it will be tracking app user's location and whenever user is nearer to any shopping mall, app will be pushed with various offers of products available in that shopping mall. Another advantage is that, whenever user clicks on any product details it will be recorded and analysed so that when user uses app for the next time he will be pushed with details of the products that user is interested in rather than pushing all the offers. Using web application shop owner can get the information in terms of percentage showing how customers are reacting to the various provided by shop owner.

**8.2 Future Enhancements**

In our project we are concerned about displaying offers present in shopping malls and we would like to improve it by focusing on small shops and display the offers provided by them. In order to improve our application we can focus on capturing habits of the User and also the shopkeepers. We would like to enhance our project by displaying various shopping malls near to his/her current location for the App users. We would like to include Shop information like Shop names, Categories, locations, description and Floor Layout in the mobile application.

**Chapter 9**

**BIBLOGRAPHY**

1. http://www.w3schools.com

2. http://www.stackoverflow.com

3. http://www.developer.android .com

4. http://www.OpenShift.com

5. http://www.tutorialspoint.com

6. http://www.cloudacademy.com