**Mobile Management System**



**Ritesh Shrestha**

**00172893**

**Computing Project**

**Level 5 Diploma in Computing**

**Email :** [riteshsht@gmail.com](mailto:riteshsht@gmail.com)

**Softwarica College of IT & E-Commerce**

***Kathmandu, Nepal***

04/14/2019

**Submitted to**: Kiran Rana

**Abstract**

In this current business environment, it’s very important to be able to respond as per client needs in most effective manner. This project allows customer to easily view all the available products but only registered users are allowed to purchase their desire products instantly using online payment Esewa or can also order by using pay on delivery option. Customer can also choose delivery options from normal delivery and fast delivery. This project enables Administrators to easily view various orders placed by the customers. They can manage customers and orders as well. Administrator can also manage user account of customers and editors.

To develop this e-commerce website, various things should be studied and understood. This include server and client-side scripting/coding techniques, programming language such as PHP and creating databases using MySQL. The objective of this project is to develop a website where customer have features of shopping cart. Shopping cart is virtual cart where customers stores their desired products. After order or payment of the product email is sent to customer about information of the product. Admin have various functions such as counters to show total sales, products, new orders ets. Admin can chat with other admins or editor easily. Admin dashboard consists of sales diagram along with various data such as lastest members, latest orders and recently added products by admin.

**Acknowledgement**

Firstly, I would like to thank Mr. Kiran Rana, my first supervisior for his extraordinary support in the project. His guidance, suggestions, constructive comments on project development process, requirement specification and so forth throughout the entire semester has been tremendously helpful. The advices he provided us made it easier for us to perform our tasks. He also provides a lot of feedback to me during the report writing which increase the overall quality of this project. In addition, I would also like to thank Softwarica College of IT & E-commerce for providing this platform to work on the Project and develop skills in current hype in technologies. I am also thankful to the college where quality learning was made possible. Finally I would like to thank my parents who were always there for me giving encouragement and guidance, and my housemates for providing me with food during long coding sessions!

Table of Contents

[Chapter 1: 11](#_Toc8087625)

[Introduction: 11](#_Toc8087626)

[Project Background: 11](#_Toc8087627)

[Problem Statement: 11](#_Toc8087628)

[Aims: 12](#_Toc8087629)

[Objectives: 12](#_Toc8087630)

[Overview of project Design 12](#_Toc8087631)

[Features 12](#_Toc8087632)

[Chapter 2: Analysis 14](#_Toc8087633)

[Introduction to analysis 14](#_Toc8087634)

[Need for Analysis: 14](#_Toc8087635)

[Object oriented analysis: 14](#_Toc8087636)

[Pitfalls and merits of your project: 14](#_Toc8087637)

[Requirements Analysis: 15](#_Toc8087638)

[Functional and Non-Functional requirements: 15](#_Toc8087639)

[NLA: 17](#_Toc8087640)

[Initial Class Diagram: 18](#_Toc8087641)

[Use Case Diagram: 19](#_Toc8087642)

[Architecture 21](#_Toc8087643)

[1. Methodology used: 21](#_Toc8087644)

[i. Requirements 21](#_Toc8087645)

[iii. Program Design 22](#_Toc8087646)

[iv. Coding 22](#_Toc8087647)

[v. Testing 22](#_Toc8087648)

[vi. Implementation 22](#_Toc8087649)

[2. Design Pattern: 23](#_Toc8087650)

[3. System Architecture: 24](#_Toc8087651)

[Chapter 3: Design 24](#_Toc8087652)

[Dynamic Modelling 25](#_Toc8087653)

[Sequence Diagram: 25](#_Toc8087654)

[Activity Diagram: 28](#_Toc8087655)

[Structural Modelling 31](#_Toc8087656)

[Final Class Diagram: 31](#_Toc8087657)

[DFD: 32](#_Toc8087658)

[Database Modelling 36](#_Toc8087659)

[ER Diagram: 36](#_Toc8087660)

[Data Dictionary: 37](#_Toc8087661)

[Chapter 4: Implementation 49](#_Toc8087662)

[Choice of language: 49](#_Toc8087663)

[Development Environment: 49](#_Toc8087664)

[Chapter 5: Testing 51](#_Toc8087665)

[White Box Testing: 51](#_Toc8087666)

[Unit Testing: 51](#_Toc8087667)

[Black Box Testing: 52](#_Toc8087668)

[Chapter 6: Other Project Issues 80](#_Toc8087669)

[Risk Management: 80](#_Toc8087670)

[Configuration Management: 81](#_Toc8087671)

[Scheduling/Gantt Chart: 85](#_Toc8087672)

[Future work: 87](#_Toc8087673)

[Limitation: 88](#_Toc8087674)

[User manual: 89](#_Toc8087675)

[User Manual 89](#_Toc8087676)

[Chapter 7: Conclusion: 97](#_Toc8087677)

[Reference 99](#_Toc8087678)

[Chapter 9: Appendix 101](#_Toc8087679)

**List of Figures**

[Figure 1: Initial Class Diagram 16](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521484)

[Figure 2: Use Case Diagram 18](#_Toc7521485)

[Figure 3: Waterfall Model 19](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521486)

[Figure 4: MVC Design Pattern 20](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521487)

[Figure 5: 3-Tier Architecture 21](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521488)

[Figure 6: Bill Sequence Diagram 23](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521489)

[Figure 7: Login Sequence Diagram 23](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521490)

[Figure 8: Item Sequence Diagram 24](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521491)

[Figure 9: Login Activity Diagram 26](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521492)

[Figure 10: Item Activity Diagram 27](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521493)

[Figure 11: Bill Activity Diagram 28](#_Toc7521494)

[Figure 12: Final Class Diagram 29](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521495)

[Figure 13: ER Diagram 33](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521496)

[Figure 14: Users Data Dictionary 34](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521497)

[Figure 15: User Type Data Dictionary 34](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521498)

[Figure 16: Item Type Data Dictionary 34](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521499)

[Figure 17: Bookings Data Dictionary 35](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521500)

[Figure 18: Items Data Dictionary 35](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521501)

[Figure 19: Bills Data Dictionary 35](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521502)

[Figure 20: Login Page Prototype 36](#_Toc7521503)

[Figure 21: Sign Up Page Prototype 36](#_Toc7521504)

[Figure 22: Home Page Prototype 37](#_Toc7521505)

[Figure 23: Gallery Page Prototype 37](#_Toc7521506)

[Figure 24: Bookings Page Prototype 38](#_Toc7521507)

[Figure 25: Categories Page Prototype 38](#_Toc7521508)

[Figure 26: About Us Page Prototype 39](#_Toc7521509)

[Figure 27: Admin Page Prototype 39](#_Toc7521510)

[Figure 28: Test Case for Register 43](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521511)

[Figure 29: Unit Testing Result for Register 44](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521512)

[Figure 30: Black Box Testing for Register 44](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521513)

[Figure 31: Black Box Testing Result for Register 45](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521514)

[Figure 32: Test Case for User Login 46](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521515)

[Figure 33: Unit Testing Result for User Login 46](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521516)

[Figure 34: Black Box Testing for Login 47](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521517)

[Figure 35: Black Box Result for Login 47](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521518)

[Figure 36: Test Case for Adding Item Type 48](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521519)

[Figure 37: Unit Testing Result for Adding Item Type 48](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521520)

[Figure 38: Black Box Testing for Adding Item Type 49](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521521)

[Figure 39: Black Box Testing Result for Adding Item Type 49](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521522)

[Figure 40: Test Case for Updating Item 50](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521523)

[Figure 41: Unit Testing for Updating Item 51](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521524)

[Figure 42: Black Box Testing for Updating Item 51](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521525)

[Figure 43: Test Case for Item Delete 52](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521526)

[Figure 44: Unit Testing for Item Delete 52](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521527)

[Figure 45: Black Box Testing Result for Item Delete 53](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521528)

[Figure 46: Black Box Testing for Item Delete 53](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521529)

[Figure 47: Test Case for Billing 54](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521530)

[Figure 48: Unit Testing for Billing 54](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521531)

[Figure 49: Black Box Testing Result for Billing 55](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521532)

[Figure 50: Test Case for Order 56](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521533)

[Figure 51: Unit Testing Result for Order 56](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521534)

[Figure 52: Black Box Testing for Order 57](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521535)

[Figure 53: Black Box Testing Result for Order 57](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521536)

[Figure 54: Test Case for Bookings 58](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521537)

[Figure 55: Unit Testing for Bookings 58](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521538)

[Figure 56: Black Box Testing for Bookings 59](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521539)

[Figure 57: Black Box Testing Result for Bookings 60](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521540)

[Figure 58: Test Case for Update Profile 61](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521541)

[Figure 59: Unit Testing for Update Profile 62](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521542)

[Figure 60: Black Box Testing for Update Profile 62](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521543)

[Figure 61: Black Box Testing Result for Update Profile 63](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521544)

[Figure 62: Test Case for URL 64](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521545)

[Figure 63: Unit Testing for URL 64](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521546)

[Figure 64: Black Box Testing Result for URL 65](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521547)

[Figure 65: Tree Chart 67](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521548)

[Figure 66: GitHub all files 68](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521549)

[Figure 67: GitHub files of Analysis 68](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521550)

[Figure 68: GitHub files of Design 69](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521551)

[Figure 69: GitHub files of Final Report 69](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521552)

[Figure 70: GitHub files of Implementation 69](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521553)

[Figure 71: GitHub files of Proposal 70](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521554)

[Figure 72: GitHub files of Testing 70](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521555)

[Figure 73: Schedules Table 71](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521556)

[Figure 74: Gantt chart 72](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521557)

[Figure 75: Laravel all files 83](#_Toc7521558)

[Figure 77: Database Migration files 84](#_Toc7521559)

[Figure 76: Database migrated in MYSQL 84](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521560)

[Figure 78: List of Models 85](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521561)

[Figure 79: List of Controllers 85](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521562)

[Figure 80: List of Views 86](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521563)

[Figure 81: Routes 87](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521564)

[Figure 82: Home Page Interface 88](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521565)

[Figure 83: Gallery Page Interface 88](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521566)

[Figure 84: About Us Page Interface 89](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521567)

[Figure 85: Products Page Interface 89](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521568)

[Figure 86: Billing Page Interface 90](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521569)

[Figure 87: Bookings Page Interface 90](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521570)

[Figure 88: Add Item Page Interface (admin) 91](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521571)

[Figure 89: Edit Item Page Interface (admin) 91](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521572)

[Figure 90: View all Bookings Interface (admin) 92](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521573)

[Figure 91: User Model Code 93](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521574)

[Figure 92: Item Types Model Code 93](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521575)

[Figure 93: Items Model Code 94](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521576)

[Figure 94: Bills Model Code 94](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521577)

[Figure 95: Register Controller Code 95](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521578)

[Figure 96: Login Controller Code 96](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521579)

[Figure 97: Item Types Controller Code 1 97](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521580)

[Figure 98: Item Types Controller Code 2 97](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521581)

[Figure 99: Items Controller Code 1 98](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521582)

[Figure 100: Items Controller Code 2 99](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521583)

[Figure 101: Items Controller Code 4 100](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521584)

[Figure 102: Items Controller Code 3 100](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521585)

[Figure 103: Booking Controller Code 1 101](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521586)

[Figure 104: Booking Controller Code 2 102](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521587)

[Figure 105: Billing Controller Code 1 103](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521588)

[Figure 106: Billing Controller Code 2 104](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521589)

[Figure 107: Billing Controller Code 3 105](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521590)

[Figure 108: Edit Profile Controller Code 106](file:///C:\Users\Roshan%20shrestha\Desktop\Final%20ReportCP.docx#_Toc7521591)

# Chapter 1:

# Introduction:

Mobile Management System is also the type of E-commerce website. E-commerce is term for any types of business. Many of the corporate companies now have made websites through which we can perform commercial trades over the web. It is reasonable to say that the online shopping is becoming common. Many ecommerce stores have come to exist which delivers different qualitative products. In context of Nepal it is a growing business. Over 2, 3 years many e-commerce company has come to exist in Nepal delivering various products.

## Project Background:

This project is based on the idea to develop buy product through websites. Studying through current situation of market in Nepal this project is going to play vital role in transaction of mobile.

Project mainly designed for business purposes where related company can keep the records of customer, product information, stock, transaction and many others feature. Different brands of mobile products will be displayed on this website which makes the related company to work on smoother and faster.

## Problem Statement:

Through e-commerce sites we can easily buy different products easily. Ecommerce brings changes in economy of the country, it increases competition which causes change in price structure of products hence causing change in consumption patterns. People save time by shopping on line.

However, there are less e-commerce sites in Nepal which sold helmets and

different bike’s gears in Nepal. Buying helmets online saves time as buyer doesn’t needs to visit different shops for buying helmets, he/she can easily choose the product and buy it from home. He/she can pay money of ordered helmets after delivery or using E-sewa

## Aims:

The goal or main aims of this project are listed below:

* Customer Satisfaction
* Display products
* While observing the product user can keep the product into the cart
* Delivery of product through cash on delivery or Online transaction

## Objectives:

Main objectives of this project are listed below:

1. Situation of market should be studied well
2. Providing good GUI to user
3. Fulfilling the target or requirement
4. Implementing protected database
5. Testing the system properly
6. Implementing proper Maintenance

## Overview of project Design

### Features

Some main features of the software are:

* **Validation:**

User login and admin login included. Specific license are provided to both.

* **Management of Customer :**

Data and information of customer are stored.

* **Data Inquiry :**

Data access consumes few time

* **Security :**

Accurate password and username is essential for the financial trade or data access. Also, improve the safety and authentication of transactions.

* **Cart system:**

User can cart the product for later purchase and still can browse other product.

* **Billing system:**

Customer can generate bill after login to the application

.

# Chapter 2: Analysis

## Introduction to analysis

According to analysis phase, the Functional and Non-Functional requirements are analyzed, and the requirements are clarified with MoSCoW Prioritization.

### Need for Analysis:

Systems need to be analyzed and it is completed by system analysts, pursues to know what public need to thoroughly analyze data input or data flow, procedure or alter data, store data and output info in the environment of a specific institute or corporate.

### Object oriented analysis:

The categorize of classes is complete and then the relationships among several classes are recognized and also requirements of the structure are resoluted.

### Pitfalls and merits:

***Merits:***

* + - The customer is provided with Online service
    - Customer desire are provided with multiple option
    - Pressure of bargain are neglected

***Pitfalls:***

* + - * online paying is lack.

## Requirements Analysis:

### Functional and Non-Functional requirements:

The requirements that requires a specific functional behavior of the system when circumstances are encountered are called functional requirements.

Non-Functional requirements are the criteria specifying requirements used to judge the operation of a system and not specific behaviors. It defines how a system is supposed to be.(Anon., n.d.)

Indexes used in the table below are:

**F**=Functional requirements

**S**=Should Have

**NF**=Non-Functional requirements

**C**=Could Have

**M**=Must Have

**Functional and Non Functional Requirements Table with MoSCoW prioritization:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. N** | **Requirements** | **MoSCoW**  **Prioritization** | **Functional(F)/ Non – Functional (NF)** |
|  | Registration | **M** | **F** |
|  | Login | **M** | **F** |
|  | Dashboard | **S** | **F** |
|  | Mobile information | **M** | **F** |
|  | inventory | **M** | **F** |
|  | Generate and print bill | **M** | **F** |
|  | Purchase information | **M** | **F** |
|  | Data (add, update, delete, view) | **M** | **F** |
|  | Product( add, update, delete, view) | **M** | **F** |
|  | View order Information | **M** | **F** |
|  | Order Online(e-sewa) | **W** | **NF** |
|  | Review | **C** | **NF** |
|  | Discount | **C** | **NF** |
|  | Exchange | **C** | **NF** |
|  | Receive order via Cash on delivery() | **M** | **NF** |
|  | Cart | **M** | **F** |
| 18. | Warranty Form | **M** | **NF** |

Table 1: Functional and non-Functional Requirements Table

## NLA:

NLA is a short form of Natural Language Analysis. The procedure of requirement analysis of a system that catagorized the nouns, adjectives and verbs as potential candidates for classes, attributes and methods in the form of natural language (English). It holds numerous forms of filtration. Nouns are nominated as potential classes whereas verbs as potential methods. The nominated classes for initial class diagram are: User, Products, Carts, order, Bill etc.

## Initial Class Diagram:

A structural or static diagram is known as class diagram which shows the static view of all the classes, constraints and their association

**Advantages:**

* Analyzing the system are time saving.
* System structure are provided with detail strcture.

**Disadvantages:**

* dynamic model is lacked.
* Collaboration

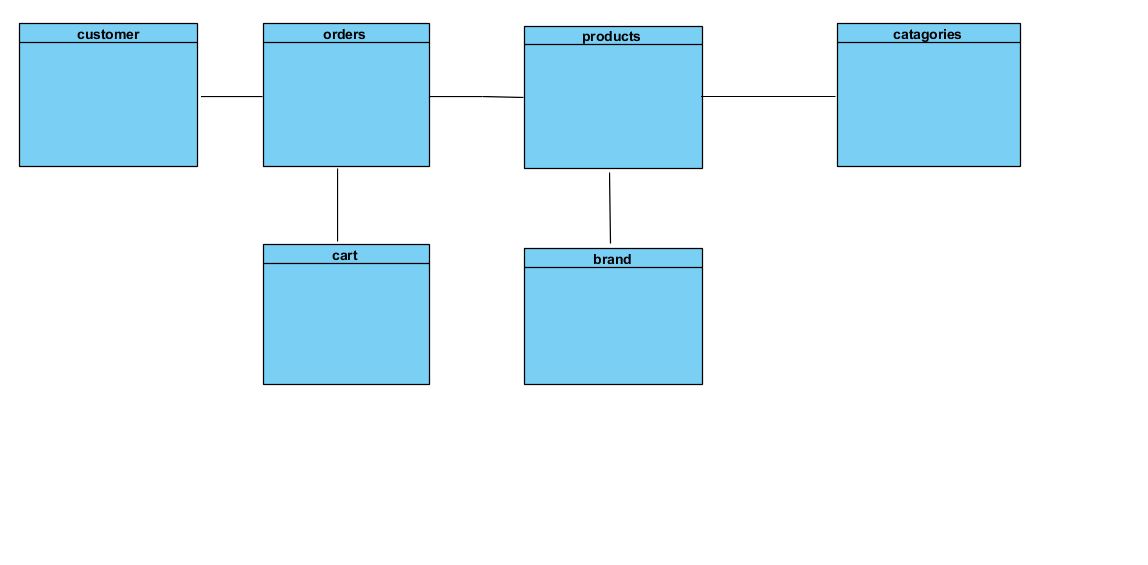


Figure 1: Initial Class Diagram

## **Use Case Diagram:**

Use case is list of actions, steps which shows the interactions between user and a system, visitor and a system and administrator and a system. It shows all the actions done by various individuals to interact with the system. Using use case, we can easily see what user is capable of. User and visitor are different entities Users and view products, purchase products and pay directly using esewa. But visitor is limited to only viewing products. Visitor need to register to have user functions. And admin obviously have all the functions such as viewing products, adding/deleting/updating products as well as adding other admins. Editors also have all the features of Admin except for user’s page which consists details of all admins and editors.

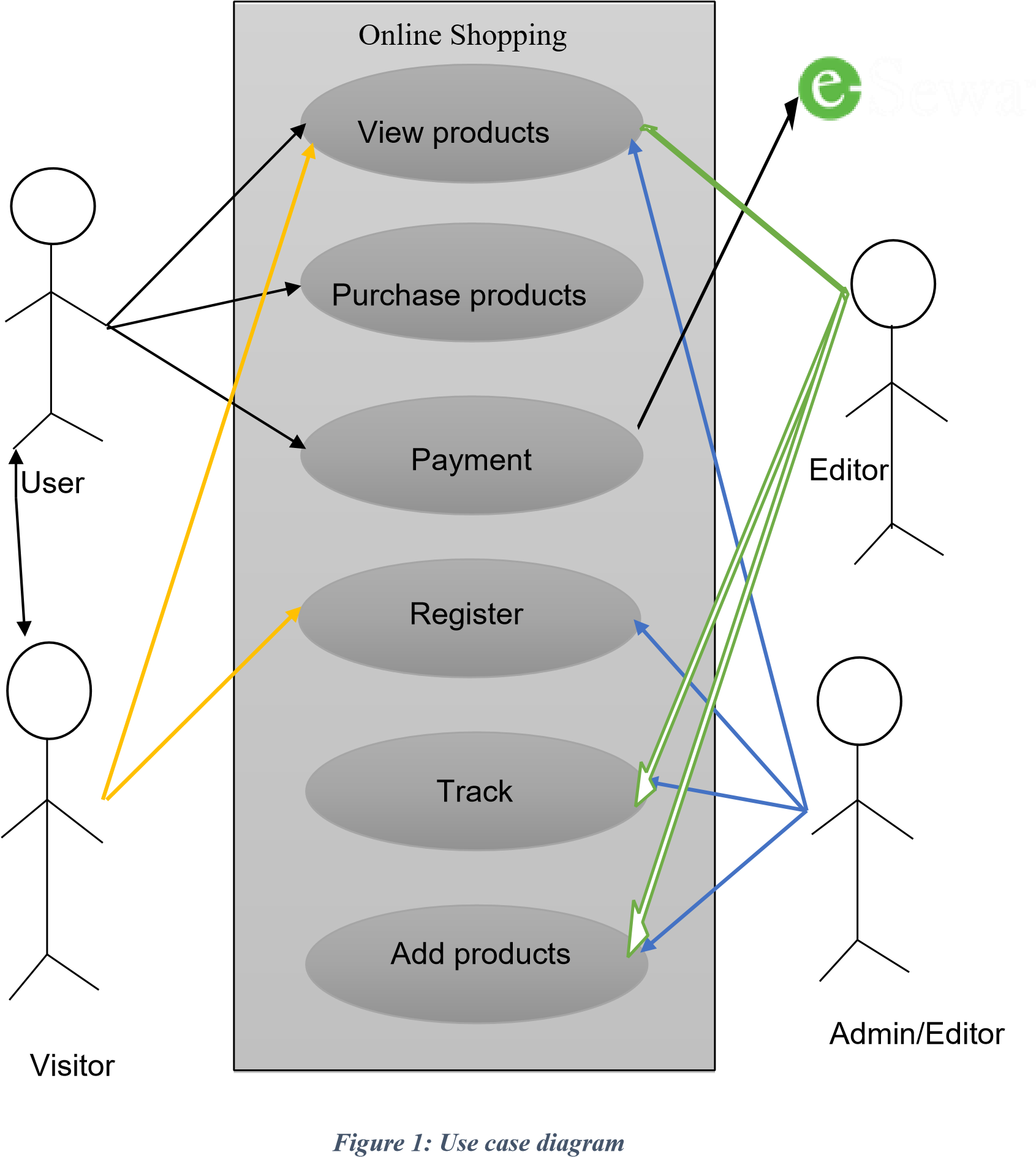


Figure 2: Use Case Diagram

## Architecture

### Methodology used:

I will be using enhanced waterfall for this project. The waterfall model divides project into six phases.

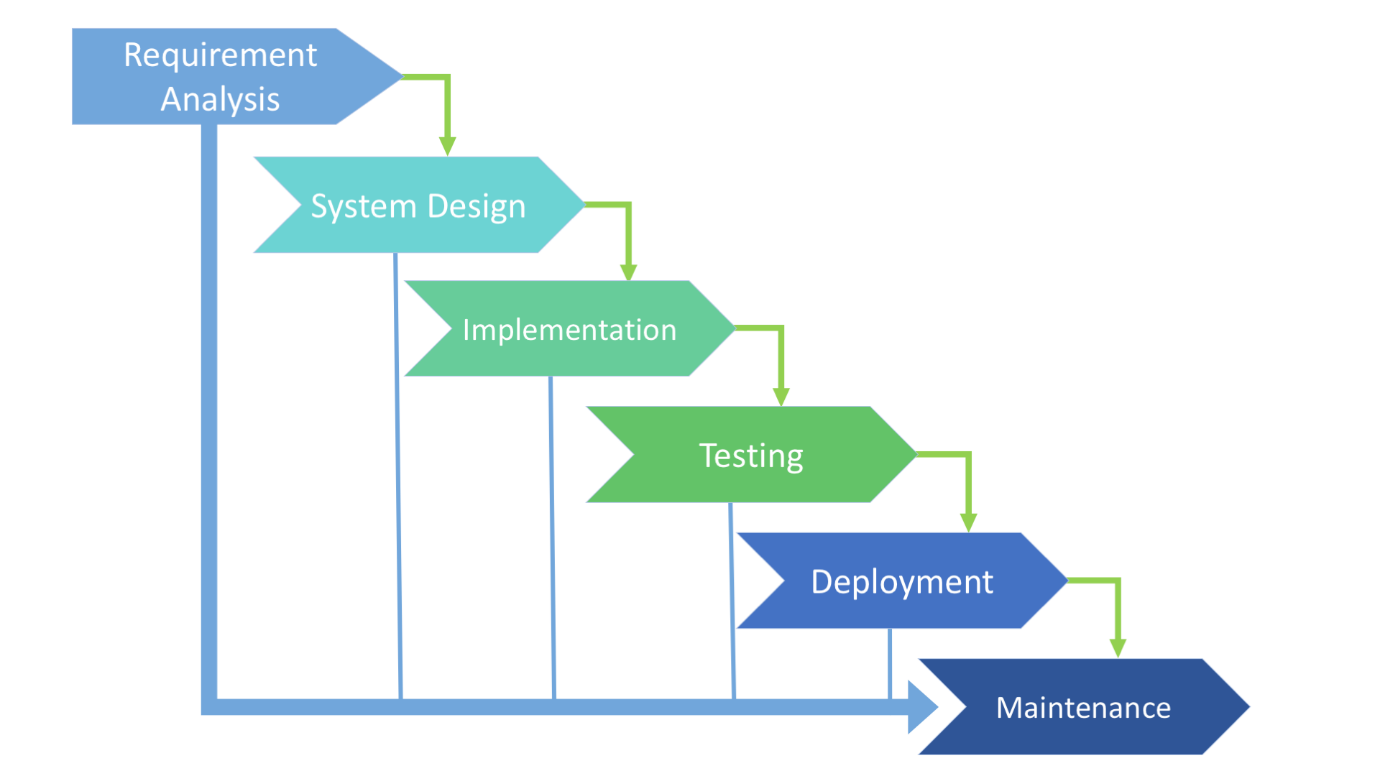


Figure 3: Waterfall Model

### i. Requirements

In this phase I will be gathering requirements that will be required for developing this project from the client and different surveys. This is the one of the important phases of system development. So, requirements need to be clear before moving to next phase.

**ii. Analysis**

In this phase I will briefly analyze gathered requirements from the client. Requirements of client will be analyzed and checked if it is feasible or not. After analyzing the requirement then we move to next phase.

### iii. Program Design

As per the requirements of the client I will then starting design for my system. First of all I will prepare use case, software architecture of the system. Then wireframes will be created and shown to client to show him how the system is going to look in future and graphical designs of the system will be also show for client satisfaction.

### iv. Coding

In this phase development work will be carried out which includes designing and back end development, first of all front-end design of the project is made using HTML, CSS, Bootstrap. Front end design will be responsive and user-friendly and later server side/back-end of the project will be done using PHP and JavaScript.

### v. Testing

Testing of the web application is done in this phase. System is debugged for any errors in this phase. The system will be reviewd to customer and supervisors for testing and check for errors.

### vi. Implementation

In this phase installation, support and maintenance of the complete system is done. After testing if there are no errors then the system is released into market. This system will be maintained in time to time.

