tur

|  |
| --- |
|  |
| Module Name: Computing Project (CP)  Hotel Order Management System  Submitted by: Sudeep Bhattarai  Submitted to: (**Sudeep Lal Bajimaya**)  NCC ID: 00175876  Batch: 23 ‘A’ |

# Abstract

There is great influence of technology towards business sector nowadays. Business in hospitality industry has been greatly influenced and competition has increased due to improved food ordering techniques and automated systems. There is necessity in client’s attention towards the business for higher achievement and also necessity of good response towards client. Hotel order management system replaces the old paper based management system to web based system that allows costumer to order through web application and monitor their order in front. The project simplifies the orders and sales through the database kept inside the system. Administration grants the role for hotel staffs and can monitor the sales and order report.

The development of this website is transected through PHP which is a programming language and MySQL as a database. The core objective of this project is to Dominate the current system of ordering with the technical and standardized system with the advantage of fast and easy ordering and billing system. The project now can calculate the day-day sales and keeps record of each sale. It is the fast and convenient way to take order and pay the order.

# Acknowledgement

A big thank to my supervisor Mr. Sudeep lal Bajimaya at first, for his big support to finalize this project. His helpful guidance and suggestion towards my thought in this project helped me throughout the project to specify the requirements and the functions with preference during the development of this project. Every session in this semester was helpful to me. His sessional advice helped us to perform the task more easily. His feedbacks on documentation and report made the quality of the project to justify. I would also like to thank Softwarica College of IT& E-commerce for providing us a good platform to work and increase the development skill professionally. Credit for providing a quality of education and knowledge in the field of study made possibility to finalize the project and also the practical experience of learning in this college was of great experience. Lastly I would also thank my other batch mates to provide references and help during the section of coding and testing.

Table of Contents

[Chapter 1: 10](#_Toc8125697)

[Introduction: 10](#_Toc8125698)

[Project Background: 10](#_Toc8125699)

[Problem Statement: 10](#_Toc8125700)

[Aims: 11](#_Toc8125701)

[Objectives: 11](#_Toc8125702)

[Overview of project Design 11](#_Toc8125703)

[Features 11](#_Toc8125704)

[Chapter 2: Analysis 13](#_Toc8125705)

[Introduction to analysis 13](#_Toc8125706)

[Need for Analysis: 13](#_Toc8125707)

[Object oriented analysis: 13](#_Toc8125708)

[Pitfalls and merits: 13](#_Toc8125709)

[Requirements Analysis: 14](#_Toc8125710)

[Functional and Non-Functional requirements: 14](#_Toc8125711)

[NLA: 15](#_Toc8125712)

[Initial Class Diagram: 16](#_Toc8125713)

[Use Case Diagram: 17](#_Toc8125714)

[Architecture 19](#_Toc8125715)

[1. Methodology used: 19](#_Toc8125716)

[i. Requirements 19](#_Toc8125717)

[iii. Program Design 20](#_Toc8125718)

[iv. Coding 20](#_Toc8125719)

[v. Testing 20](#_Toc8125720)

[vi. Implementation 20](#_Toc8125721)

[2. Design Pattern: 21](#_Toc8125722)

[3. System Architecture: 22](#_Toc8125723)

[Chapter 3: Design 22](#_Toc8125724)

[Dynamic Modelling 23](#_Toc8125725)

[Sequence Diagram: 23](#_Toc8125726)

[Activity Diagram: 26](#_Toc8125727)

[Structural Modelling 29](#_Toc8125728)

[Final Class Diagram: 29](#_Toc8125729)

[DFD: 30](#_Toc8125730)

[Database Modelling 32](#_Toc8125731)

[ER Diagram: 32](#_Toc8125732)

[Data Dictionary: 33](#_Toc8125733)

[Chapter 4: Implementation 43](#_Toc8125734)

[Choice of language: 43](#_Toc8125735)

[Development Environment: 43](#_Toc8125736)

[Chapter 5: Testing 45](#_Toc8125737)

[Unit Testing: 45](#_Toc8125738)

[White Box Testing: 45](#_Toc8125739)

[Black Box Testing: 45](#_Toc8125740)

[Chapter 6: Other Project Issues 65](#_Toc8125741)

[Risk Management: 65](#_Toc8125742)

[Configuration Management: 67](#_Toc8125743)

[Scheduling/Gantt Chart: 69](#_Toc8125744)

[Limitation: 71](#_Toc8125745)

[User manual 72](#_Toc8125746)

[Chapter 7: Conclusion 80](#_Toc8125747)

[Reference 81](#_Toc8125748)

[Chapter 9: Appendix 83](#_Toc8125749)

**List of Figures**

[Figure 1: Initial Class Diagram 15](#_Toc8125750)

[Figure 2: Use Case Diagram 17](#_Toc8125751)

[Figure 3: Waterfall Model 18](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125752)

[Figure 4: MVC Design Pattern 20](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125753)

[Figure 5: 3-Tier Architecture 21](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125754)

[Figure 6: Login Activity Diagram 26](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125755)

[Figure 7: Activity Diagram to generate bill 27](#_Toc8125756)

[Figure 8: Final Class Diagram 28](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125757)

[Figure 9 : Level 0 DFD Diagram 29](#_Toc8125758)

[Figure 10: Level 1 DFD Diagram 30](#_Toc8125759)

[Figure 11: ER Diagram 32](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125760)

[Figure 12: Customer Type 32](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125761)

[Figure 13: Item Type Data Dictionary 32](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125762)

[Figure 14: Customer Data Dictionary 33](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125763)

[Figure 15: Bills Data Dictionary 33](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125764)

[Figure 16: Carts Data Dictionary 33](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125765)

[Figure 17:Order Data Dictionary 34](#_Toc8125766)

[Figure 18: Login Page Prototype 35](#_Toc8125767)

[Figure 19: Prototype of Sign Up 36](#_Toc8125768)

[Figure 20: Prototype of Home Page 37](#_Toc8125769)

[Figure 21: Gallery Page Prototype 38](#_Toc8125770)

[Figure 22: Prototype of Carts Page 39](#_Toc8125771)

[Figure 23: prototype of order page 40](#_Toc8125772)

[Figure 24: Prototype of Admin Page 41](#_Toc8125773)

[Figure 25: Test Case for Register 46](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125774)

[Figure 26: Unit Testing Result for Register 47](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125775)

[Figure 27: Registration of Black Box Testing 47](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125776)

[Figure 28:Log in form 48](#_Toc8125777)

[Figure 29: Test Case for Adding Item Type 49](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125778)

[Figure 30: Unit Testing Result for Adding Item Type 49](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125779)

[Figure 31: Adding Product Black Box Testing success 50](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125780)

[Figure 32: Added product for black box testing 51](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125781)

[Figure 33: Test Case for Updating Item 52](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125782)

[Figure 34: Unit Testing for Updating Item 53](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125783)

[Figure 35: Black Box Testing for Updating icon 53](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125784)

[Figure 36: edit/update success 53](#_Toc8125785)

[Figure 37: Test Case for Item Delete 54](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125786)

[Figure 38: Unit Testing for Item Delete 54](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125787)

[Figure 39: Black Box Testing for Item Delete 55](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125788)

[Figure 40: Black Box Testing Result for Item Delete 55](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125789)

[Figure 41: Test Case for cart 56](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125790)

[Figure 42: Unit Testing for cart 57](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125791)

[Figure 43: Black Box Testing Result for cart fails 57](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125792)

[Figure 44: Test Case for Order 58](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125793)

[Figure 45: Unit Testing Result for Order 59](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125794)

[Figure 46: placing Order 59](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125795)

[Figure 47: Order success 59](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125796)

[Figure 48: Search test case 60](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125797)

[Figure 49: search Unit Testing 60](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125798)

[Figure 50: search s7 61](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125799)

[Figure 51:Search pass 61](#_Toc8125800)

[Figure 52: Test Case for view order 62](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125801)

[Figure 53: view order Unit Test pass 62](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125802)

[Figure 54: view order of Black Box Testing 63](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125803)

[Figure 55: Tree Chart 66](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125804)

[Figure 56: Schedules Table 68](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125805)

[Figure 57:Gant Chart 69](#_Toc8125806)

[Figure 58: Laravel all files 82](#_Toc8125807)

[Figure 59: Database migrated in MYSQL 83](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125808)

[Figure 60: List of Models 84](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125809)

[Figure 61: Controllers list 85](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125810)

[Figure 62: Views list 86](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125811)

[Figure 63: Routes 87](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125812)

[Figure 64: Home page 88](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125813)

[Figure 65:Gallery 88](#_Toc8125814)

[Figure 66: Cart page 89](#_Toc8125815)

[Figure 67:Order Page 89](#_Toc8125816)

[Figure 68:Add product(admin) interface 90](#_Toc8125817)

[Figure 69:Added product interface 90](#_Toc8125818)

[Figure 70: User Model Code 91](file:///C:\Users\nextu\Desktop\submission\00172893_Ritesh%20Shrestha_CP.docx#_Toc8125819)

# Chapter 1:

## Introduction:

We are living in the era of automation. Business in the hospitality industry has been greatly influenced and competition has increased due to improved food ordering techniques and automated systems. In order to effectively run a restaurant or hotels, time saving and cost optimizations are essential. Reduction in time by a few seconds for each table can speed up order processing, increase efficiency and boost profits. My project is to create an online hotel ordering and billing system website for an additional amenity in luxurious hotels. In order to create my project, I will be using a specific programming language PHP, MySQL as a database, JavaScript, bootstrap, jQuery.

## Project Background:

The food ordering system, till a few years ago, was a completely manual process where a waiter used to note down orders from the customers using pen and paper, take the orders to the kitchen, bring the food and make the bill. Although this system was simple it required extensive investment in purchase and storage of paper, large manpower and also was prone to human errors and greater time consumption.

In order to overcome these limitations in manual system, a system can be developed like creating a websites containing all the features of ordering and billing in the hotel where costumer can get efficient and effective service and get notified about the process. User get in touch with the counter table and kitchen screen about the order taken and bills are counted automatically in each table of the order.

## Overview of the Project:

### Features

Following features will be included on the system:

* **Provides notification of the activity performed by the user.**

The activity of taking order and kitchen process are notified to the costumer on the screen

* **Provide an interface to add product and their respective price.**

The products and their prices are detailed on the interface to select as an order.

* **Provides total sales between selected period.**

The sales report is captured by the time

* **Calculates the amount of the order.**

Amount of the order is calculated instantly.

* **Print the bill of the order.**

The bill of the order can be printed simultaneously.

* **Orders to be made ready is displayed in cook display board.**

The taken order is displayed on the screen in the kitchen board.

* **Customer order status is displayed in the screen near to the table.**

The ordered items with its status is shown in the screen near the table

## Justification of the Project:

Within this all features the hotel can create a clear and convenient environment of services within the costumer along with the problem listed above is fixed by the additional features of note given by the costumer for their health purposes. User can bear their fast services and interact positively with the costumer. Some more features of taking reviews and feedbacks are added so that the problem and unsatisfaction can be detected and hotel can take actions against it.

## Aims and Objectives:

**Aims**

Fast service and easy ordering and monitoring is the main aim of the project. Replacing the paper note ordering system the aim to save time and fast delivery can optimize the profit by growth in sales. Some of the point listed aims are:

* To provide fast ordering and serving system in a standardized pattern.
* To provide monitoring system to inform about the order in each of the table.
* To make the portable ordering system which helps user to take order in different table at a time and give order to the kitchen at a time with notification.

**Objectives**

Dominating the current system of ordering with the technical and standardized system with the advantage of fast and easy ordering and billing system the functions are featured as an objective in this system some of the objectives of this management systems are listed below:

* Creating the database that is normalized after understanding the attribute and relation between the entities.
* Designing user friendly UI which is simple and easy to use.
* Providing robust login system for admin users.
* Design a simple interface to add different product and its price.
* Designing a cook display board, table display board and order booking interface.
* Making the interfaces dynamic.
* Making the interface to update automatically.

# Chapter 2: Analysis

## Introduction to analysis

A problem solving technique that ensures the functional and non-functional requirements in a system and collecting the problem, threats and fact is known as Analysis.

## Analysis methodology:

Usually two types of analysis methodology are used one is hard and another is soft. Both large and small projects can be made using this methodology. Since, our project is small but can be a large project in future I have used hard methodology for the solution.

**DFD**

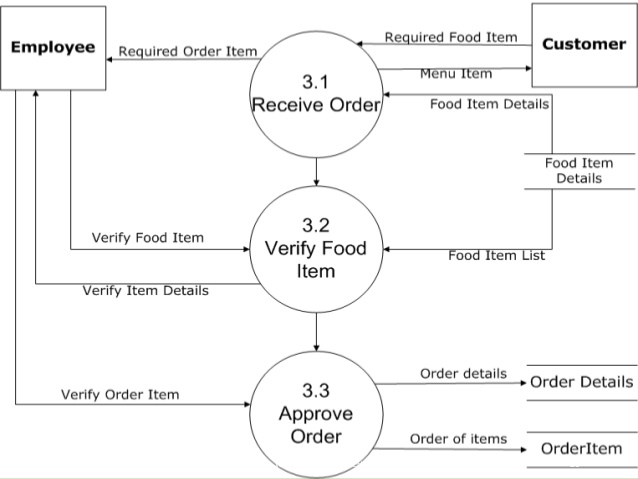
****

Table 1: DFD

## Information gathering methodology

Questionnaires and interviews were done with various groups of people before conducting the development process to gather information.

### Questionnaire

A series of questions are made and asked to the fresher’s and their comments and opinions are gathered as an information in this methodology.

* Rating in the satisfaction of the software with classified features and functions.

|  | **Very Satisfied** | **Somewhat Satisfied** | **Medium** | **Unsatisfied** | **Very Unsatisfied** |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | |  |
| **Hardware requirements** |  |  |  |  |  |  |
|  | | | | | |  |
|  | | | | | |  |
| **Interface design** |  |  |  |  |  |  |
|  | | | | | |  |
| **Easiness of using the software** |  |  |  |  |  |  |
|  | | | | | |  |
| **Quality** |  |  |  |  |  |  |
|  | | | | | |  |
| **Feedbacks collection** |  |  |  |  |  |  |
|  | | | | | |  |
| **It can be used from anywhere** |  |  |  |  |  |  |
|  | | | | | |  |
|  | | | | | |  |
|  | | | | | |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  | | | | | |  |
|  |  |  |  |  |  |  |
|  | | | | | |  |

## Feasibility study

Analyzing the project whether it is feasible every potential factors like cost, time is especially known as feasibility study. It is one of the important part in analysis that helps in development process to take place in a flow of smoothness. It also verifies if the project is socially, economically, legally and technically fit or not. The weakness and strength of the project can be identified with a good feasibility study that helps to decrease the rate of system issues and defects.

1. Legal feasibility:

It denotes the legal issues that may occur during the deployment.as our project is not so big and doesn’t follow e-commerce. There may not occur any legal issue. Boundaries are kept inside the company who use it for the internal booking and order system.