### Design Pattern:

There are lots of pattern like creational, behavior, Factory, Repository, MVC etc. Among them I like to use MVC pattern which is Model View Controller. I choose this pattern because It is simple and best way to separate the logic of this application. The key features of this pattern are:

* The view cooperates with the customer
* customer input is controlled by the controller
* Updates and Information are received to the model
* State of Model is checked by the view and respond It accordingly
* View waits another interaction.

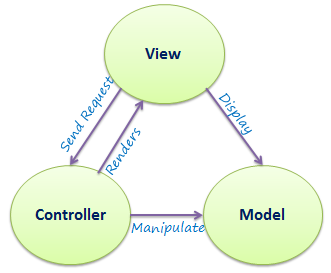
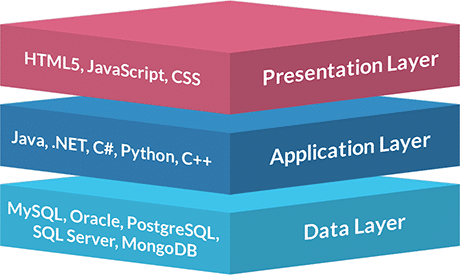


Figure 4: MVC Design Pattern

MVC had indeed moral philosophy. Code arrangement and design are simpler, Establishing the software maintainable. The view is in the view files, the logic in the template, and the controller handles them all.

### System Architecture:

System architecture is the art of design which consist of different architecture tier like N-tier, Two-tier, and three-tier. So I have chosen to use 3-tier architecture.



# Chapter 3: Design

Figure 5: 3-Tier Architecture

## Dynamic Modelling

### Sequence Diagram:

Sequence Diagrams are collaboration diagrams which retain records of processes and display elements that are prepared according to object (horizontally) and time (vertically).

**Justifications:**

I have provided sequence diagram to display the system active object interaction.

**Advantages:**

* used for object-oriented analysis to confirm class diagrams against use cases.

**Disadvantages:**

* It needs a completely well-defined class model.

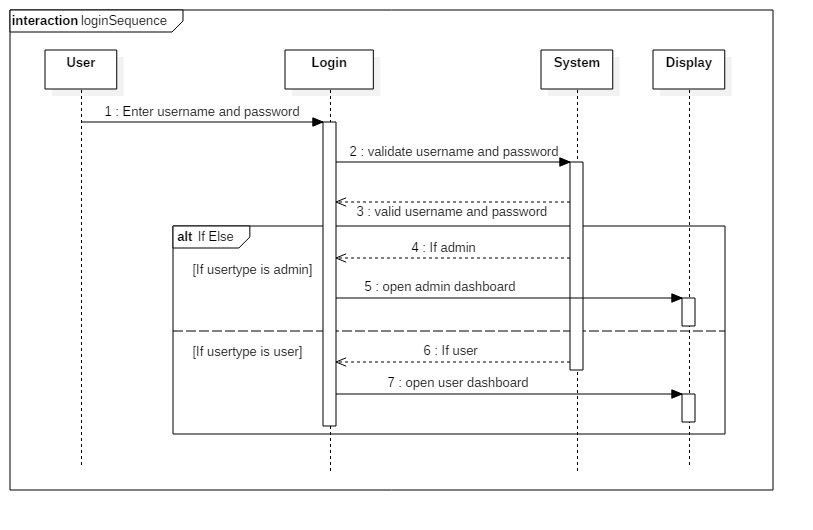
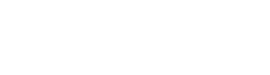
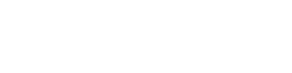


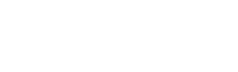
Figure 6: Login Sequence Diagram



ShoppingCart



Order



Item



create



getTotal



totalprice



getPrice



Calculate total



Itemprice

Figure 7: Sequence Diagram

This is the sequential diagram of shopping cart. A virtual cart is created at first for selected products then shoppingCart asks for getTotal to Order. Order ask Price from item then Itemprice is returned to Order. Then based on itemprice it calculates total and return total price to shopping cart.

### Activity Diagram:

An **activity diagram** is a graphical representation of an executed set of procedural system activities. It describes parallel and conditional activities and use cases and system functions comprehensively.(Anon., n.d.)

**Justifications:**

Here, I have drawn activity diagram to show message flow from one activity to another. It captures the dynamic behavior of the system. It is also used to draw the activity flow of a system, describes the sequences from one activity to another. It also describes the parallel, branched and concurrent flow of the system.

**Advantages:**

* Since it is the most user-friendly diagram. So, generally regarded as an essential tool.
* It helps to display multiple conditions and actors within a work flow through the use of swim lanes.
* These diagrams are normally easily comprehensive for both analysts and stakeholder.

**Disadvantages:**

* These diagrams can lead the over complex which might affect the user-friendly nature.
* These diagrams do not give the detail about how object behave or collaborate.

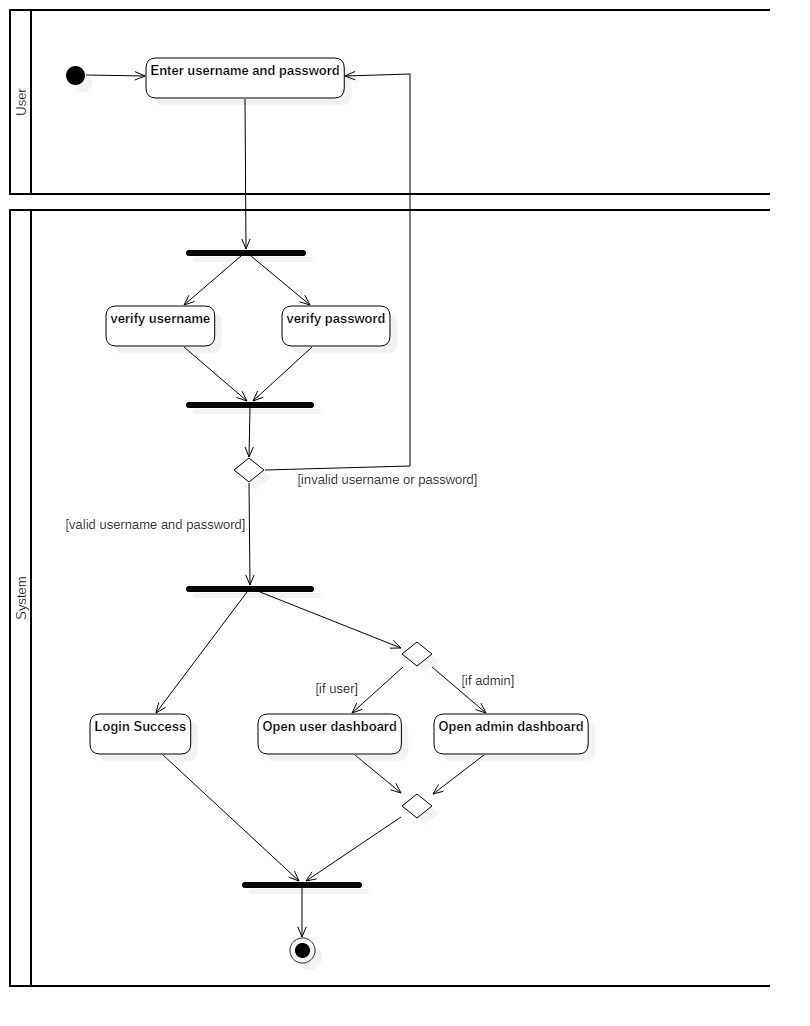
****

Figure 6: Login Activity Diagram

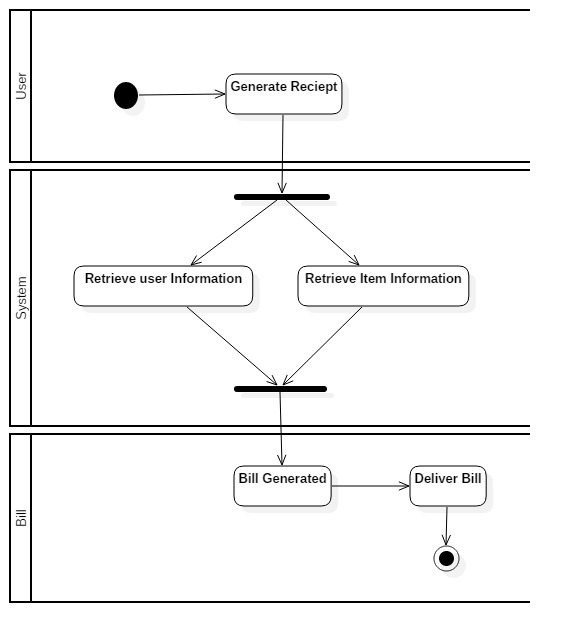


Figure 7: Activity Diagram to generate bill

## Structural Modelling

### Final Class Diagram:

Final class Diagram is the part of Design phase for the identification of classes and operations along with their relationships.

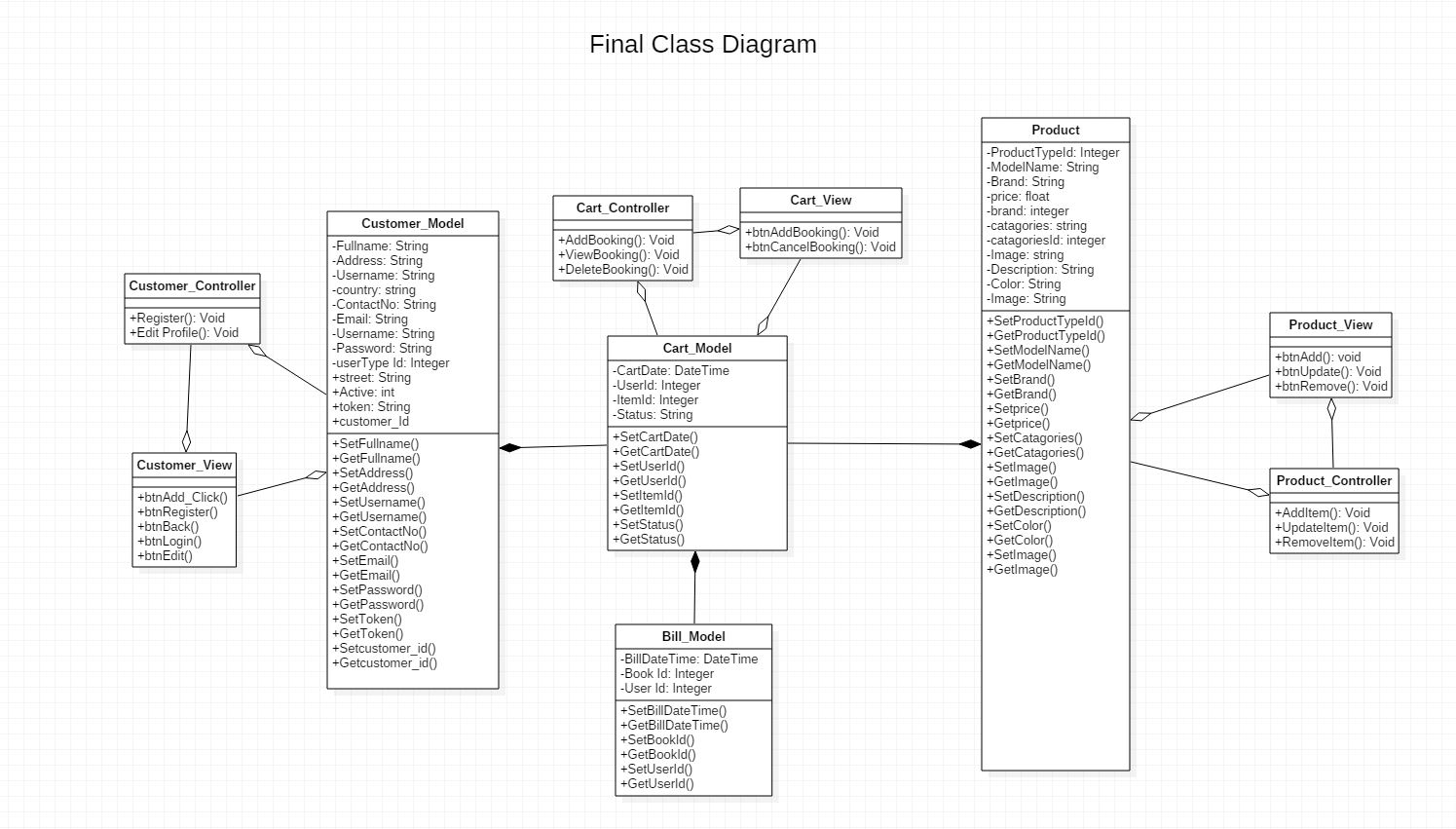
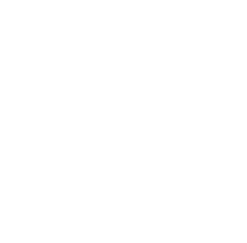


Figure 8: Final Class Diagram

### DFD:

A data flow diagram (DFD) charts the info movement for any process. levels and layers are used, it can dive into further aspect consecutively. DFD levels are set at 0, 1 or 2, and regularly go to or ahead of level 3. Here, I have drawn 0 level DFD and 1 level.

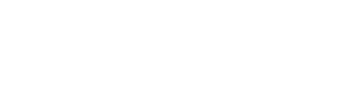
The Level 0 Data Flow Diagram or context diagram is shown below:



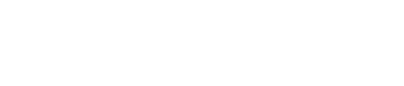
E

-

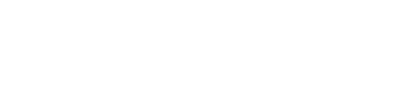
commerce



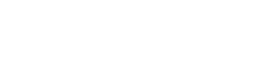
Database



Administrator



User



Visitor



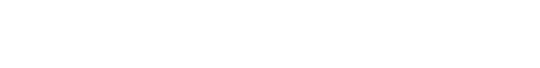
Website management



Website activity



Purchase



Retrieve purchase detail and

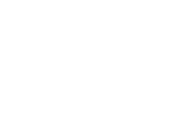
orderdetails



Figure 9 : Level 0 DFD Diagram

|  |  |
| --- | --- |
| D1 | Customer records |

Level 1 Data flow diagram



Customer

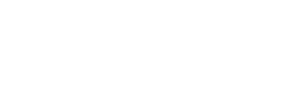


Enquiry



Availability

checked



Registration success

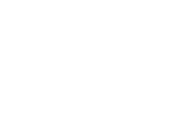
message



1



Customer details



Customer



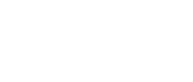
Check

products



Availability

checked



Product available

message



2



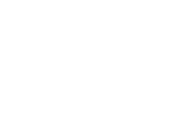
D2



Product records



Product details



Customer



Order product



Product

purchased



Receipt



3



Figure 10: Level 1 DFD Diagram

## Database Modelling

### ER Diagram:

An ***Entity-Relationship Diagram (ERD)*** is a technique of data modeling that graphically depicts the entities and the relationship between them in an information system

**Justifications:**

It is used for the ideas to visualize database design, so as to recognize the errors and corrections are made before executing the modifications in database.

**Advantages:**

* If the relationship of entities and attributes are known, then It is very simple to analyze the system.
* Visual representation.
* for database designer it is one of the effective tool.

**Disadvantages:**

* limited specification and constraints.
* Data manipulation are difficult to show.

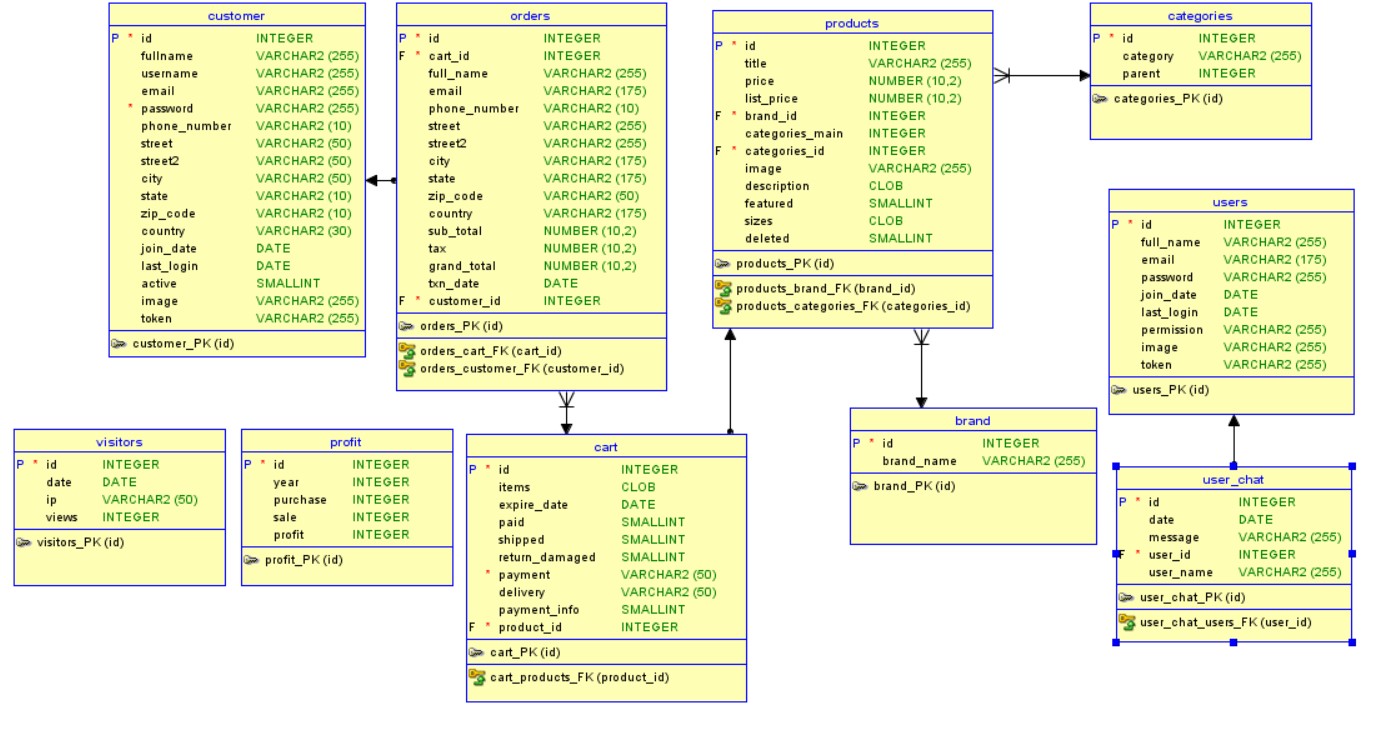


Figure 11: ER Diagram

## Data Dictionary:

Figure 12: Customer Type

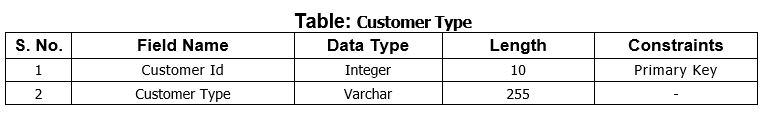


Figure 13: Item Type Data Dictionary

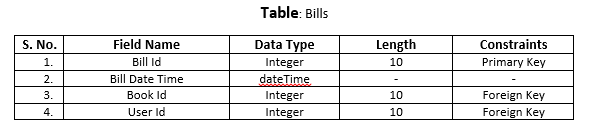
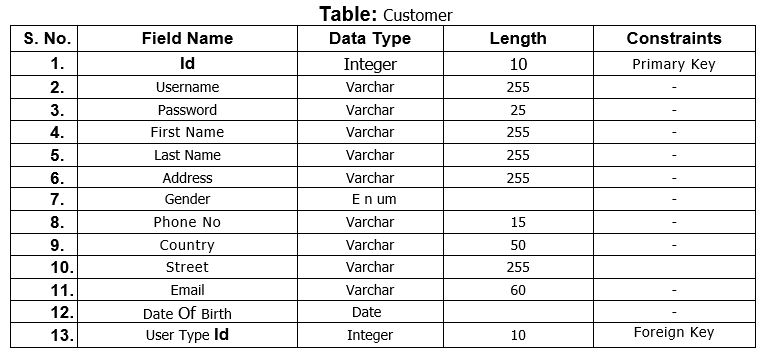
****

Figure 14: Customer Data Dictionary

Figure 15: Bills Data Dictionary

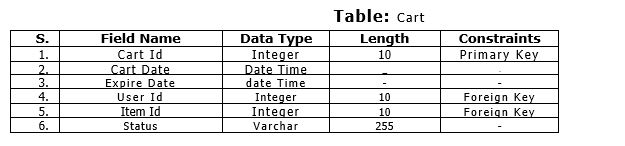


Figure 16: Carts Data Dictionary

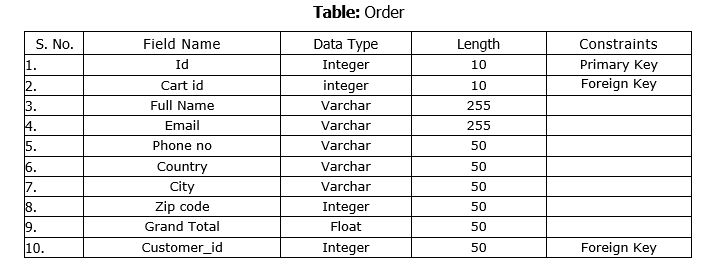


Figure 17:Order Data Dictionary

**UI Modeling-Prototype**

Prototype are created according to client requirements. It is just rough structure of how the system is going to look after development.

1. **Login Page:**



Figure 18: Login Page Prototype

1. **Registration Page for new user:**

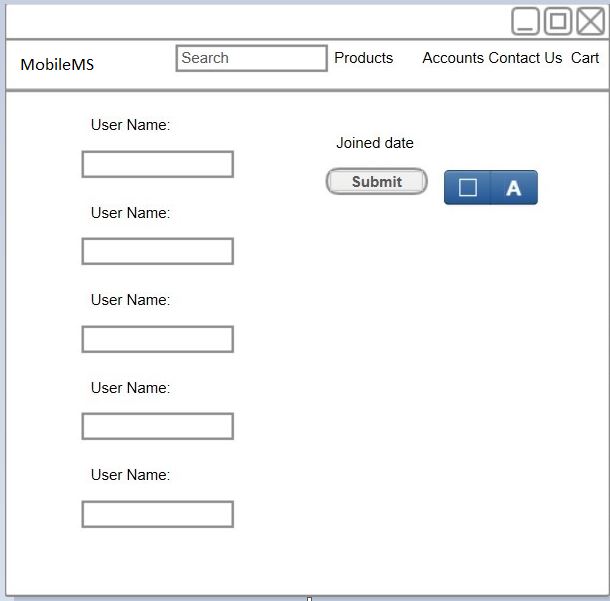


Figure 19: Prototype of Sign Up

1. **Home Page:**

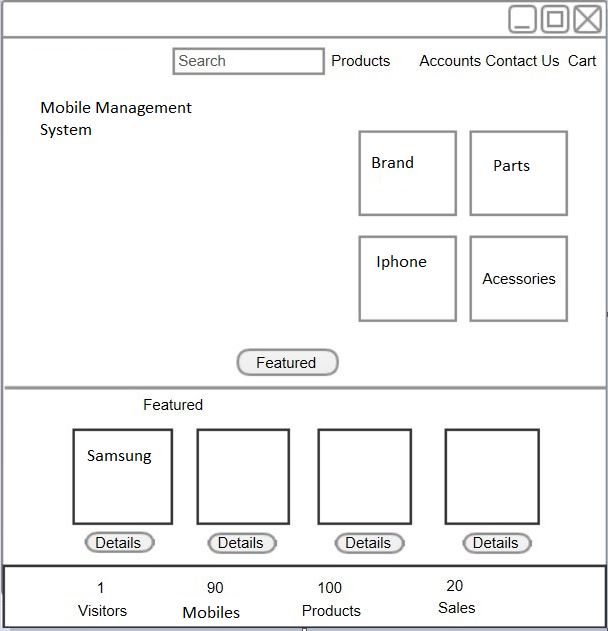


Figure 20: Prototype of Home Page

1. **Gallery Page:**

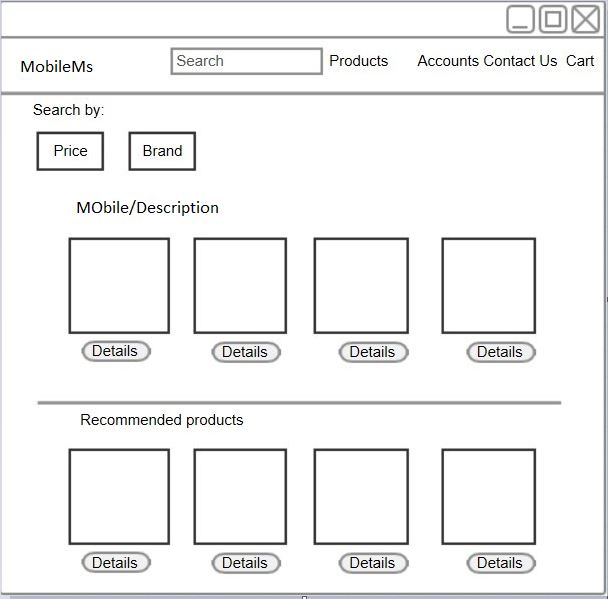


Figure 21: Gallery Page Prototype

1. **Carts Page**

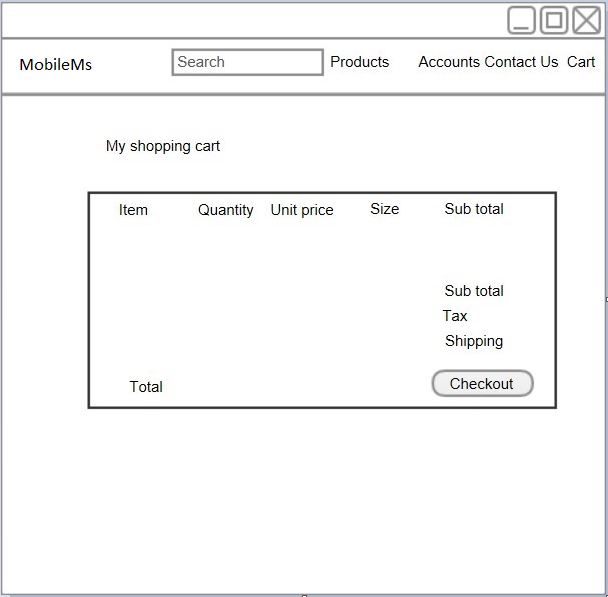


Figure 22: Prototype of Carts Page

1. **Order Page:**

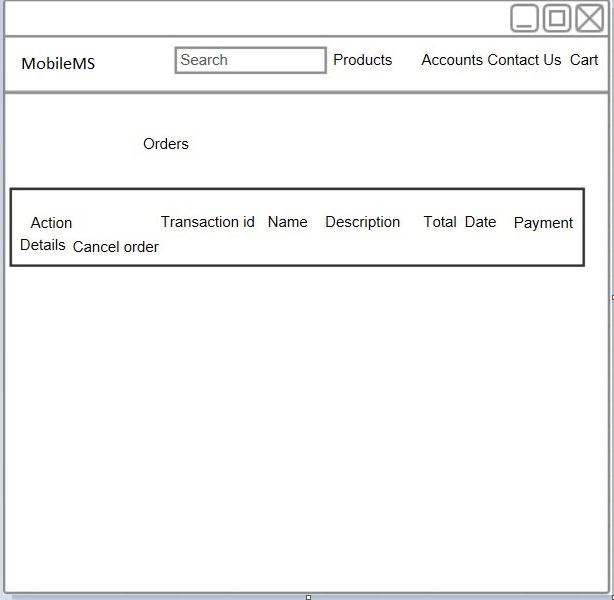
****

Figure 23: prototype of order page

1. **Admin Page**

****

Figure 24: Prototype of Admin Page

# Chapter 4: Implementation

In this phase, we start to make the product on the basis of analysis and design phase. Coding is the main objective in this phase.