1. Economic Feasibility: It defines the investment of the party to the project weather it is cost benefit or not.as our project needs monitoring hardware in the kitchen and the tables for viewing orders there may be leaved the cost in hardware even though it is a small project so can be done in a little amount and can be completed within the deadline it is feasible economically.
2. Legal Feasibility: This includes all the rules and regulation leaved under software. Government track and record in legal boundaries is known as legal feasibility. Development process is under rules so it is ok in legal feasibility.
3. Technical Feasibility: It studies technical requirement of the project under the company. We have required technical needs for this process but can be need of monitors though it can be technically feasible.
4. Social Feasibility: Impacts of the project towards society is social feasibility. System is for internal use so the social issue may not arise due to the project.

## Software Requirement Specification (SRS)

The system functionality and the expectation towards the software describes SRS (Software Requirement Specification). Some of the SRS for the project are:

1. Functional Requirement.

This requirement means the exceptional need of the project like It must be able to perform. So, some of the requirements with the view point of the user are:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id** | **Title** | **Description** | **Rational** | **Dependency** |
| F1 | Login | User assigned can login to the system | To use the functions that are assigned. | F1 |
| F2 | View Order | Users can view orders taken and the detail of order. | To view food and drinks order | F2,F3 |
| F3 | Add Order | Admin and Waiter can add order . | To add order for delivery. | F3 |
| F4 | Update order | Admin and Waiter can Update or cancel order. | To update ,add and delete order. | F4.F3,F5 |
| F5 | Delete Order | Admin and waiter can delete their orders . | Deleting the orders. | F5,F4 |
| F6 | Pay | Admin should transfer order to sales through Pay after the payment | To take payment from costumer | F2 |
| F7 | Add categories of food | Categories of food should be able to be added by admin. | In order to take order food categories are listed by food. | F7 |
| F8 | Delete categories | Added categories must be able to be deleted.. | In absence of food items in menu food categories are deleted. | F8,F7 |
| F9 | Update categories | The categories defined can be updated/edited. | To update the price and replace food to take order. | F9,F7 |
| F10 | Add table | Admin can add the table. | To order the food according to table. | F10 |
| F11 | View table | Admin can view the number of table assigned | To view tables . | F11,F10 |
| F12 | Delete table | Admin can remove table. | To remove tables . | F12,F11 |
| F13 | Add food | Admin can add food variety with detail | To take order from the costumer | F13 |
| F14 | Delete Food | Admin should be able to delete the food item that are not made. | To delete food that cannot be cooked | F14,F13 |
| F15 | Update Food | Admin should be able to update food items to replace and change price | To update food items. | F15,F13 |
| F16 | View Sales | Admin should be able to view sales detail | To view sales date and sales item | F16 |
| F17 | Notification | Admin should be able to send notifications about the functions used | To review the function | F17.F2,F15 |

1. Non-functional requirements

The detonation of behavior and keeping barrier towards the functionality is known as Non-functional requirement. Some of the non-functional requirements are

|  |  |  |
| --- | --- | --- |
| **Id** | **Title** | **Description** |
| NF1 | Security | Only the authorized person should be able to use the system. |
| NF2 | User Friendly | The system developed should be user friendly that satisfies the user in its use |
| NF3 | Performance | System quality is cleared with its performance that it should be fast and easy. |
| NF4 | Reliability | Reliability denotes the trust of user to the system against security. |
| NF5 | Maintainability | System may get error at any time so there must be maintainability of the system. |
| NF6 | Scalability | Scalability defines the system growth and its need as the data in the system may grow. |
| NF7 | Usability | Everyone from each and every background should be able to use this system. |
| NF8 | Serviceability(Supportability) | Support and service on the system for its fault is serviceability which is important. |
| NF9 | Availability | System must be available from any place to any device which is known as availability. |

1. MoSCoW Prioritization

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Requirement** | **MoSCoW** | **Rational** |
| F1 | Add User | Should have | To assign user by the admin |
| F2 | Login | Must have | To use the system overall |
| F3 | View Order | Must have | To view the order taken for sales |
| F4 | Add Order | Must have | To add order |
| F5 | Update Order | Should have | To edit Orders taken |
| F6 | Delete order | Could have | To cancel the order |
| F7 | Add categories | Must have | To add categories of food items |
| F8 | Delete categories | Should have | To delete categories of food |
| F9 | Update categories | Should have | To update categories of the food items |
| F10 | View table | Must have | To view table. |
| F11 | Add table | Must have | To add tables that is according to the requirement |
| F12 | Add Food | Must have | To add food items |
| F13 | Modify food | Should have | To modify food items. |
| F14 | Delete food | Could have | To delete food items. |
| F15 | View sales | Must have | To view sales items after order and payment |
| F16 | Pay | Must have | To get payment after order delivery |
| F17 | Sending notification | Could have | To send notifications of function to the user |
| NF1 | Security | Must have | Only the authorized person should be able to use the system. |
| NF2 | User Friendly | Should have | The system developed should be user friendly that satisfies the user in its use |
| NF3 | Performance | Must have | System quality is cleared with its performance that it should be fast and easy. |
| NF4 | Reliability | Must have | Reliability denotes the trust of user to the system against security. |
| NF5 | Maintainability | Must have | System may get error at any time so there must be maintainability of the system. |
| NF6 | Scalability | Should have | Scalability defines the system growth and its need as the data in the system may grow. |
| NF7 | Usability | Must have | Everyone from each and every background should be able to use this system. |
| NF8 | Serviceability(Supportability) | Must have | Support and service on the system for its fault is serviceability which is important. |
| NF9 | Availability | Must have | System must be available from any place to any device which is known as availability. |

1. Hardware needed:

Processor: Intel® Core™ i5-2430M CPU @2.40 GHZ

RAM: 4GB above

Monitor: VGA Color Monitor

1. Software needed:

Operating System: Windows 7 or greater

Developing tool: PHP my admin, Bracket

Database: PHP MY ADMIN

## Use case Diagram

A diagrammatic presentation of system uses by the user is known as use case diagram. Symbols like connectors, actors, package, objects etc. Use case diagram for this system is shown below:

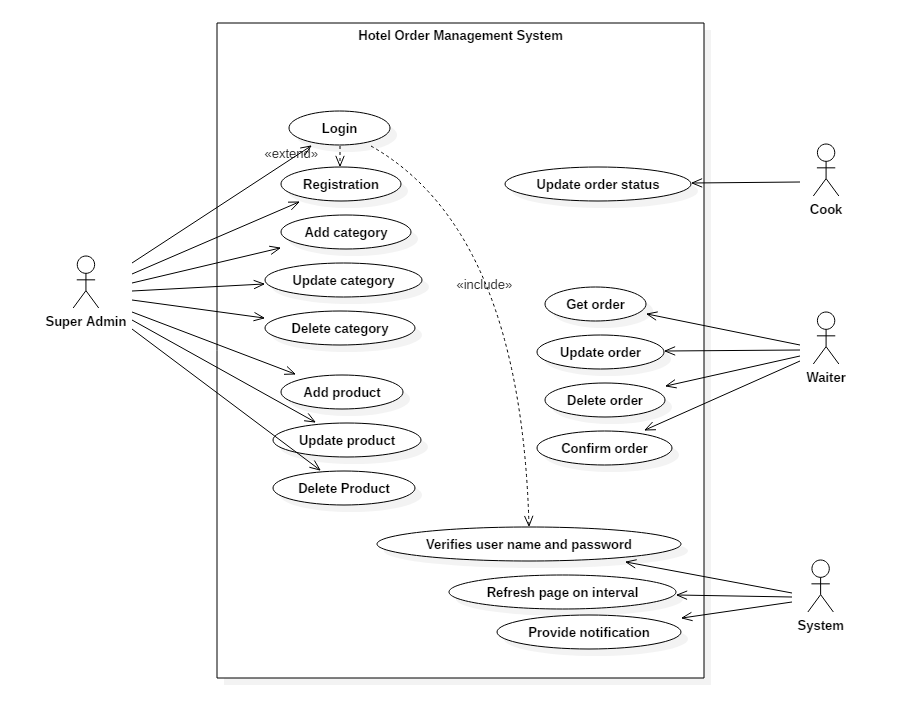


Figure 3: Use case diagram

In the illustrated diagram Super admin, Cook and Waiter are shown as a primary actor who work together to collaborate with each other. Super admin as an actor in the system only has full authority to edit information’s while Cook and the Waiter are register by the admin and given access to only limited functionality of the system. They can login to the system with the Username and password they are assigned to. Only after the assignation of the roles they can make order and go on the Hotel order and payment process.

Login: Login feature is available to Cook , waiter and the admin itself to access the system and Cook and the waiter have access to take and serve order while Admin views the order and make payment.

View Order: This function is used by all the users to view taken order.

Pay: Super admin as a user have access to this option to validate the order passed and take payment.

Add categories: The categories of the food items are added by the admin as a menu in the system so that waiter can make order through.

Add Food: The food items are added with categories that helps waiter to take order.

Update Categories and Food: This feature is accessed only by the admin to update the categories of food and food in the menu of the Hotel.

Delete Categories and Food: This feature is accessed by the admin user to delete the unavailable food when needed.

## Scenario

Hotel Order Management System is a system that process the entire KOT to Bill management with the advanced technique of ordering and processing system. the system must include Order taking system that has to be rolled by the Waiter and View Order function for the Cook to process the order. System must also include categories and Food list as a menu given to the Waiter for taking order. The system also includes the admin panel that can view the orders, make payment and edit and cancel the order when needed.

## NLA

NLA also known as natural language analysis is a process of generating verb, noun and adjectives from the given scenario and afterwards changed it to classes, attributes and methods. For this system we have used to remove the duplications, irrelevancy.

* Nouns

|  |  |  |  |
| --- | --- | --- | --- |
| Nouns | | | |
| Train | Ticket | System | System |
| Reservation | Process | Costumer | Passenger |
| Registration | Login | Customer | Account |
| Book | Ticket | Reservation | Online |
| System | Payment | System | Admin |
| Panel | Features | Bookings | Status |
| Tickets | Time | Arrival | Train record |

|  |  |  |  |
| --- | --- | --- | --- |
| Unique nouns | | | |
| Train | Ticket | System | Reservation |
| Process | Customer | Registration | Login |
| Account | Booking | Online | Admin |
| Payment | Panel | Features | Train record |
| Time | Arrival | Passenger |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Irrelevant classes removed | | | |
| Customer | Ticket | System | Reservation |
| Passenger | Customer | Train record |  |
| Payment | Booking | Admin |  |

|  |  |
| --- | --- |
| Synonyms | |
| * Passenger * Customers | * Booking * Reservation |

|  |  |  |
| --- | --- | --- |
| **Noun identification for candidate class** | **Selected as candidate class** | **Justification for selection or rejection as candidate class** |
| Admin | Yes | It play significant role in the system. |
| Passenger | Yes | It play significant role in the system. |
| Payment | Yes | It specifies requirements. |
| Train records | Yes | It provides all the details information regarding train. |
| Ticket | Yes | It play significant role in the system. |
| System | Yes | Requirements specifies and act as main class in the system. |
| Booking | No | It is the method. |
| Customer | No | Alternative to passengers. |
| Train | No | Irrelevant to the requirements and it is also the property of flight list. |
| Time | No | Irrelevant to the requirements. |

|  |
| --- |
| Selected Candidate class |
| * Passengers * Ticket * Admin * System * Train records * Payment |

* Verbs

|  |  |  |  |
| --- | --- | --- | --- |
| Verbs | | | |
| Booking | Update | Include | Logging |
| Can | Make | Update | Delete |
| Include | Make | Should | Contain |
| Will | View | Consisting | Managing |
| Add | Edit | Delete | Changing |

|  |  |  |  |
| --- | --- | --- | --- |
| Unique Verbs | | | |
| Booking | Update | Include | Logging |
| Can | Make | Edit | Delete |
| Should | Contain | Will | View |
| Consisting | Managing | Add | Edit |
| Delete | Changing |  |  |

|  |  |  |
| --- | --- | --- |
| Irrelevant classes removed | | |
| Booking | Changing | Managing |
| Delete | Edit | View |
| Update | Add | Logging |

|  |
| --- |
| Synonyms |
| * Edit * Update * Changing * Managing |

|  |
| --- |
| Selected Method |
| * Add * Edit * Delete * View |

## Class Diagram

The structural representation of system classes, attributes, their relation and method applied on object as class diagram. It is also known as the most important diagram in UML that shows the whole system structure and components indeed it is used in this project with reliability.

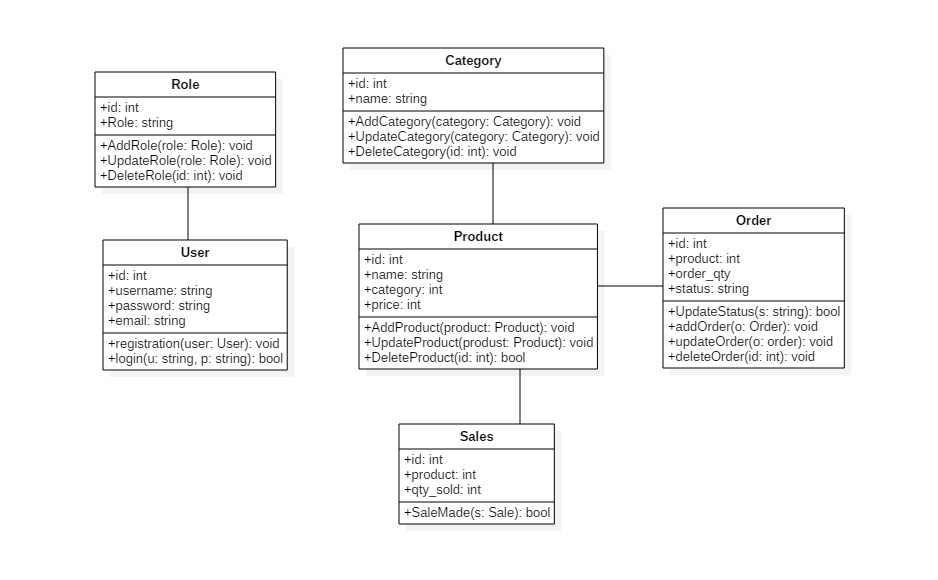
Initial Class Diagram for Hotel Order Management System

Figure 5: Class Diagram

## Architecture

#### Methodology used:

The entire project has been involved in a Waterfall model where various steps of development are involved in next to next format. In this method of development sequential pattern of steps is involved where the steps are completed and only moved to next steps of process where each of the step have its distinct role to be done.

Firstly, every requirement is listed with class then The system is designed according to the requirements listed. After while Implementation is done where the entire system is formed or developed. Next step includes testing in which overall system are tested for additional requirements needed then the product is released. Lastly the maintenance and upgrades are added that were detected while testing.

* **Advantages of Waterfall Model are:**

1. Understandable and easy to use,
2. For small schemes where supplies are well understood it is more effective.
3. Task can be arranged easily.

* **Disadvantages of Waterfall Model are:**

1. Doubt are unclear with maximum possibility of risk.
2. Accommodation to changing requirement are not suitable.
3. The process model is not efficient with the projects where requirements are at moderate to high risk of replacement and changes.