### Choice of language:

For the implementation of product, I chose PHP (Hypertext Preprocessor) language. It is an HTML embedded Web scripting language. It allows Web developers to write generated pages quickly with ease. It is also great for creating database-driven Web sites. It also has a number of unique features and features and specific function.(Anon., n.d.)

### Development Environment:

1. PHP

It is a general-purpose server-side scripting language designed for web development especially but also used as general-purpose programming language. PHP is better for my project because it provides an open source, cross-platform tools for creating a dynamic website.

It’s suitable for ecommerce websites where there are

1000’s of products, here we can easily retrieve products from the database using PHP. It is also cross platform which means PHP scripts will run on UNIX, Linux or Windows server. PHP is very compatible with MySQL. PHP is the heart of this project. PHP is used for various (CRUD) CREATE READ UPDATE DELETE) functionality in this project. PHP. Using PHP this web application is dynamic anyone without knowledge of programming and designing can easily run this web application to run his/her company.

1. MySQL

To store data, we need database. MySQL is one of big brands used to communicate with database. It is standard language for relational database management system. SQL is required for creating new databases to store information, records, etc.

##### Frameworks

i. Bootstrap

Bootstrap is popular framework for designing websites. It is used world widely by millions of designers. Bootstrap makes coding faster and easier. Bootstrap are for high styles and responsive design.

##### IDE

i. PhpStorm

It is cross-platform IDE for PHP. It provides editor through which we can easily code for PHP, HTML and JavaScript. It provides features such as errors prevention and automated refactoring’s for PHP and JavaScript code.

Code & UI screenshot are given in the **Appendix.**

# Chapter 5: Testing

Testing refers to the evaluation of software developed in order to ensure the quality of product and to identify defects. There are many types of testing approaches but, in this project I have done **white box** and **black box** testing only.

## White Box Testing:

It is also called Structural or Glass Box Testing. It's based on the structure of internal code applications. In white-box testing, programming skills along with an internal perspective of the system is used to design test cases. The testing is carried out at the unit level. Unit testing is done as white box testing.

### Unit Testing:

**Unit testing** is a level of software testing where testing of components of a software is done. The purpose is to validate that each unit of the software performs as it is designated to be. A unit is the smallest testable part of any software with one or several inputs and usually one output.

**Advantages of White Box Testing:**

1. Optimizing code by finding hidden errors.
2. Cases for white box testing can be easily automated.
3. Testing is more thorough as it usually covers all code paths.
4. Even if GUI is not available, testing can start early in SDLC.

**Disadvantages of White Box Testing:**

1. Testing the white box can be quite complex and costly.
2. Developers who typically run test cases in the white box detest it. Developers ' white box testing is not detailed can result in production errors.
3. White box testing requires professional resources, with detailed programming and implementation understanding.
4. White-box testing is time-consuming, it takes time for larger programming applications to fully test.

## Black Box Testing:

` It is also called as Behavioral/Input-Output Testing. It is a software testing method in which the functionality of the software under test is evaluated without looking at the internal code structure.

**Advantages of Black Box Testing:**

* Tester can be non-technical.
* There is no need for the tester to have detailed functional knowledge of system.
* Testing helps to identify contradictions and ambiguousness in functional specifications.
* Test cases can be designed as soon as the functional specifications are complete.

**Disadvantages of Black Box Testing:**

* Test cases are challenging to design without having clear functional specifications.
* If the test cases are not developed based on specifications then it would be difficult to identify tricky inputs.
* There are chances of having unidentified paths during the testing process.
* There is a high probability of repeating tests already performed by the programmer.

Numbers of test cases should be created for the unit testing. As this is my leaning project, here I have created only 10 test cases for the unit testing as shown below:

1. **User Registration**

***Test log:***

|  |  |
| --- | --- |
| Objective | To login into the system for user customer. |
| Action | By entering customer login details |
| Expected Result | Login and enter the system with customer details |
| Actual Result | Logged in into the system with customer details |
| Conclusion | Successful |

Table 3: Test Log for User Registration

***Unit Testing:***

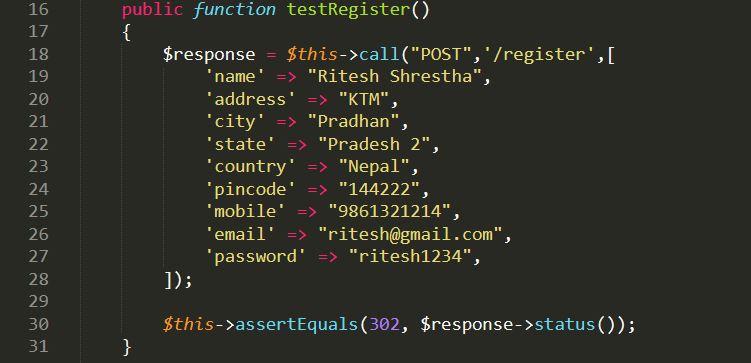
****

Figure 25: Test Case for Register

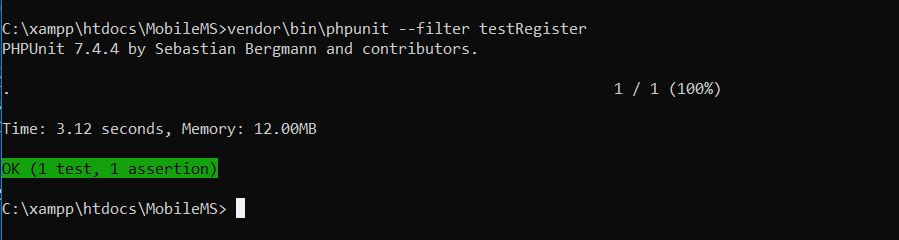


Figure 26: Unit Testing Result for Register

***Black Box Testing:***

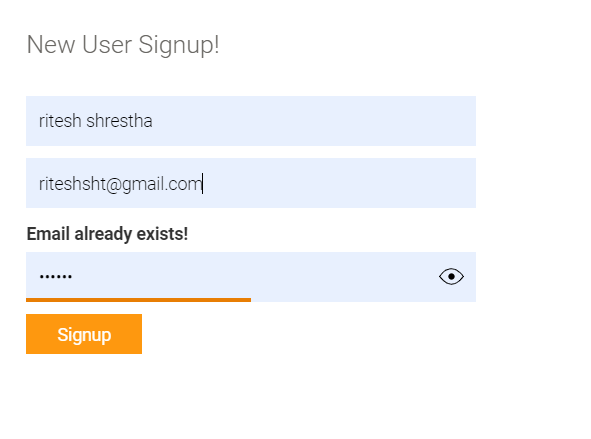


Figure 27: Registration of Black Box Testing

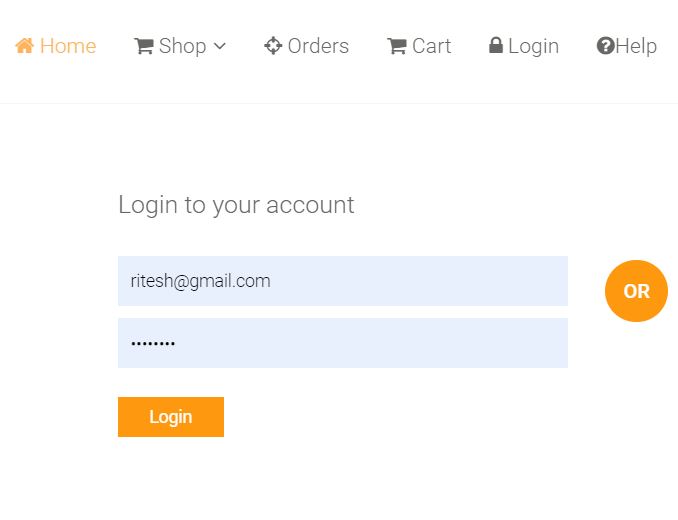
****

Figure 28:Log in form

1. **Add Item Type**

***Test log:***

|  |  |
| --- | --- |
| Objective | To insert product details |
| Action | By entering product details through the form |
| Expected Result | Insertion of product details in the table |
| Actual Result | Product details inserted in the products table |
| Conclusion | Successful |

Table 5: Test Log for Adding Item Type

***Unit Testing:***

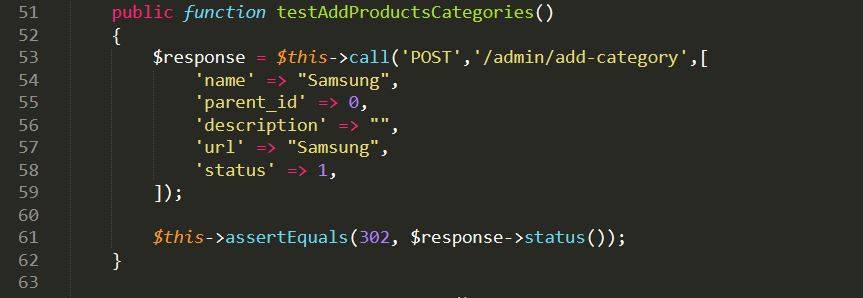
****

Figure 29: Test Case for Adding Item Type

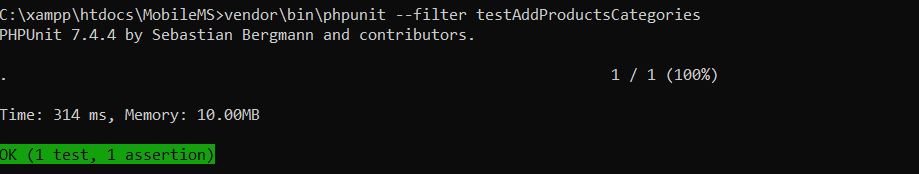
****

Figure 30: Unit Testing Result for Adding Item Type

***Black Box Testing:***

**Product add**

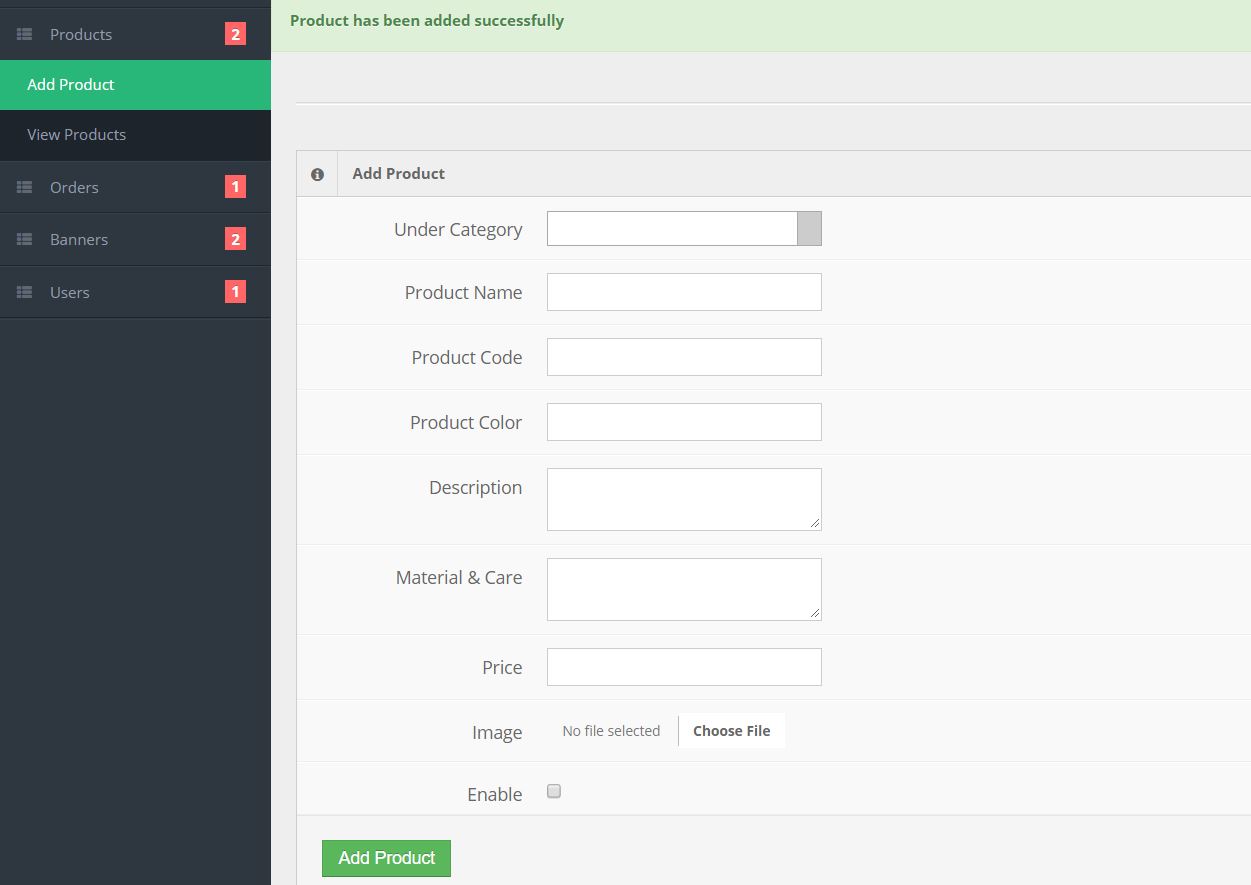
****

Figure 31: Adding Product Black Box Testing success

**View of added product**

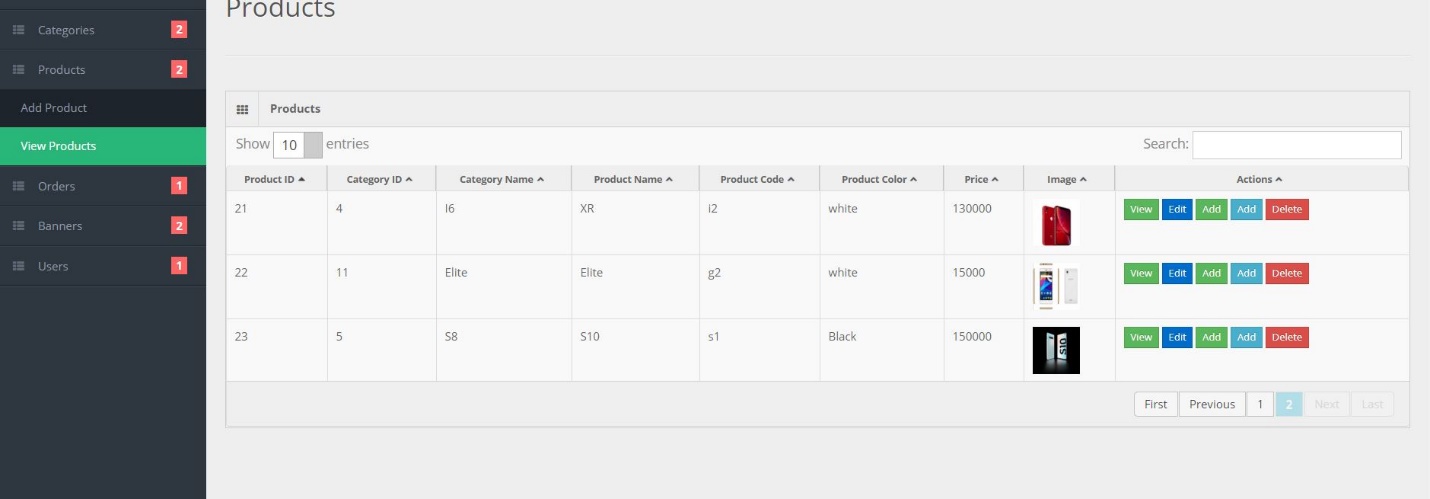
****

Figure 32: Added product for black box testing

* + - 1. **Item Update**

***Test log:***

|  |  |
| --- | --- |
| Objective | To edit and update products details |
| Action | By clicking on the edit button and update customer details from the table |
| Expected Result | Update products details from the table |
| Actual Result | Product details update sucessfully from the table |
| Conclusion | Successful |

Table 6: Test Log for Updating Item

***Unit Testing:***

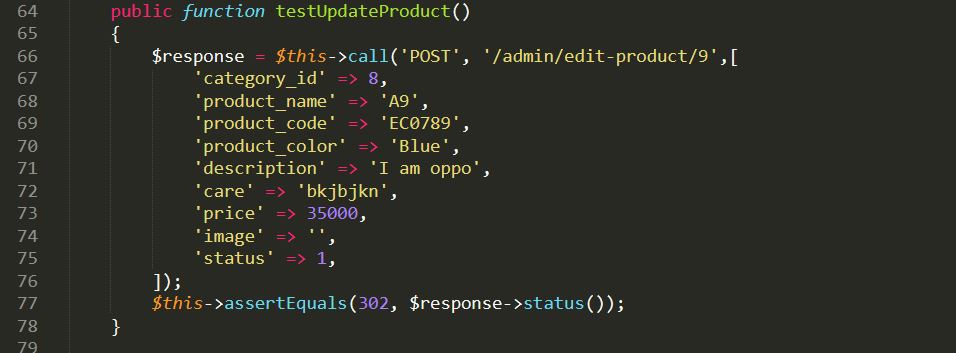
****

Figure 33: Test Case for Updating Item

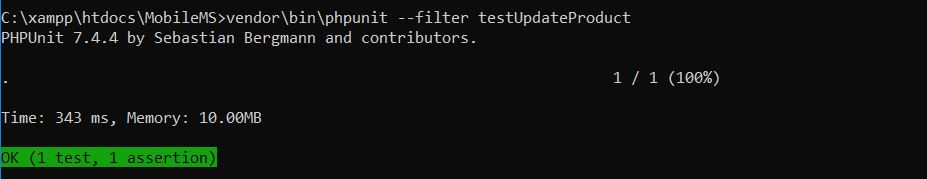
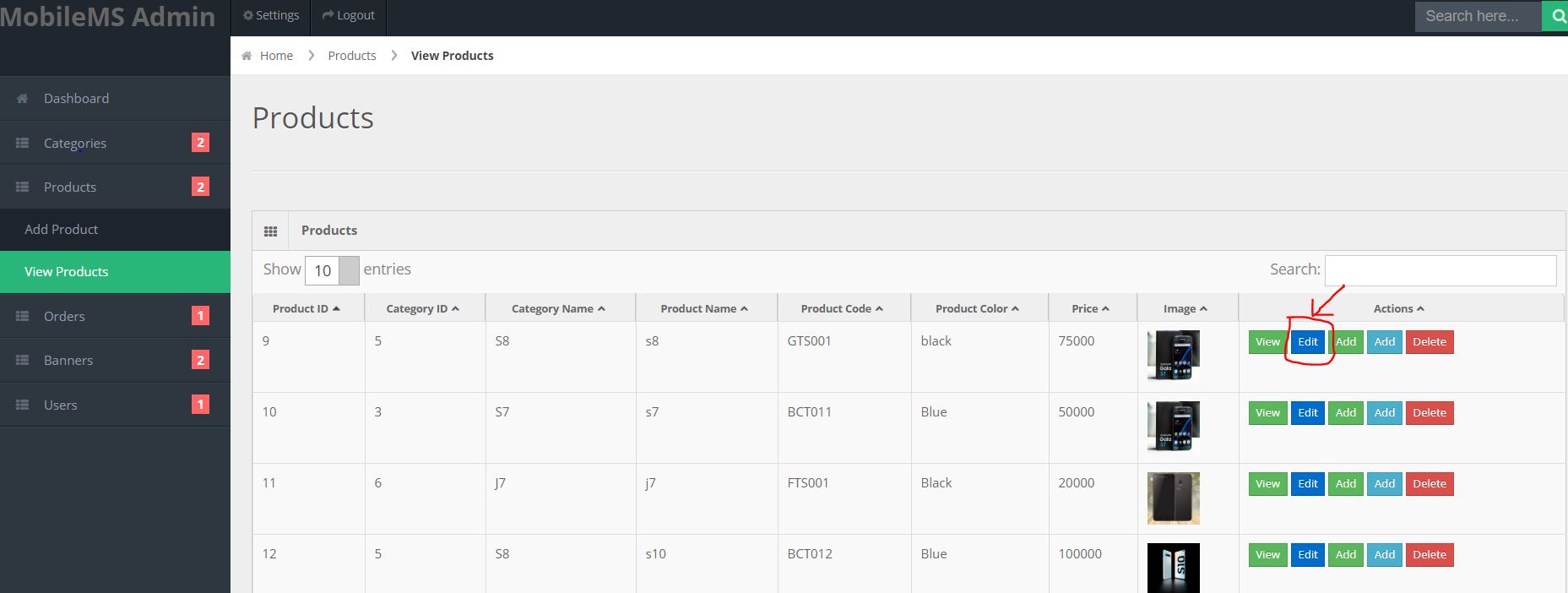
******

Figure 34: Unit Testing for Updating Item

***Black Box Testing:***

****

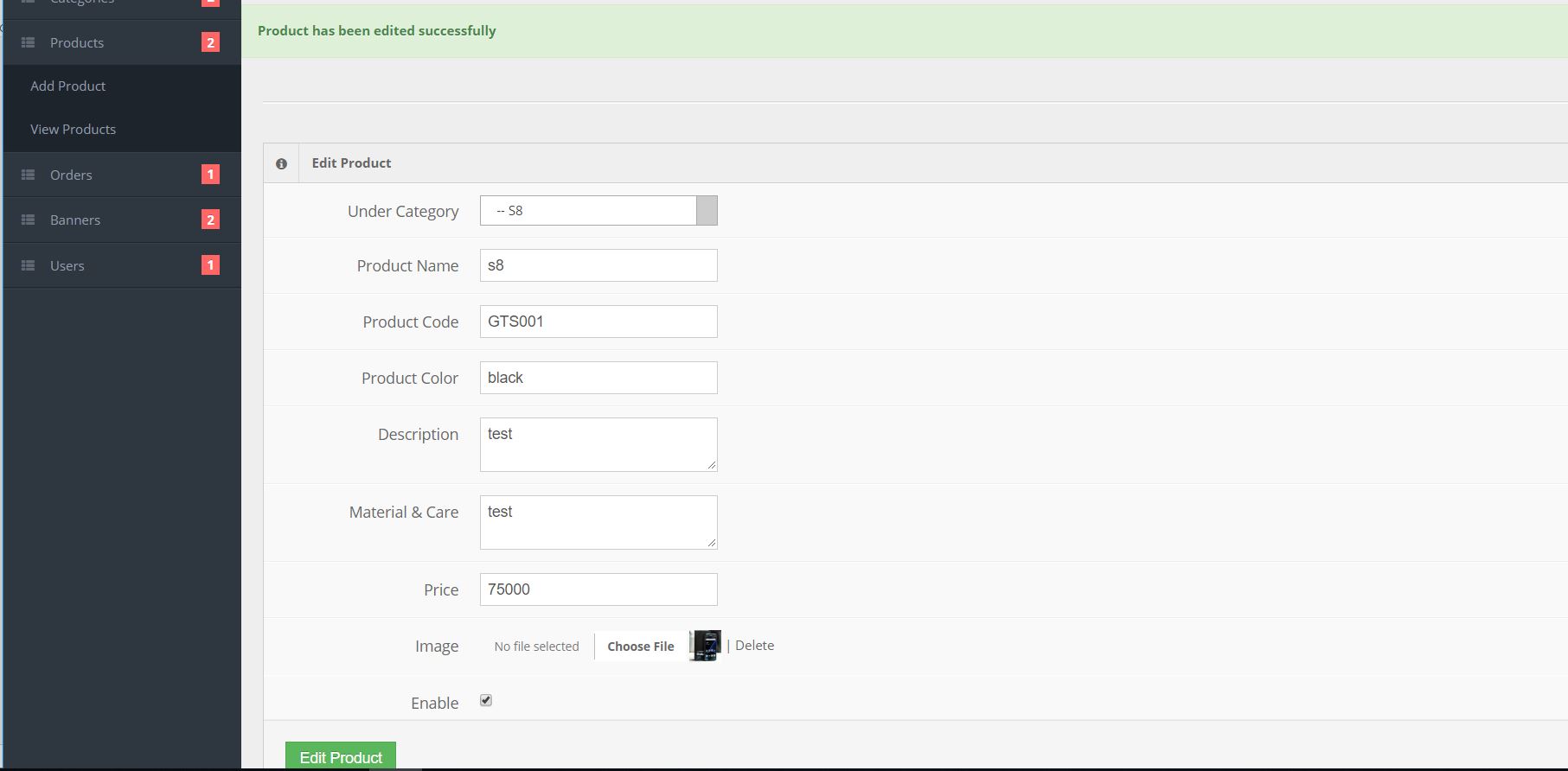
****

Figure 35: Black Box Testing for Updating icon

Figure 36: edit/update success

* + - 1. **Item Delete**

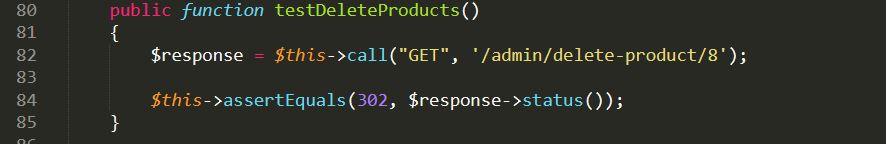
***Test log:***

|  |  |
| --- | --- |
| Objective | To delete products selected product |
| Action | By clicking on the delete button to delete product from table |
| Expected Result | Delete products details from the table |
| Actual Result | Product details deleted sucessfully from the table |
| Conclusion | Successful |