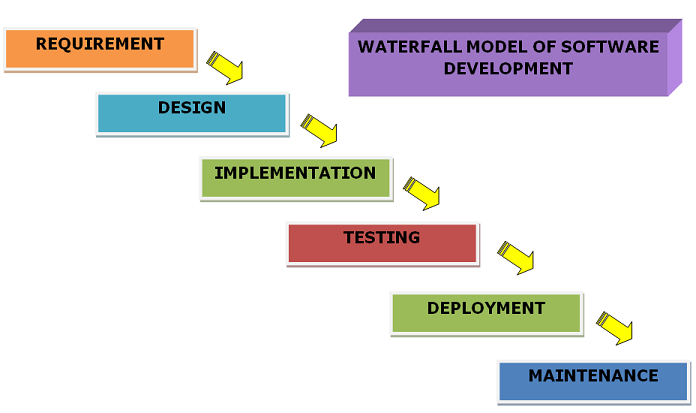


Figure 1: Software development life cycle

Waterfall model of SDLC is accurate methodology for this project as the project tends to be less complex and small scaled oriented.

#### Design Pattern

**Model View Controller (MVC)** design pattern is used as the design pattern for this project as it is commonly used mostly of the programming language nowadays which is effective and easy to understand. Both desktop-based and web based can use MVC design pattern commonly used by most of the programming language. Here **Model** represents an object which has logic to update the data changes. **Views** represents the visualization of the data contained and **Controller** represents the act of both model and view that controls data flow into the objects and update the data changes.

**Advantages of MVC pattern are:**

1. It offers support for quick and parallel development that helps in developing web application by dividing the work where View and Controller are worked by different developer.
2. It supports Asynchronous Technique in web application that it can be madeto work even with PDF files, site that runs only on the specific browser, and also for desktop widgets.

**Disadvantages of using MVC pattern are:**

1. As they each are isolated the complexity is high in development of the application using this pattern there is no idea of work done by model view and controller.
2. With the application’s performance and architecture, it is not suitable for the small application which gives adverse effect.

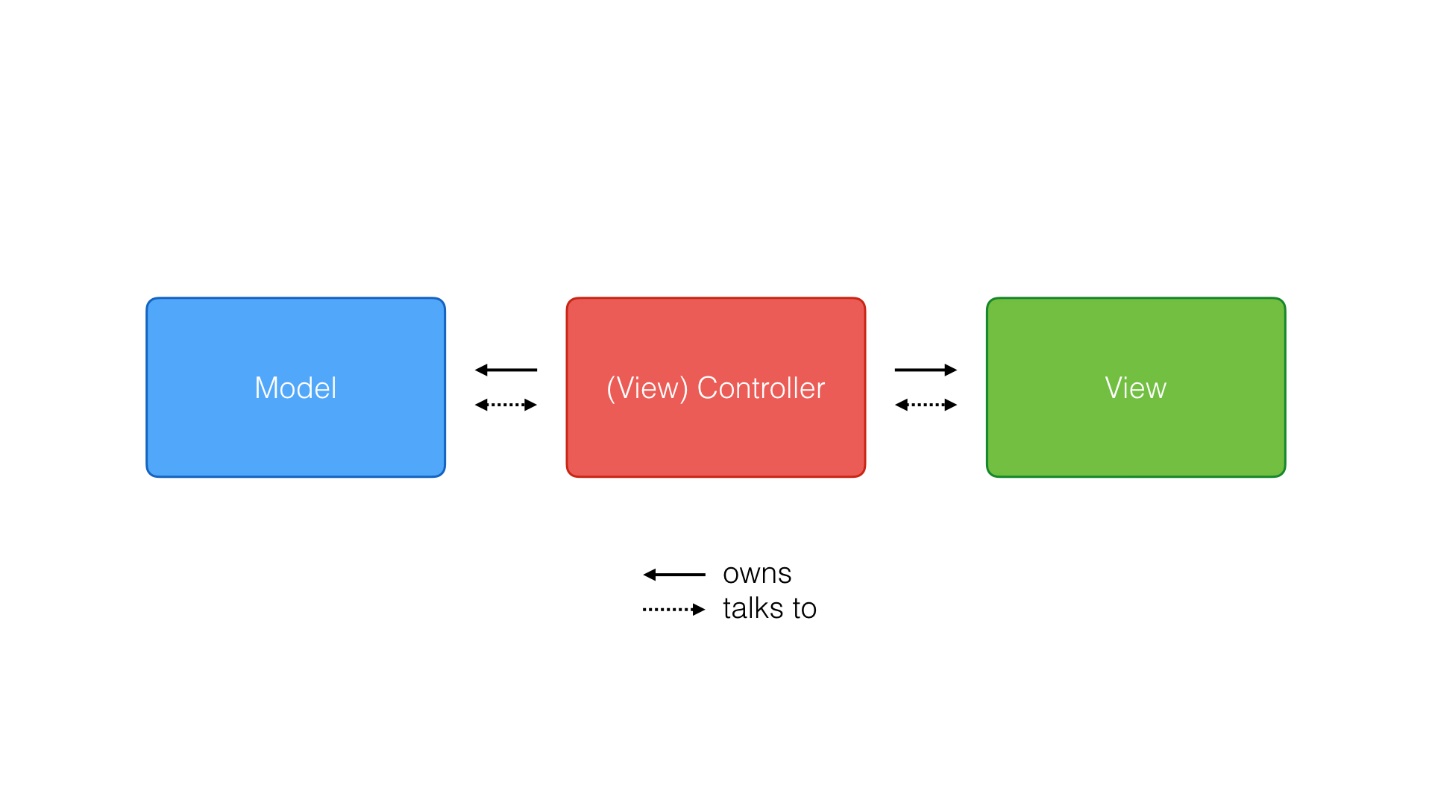


Figure 2: Design pattern used (MVC)

#### System Architecture

A system architecture defines the fundamental union of the system with its components ambled with their relationships and the environment with the theme of governing its design and evolution that helps to briefly system behavior, system view and its structure which is conceptual. Three-tier Architecture is used for this project like:

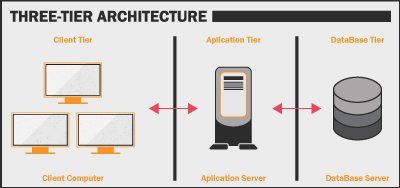


Figure 3: System Architecture (3- tier architecture)

1) Client tier: The credit of building the system with HTML, CSS and JavaScript in the system and deploying in the computer is known as Presentation Tier.

2) Application tier: The tier that supports the core functionalities and support that also contains business logic is known as Application tier.

3) Data tier: The tier that contains programs and the database to read and write onto is Data tier which manages the system of read and write to database managed by MYSQL, Microsoft SQL Server etc.

# Chapter 3: Design

## 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.

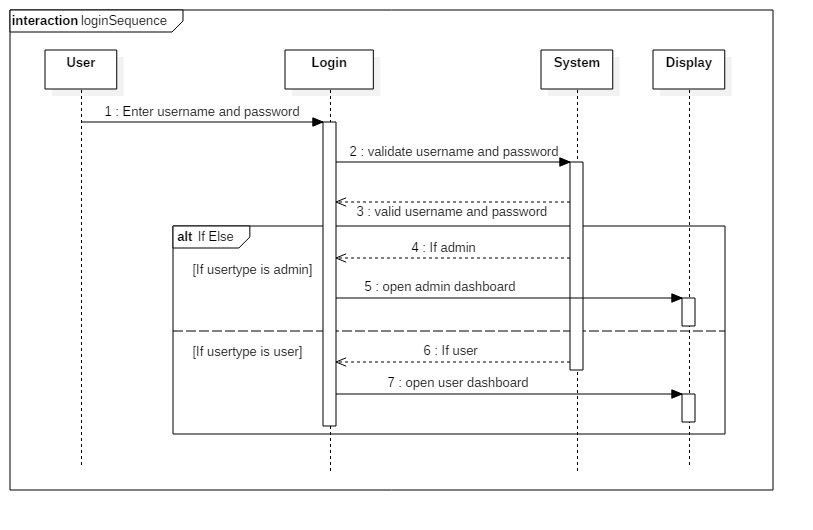
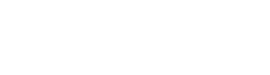
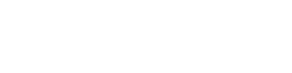


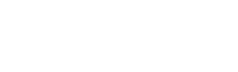
Figure 6: Login Sequence Diagram



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.

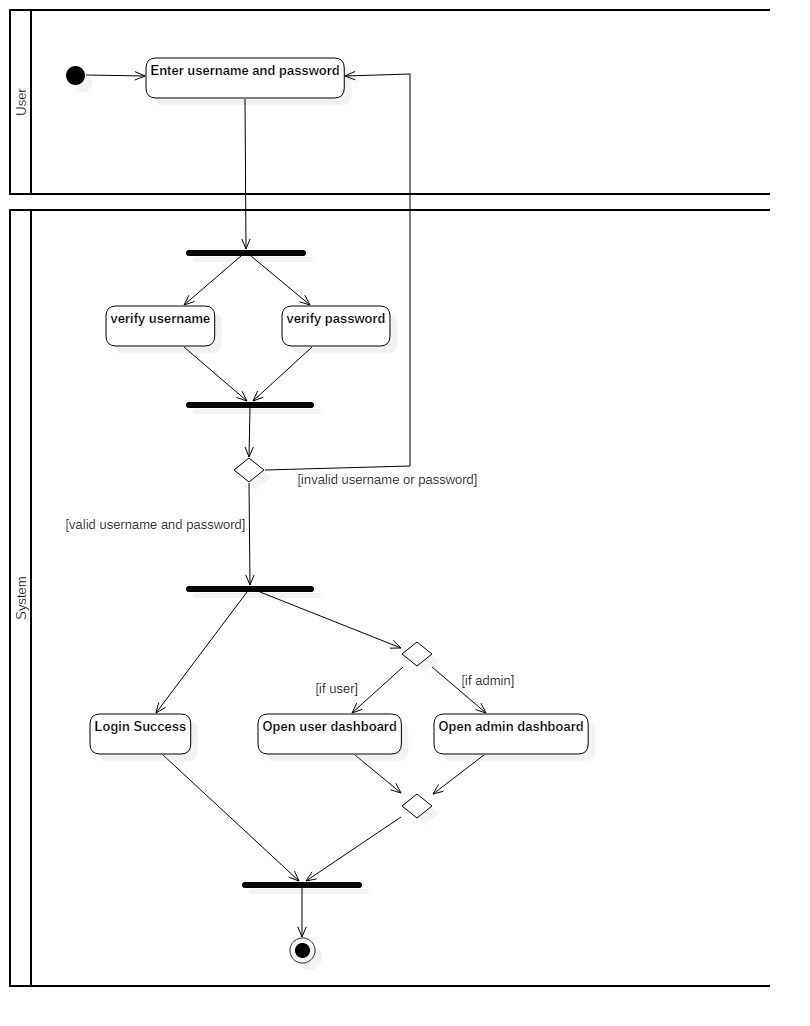
****

Figure 6: Login Activity Diagram

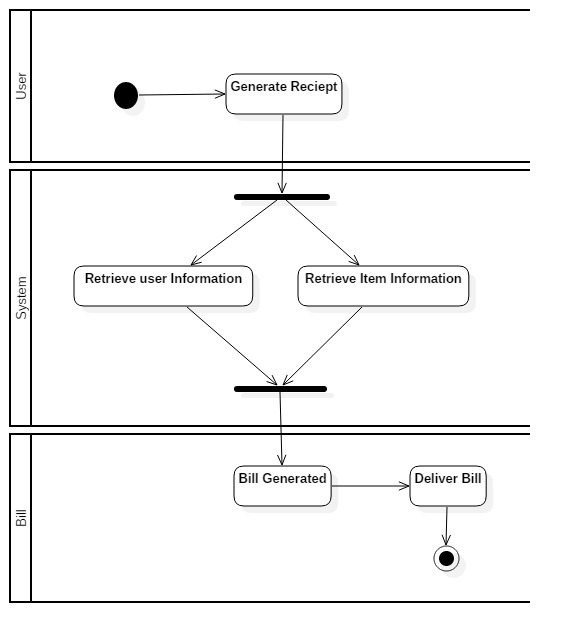


Figure 7: Activity Diagram to generate bill

## 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.

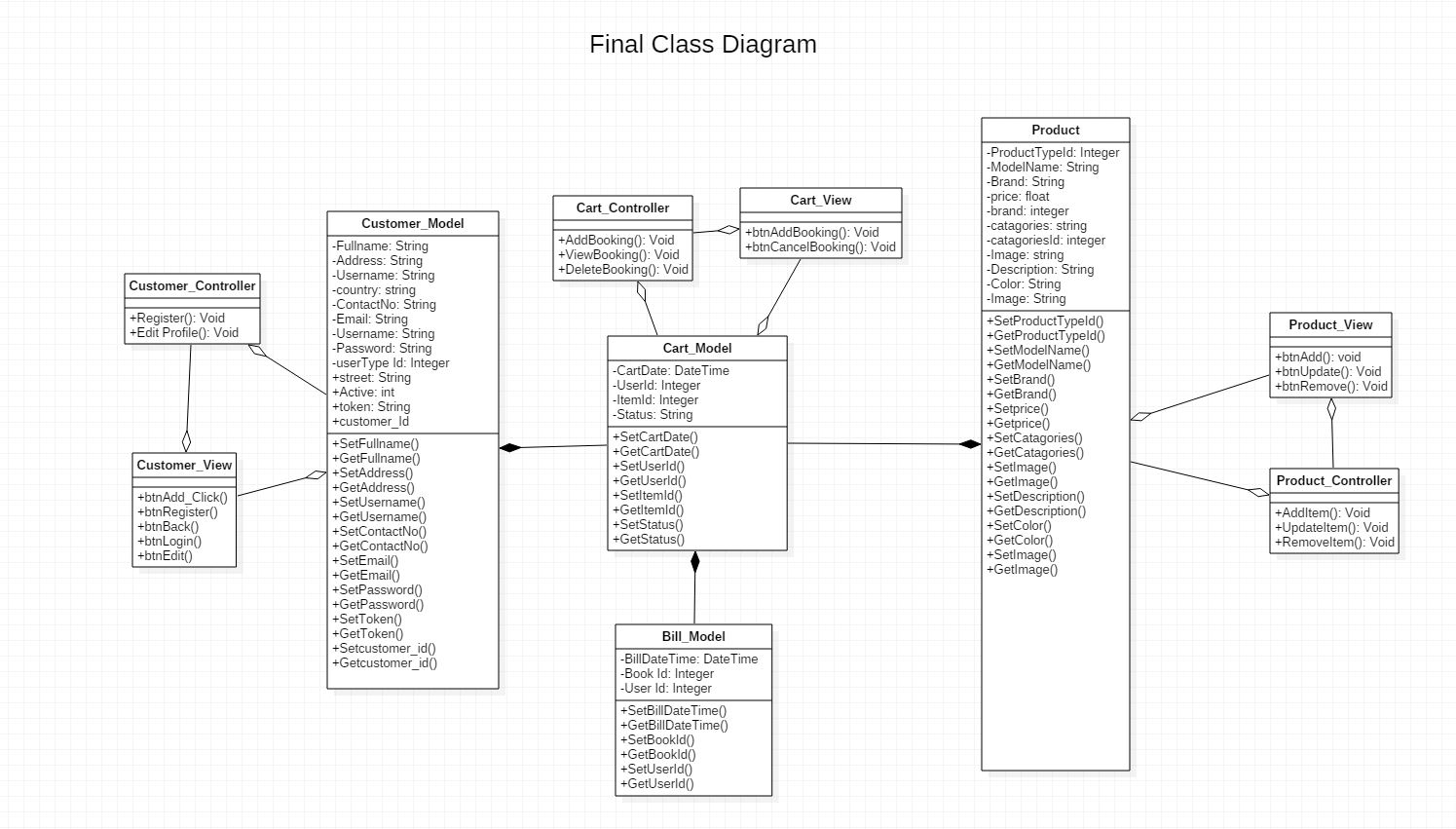
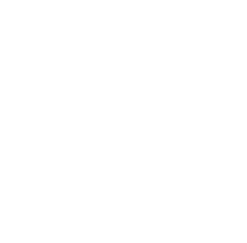


Figure 8: Final Class Diagram

### 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.

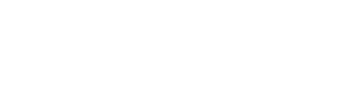
The Level 0 Data Flow Diagram or context diagram is shown below:



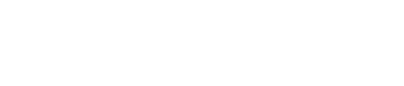
E

-

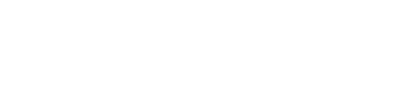
commerce



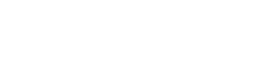
Database



Administrator



User



Visitor



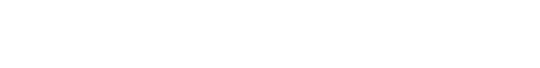
Website management



Website activity



Purchase



Retrieve purchase detail and

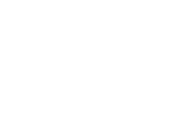
orderdetails



Figure 9 : Level 0 DFD Diagram

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

Level 1 Data flow diagram



Customer

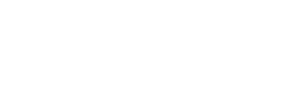


Enquiry



Availability

checked



Registration success

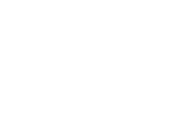
message



1



Customer details



Customer



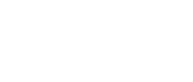
Check

products



Availability

checked



Product available

message



2



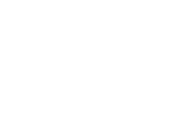
D2



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.

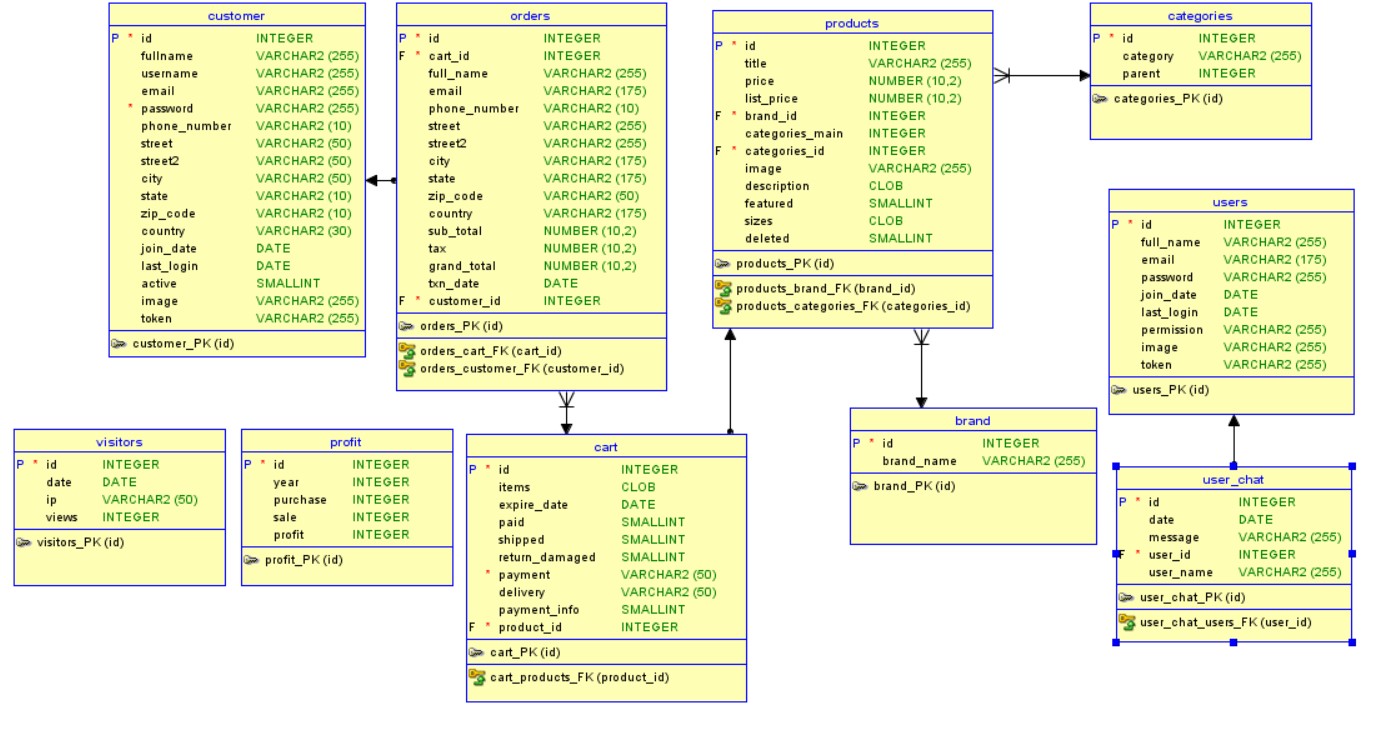


Figure 11: ER Diagram

## Data Dictionary:

Figure 12: Customer Type

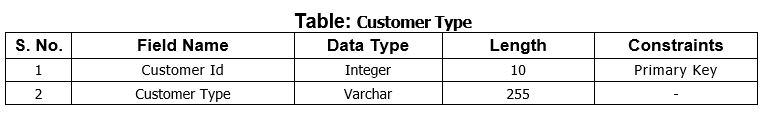


Figure 13: Item Type Data Dictionary

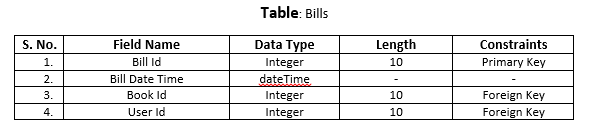
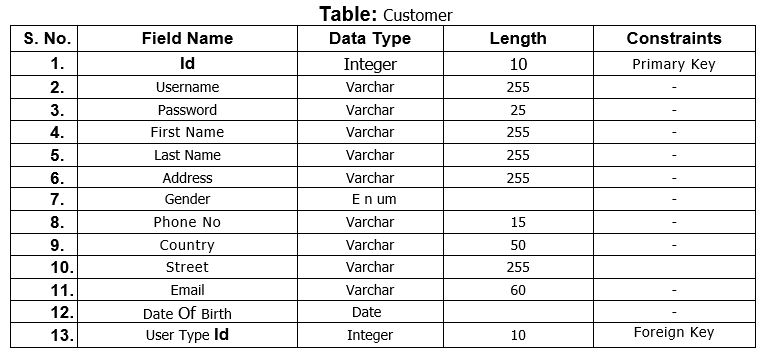
****

Figure 14: Customer Data Dictionary

Figure 15: Bills Data Dictionary

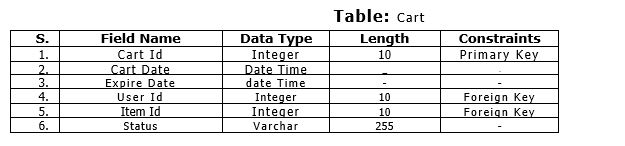


Figure 16: Carts Data Dictionary

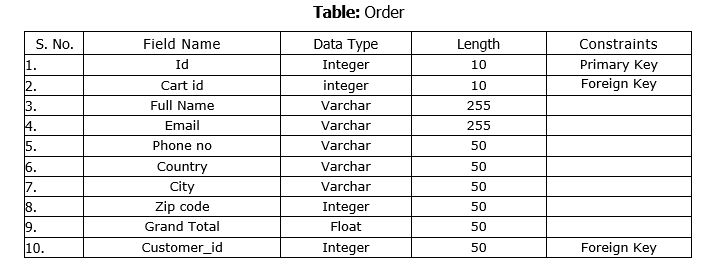


Figure 17:Order Data Dictionary

**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:**

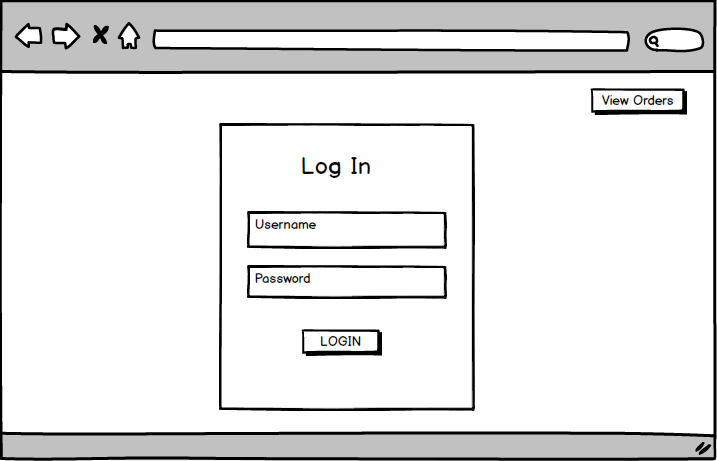


Figure 18: Login Page Prototype

Figure 19: Prototype of Sign Up

1. **Home Page:**

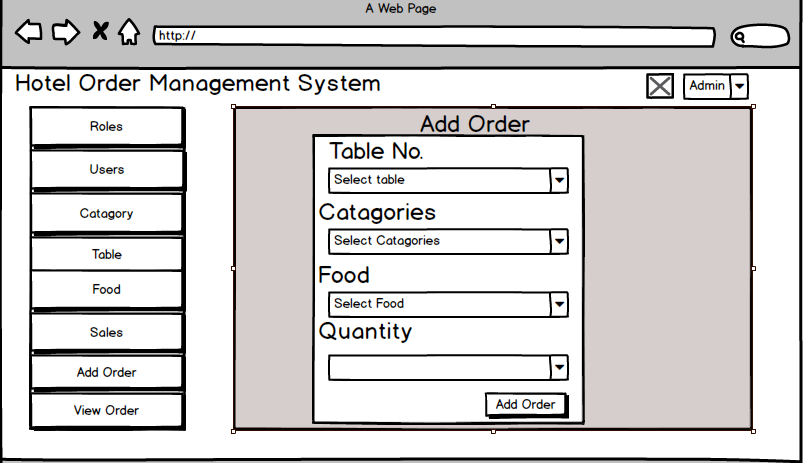


Figure 20: Prototype of Home Page

1. **Add User Page:**

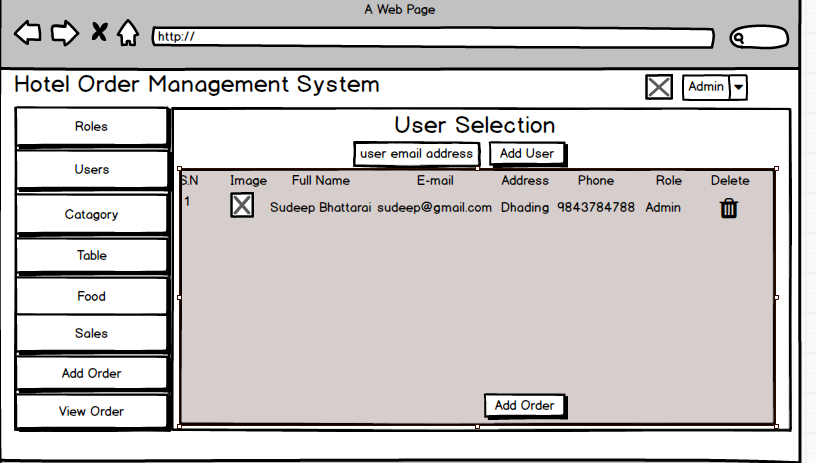


Figure 21: user selection Prototype

1. **View Order**

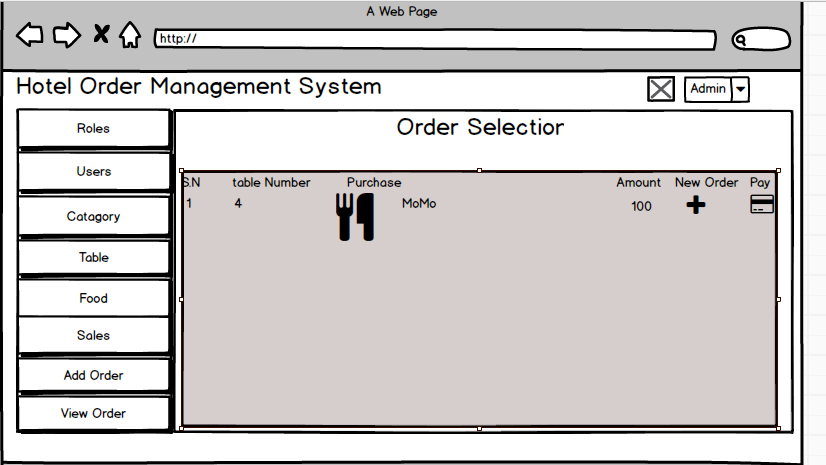


Figure 22: Prototype of order Page

1. **Sales Page:**

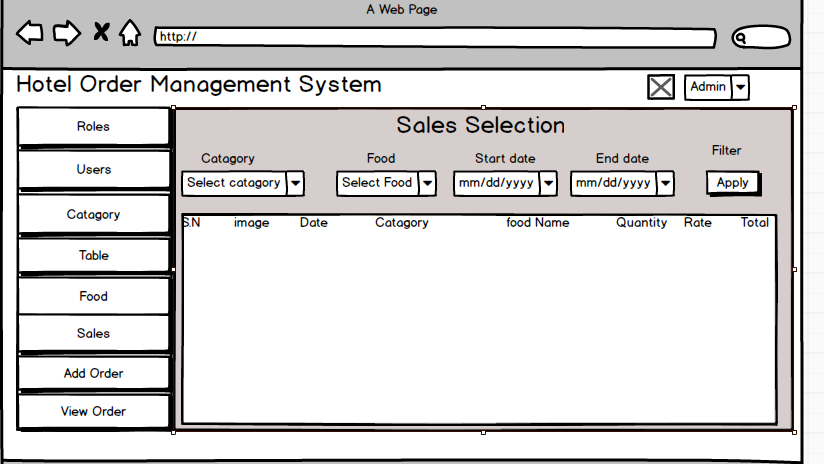


Figure 23: prototype of Salesr page

1. **Waiter Page**

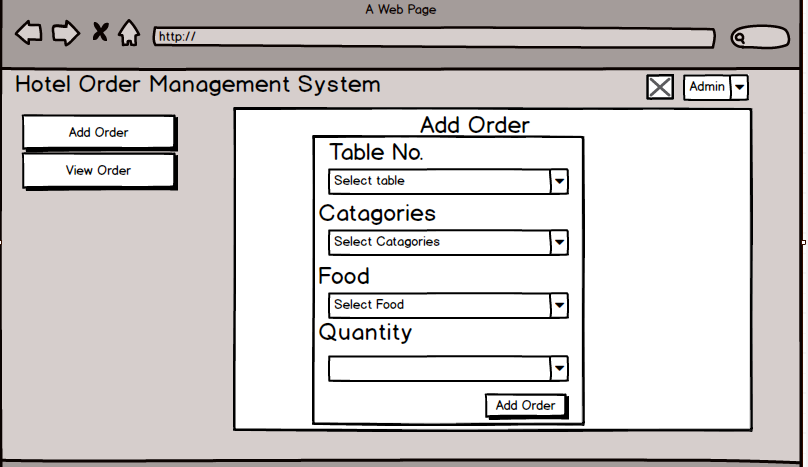


Figure 24: Prototype of Waiter Page

# Chapter 4: Implementation

The main objectives of this session is the code. After Analysis and Design Implementation is the process in software development life cycle.