Table 7: Test Log for Item Delete

***Unit Testing:***

Figure 37: Test Case for Item Delete

****

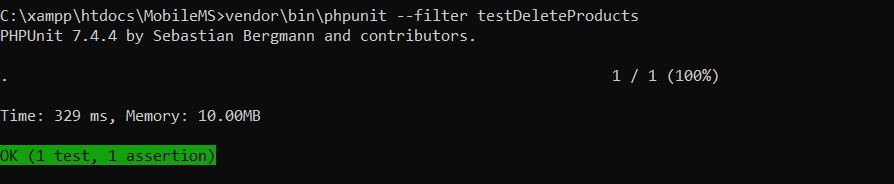
****

Figure 38: Unit Testing for Item Delete

***Black Box Testing:***

******

Figure 39: Black Box Testing for Item Delete

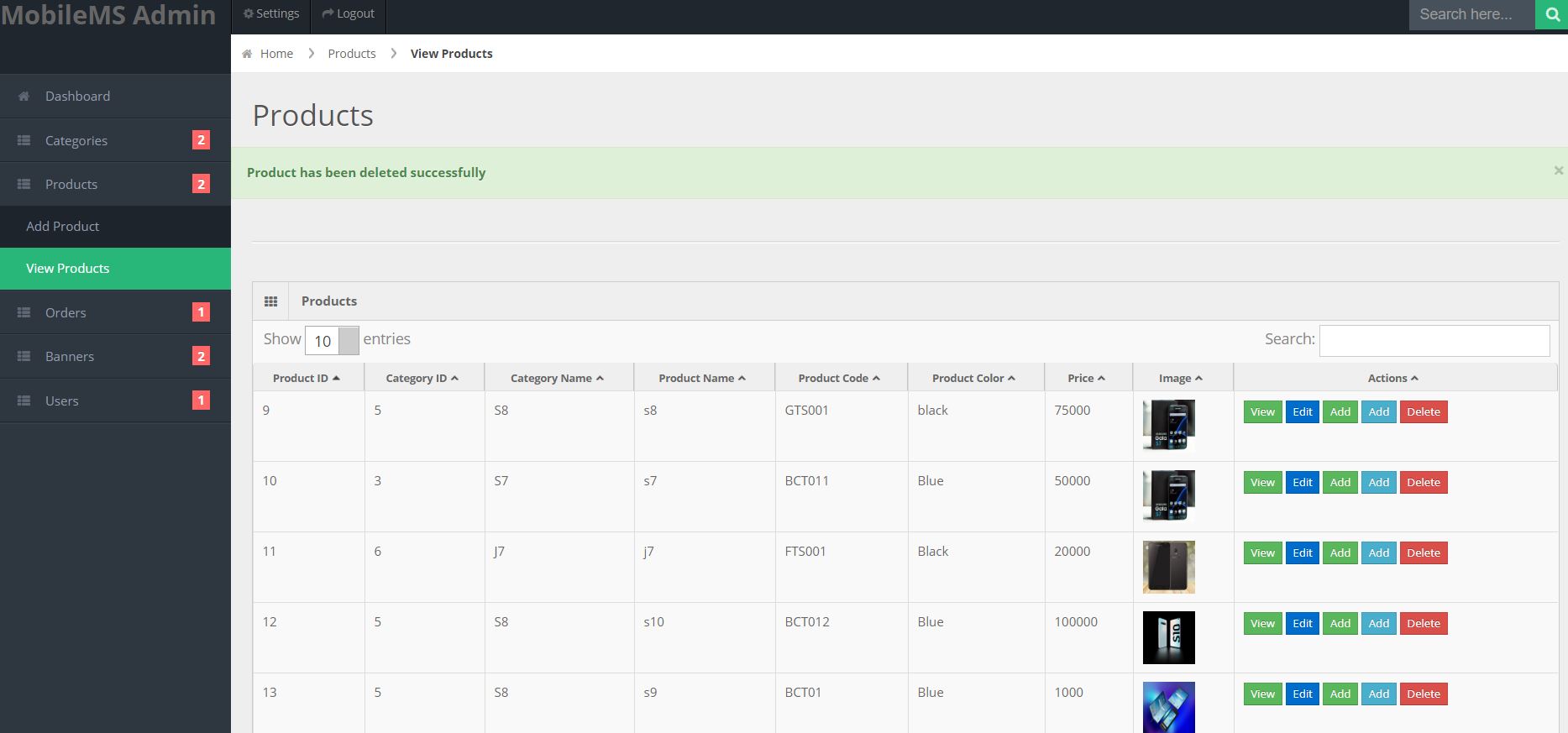
******

Figure 40: Black Box Testing Result for Item Delete

1. **cart**

***Test log:***

|  |  |
| --- | --- |
| Objective | To add selected products into cart |
| Action | By clicking on the add to cart button to add product into cart table |
| Expected Result | Add products details to cart |
| Actual Result | Product details added sucessfully to cart |
| Conclusion | Successful |

Table 8: Test Log for cart

***Unit Testing:***

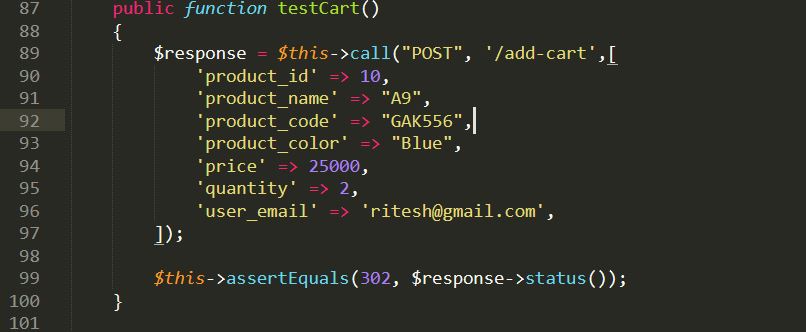


Figure 41: Test Case for cart

****

Figure 42: Unit Testing for cart

***Black Box Testing:***

****

Figure 43: Black Box Testing Result for cart fails

1. **Order**

***Test log:***

|  |  |
| --- | --- |
| Objective | To order selected products from cart along with payment method and delivery options |
| Action | By ordering selected products |
| Expected Result | Display orders ordered by customers |
| Actual Result | Thank you page showing message about ordered product |
| Conclusion | Successful |

Table 9: Test Log for Order

***Unit Testing:***

****

Figure 44: Test Case for Order

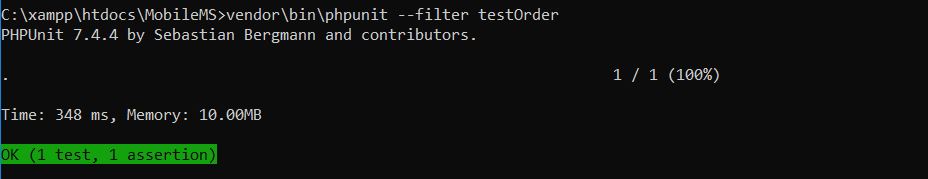
****

Figure 45: Unit Testing Result for Order

***Black Box Testing:***

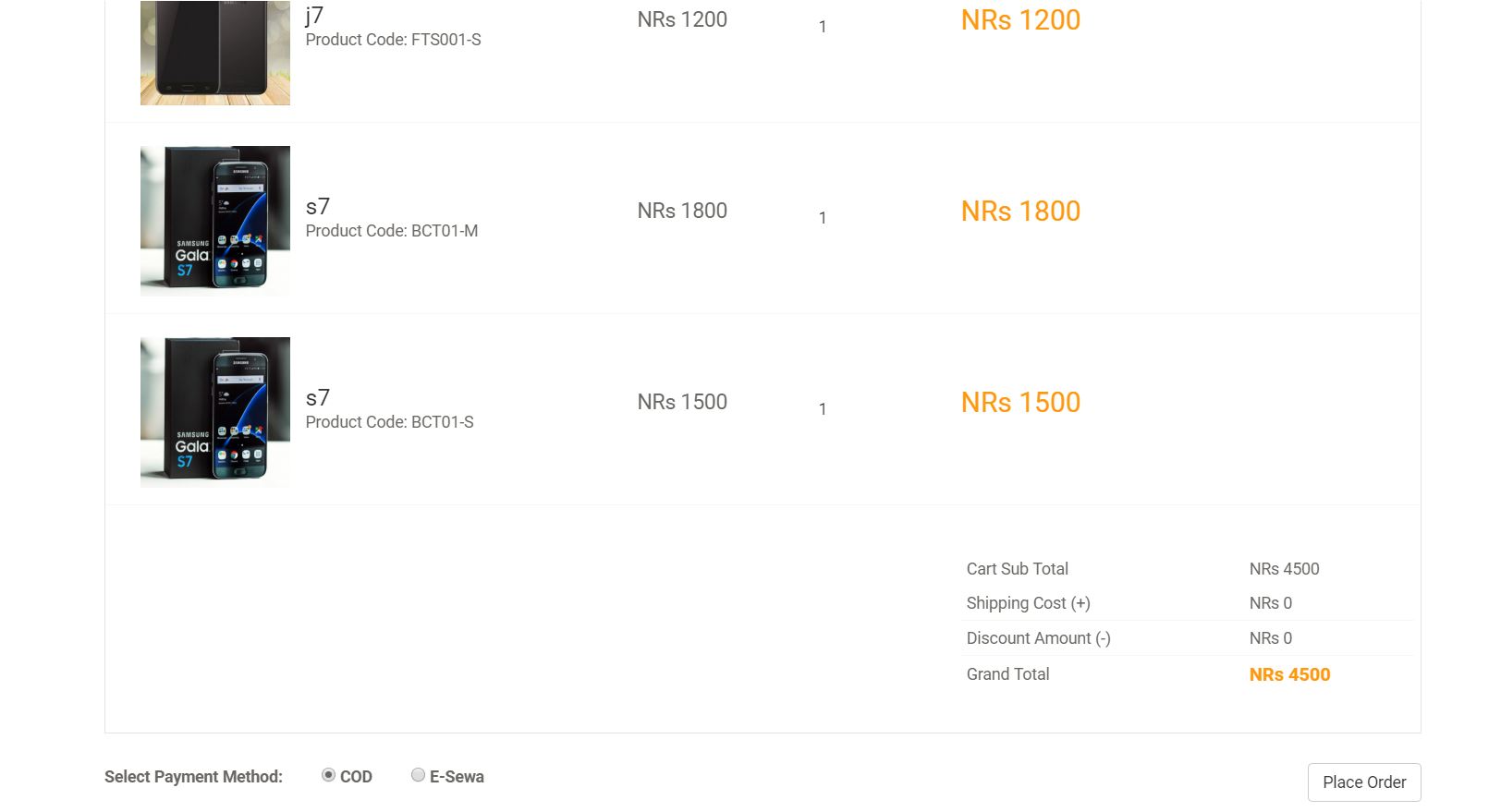


Figure 46: placing Order

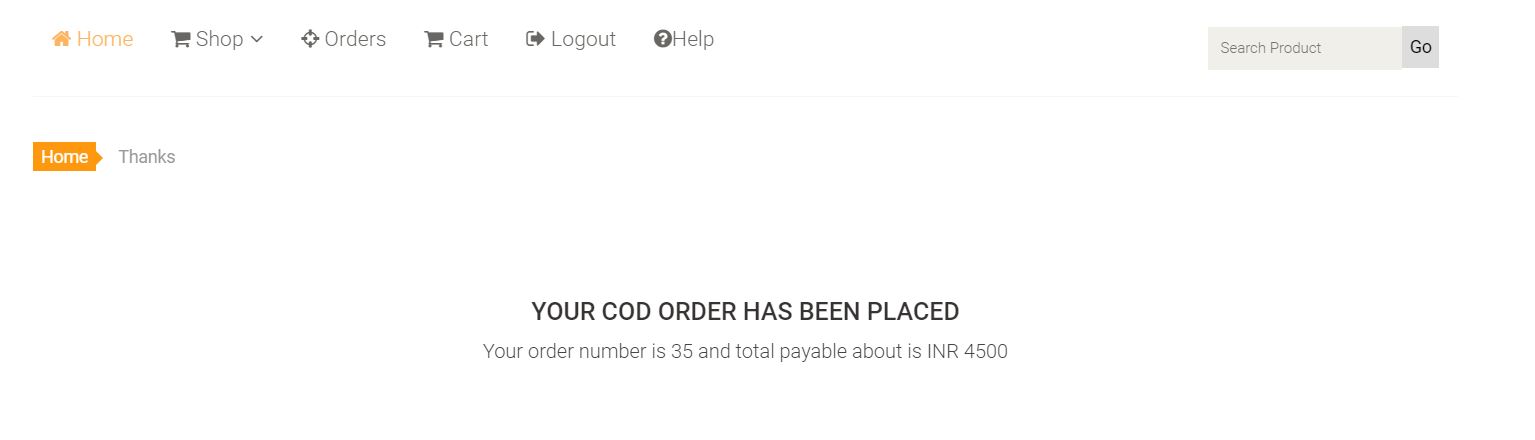
****

Figure 47: Order success

1. **Search**

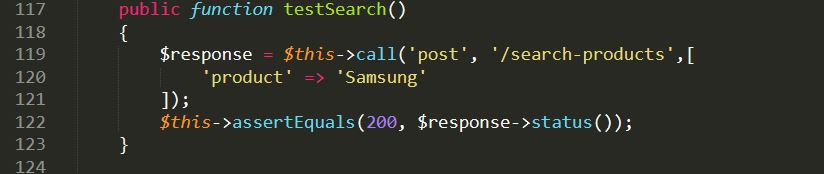
***Test log:***

|  |  |
| --- | --- |
| Objective | To allow to enter search option |
| Action | By clicking on search button it allows customer to view products. |
| Expected Result | Checks expected search result. |
| Actual Result | User views the exact search word. |
| Conclusion | Successful |

Table 10: Search test log

***Unit Testing:***

Figure 48: Search test case

****

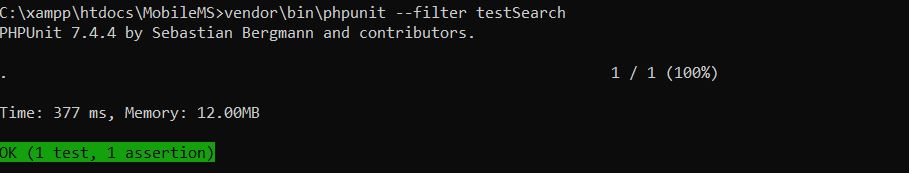
****

Figure 49: search Unit Testing

***Black Box Testing:***

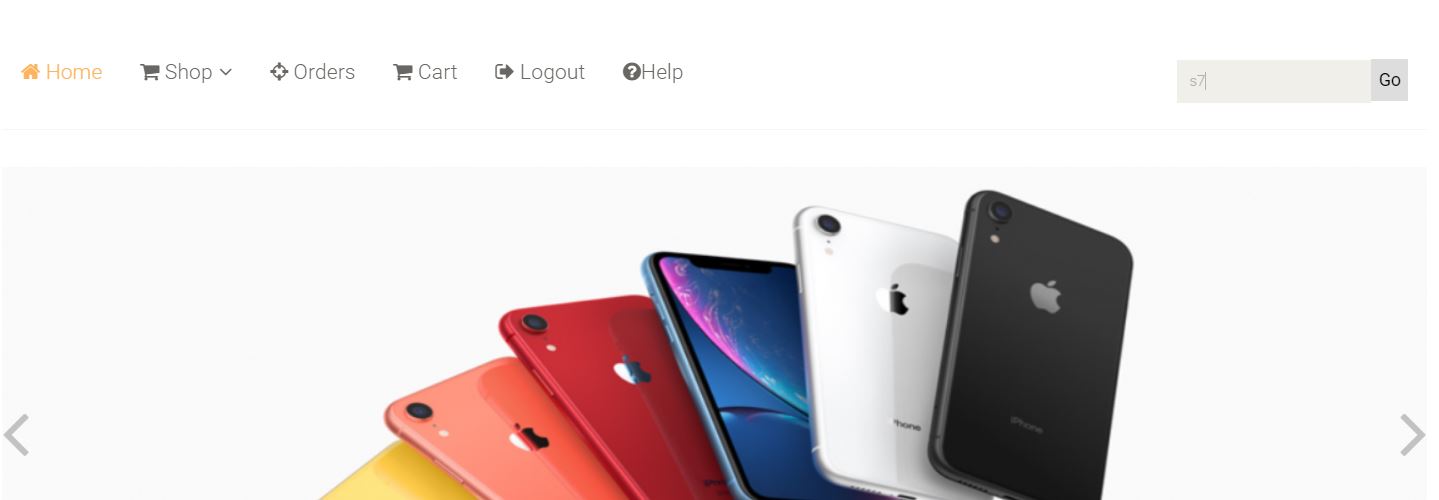


Figure 50: search s7

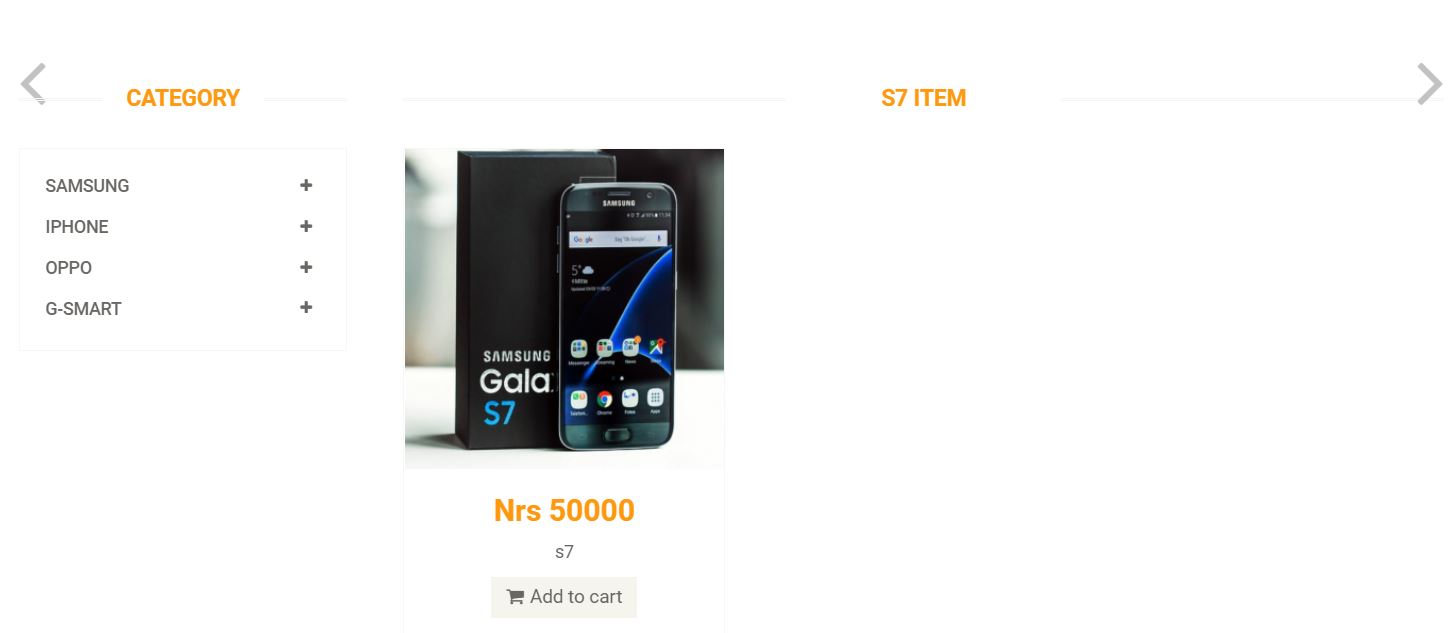
****

Figure 51:Search pass

**X. View order**

***Test log:***

|  |  |
| --- | --- |
| Objective | To see all orders pending to be delivered |
| Action | By showing all pending orders |
| Expected Result | Display orders that are pending to be delivered |
| Actual Result | Orders page showing pending orders |
| Conclusion | Successful |

***Unit Testing:***

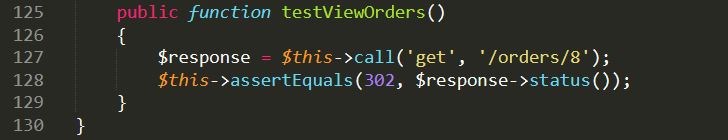
******

Figure 52: Test Case for view order

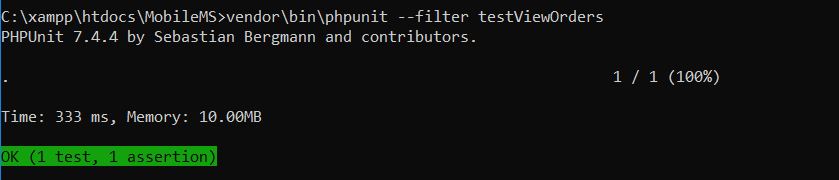
******

Figure 53: view order Unit Test pass

***Black Box Testing:***

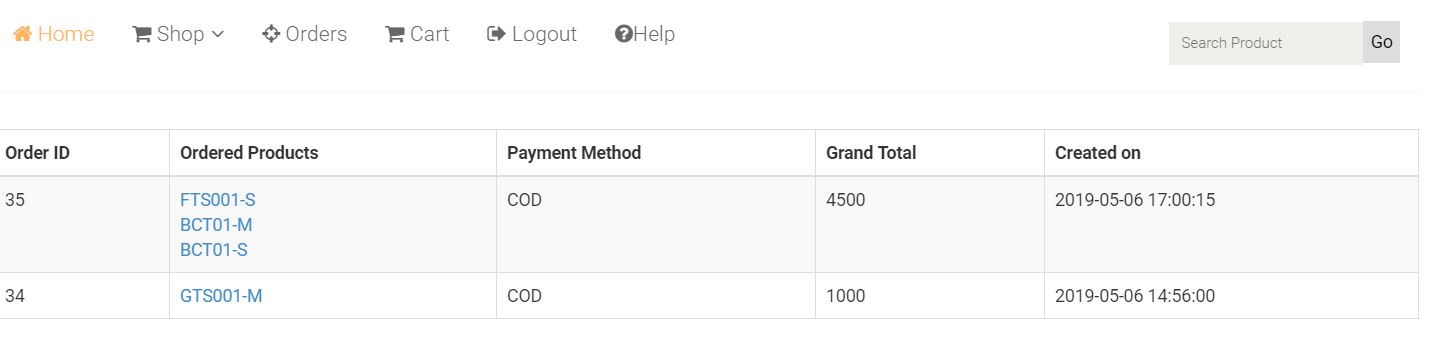
****

Figure 54: view order of Black Box Testing

# Chapter 6: Other Project Issues

# **Risk Management:**

Risk management is used to manage loss which can be anything like cost increase while developing, poor software quality, extending time period. Its main features are:

* Risk Identify
* Cut the impact of risk
* Monitor risk

|  |  |
| --- | --- |
| **Likelihood** | **Value** |
| Low | 1 |
| Medium | 2 |
| High | 3 |

Table 1: Likelihood

|  |  |
| --- | --- |
| **Consequences** | **Value** |
| Very low | 1 |
| Low | 2 |
| Medium | 3 |
| High | 4 |
| Very high | 5 |

Table 2: Consequences

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risks** | **Likelihood** | **Consequence** | **Impact** | **Actions** |
| Failed system | 1 | 4 | 4 | Maintain proper back up |
| Requirement alteration | 3 | 4 | 12 | Need to give proper contract. |
| Hacking | 2 | 2 | 4 | Proper security should be implemented. |
| Improbable budget | 2 | 4 | 8 | Estimate proper plan and budget |
| Virus and spam | 1 | 2 | 2 | Trusted antivirus should be install and should block unauthorized access. |
| Natural disaster | 5 | 4 | 5 | Keep backup also use cloud back up. |
| Quit job | 2 | 3 | 6 | Need of proper contract to the staff. |

Table 3:Risk Management

# **Configuration Management:**

Software management is a discipline containing of procedures and methods often used by administrations to manage the changes presented to its software products.

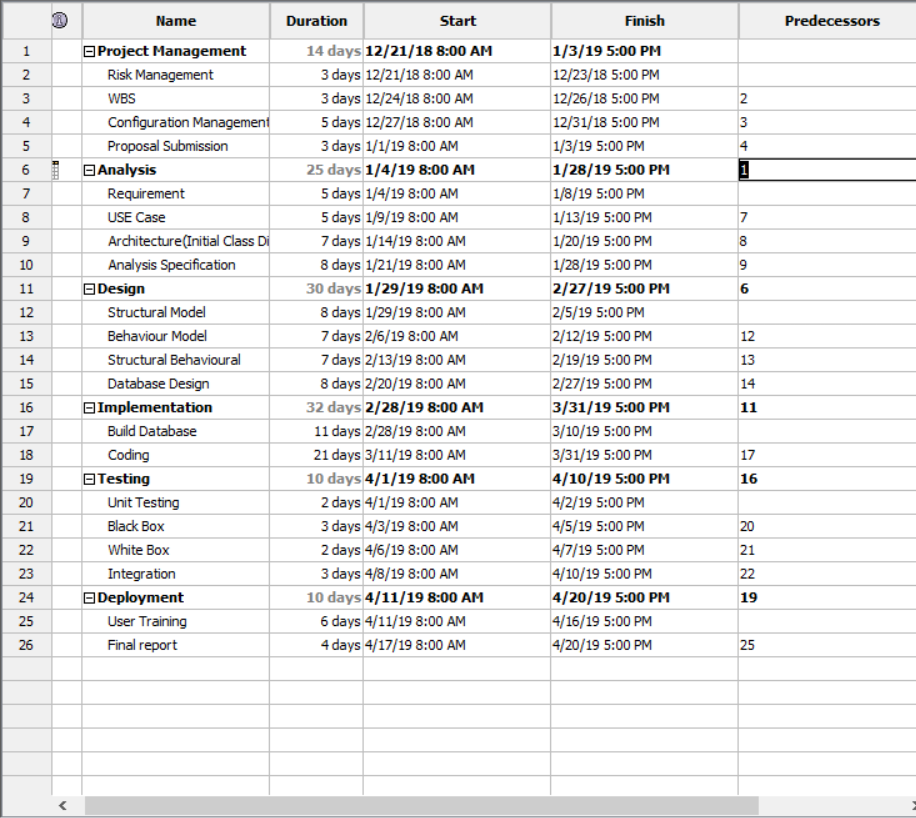
Figure 55: Tree Chart

Main aims to control changes introduced to complex software systems through dependable version selection and version control.

**Github**

## **Scheduling/Gantt Chart:**

The time schedule for my project is represented in a diagram of Gantt chart with milestones.

****Figure 56: Schedules Table

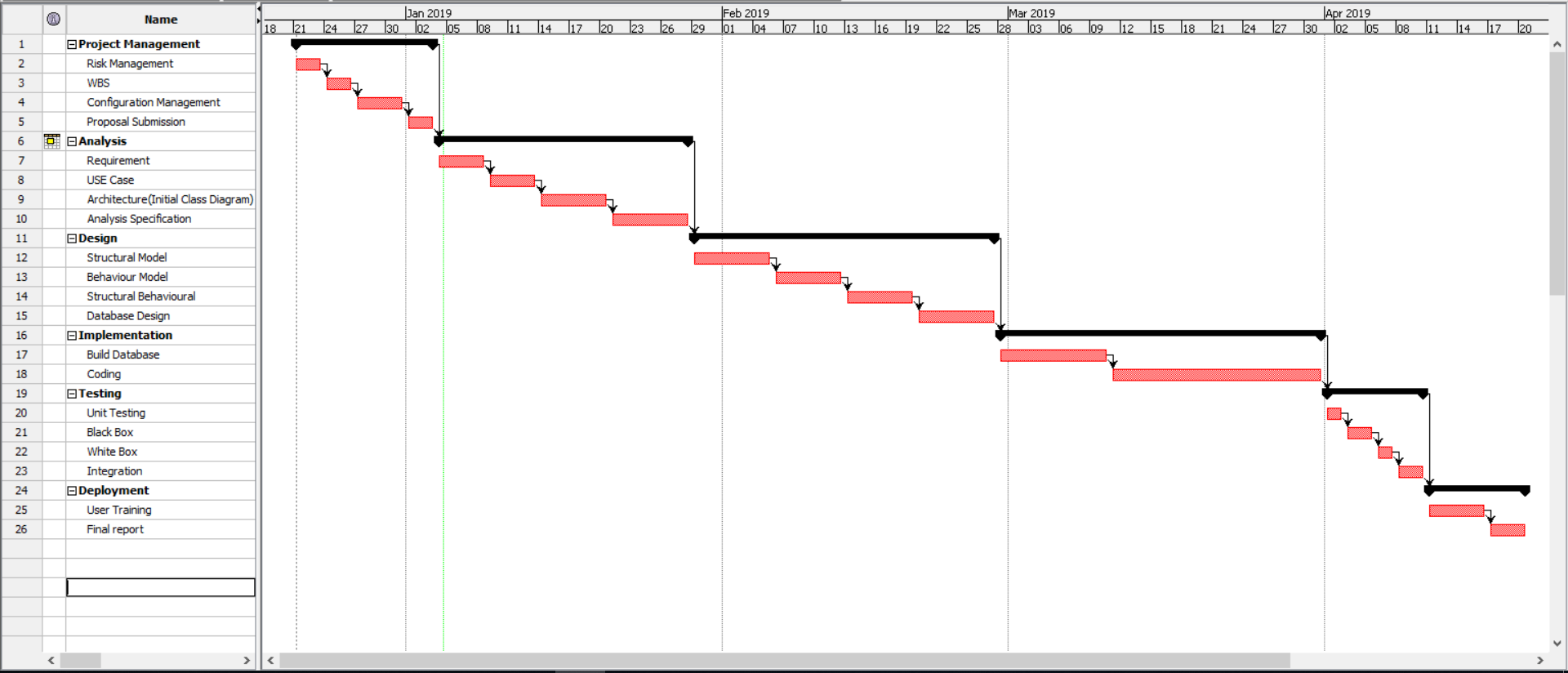
****

Figure 57:Gant Chart

**Future work:**

In future I am hoping to add various functions. First of all, I will fully integrate esewa or khalti with this project. I was confused which payment method should I keep for this project. Due to limited time I couldn’t create various charts. I could improve or create more and more charts, bar graphs of monthly sales, yearly sales with profit or loss. For this annual profit needs to be calculated which is the outcome of Purchase and sale. If purchase is greater than profit is generated and shown in chart with amount. I am looking forward to creating this feature in future. This website is very user-friendly and responsive. It is responsive for various devices but due to lack of time I couldn’t do responsive tests in various device. I will make this application responsive, so it can support on almost every device.