### Choice of language:

While implementing the codes and database creation is done where I have chosen PHP as a programming language and MySQL as a database. The main objective of choosing PHP is that it allows developer to develop the page with ease and quickly and also it is known as a great platform to create database driven website with a lot of features so I have chosen PHP as a programming language for this project.

### Development Environment:

1. PHP

A programming language used widely for web development that can be embedded into HTML is PHP. It is used widely, cast to general propose for the developer that is open sourced. It provides cross-platform tools to create dynamic website so I have chosen this programming language. It is suitable for all platforms like UNIX, Linux or Windows server. Due to its compatibility with MYSQL there is plus point to use this language in this project. It is also used for CRUD functionality. So, Using PHP any company or Individual can run their web page in a creative way with creative designing.

1. MySQL

Database mean the platform to store data and it is important to have database for any webpage that need to store data flowed through it. MySQL is a platform where database can be managed.

##### Frameworks

1. Bootstrap

For the development of responsive and mobile-first websites the commonly used CSS framework called Bootstrap is used. Mostly the developer uses this framework to design their websites that helps in dynamic design of the website and also it helps in fast coding.

1. Text Editor

For the coding Bracket as a text editor is used

# Chapter 5: Testing

Evaluation of developed software to ensure the quality of the products and defects on it is known as Testing. There are other approaches of testing methods like Black box testing and white Box testing.

## 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.

**Advantages of White Box Testing:**

1. Optimizing code by identifying unseen faults.
2. Cases for white box testing can be easily automated.

**Disadvantages of White Box Testing:**

1. Testing the white box can be pretty composite and expensive.
2. White-box testing is slow; it takes time for bigger programming applications.

## Black Box Testing:

`It is also called input-output / behavioral testing. It is a method of software testing in which the functionality of the test software is evaluated without looking at the structure of internal code.

**Advantages of Black Box Testing:**

* Could be a non-technical test tester.
* · Detailed functional knowledge of the system is not required by the tester.
* Testing aids in the identification of contradictions and ambiguity in functional specifications.
* Upon completion of the functional specifications, each test case can be designed.

**Disadvantages of Black Box Testing:**

* Along with test cases, design is challenging without clear functional specifications.
* · If test cases are not developed on the basis of specifications then tricky inputs would be hard to identify.

Different test cases are 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 2: Test Log for User Registration

***Unit Testing:***

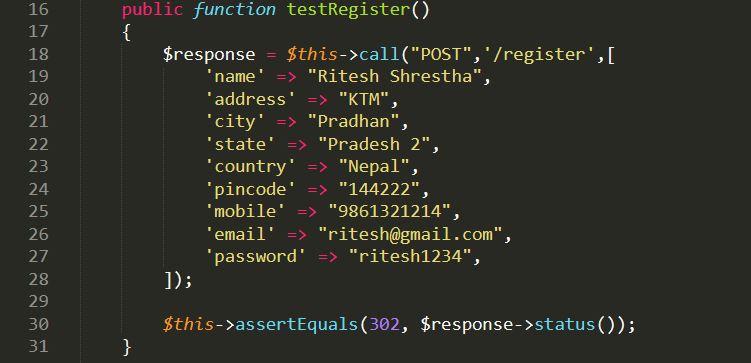
****

Figure 25: Test Case for Register

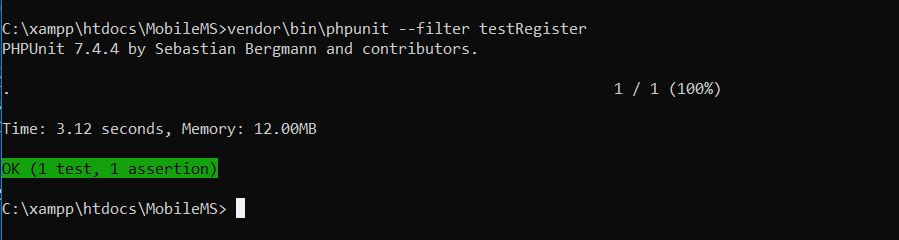


Figure 26: Unit Testing Result for Register

***Black Box Testing:***

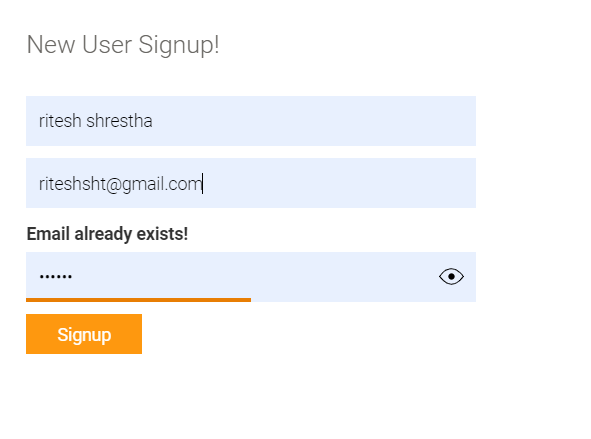


Figure 27: Registration of Black Box Testing

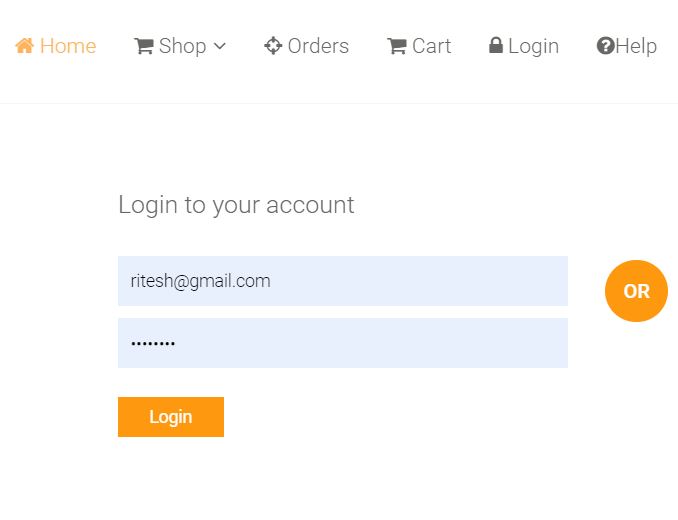
****

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 3: Test Log for Adding Item Type

***Unit Testing:***

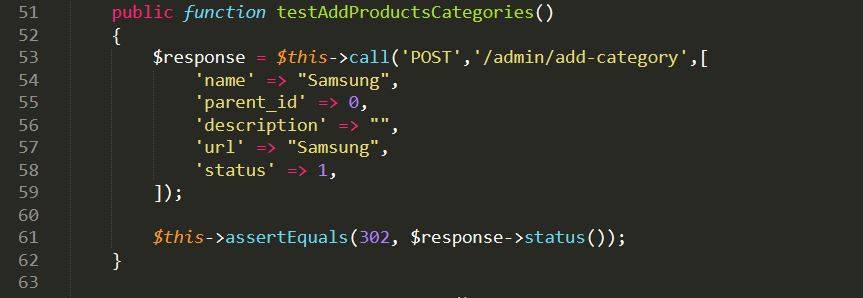
****

Figure 29: Test Case for Adding Item Type

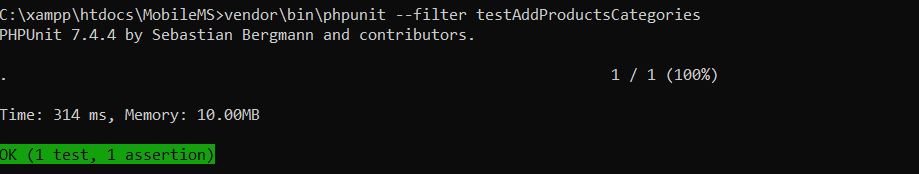
****

Figure 30: Unit Testing Result for Adding Item Type

***Black Box Testing:***

**Product add**

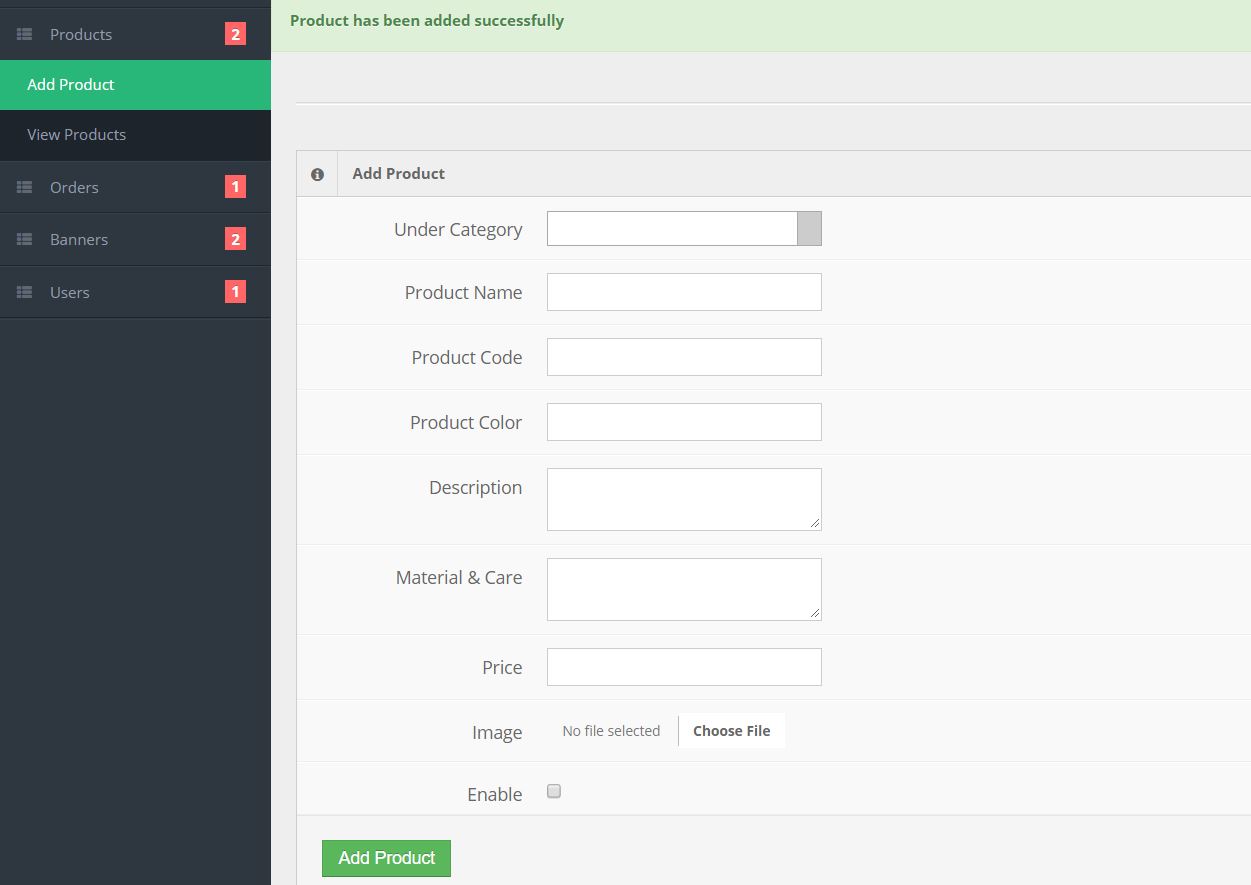
****

Figure 31: Adding Product Black Box Testing success

**View of added product**

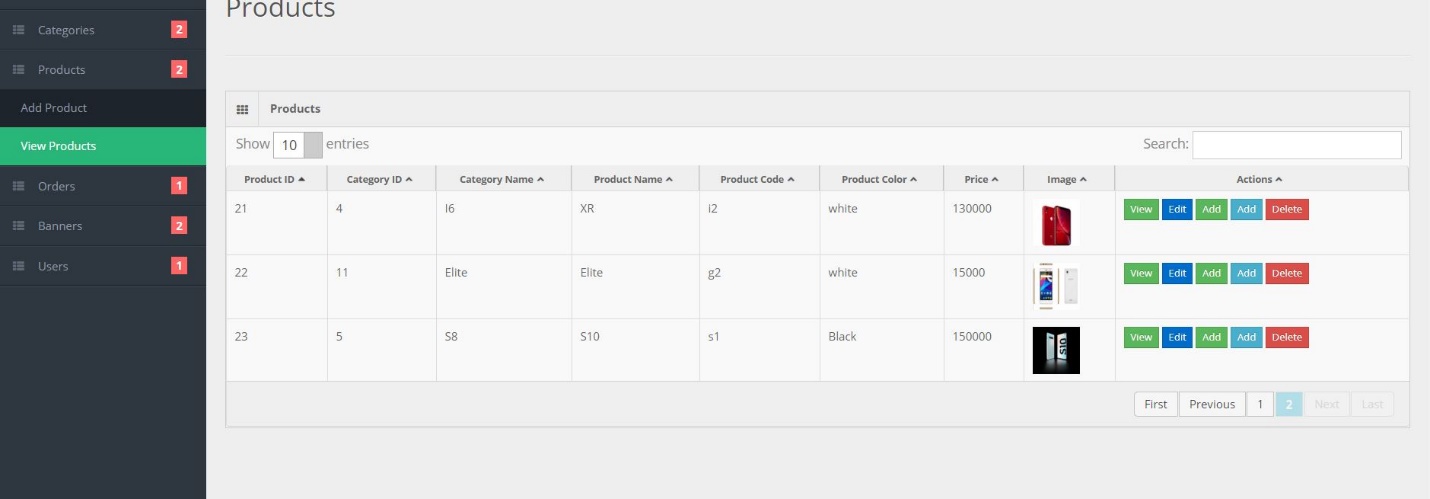
****

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 4: Test Log for Updating Item

***Unit Testing:***

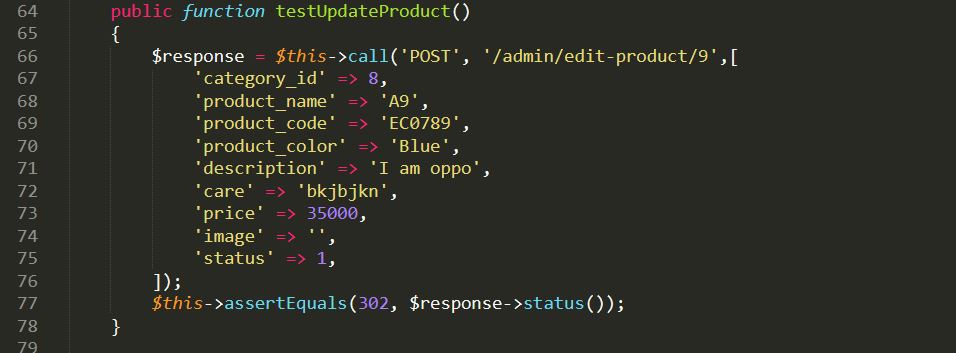
****

Figure 33: Test Case for Updating Item

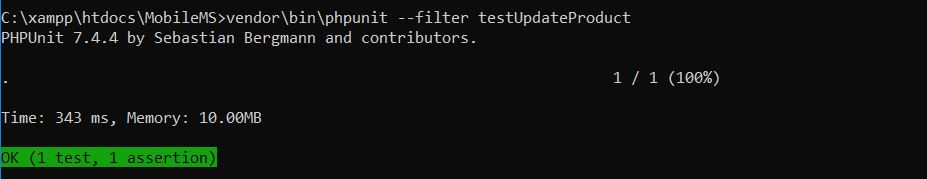
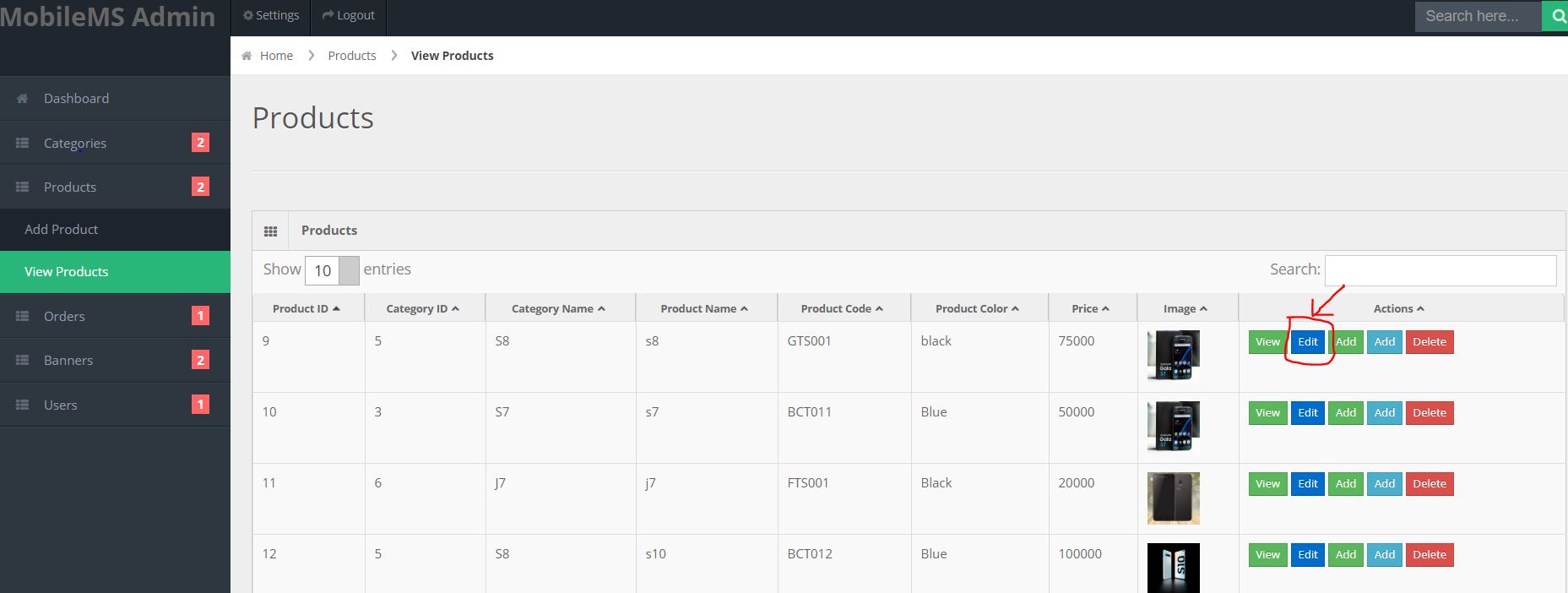
******

Figure 34: Unit Testing for Updating Item

***Black Box Testing:***

****

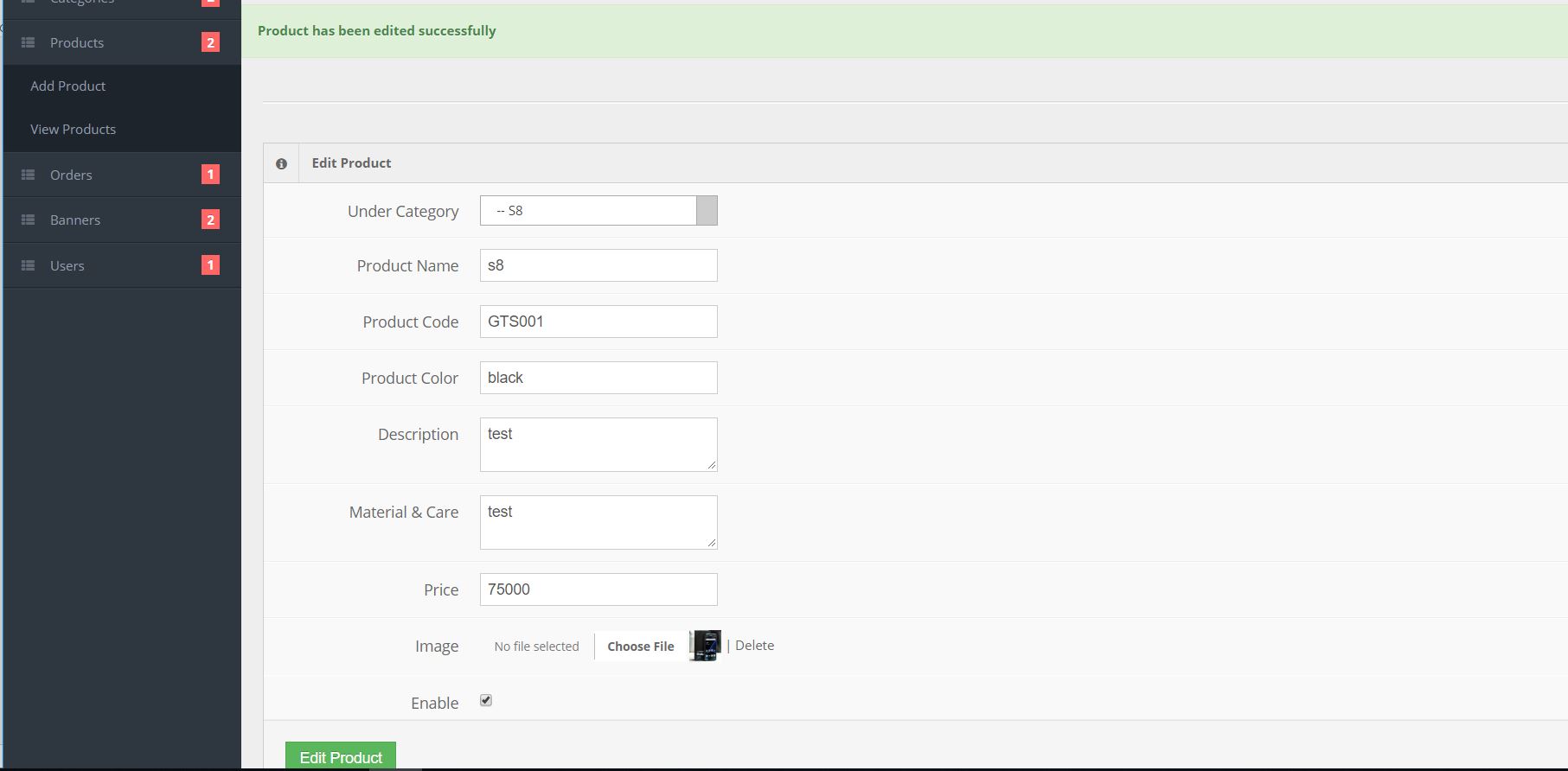
****

Figure 35: Black Box Testing for Updating icon

Figure 36: edit/update success

* + - 1. **Item Delete**

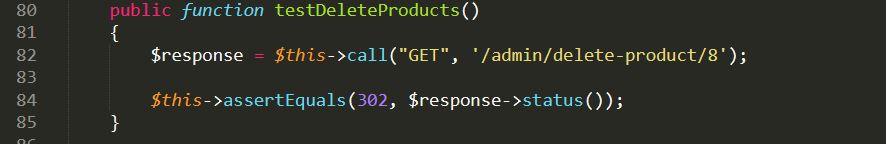
***Test log:***

|  |  |
| --- | --- |
| 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 5: Test Log for Item Delete