I am also looking forward to creating mobile application for this system through this web application can be executed. Now a day’s android had taken the market. So, I will create android application for this project.

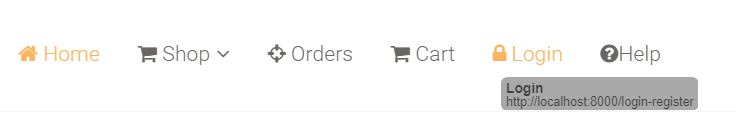
## Limitation:

With overall the most prioritized requirements this project has been completed. The project lacks some features like online payment though it is included it’s still got some issues with the transaction.

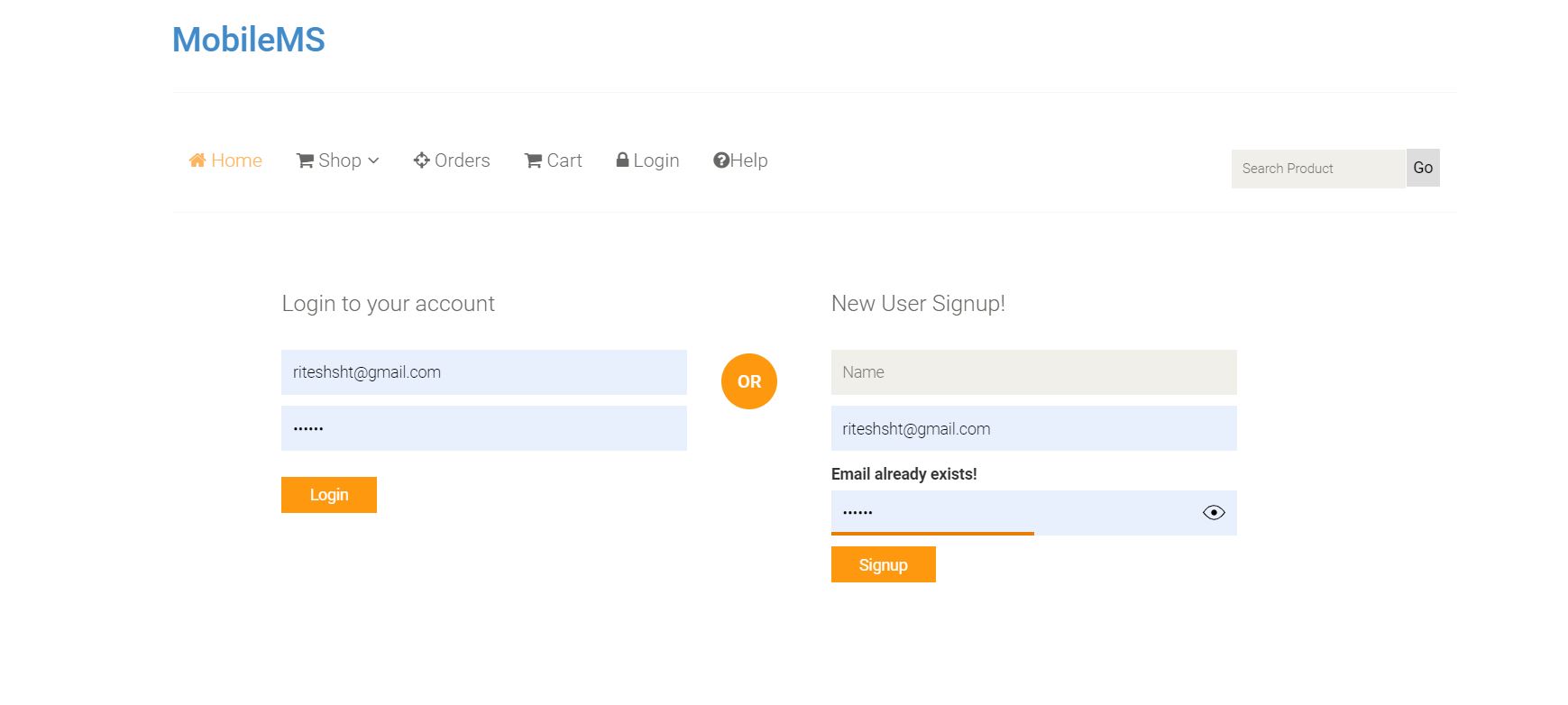
## User manual

1. **REGISTER/SIGN UP.**

* To register or Signup, Click on “LOGIN” in the navigation bar.

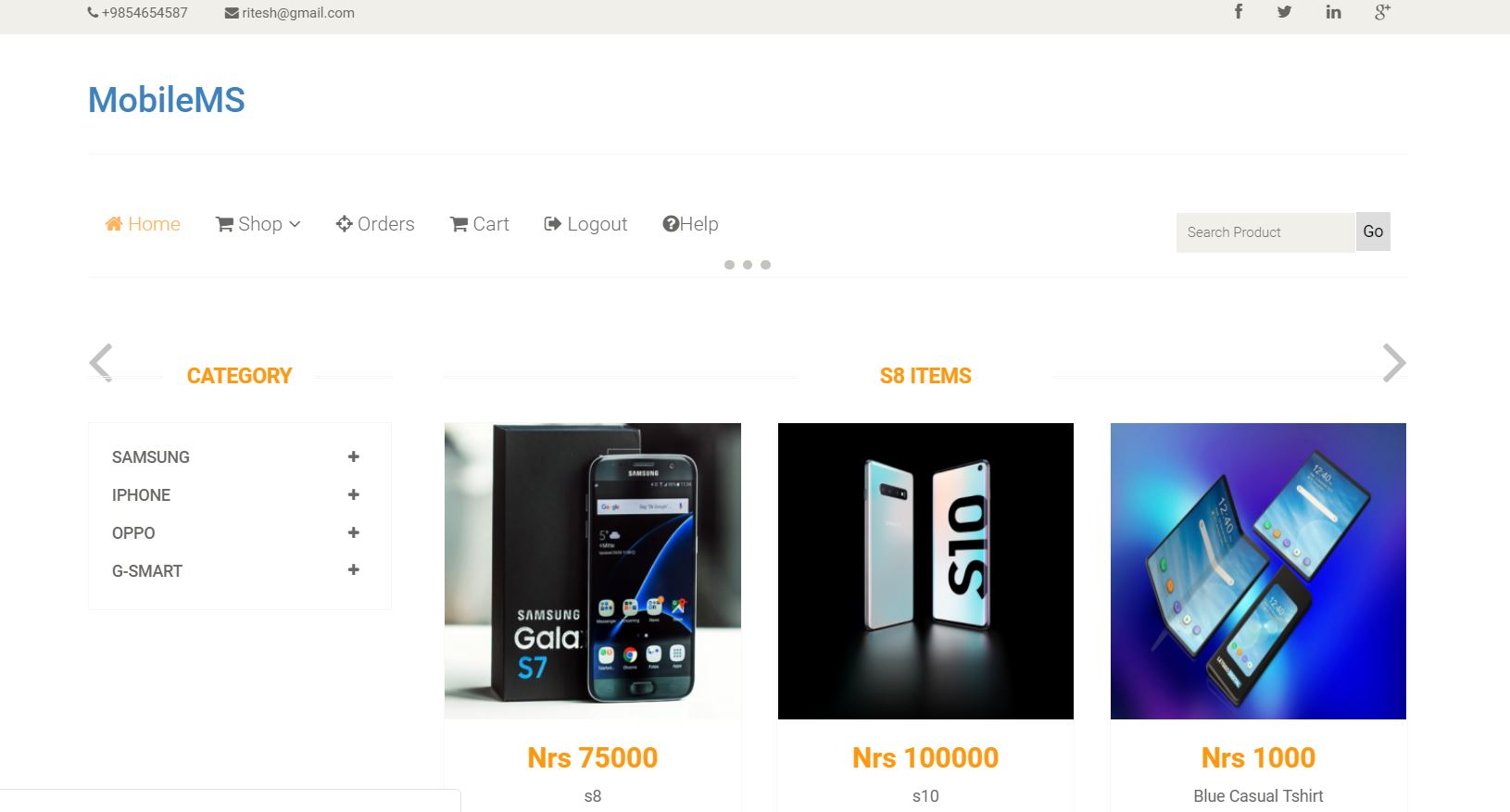


* The form is displayed below.
* Fill all the details in “New User Signup” to get registered for unregistered users.
* Or fill the details in “Login to your account” after for registered users.



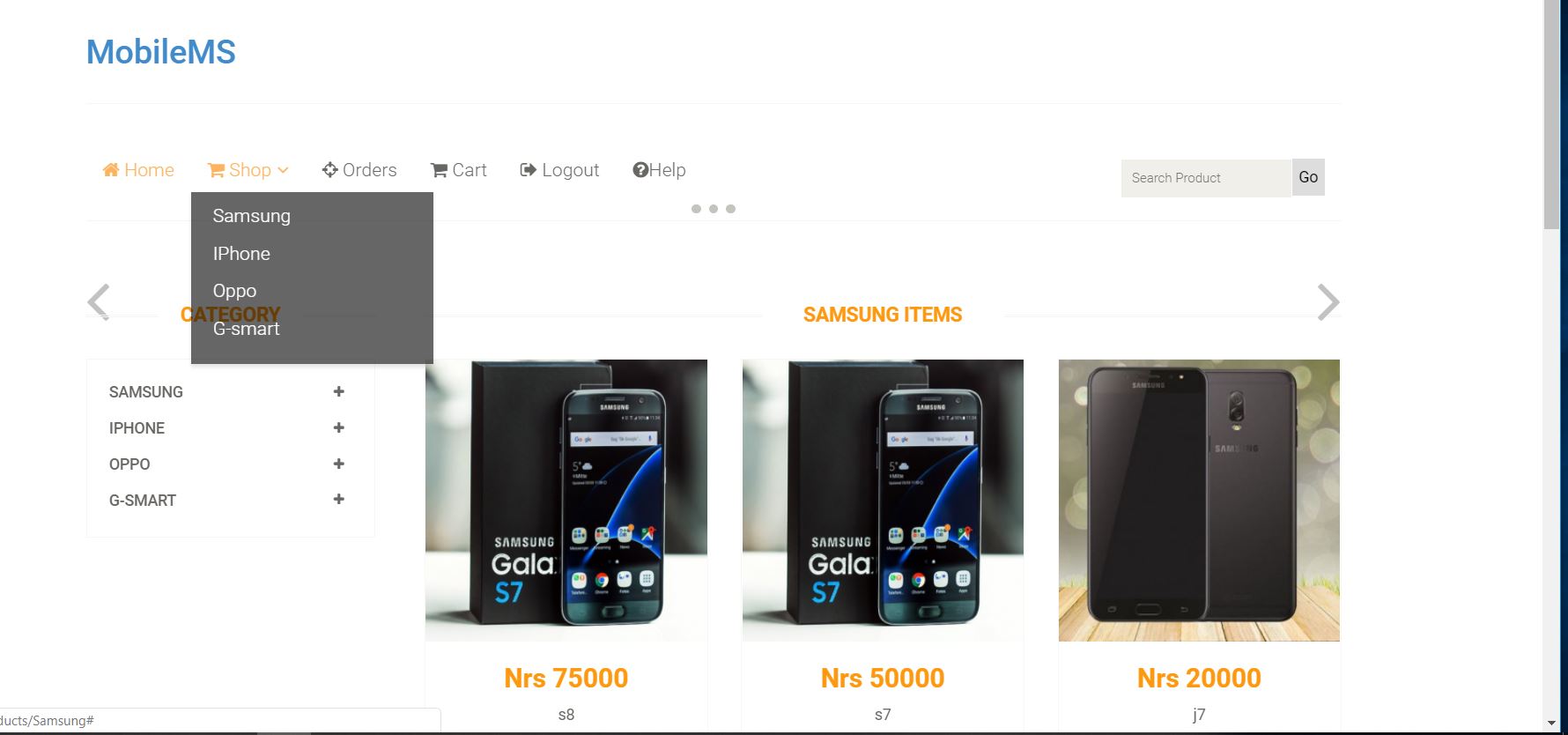
1. **HOMEPAGE**

* The home page is available for both registered or registered use. After successfully log in user can gain access to buy the product.



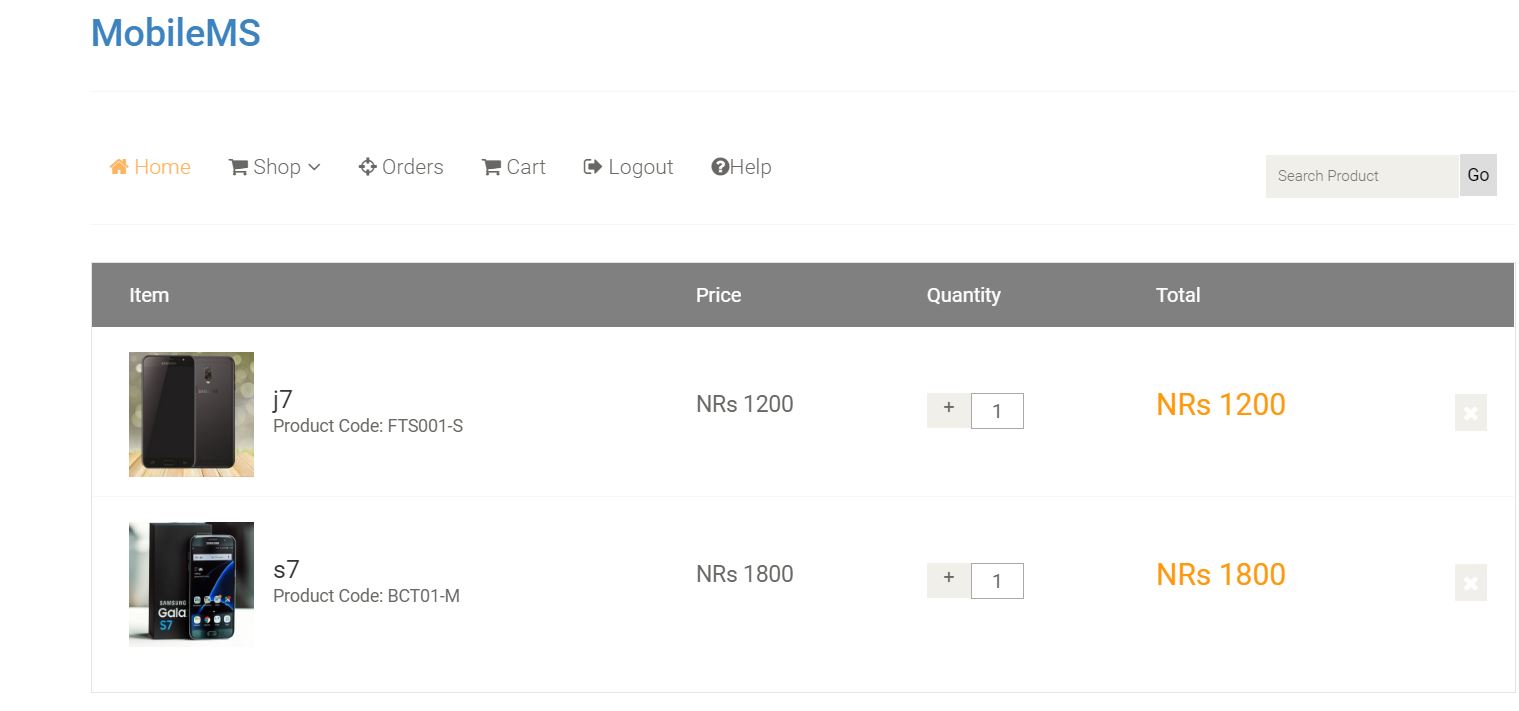
1. **SHOP**

* To view the shop page, click on “SHOP” button, shown below.
* After clicking the button, different brands option will be displayed accordingly.



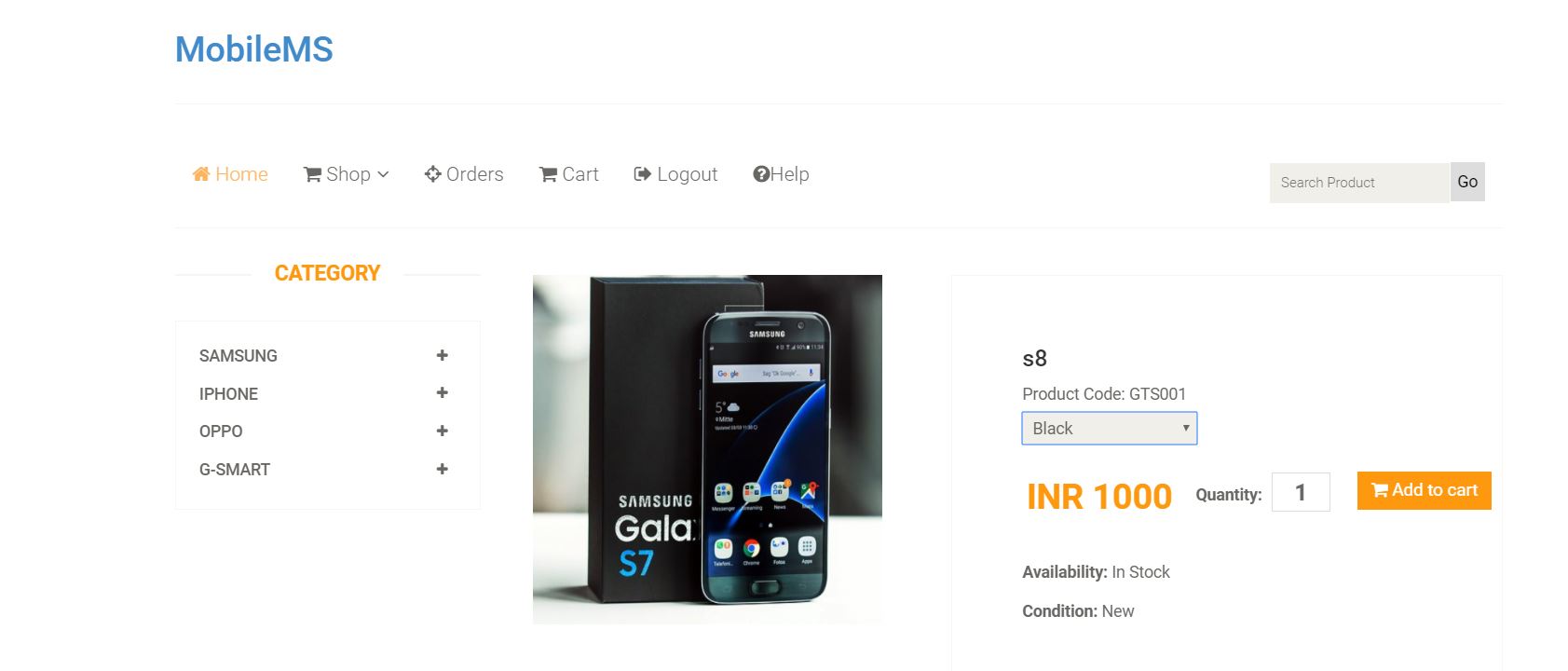
1. **CART**

* To view your cart, before purchasing the product, click on “Cart” in the navigation bar.

****

1. **TO PURCHASE**

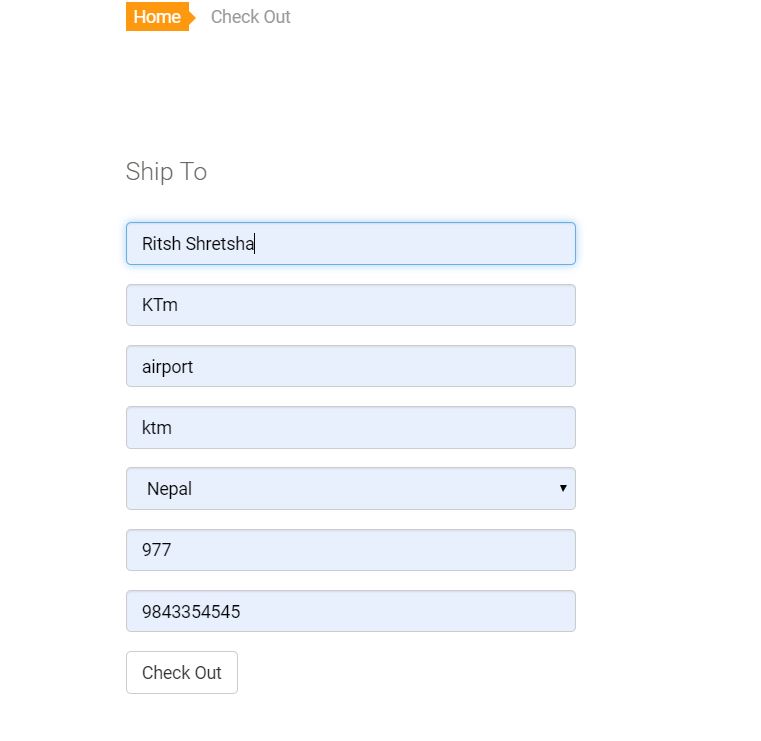
* Click on any item to purchase, where the price and the availability of the item will be displayed which is shown below.
* Item quantity can also be changed accordingly.
* Click on “Add to Cart” button, for further process.

****

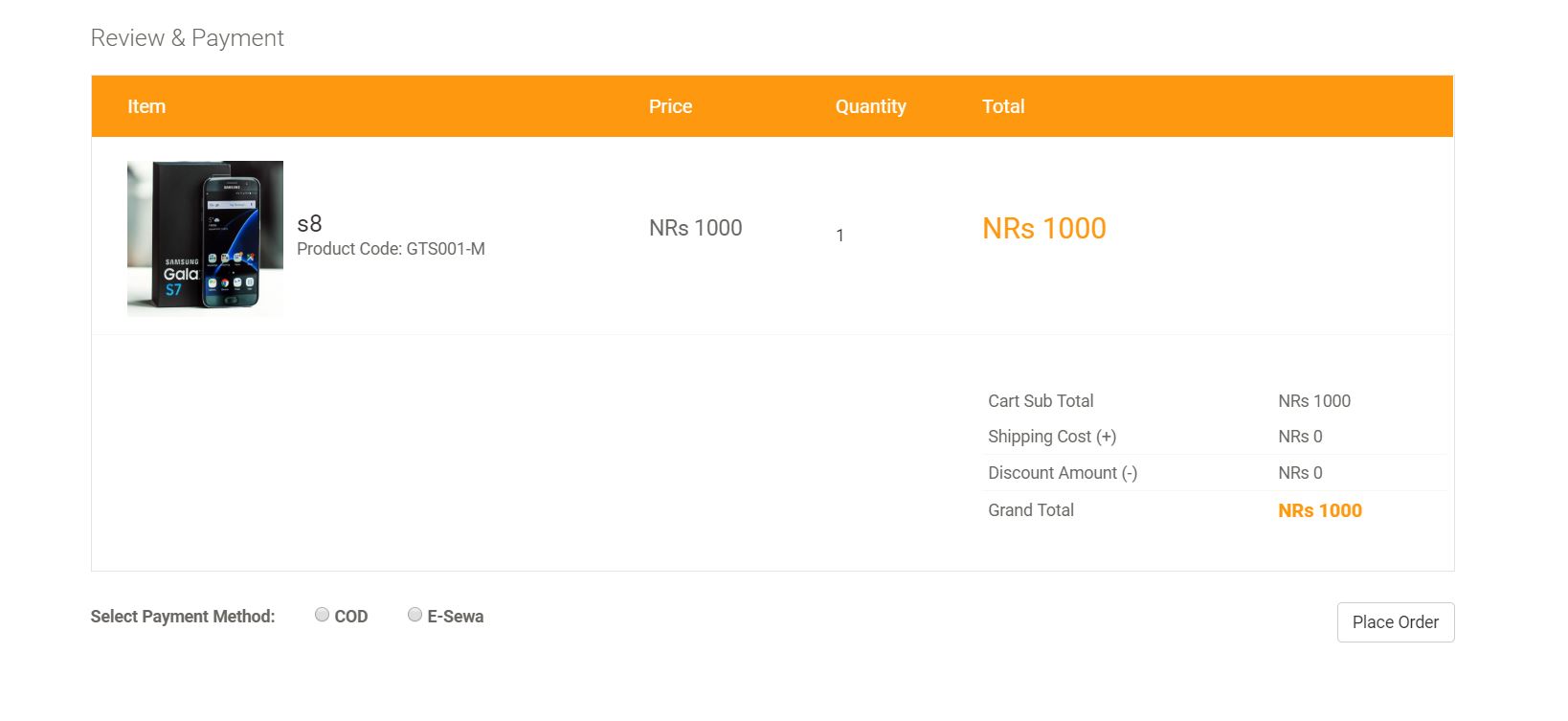
* The confirmation page will be displayed which is shown below.
* Click on “Check out” button, to confirm the purchase.

****

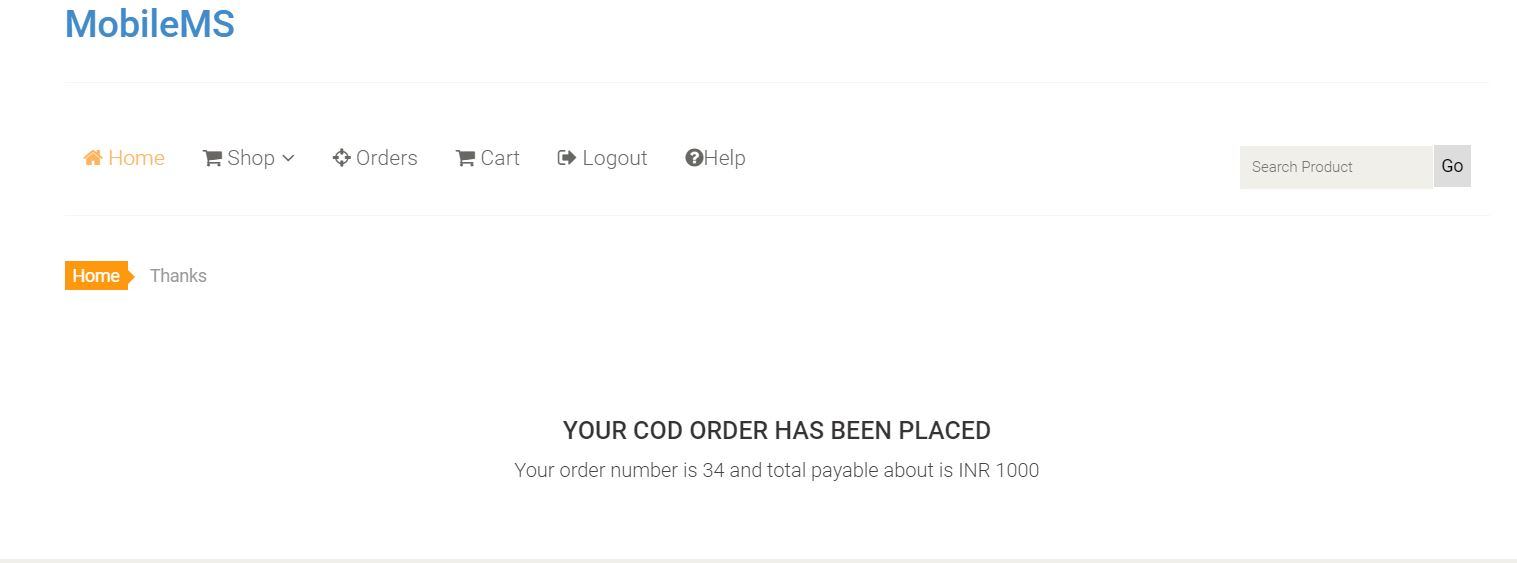
* Fill up all the required details, to get the ordered item, shown below.