***Unit Testing:***

Figure 37: Test Case for Item Delete

****

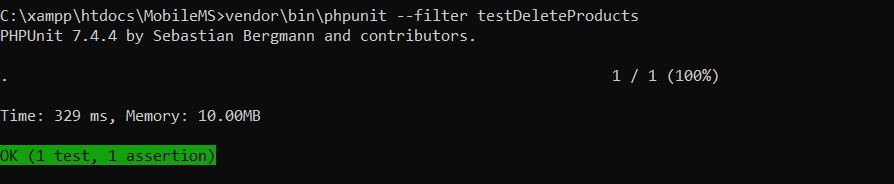
****

Figure 38: Unit Testing for Item Delete

***Black Box Testing:***

******

Figure 39: Black Box Testing for Item Delete

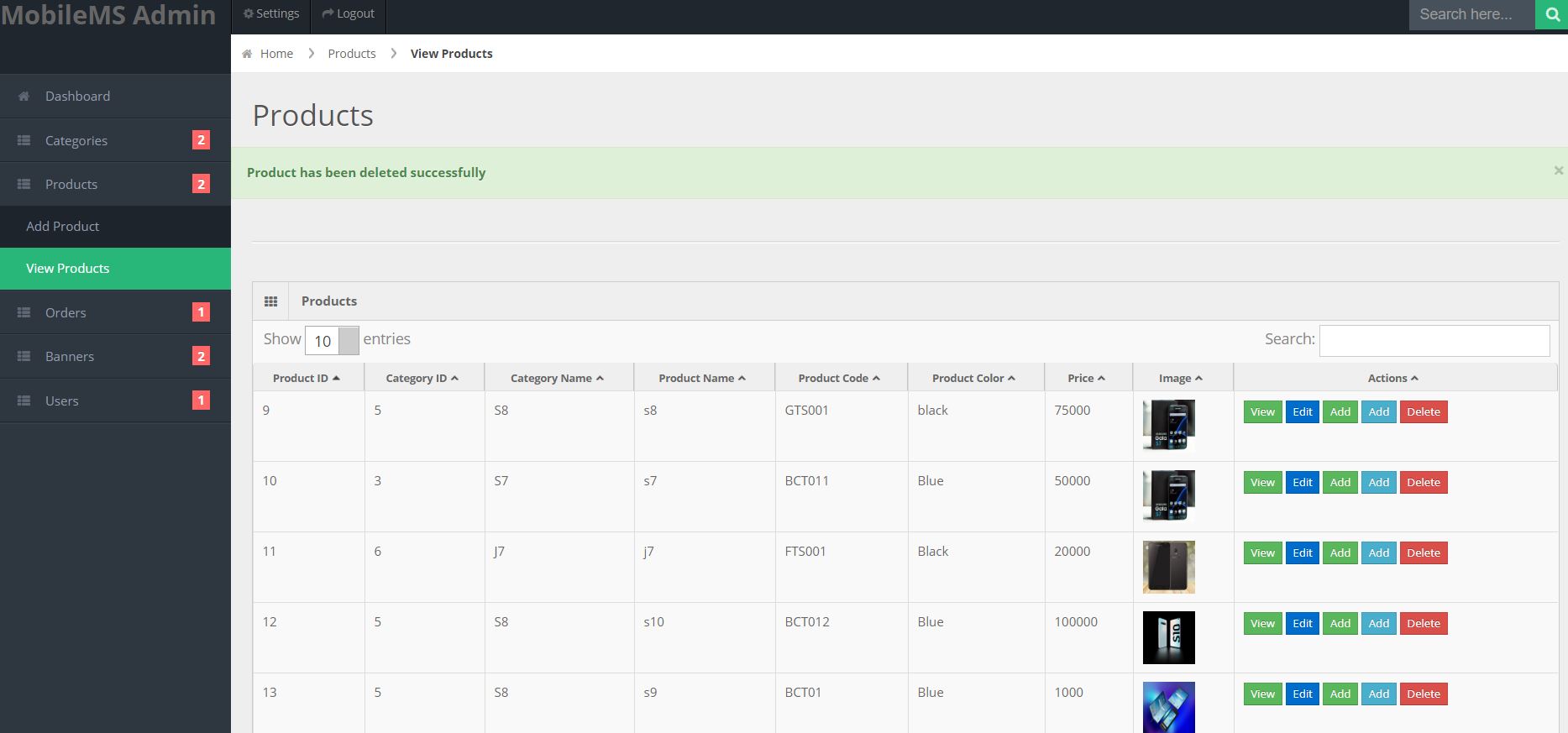
******

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 6: Test Log for cart

***Unit Testing:***

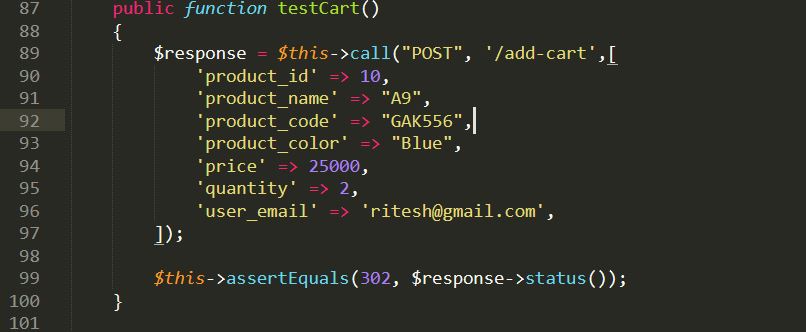


Figure 41: Test Case for cart

****

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 7: Test Log for Order

***Unit Testing:***

****

Figure 44: Test Case for Order

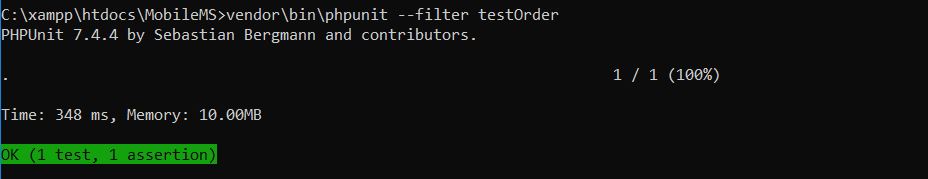
****

Figure 45: Unit Testing Result for Order

***Black Box Testing:***

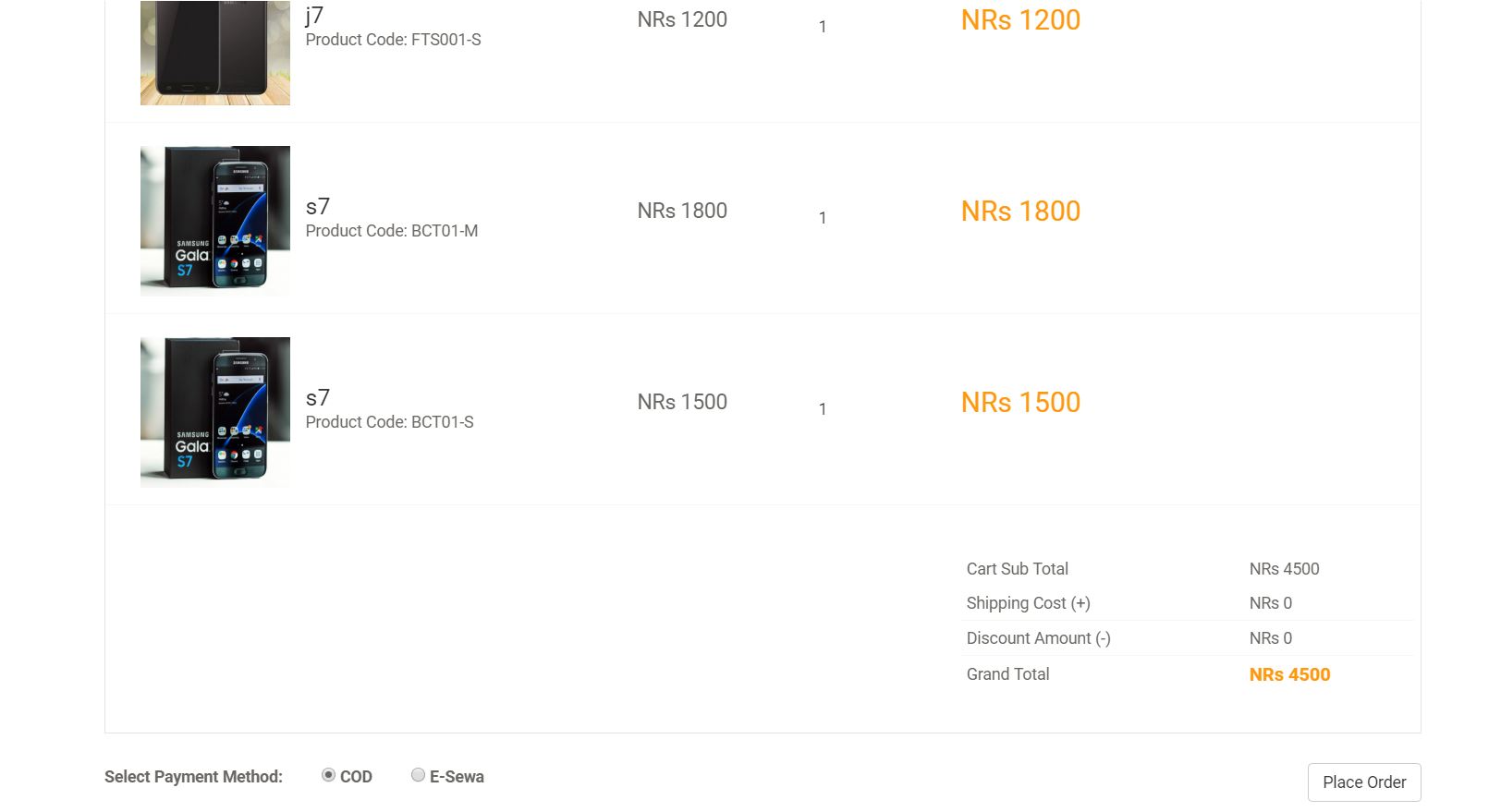


Figure 46: placing Order

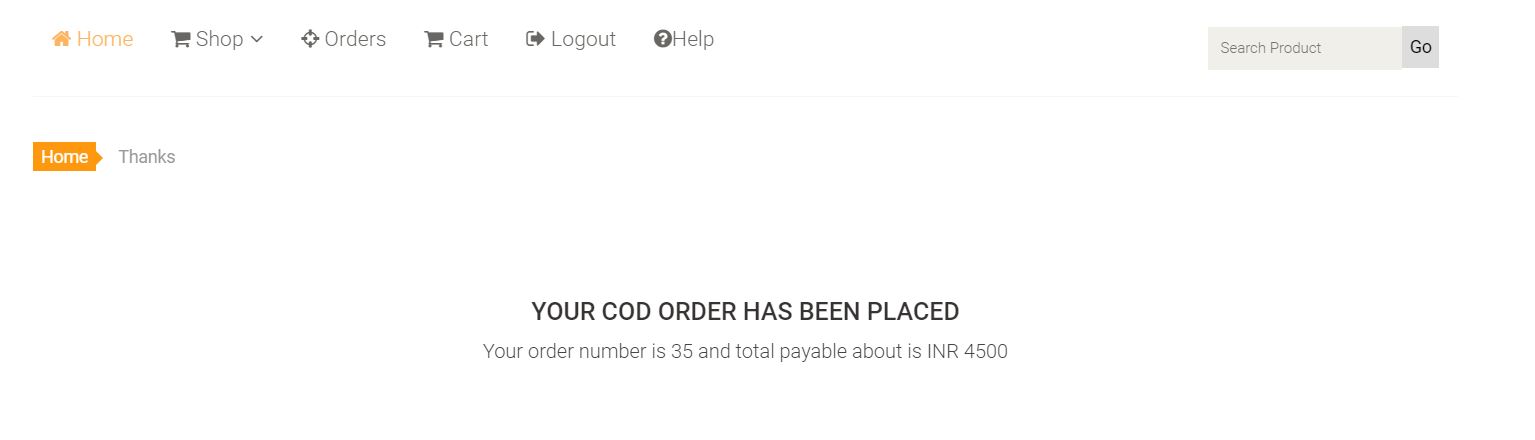
****

Figure 47: Order success

1. **Search**

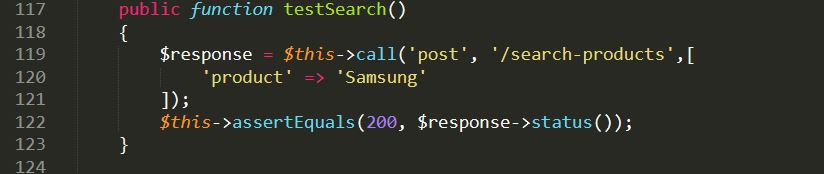
***Test log:***

|  |  |
| --- | --- |
| 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 8: Search test log

***Unit Testing:***

Figure 48: Search test case

****

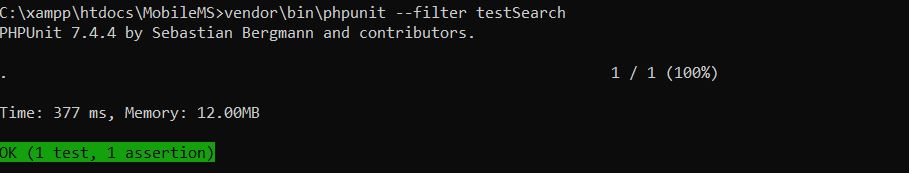
****

Figure 49: search Unit Testing

***Black Box Testing:***

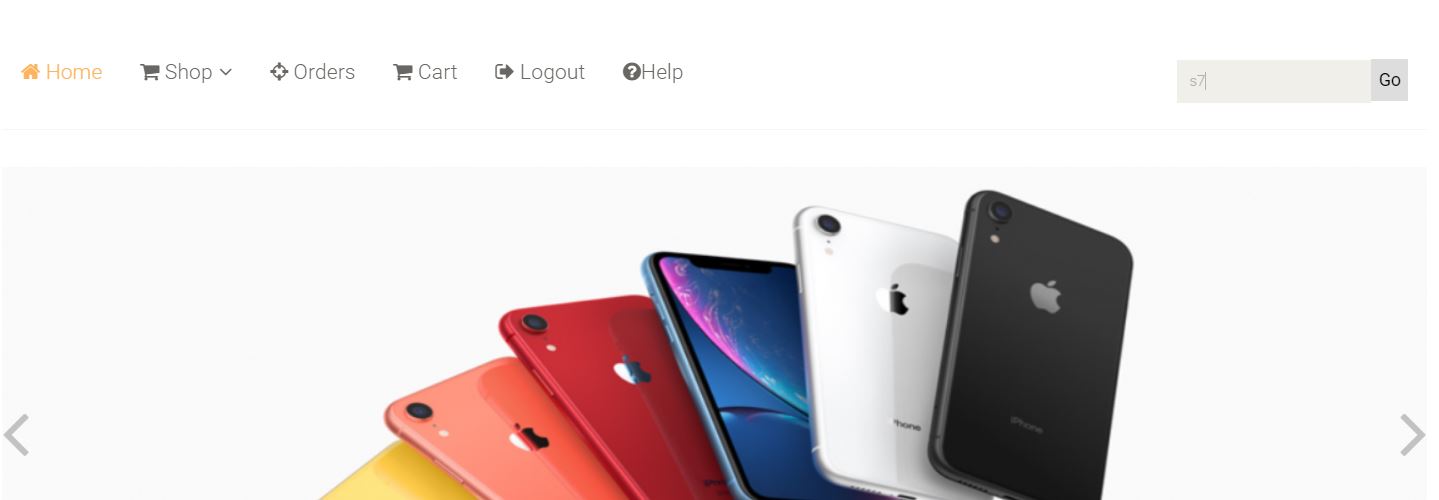


Figure 50: search s7

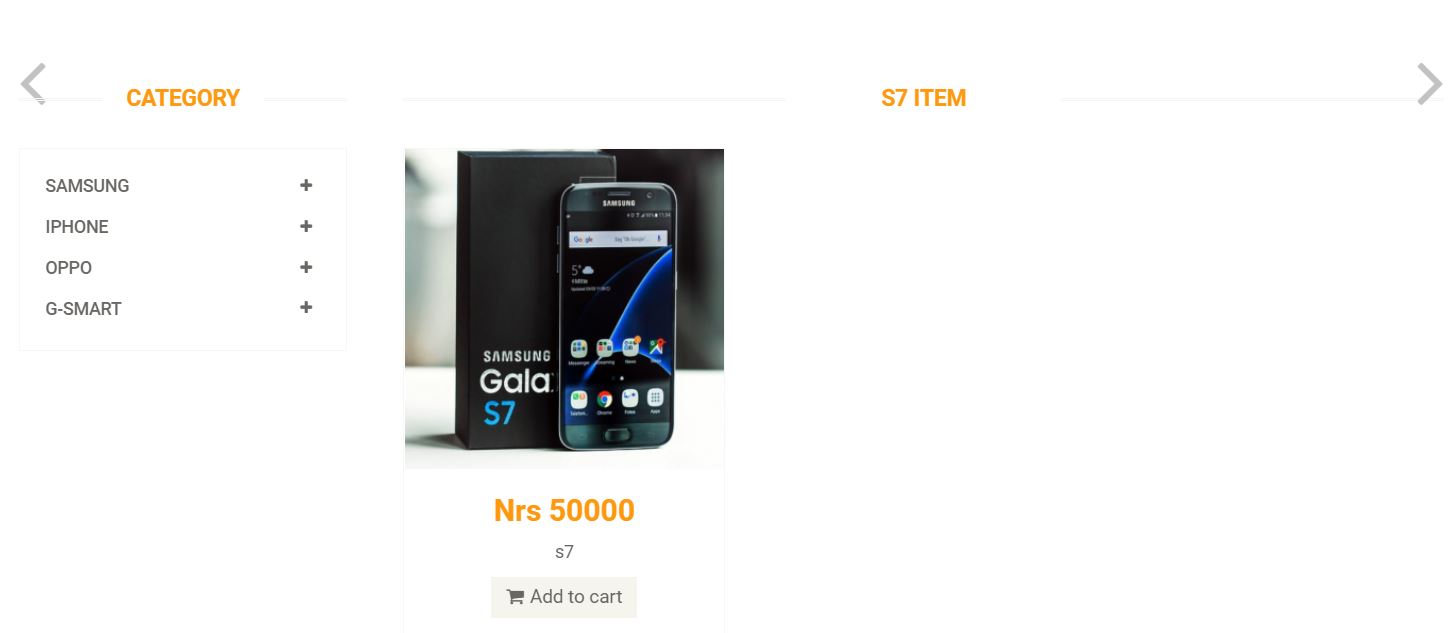
****

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:***

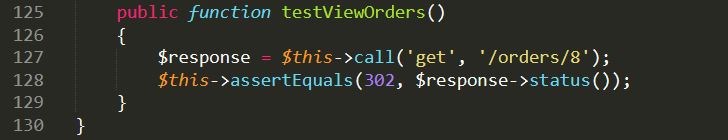
******

Figure 52: Test Case for view order

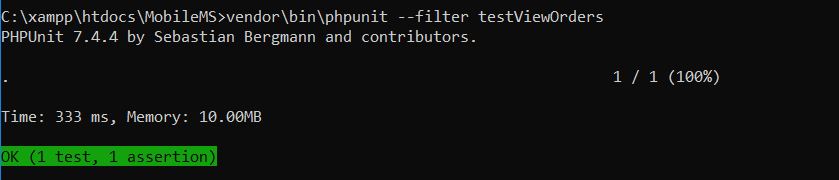
******

Figure 53: view order Unit Test pass

***Black Box Testing:***

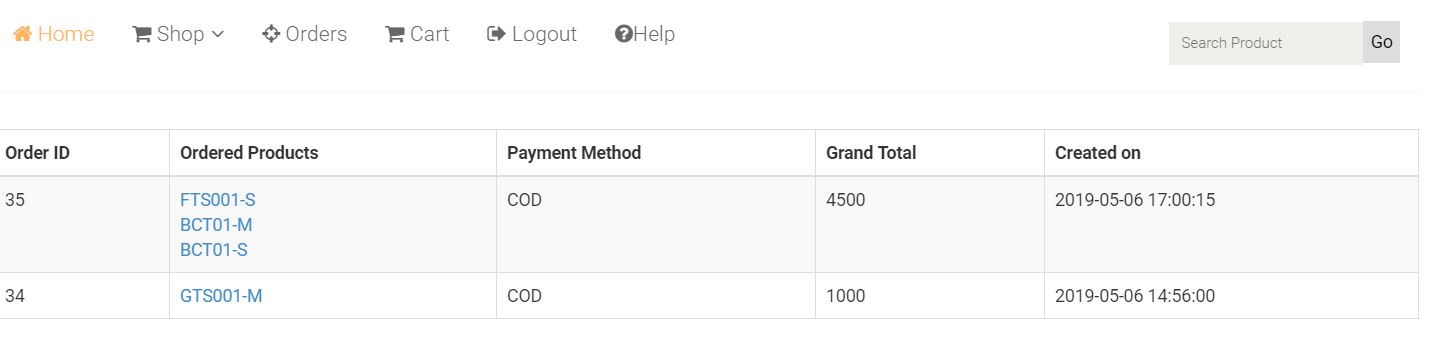
****

Figure 54: view order of Black Box Testing

# Chapter 6: Other Project Issues

# **Risk Management:**

The identification and controls over the threats of an organization’s investment and grossing that arises from the extreme variety of sources including all the financial uncertainty, legal labiates, strategic management errors, natural disasters of other type of accidents is known as Risk management.