****

* Click on “Check out” to select the payment method accordingly.
* After confirming the details, click on “Place Order” button to order the product.

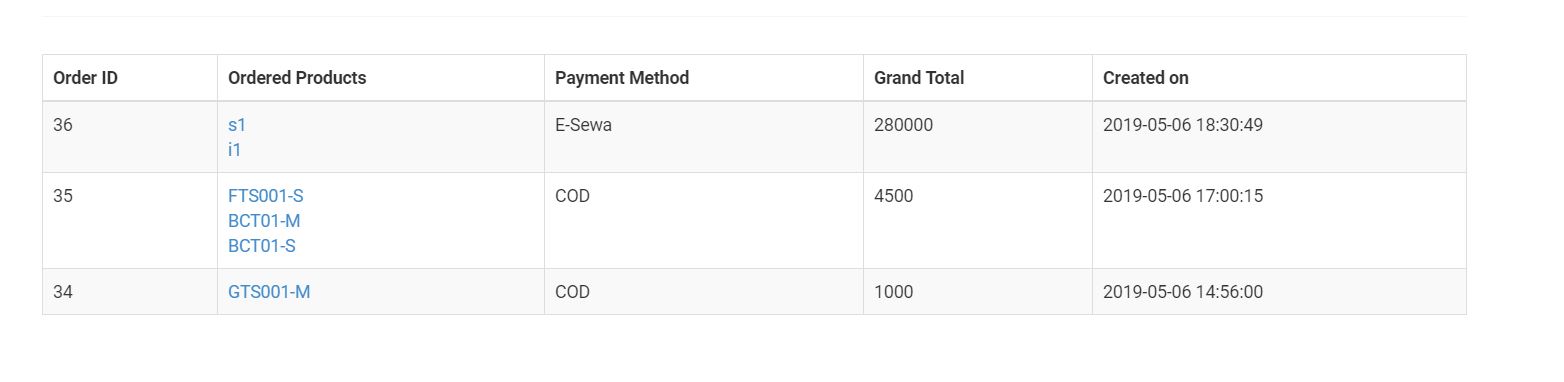
****

* After placing the order, following information given below will be displayed.

****

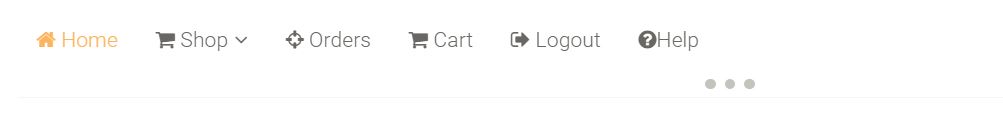
1. **ORDER**

* To view your orders, click on “Order” in the navigation bar, shown below.

****

1. **LOGOUT**

* To logout, form the page, click on “Logout” button from the navigation bar.

****

# Chapter 7: Conclusion

In this module “Project” we get to choose what we want to create as our final project. I choose PHP as programming language for developing ecommerce web application for **Mobile Management System** because PHP is very suitable for this type of project. PHP is server-side scripting language. This web application of Mobile MS was created with continuous research and development. This was the biggest project I had ever worked in. It took me lot of time for developing design and back-end of this project. This web application of Mobile MS is an ecommerce online application were anyone can easily browse and search various products available. Visitors can search products as well as sort different products as of price, categories and brand which helps them to search their products easily. Visitors can add products in cart for further purchase but requires to logged in to purchase that products. The need to login otherwise they won’t be able to order products which are added to cart.

Customer have various other functions such as they can choose payment options from cash on delivery and esewa option and they can also choose delivery options from normal delivery and fast delivery. There are some verifications after the products ordered by customer which is done by admin or other staffs. This web application is very user-friendly and responsive. Admin has various functions such as adding/updating/products, managing orders, verifications, manage other staffs/editors etc. Admin can view sales by month, yearly sales chart and many mores. Admin has been provided with lots of features. Anyone without knowledge of programming can easily run this entire web application.

# Reference

Bootstrap Grid System. 2018. *Bootstrap Grid System*. [ONLINE] Available at: [https://www.w3schools.com/bootstrap/bootstrap\_grid\_system.asp.](https://www.w3schools.com/bootstrap/bootstrap_grid_system.asp) [Accessed 10 march 2019].

Responsive Web Design Media Queries. 2018. *Responsive Web Design Media Queries*.

[ONLINE] Available at: [https://www.w3schools.com/css/css\_rwd\_mediaqueries.asp.](https://www.w3schools.com/css/css_rwd_mediaqueries.asp)

[Accessed 2 April 2019].

Bootstrap Carousel. 2018. Bootstrap Carousel. [ONLINE] Available at:

https://www.w3schools.com/bootstrap/bootstrap\_carousel.asp. [Accessed 5 April 2019].

JavaScript JSON. 2018. JavaScript JSON. [ONLINE] Available at:

https://www.w3schools.com/js/js\_json.asp. [Accessed 10 April 2019].

Bootstrap Carousel. 2018. Bootstrap Carousel. [ONLINE] Available at:

https://www.w3schools.com/bootstrap/bootstrap\_carousel.asp. [Accessed 12 April 2019].

CSS Animations. 2018. *CSS Animations*. [ONLINE] Available at:

[https://www.w3schools.com/css/css3\_animations.asp.](https://www.w3schools.com/css/css3_animations.asp) [Accessed 13 April 2019].

PHP 5 Tutorial. 2018. *PHP 5 Tutorial*. [ONLINE] Available at:

[https://www.w3schools.com/php/default.asp.](https://www.w3schools.com/php/default.asp) [Accessed 15 April 2019].

JavaScript Tutorial. 2018. *JavaScript Tutorial*. [ONLINE] Available at: [https://www.w3schools.com/js/default.asp.](https://www.w3schools.com/js/default.asp) [Accessed 15 April 2019].

JavaScript Tutorial. 2018. *JavaScript Tutorial*. [ONLINE] Available at: [https://www.w3schools.com/js/default.asp.](https://www.w3schools.com/js/default.asp) [Accessed 16 April 2018].

Creating a User Login System with PHP and MySQL - Tutorial Republic. 2018. *Creating a User Login System with PHP and MySQL - Tutorial Republic*. [ONLINE] Available at: [https://www.tutorialrepublic.com/php-tutorial/php-mysql-login-system.php.](https://www.tutorialrepublic.com/php-tutorial/php-mysql-login-system.php) [Accessed 20 April 2019].

# Chapter 9: Appendix

All the screenshots of implementation part are shown below:

***Laravel files on Sublime Editor:***

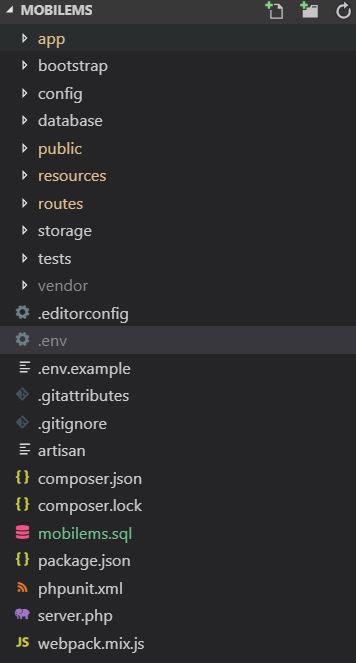


Figure 58: Laravel all files

***Database migrations:***

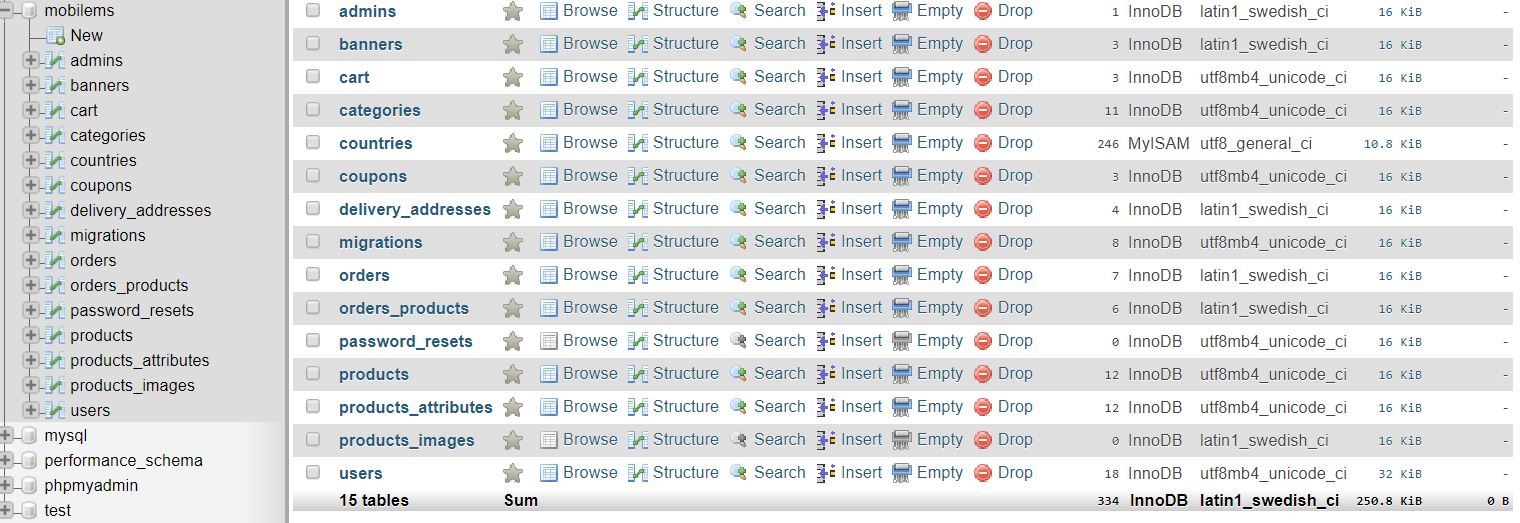


Figure 59: Database migrated in MYSQL

***Models on Laravel:***

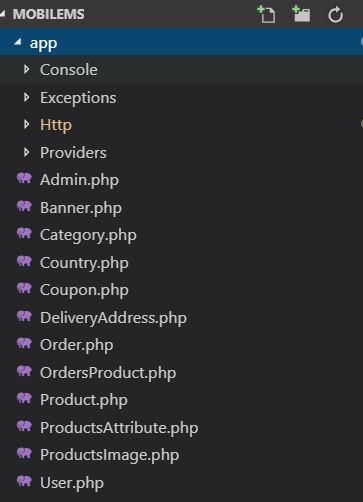
******

Figure 60: List of Models

***Controllers on Laravel:***

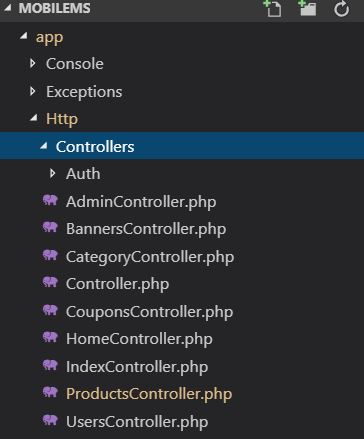
******

Figure 61: Controllers list

***Laravel views:***

******

Figure 62: Views list

***Routes:***

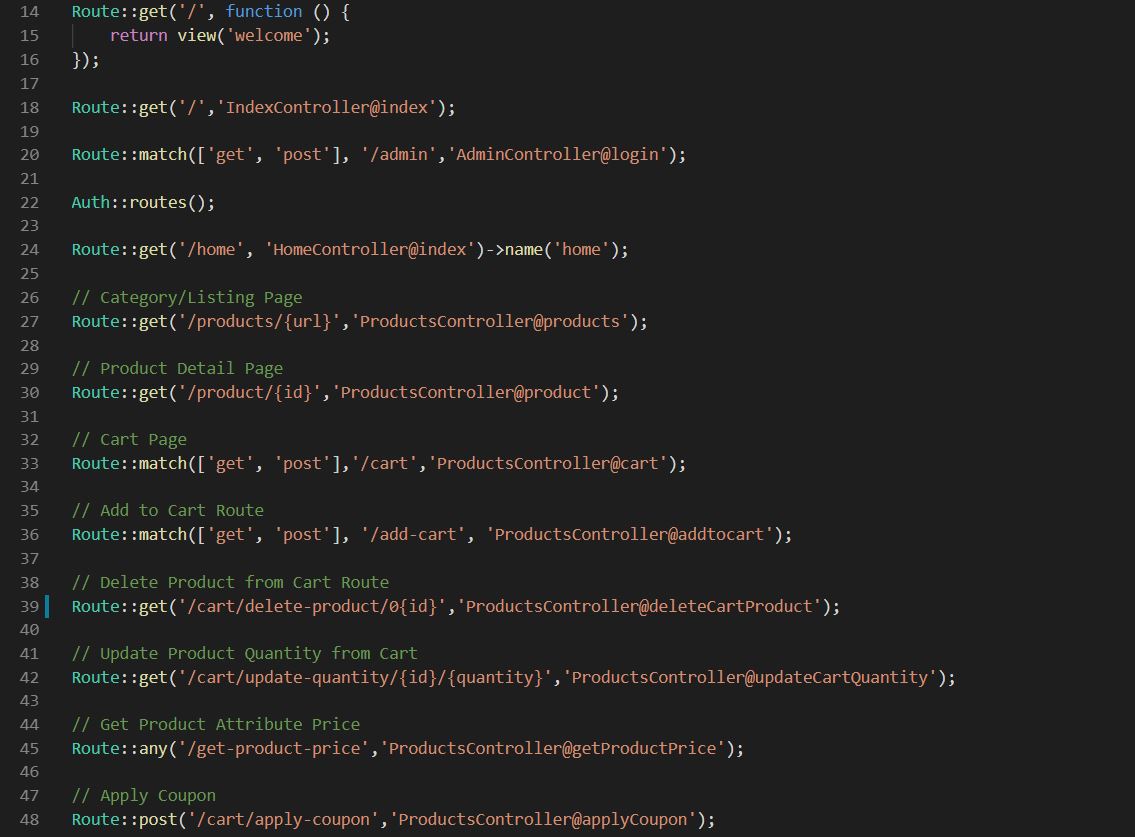
******

Figure 63: Routes

***UI Design***

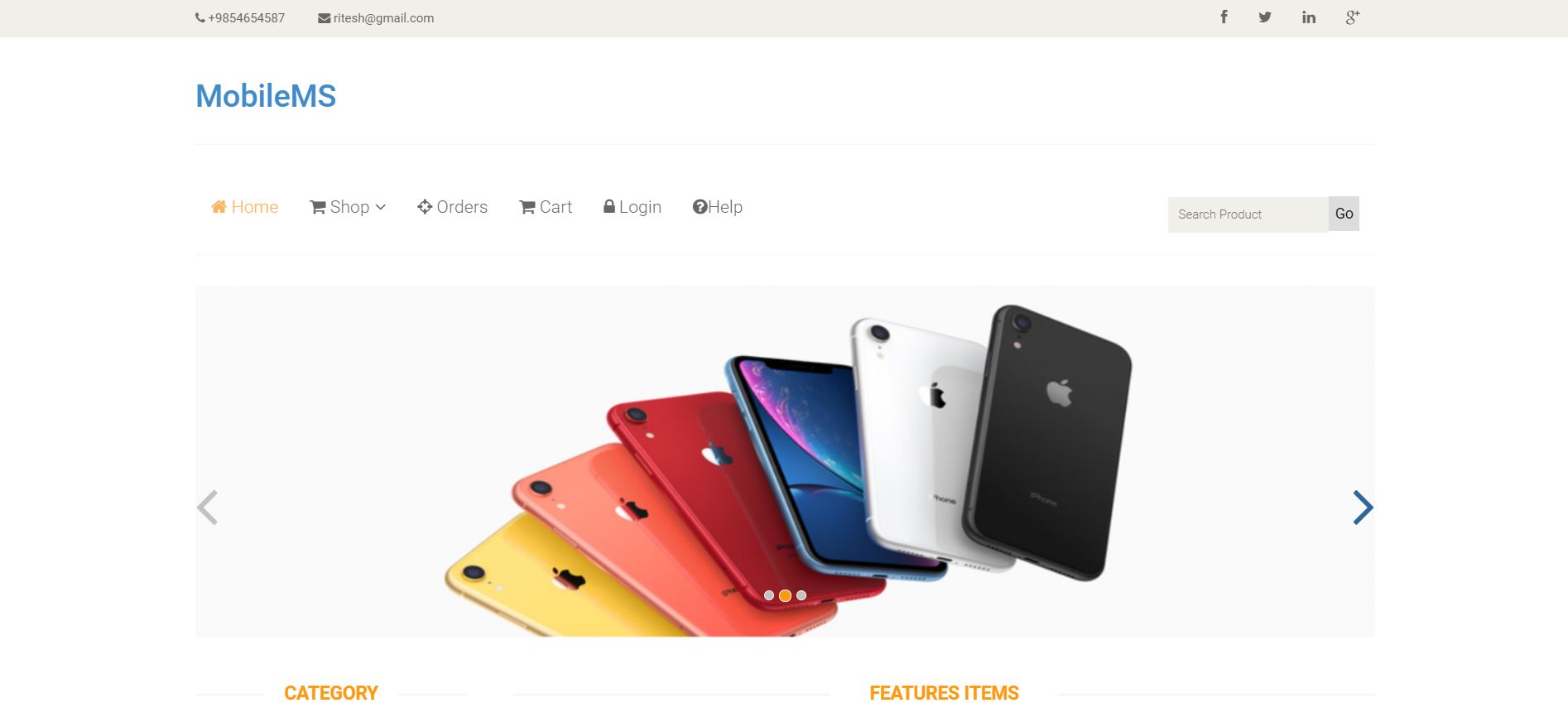
******

Figure 64: Home page

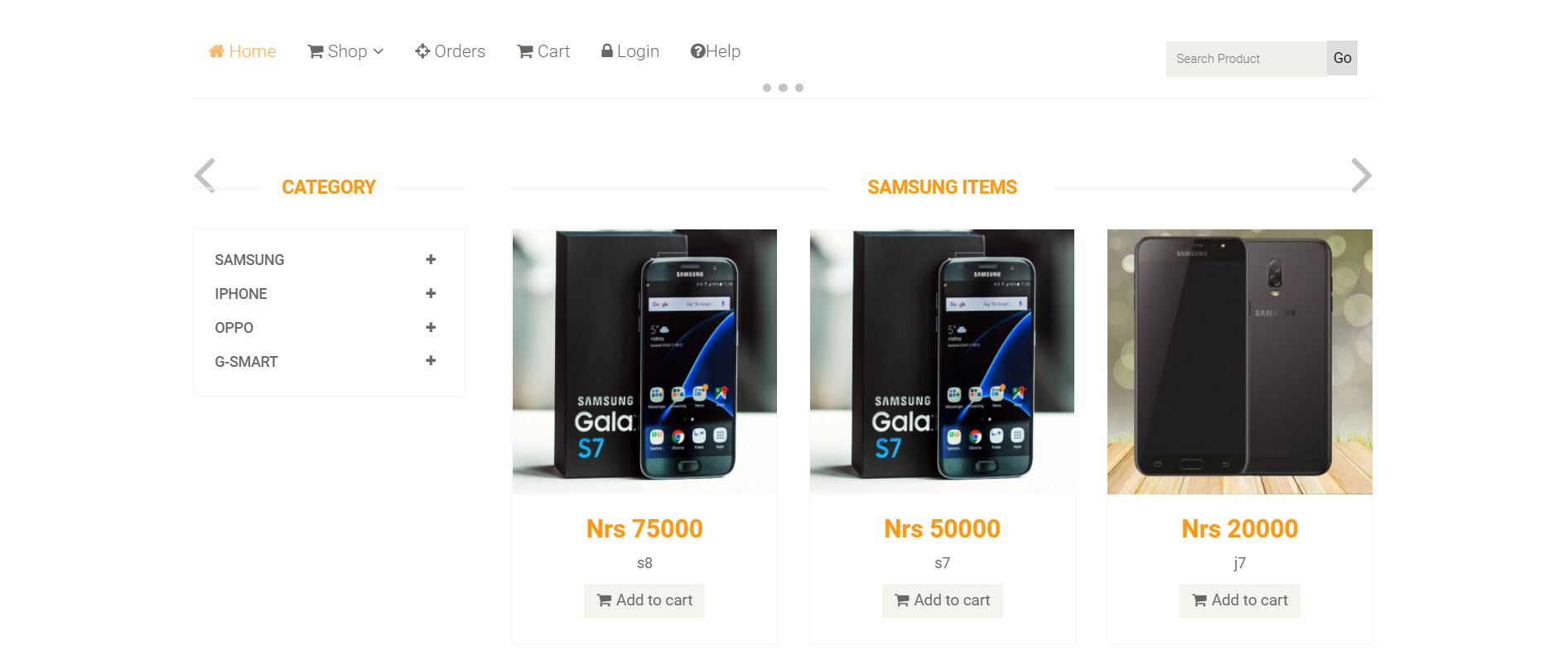
******

Figure 65:Gallery



Figure 66: Cart page

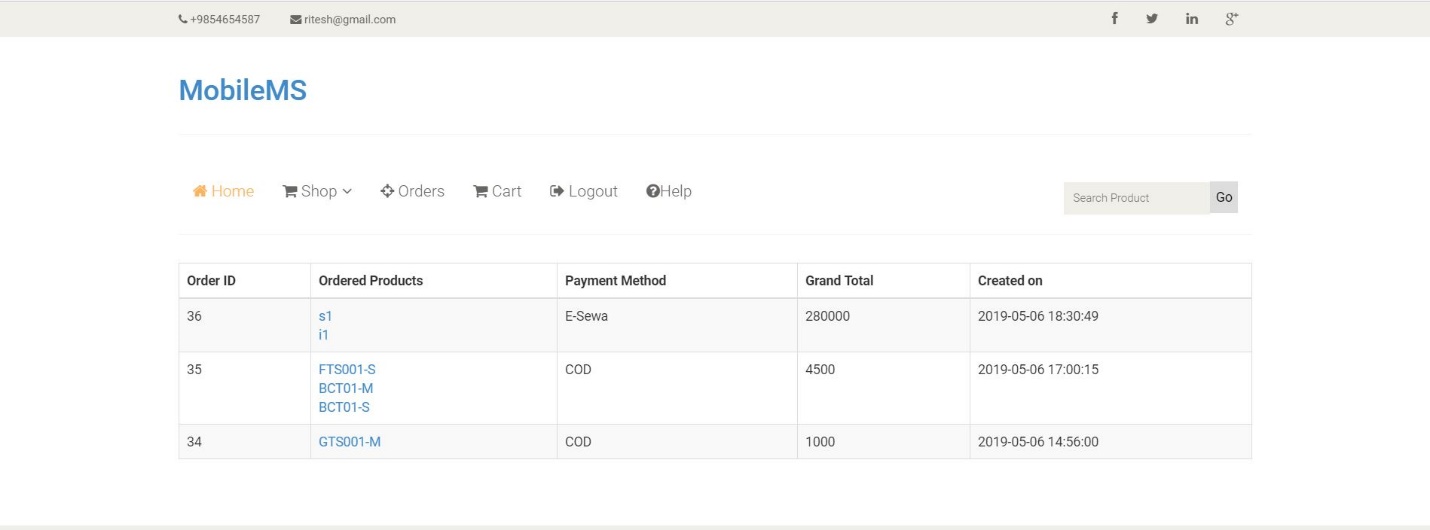


Figure 67:Order Page

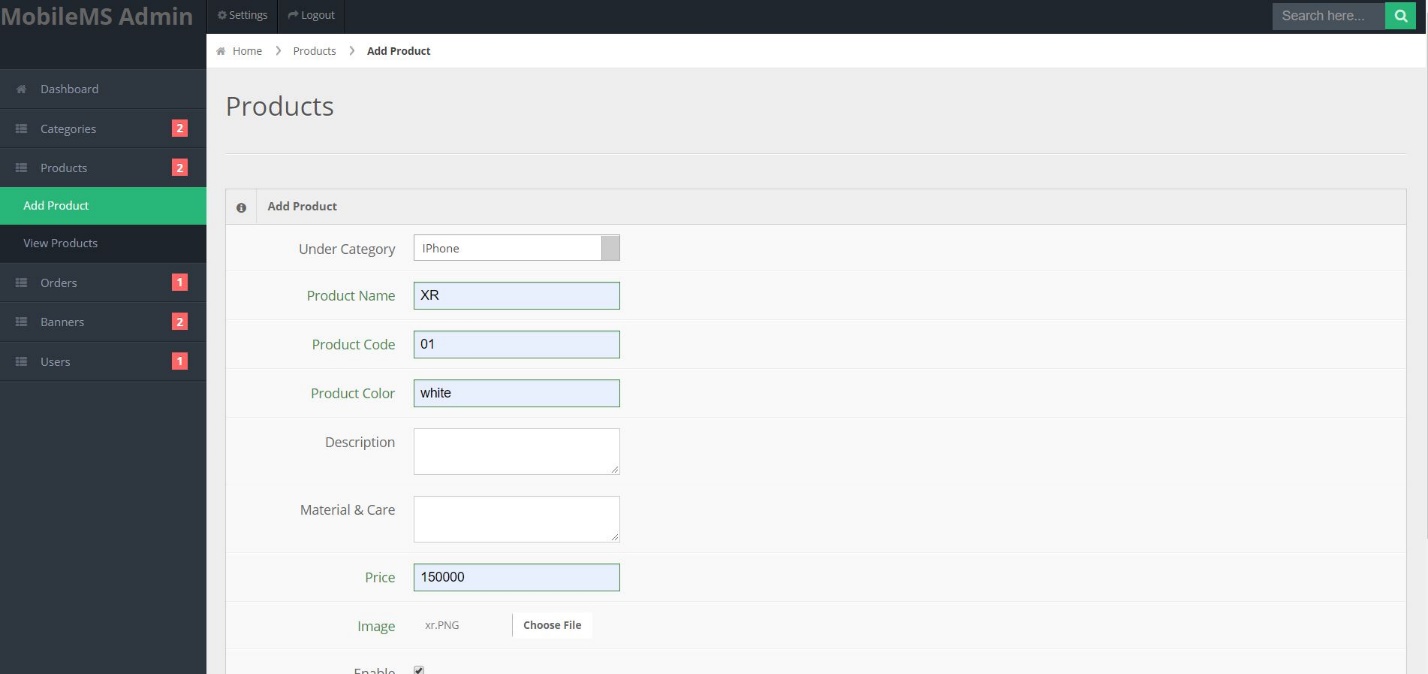


Figure 68:Add product(admin) interface

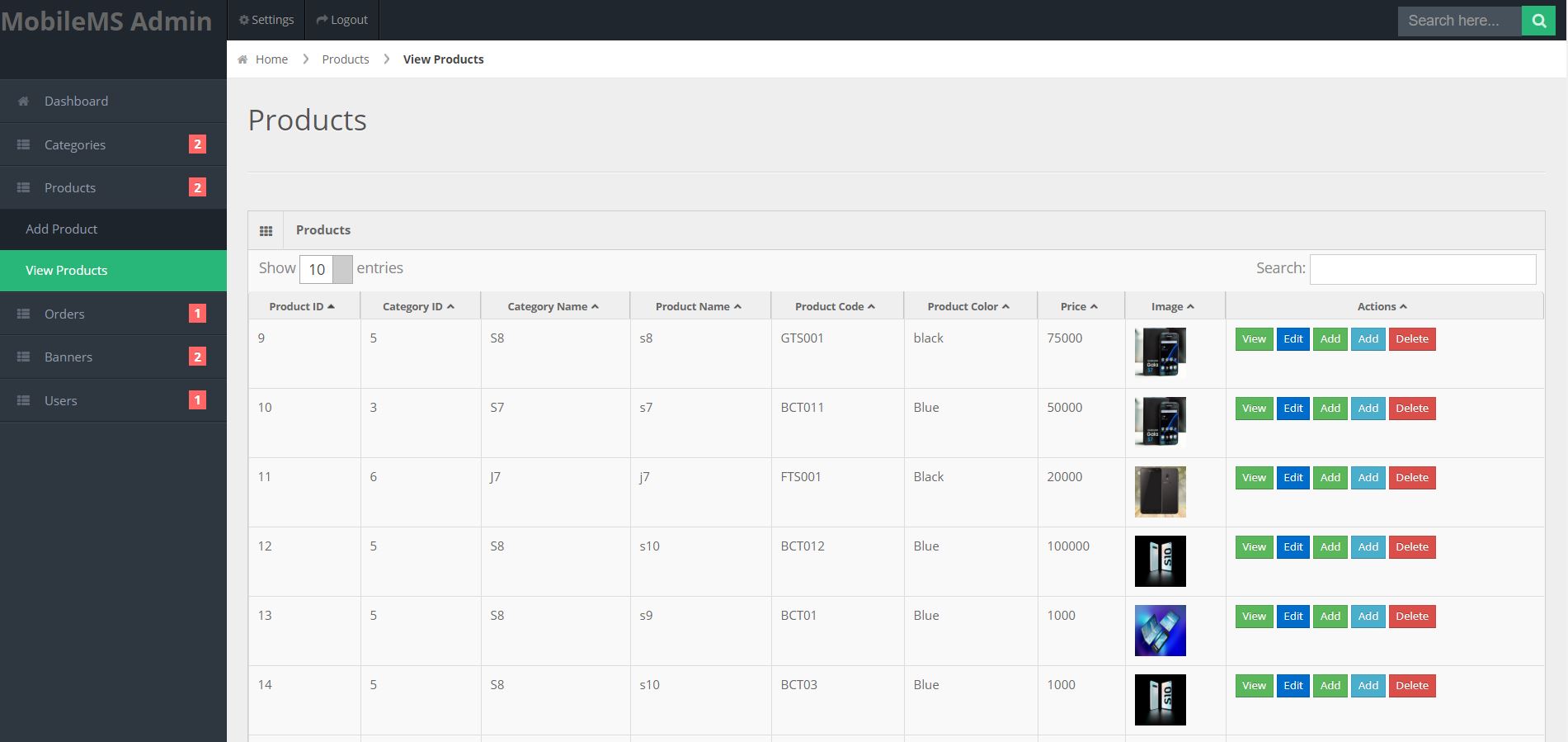


Figure 69:Added product interface

***Users Model:***

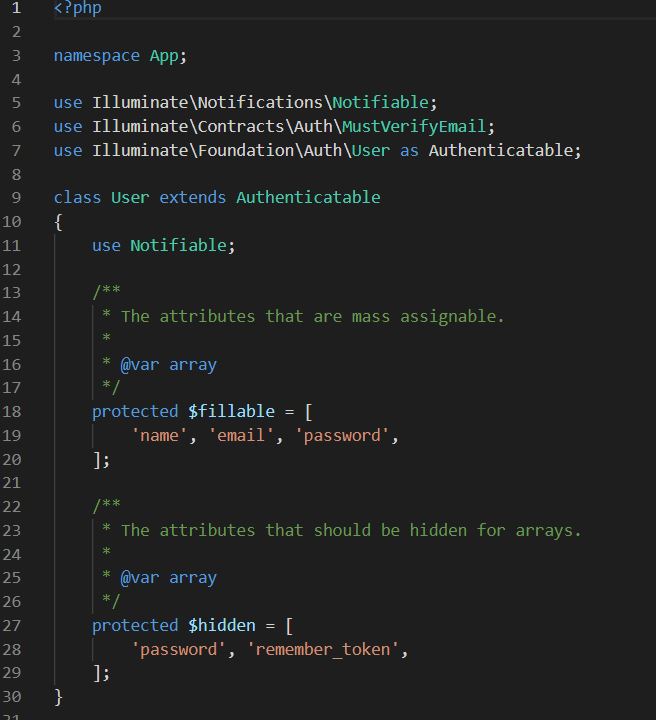
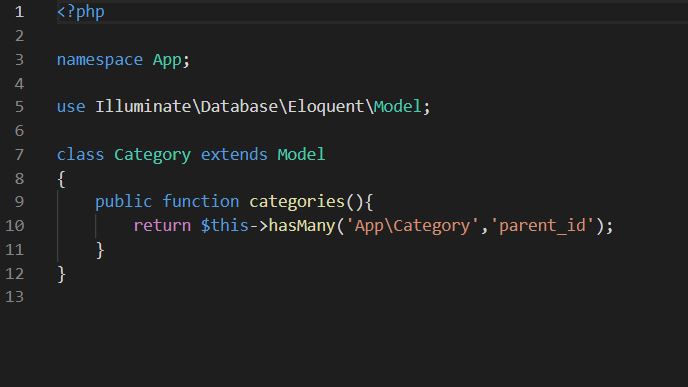
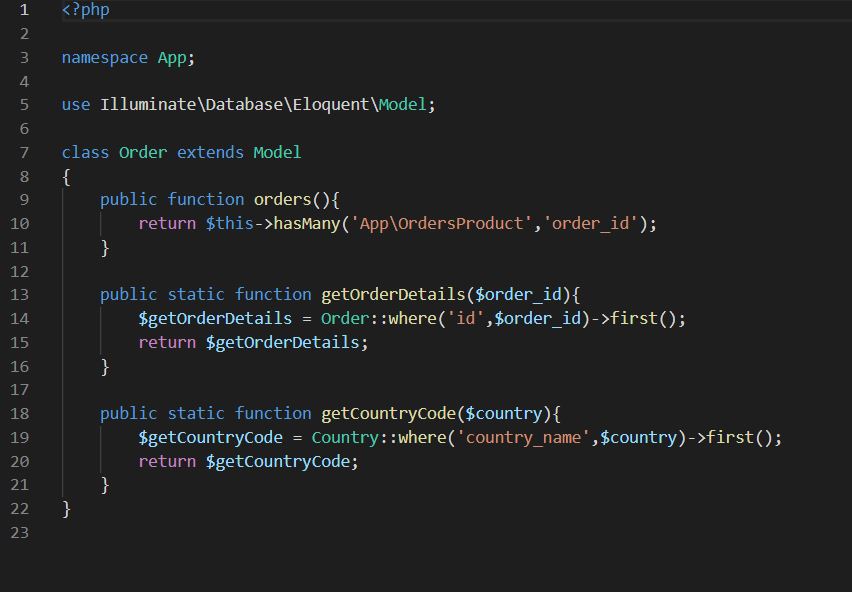
****

Figure 70: User Model Code

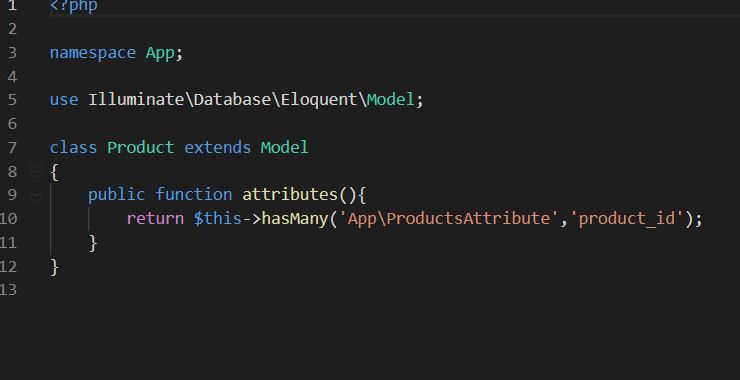
Categories



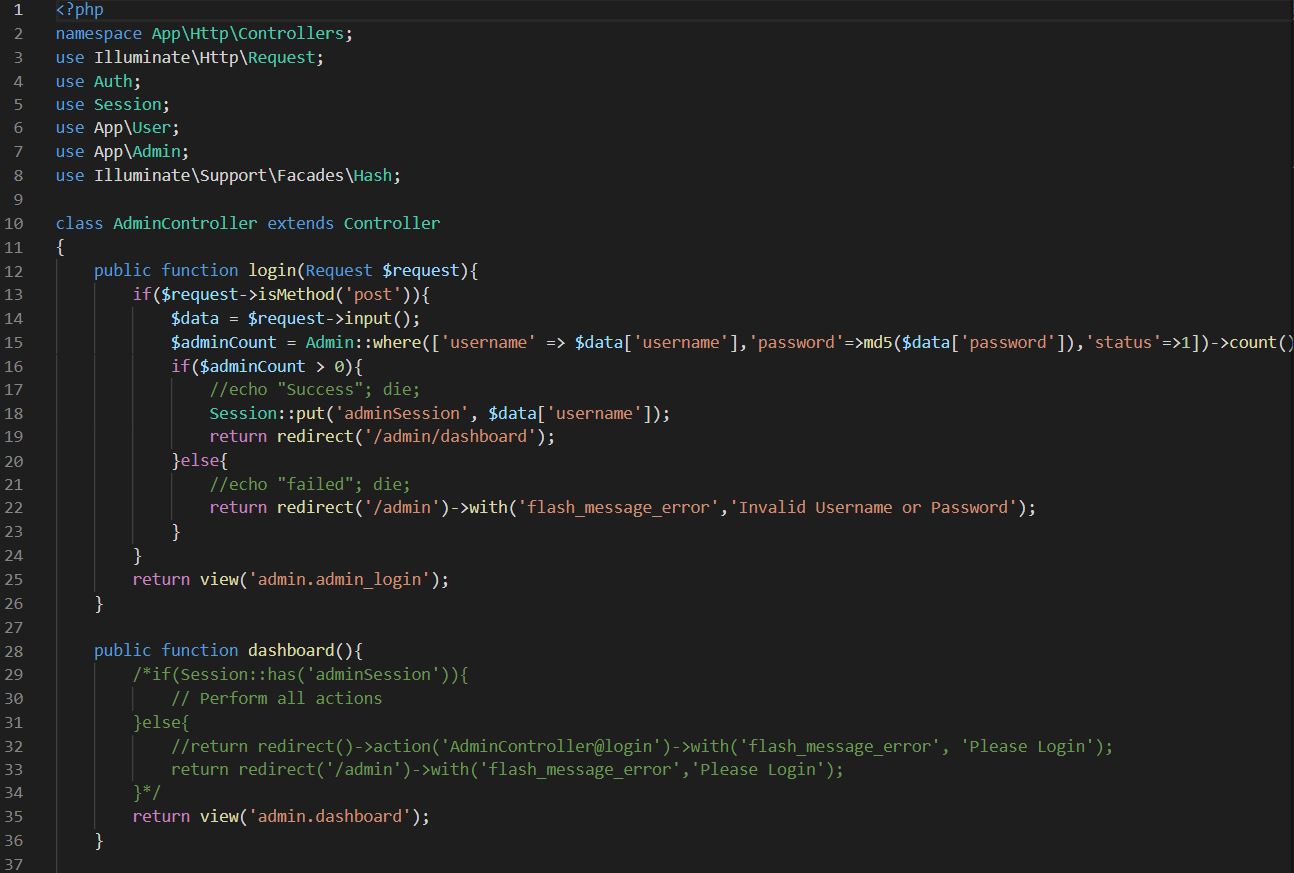
Order



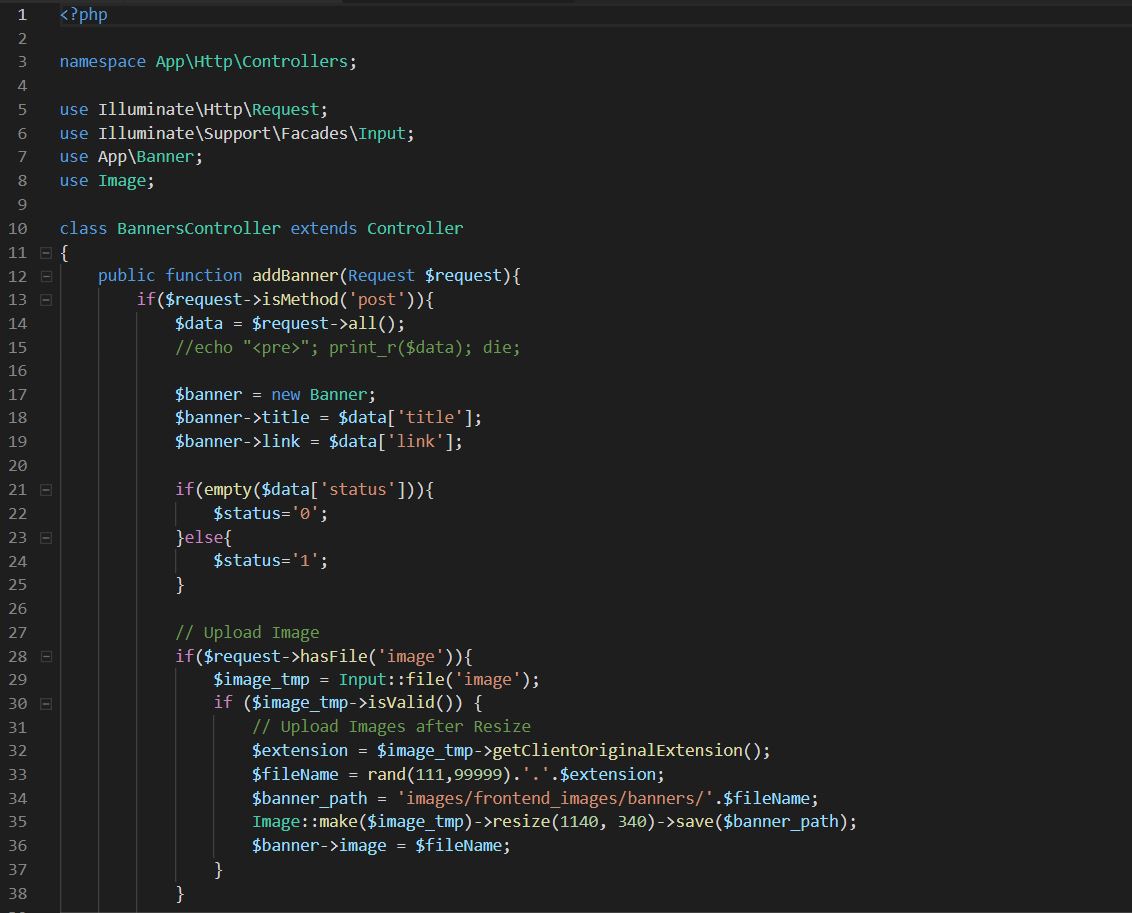
Attributes



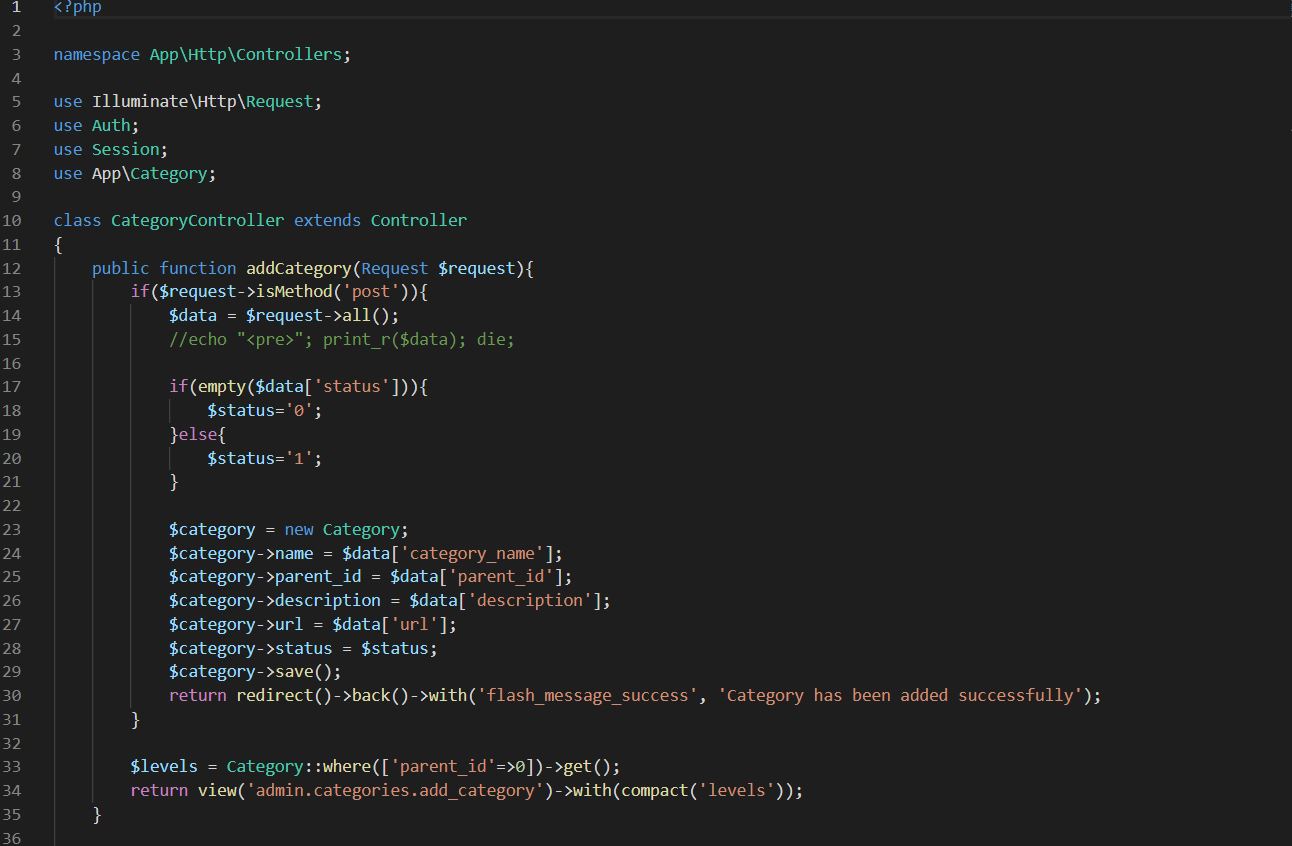
Admin Controller



Banner controller

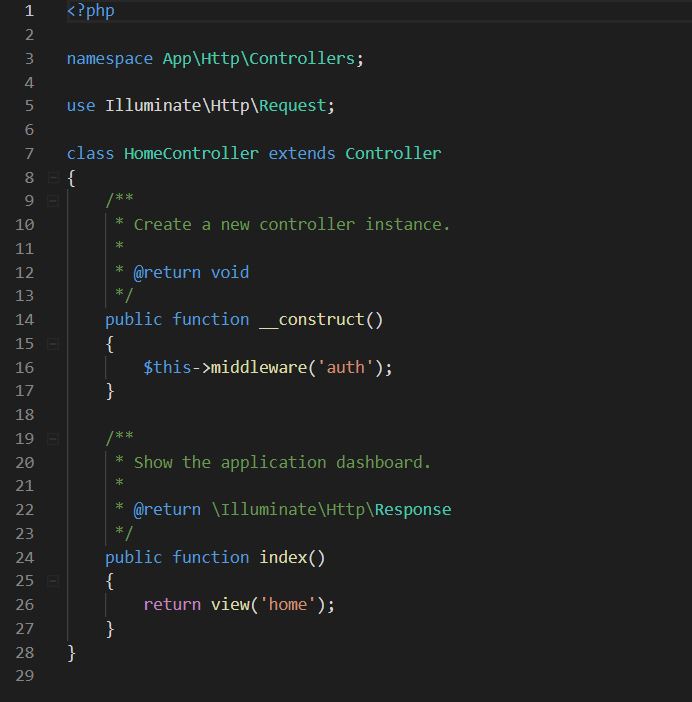


**Category Controller**

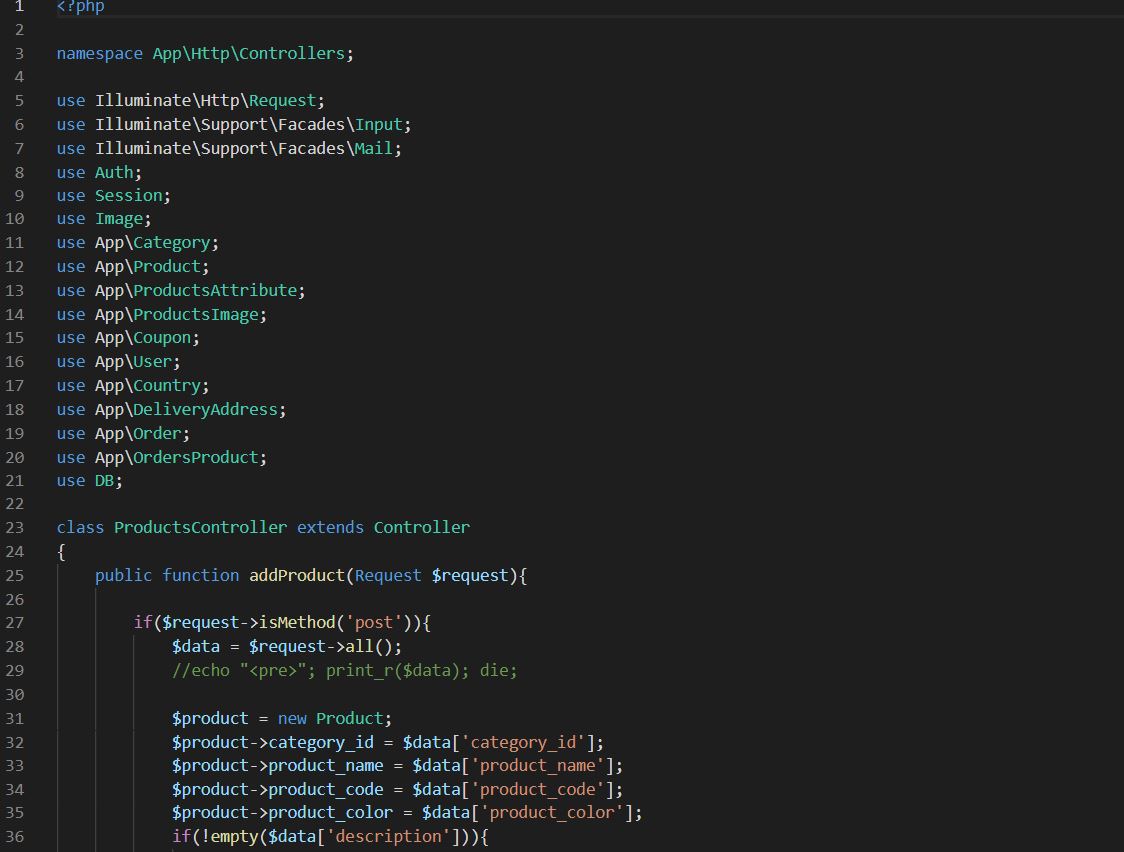


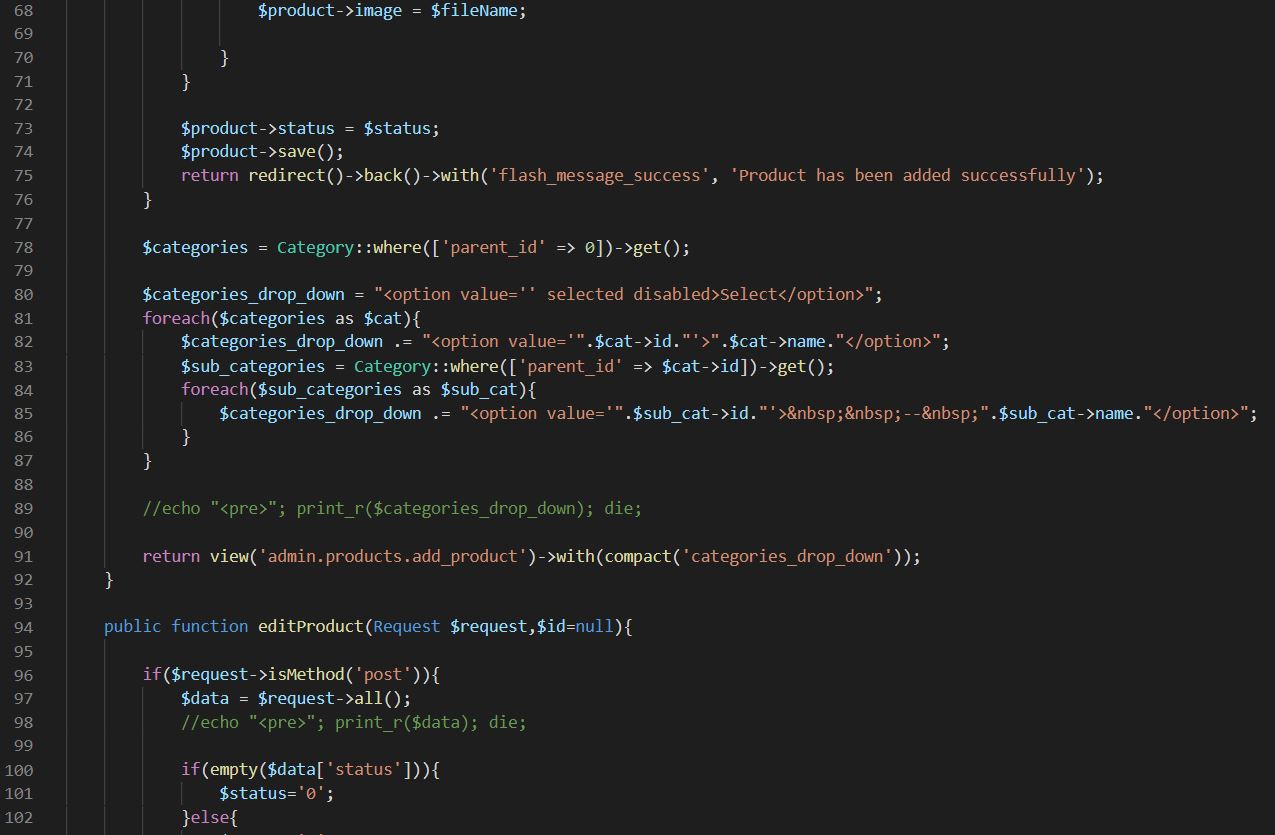
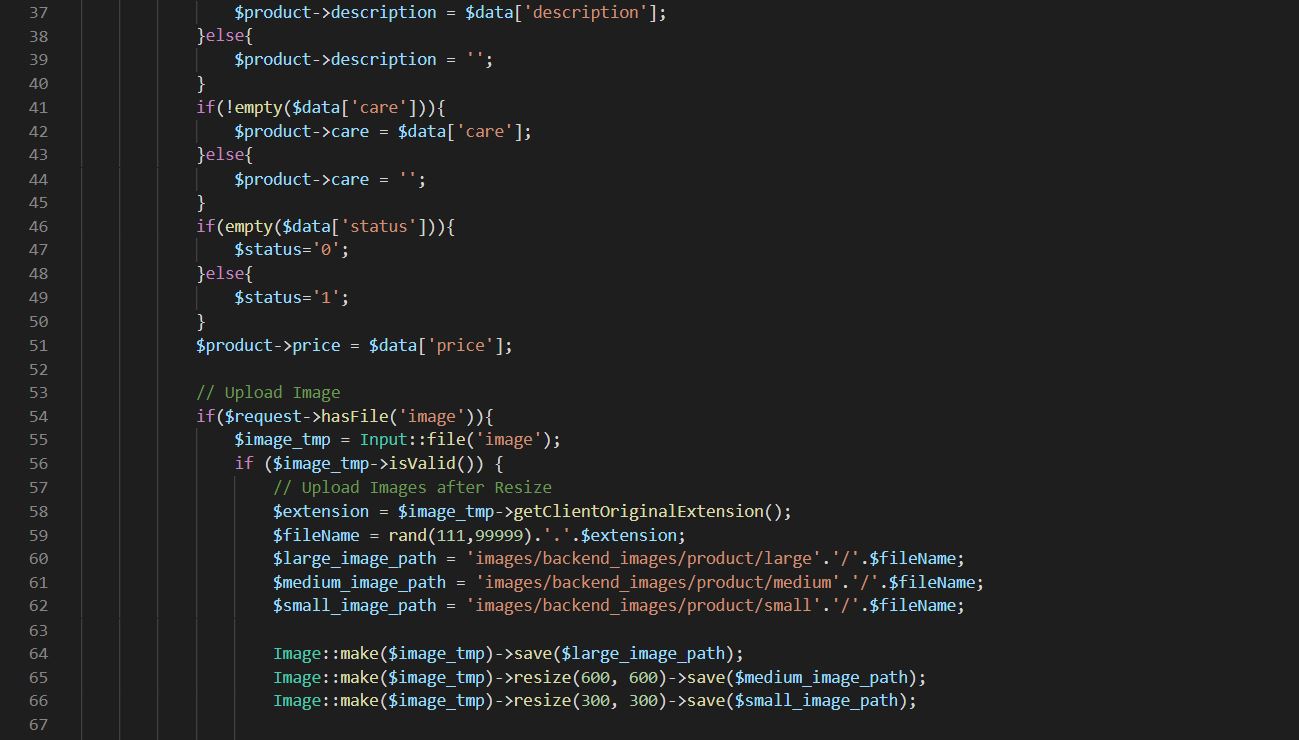


**Home Controller**

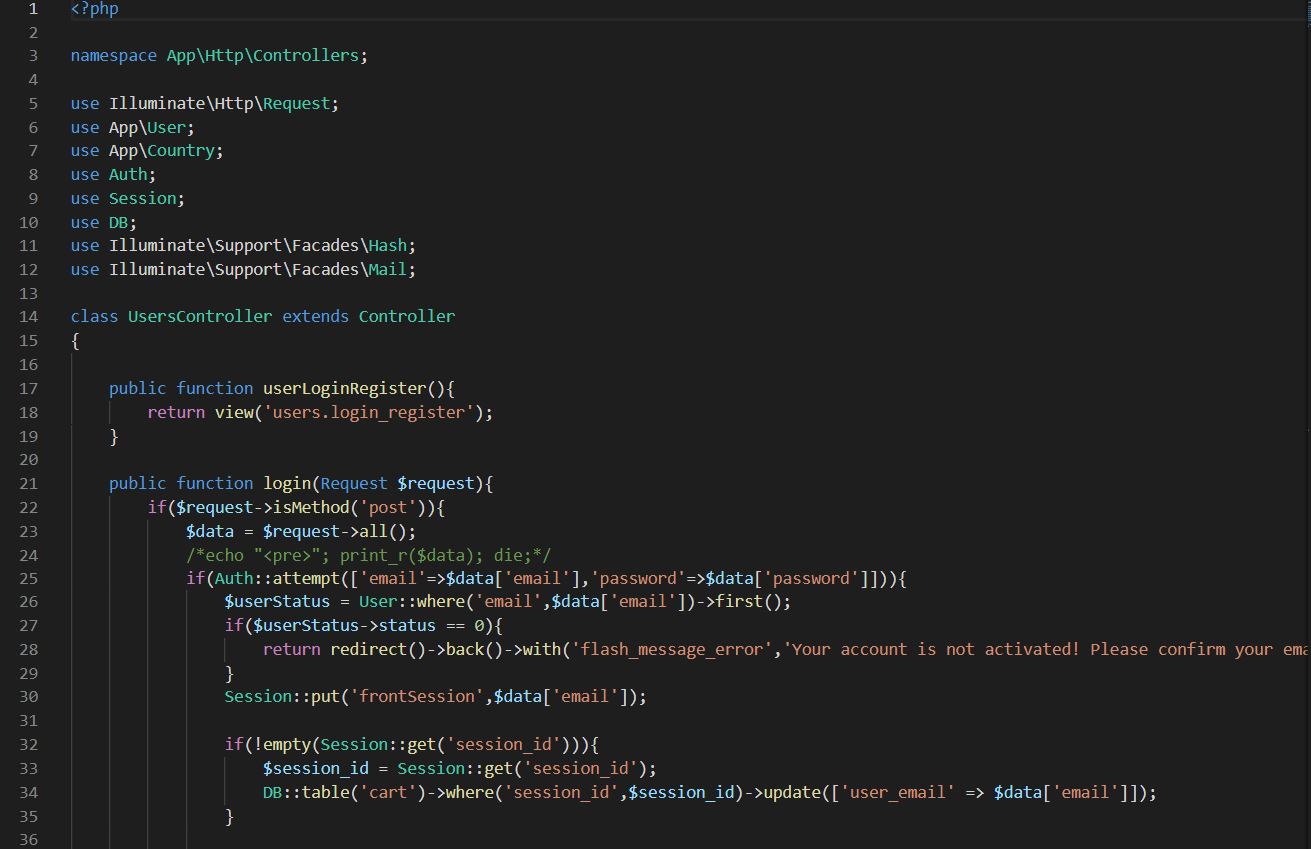


**Product Controller**

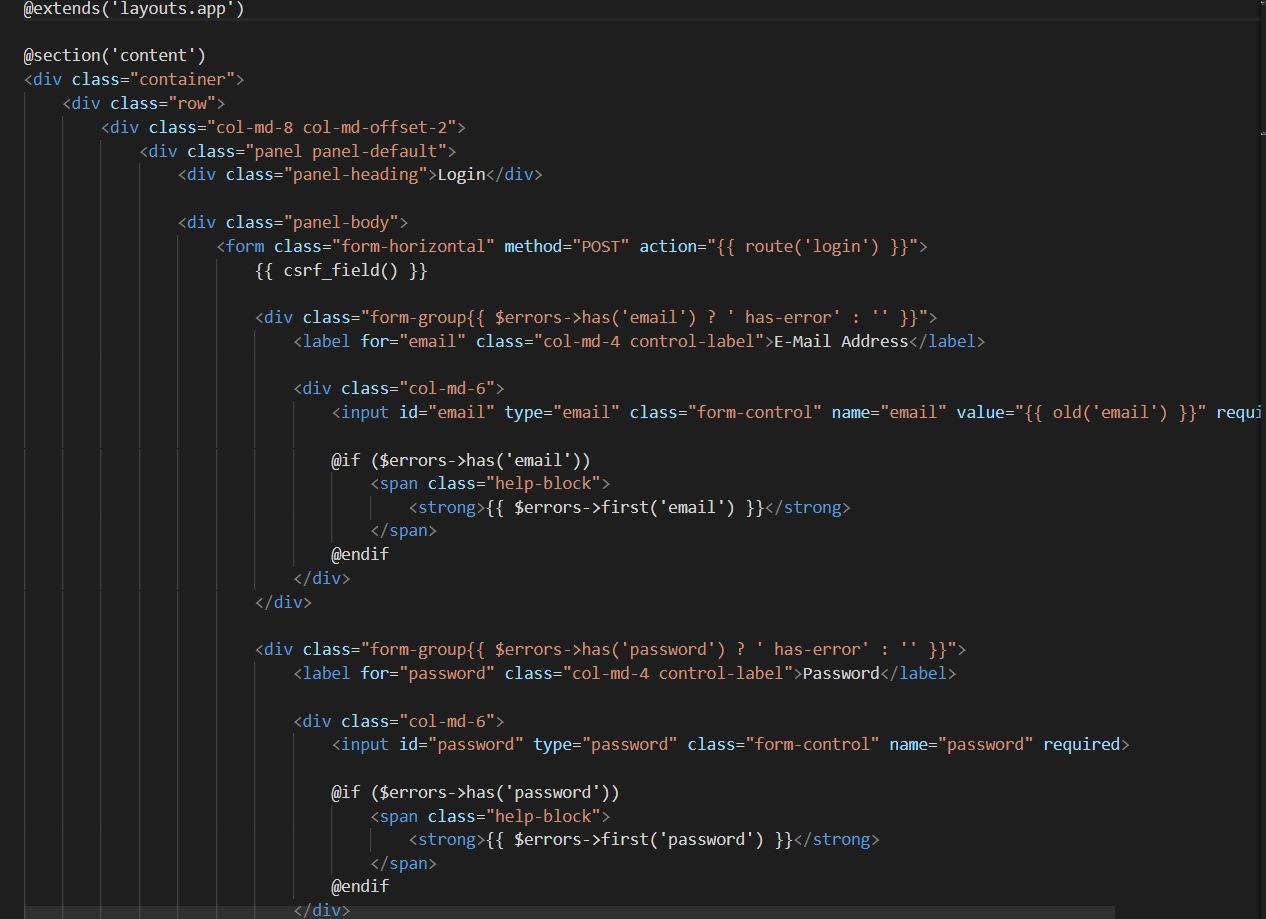
******

******

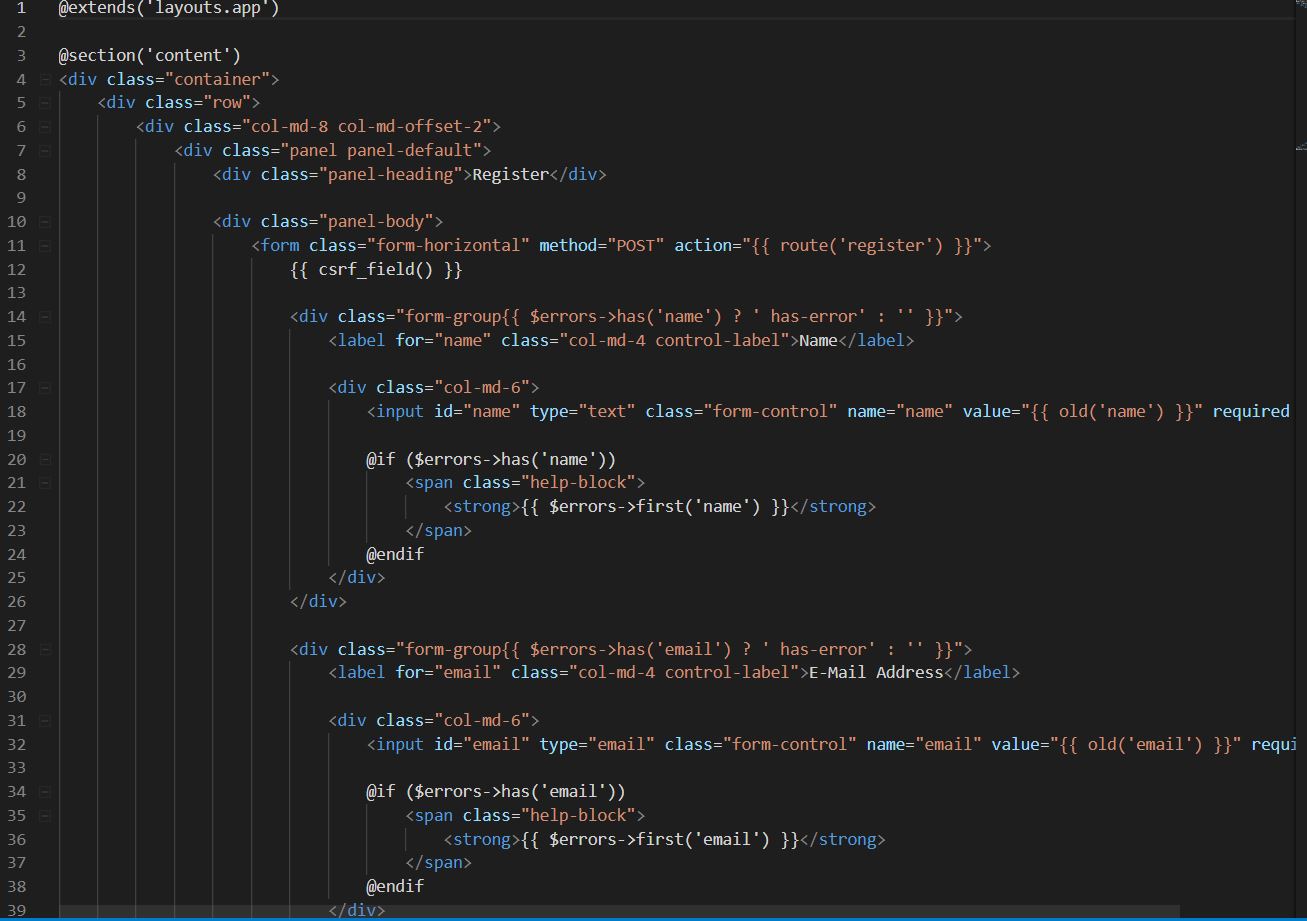
***User Controller***

******

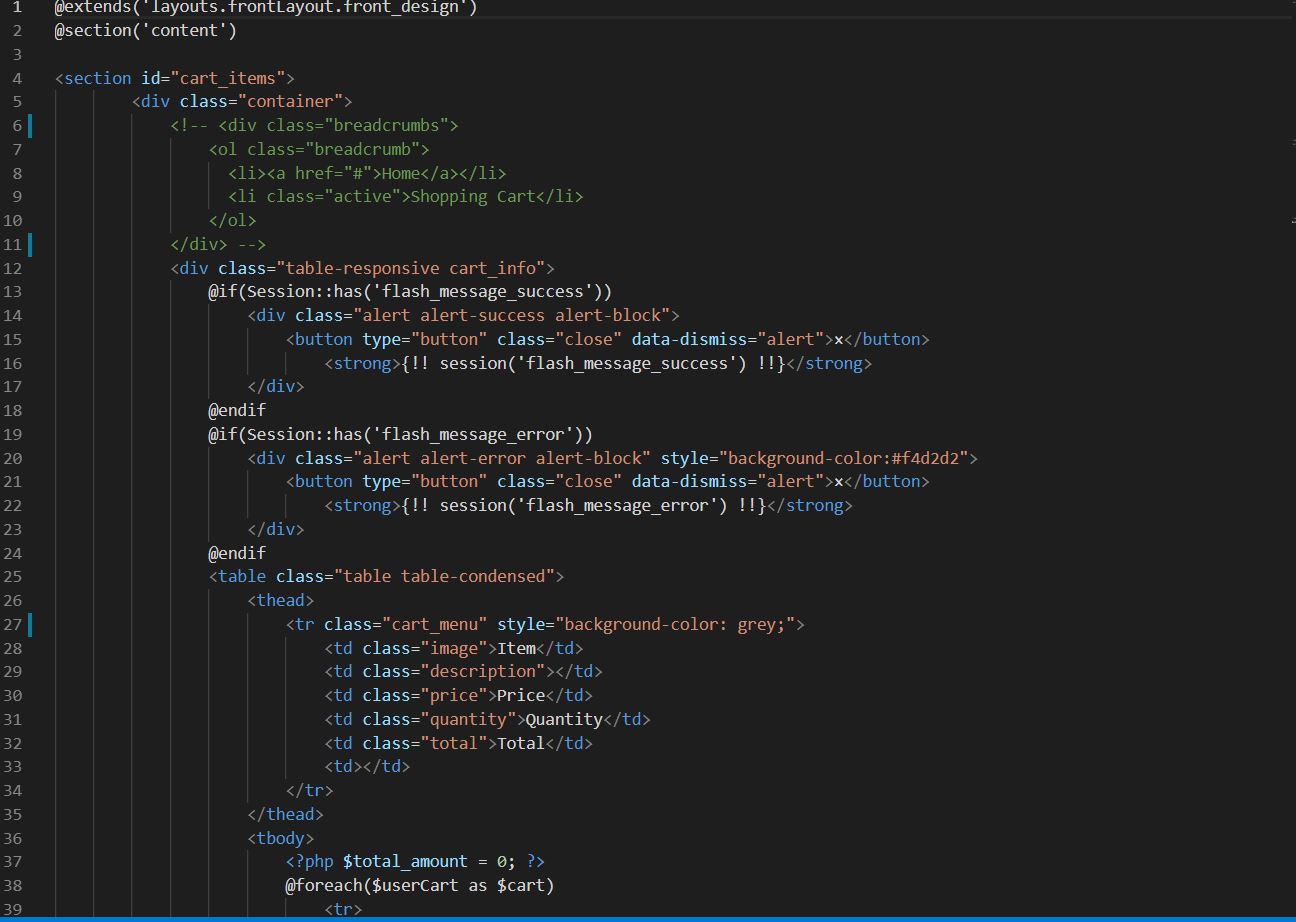
***Login Blade***

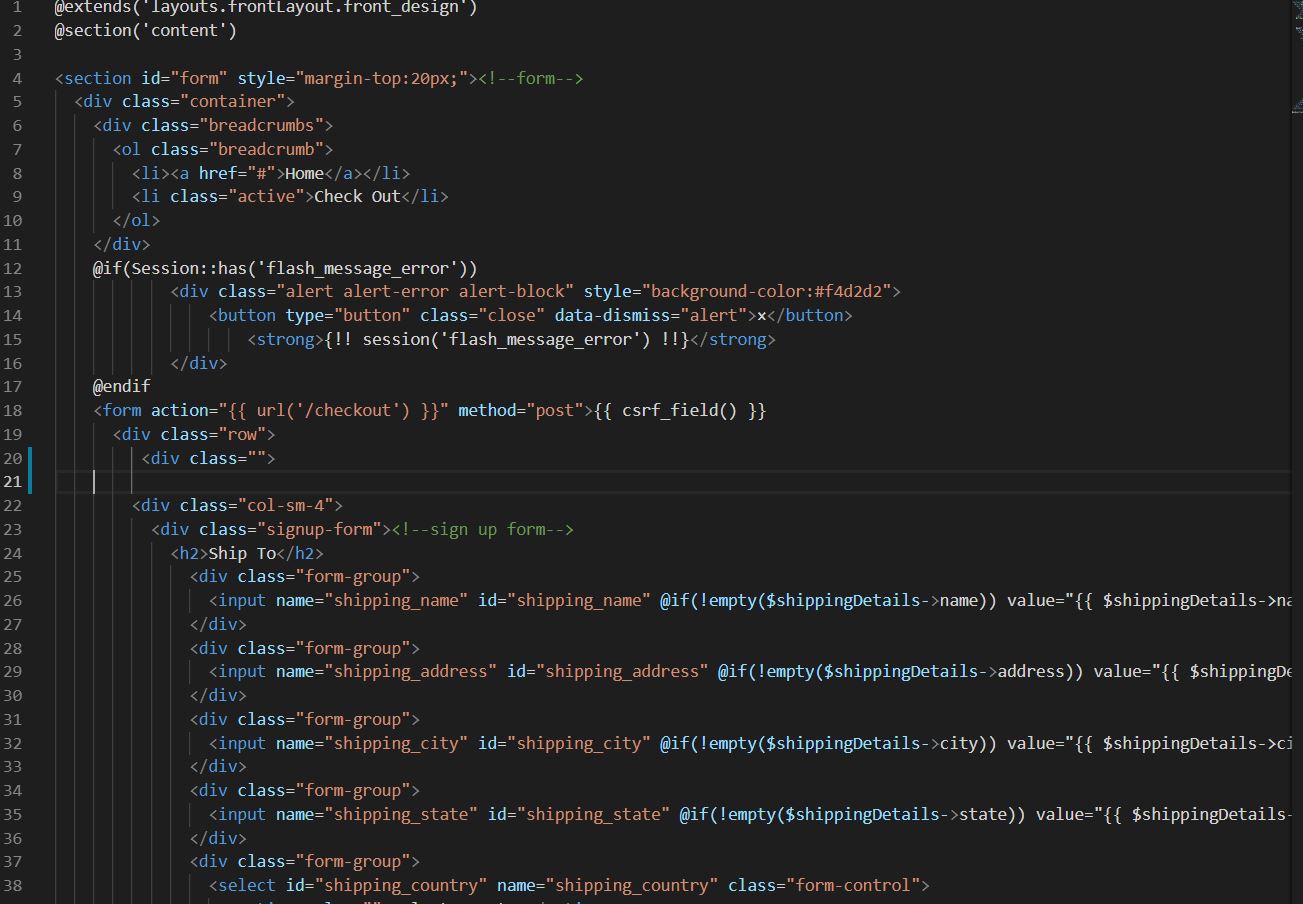
******

***Register Blade***

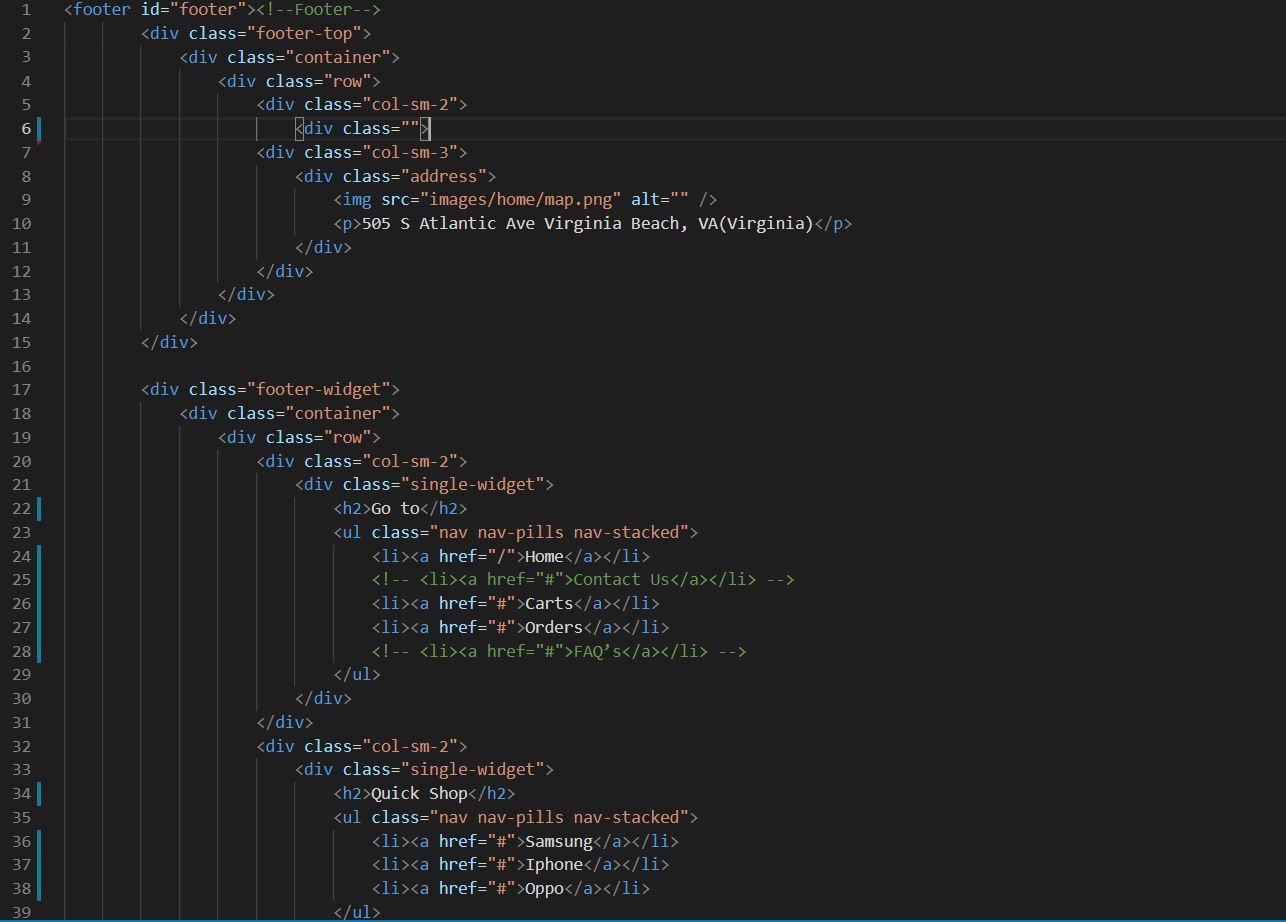
******

***Cart Blade***

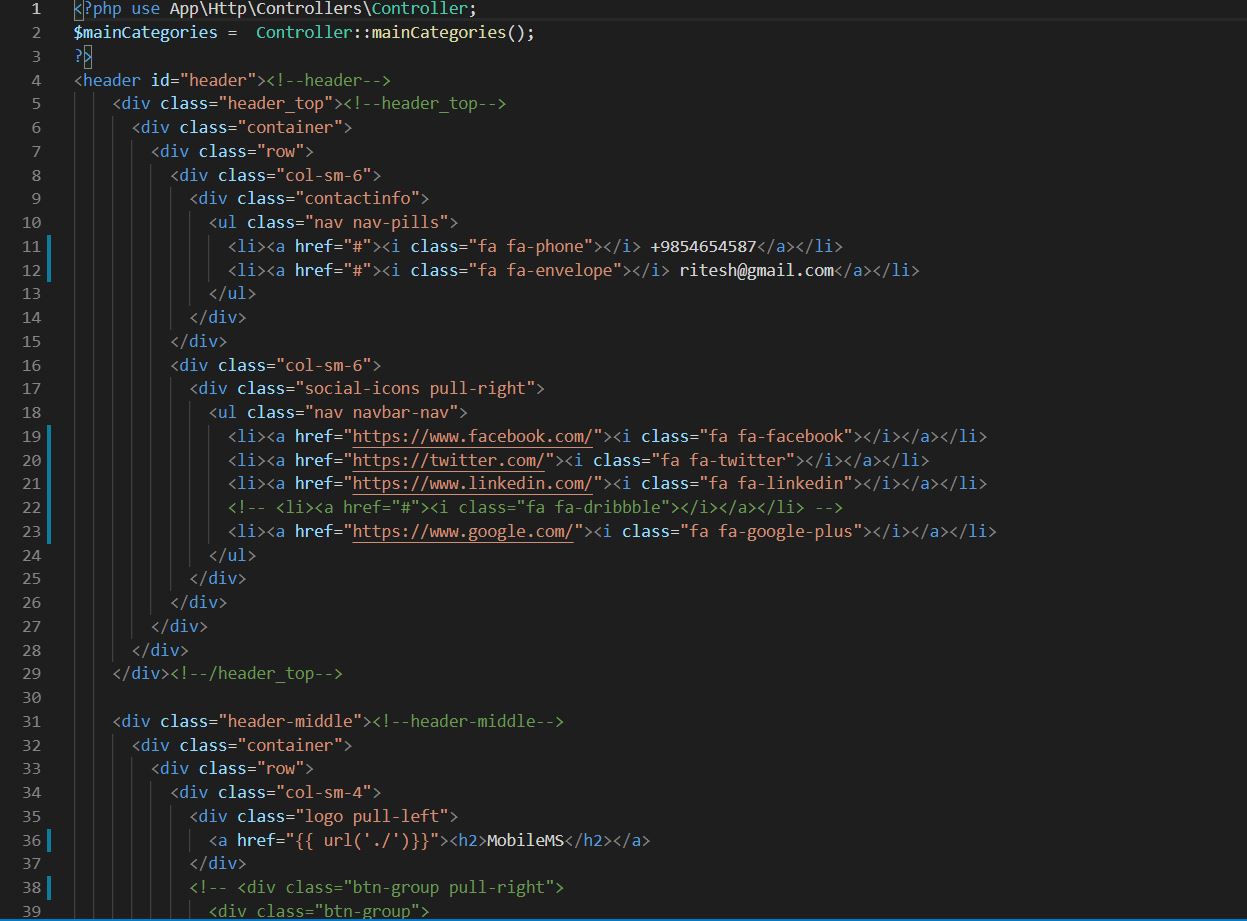
******

***Checkout Blade***

***Footer Blade***

******

***Header Blade***

******

**The End**