**Identify the Risk**

Identification of the risk is the most part in a project. Different techniques are used to identify the Risk like:

1. **Analyze the Risk**

We analyze the potential of harming a project of a risk and understand the management system goals and objectives with the likelihood and consequences.

1. **Evaluate or Rank the Risk:**

After the consequences and likelihood are analyzed we bother its acceptance if is serious enough for the warrant treatment.

1. **Treat the Risk**

The highest ranked risk is accessed and planned for the treatment for minimizing the negative risks.

1. **Review the Risk**

In this process we take the risk as a point and trace the review for the risk.

The impact on the system can be calculated by using following formulae

**Impact = Likelihood X Consequences**

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

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

**Risk Management table:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **RID** | **Risk Name** | **Likelihood** | **Consequence** | **Impact** | **Solution** |
| 1 | Bug on the Code | 2 | 2 | 6 | Proper testing should be done before deployment. |
| 2 | Server failure | 1 | 5 | 5 | Backup should be done time to time |
| 3 | Untrained Developer | 1 | 5 | 5 | Proper training should be given. |
| 4 | Malware/ Virus | 2 | 5 | 10 | Antivirus and anti-malware software should be installed. |
| 5 | Strategy Risk | 2 | 5 | 10 | A well plan project should be implemented |
| 6 | External Factors | 1 | 3 | 3 | Proper safety measures for hazards. |

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 9: Likelihood

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

Table 10: 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 11: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.

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

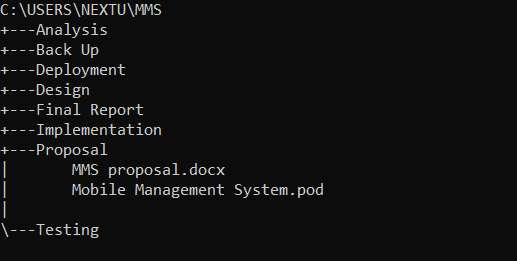
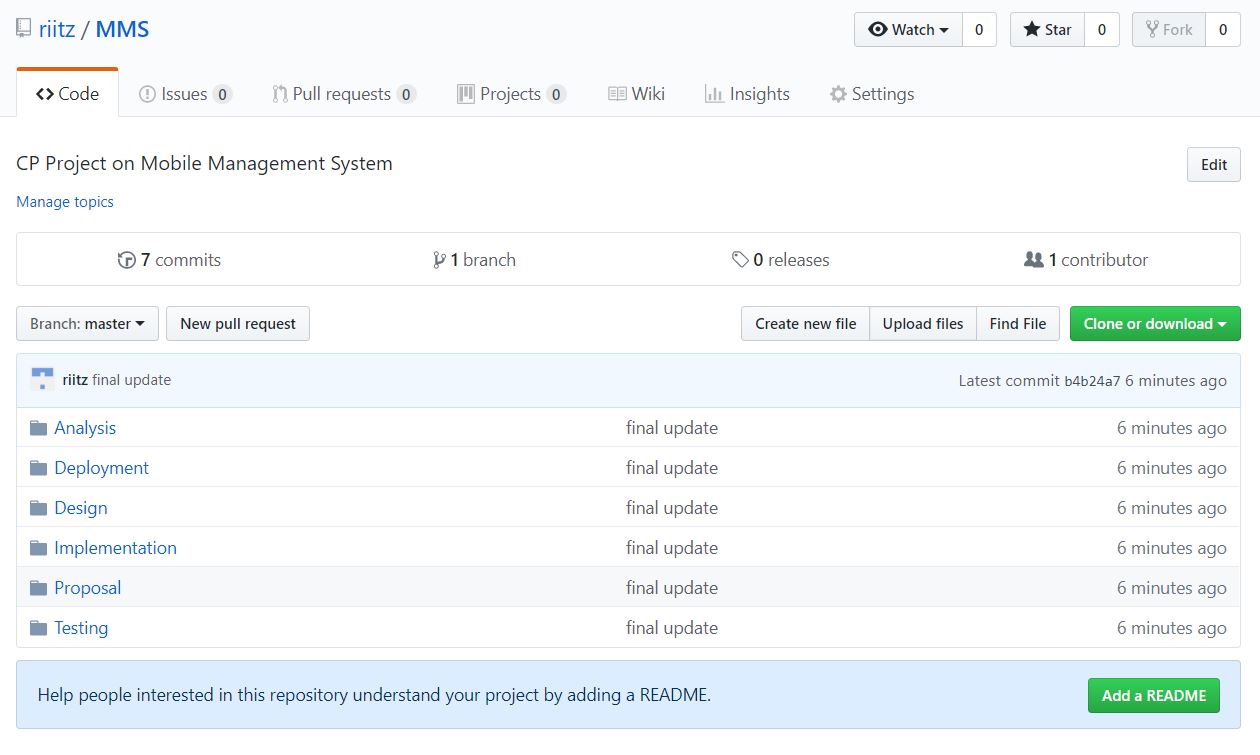


Figure 55: Tree Chart

**GitHub**

Milestones

It is often known as the planning or scheduling of the contents of the task of the project that helps to complete any project in required interval of time. Milestone is also known as the visual indicator of the progress of the project towards its objectives which is also helpful in understanding the structure of the project.

|  |  |  |  |
| --- | --- | --- | --- |
| **Hotel Order Management System** | | | |
| **WBS number** | **Milestone** | **Date (2019)** | **Days** |
| **1** | **Proposal** | **(16 June – 1 July )2019** | **16 days** |
| 1.1 | Scope and objectives | 16 June- 19 June | 4 days |
| 1.2 | WBS, Milestone and Gantt Chart | 20 June – 23 June | 4 days |
| 1.3 | Risk Management | 24 June – 26 June | 3 days |
| 1.4 | Configuration Management | 27 June – 29 July | 3 days |
| 1.5 | Submission | 30 July– 1 July | 2 days |
| **2** | **Analysis** | **(2 July- 29 July) 2019** | **28 days** |
| 2.1 | Feasibility study | 2 July- 13 July | 12 days |
| 2.2 | Requirement Specification | 14 July – 26 July | 13 days |
| 2.3 | Use Case Diagram | 27 July – 29 July | 3 days |
| **3** | **Design** | **(30 July – 29 August) 2019** | **31 days** |
| 3.1 | Structural Model  [Class Diagram] | 30 July – 6 August | 8 days |
| 3.2 | Behavior Model | 7 August – 14 August | 8 days |
| 3.3 | Database design  [ER diagram]  [Data-Dictionary] | 15 August – 22 August | 8 days |
| 3.4 | User Interface (UI) Design | 23 August - 29 August | 7 days |
| **4** | **Coding** | **(30 August – 20 September) 2019** | **22 days** |
| 4.1 | Build Database | 30 August – 7 September | 9 days |
| 4.2 | Implementation of code | 8 September – 20 September | 13 days |
| **5** | **Testing** | **(21 September – 30 September) 2019** | **10 days** |
| 5.1 | Black Box Testing | 21 September- 23 September | 3 days |
| 5.2 | Unit Testing | 24 September – 27 September | 4 days |
| 5.3 | Validation Testing | 28 September- 30 September | 3 days |
| **6** | **Documentation** | **(1 October – 12 October) 2019** | **12 days** |
| 6.1 | User Manual | 1 October – 8 October | 8 days |
| 6.2 | Final Report | 9 October - 12 October | 4 days |

## 4.3 Scheduling / Gannt Chart

The scale of time leaved for the project is represented as a gannt chart with its milestone.

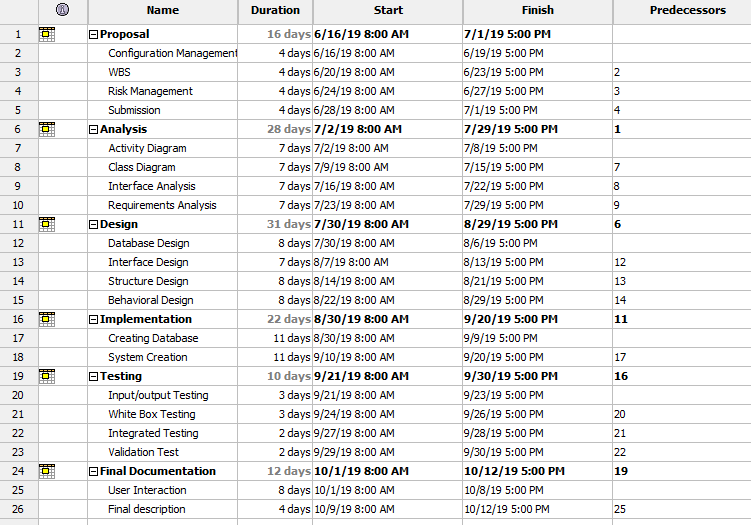


Figure 4: WBS in Project Libre

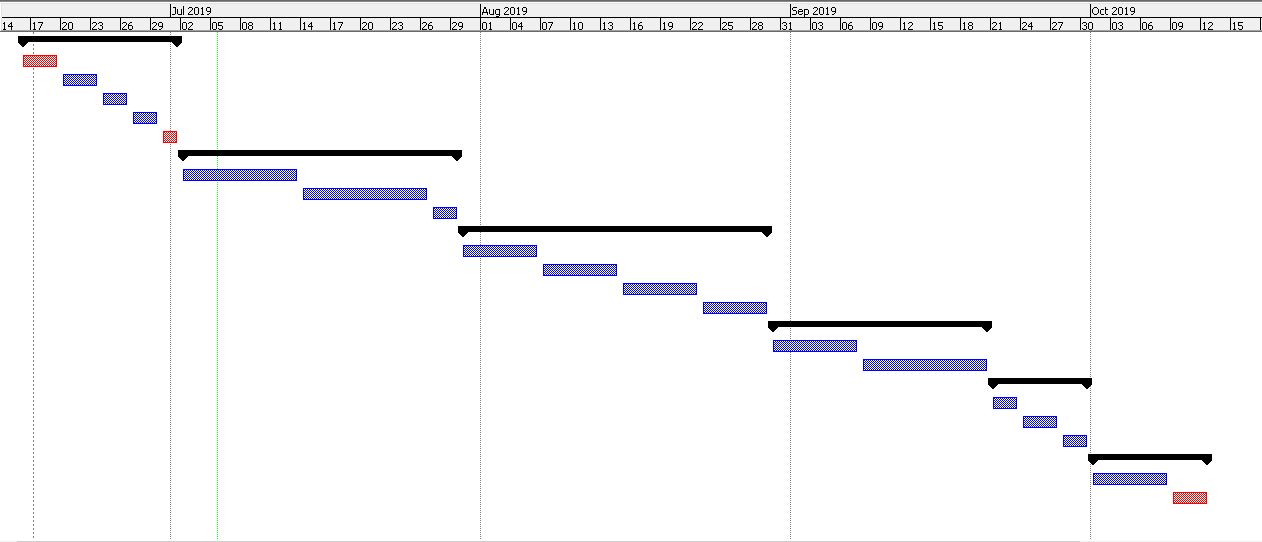


Fig: Gnannt chart

**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.

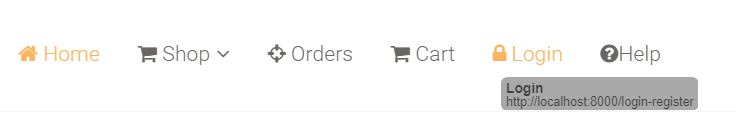
## Limitation:

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.

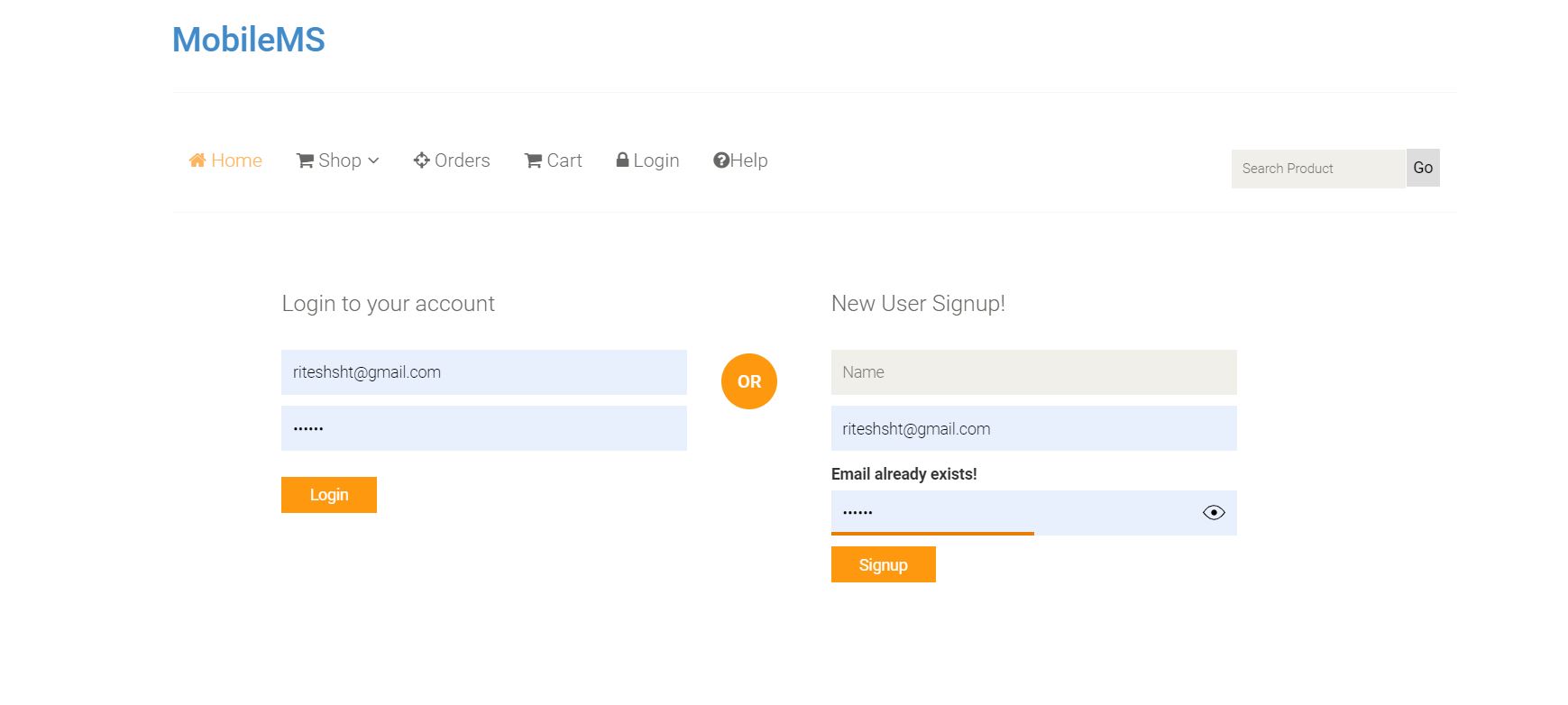
## User manual

1. **REGISTER/SIGN UP.**

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

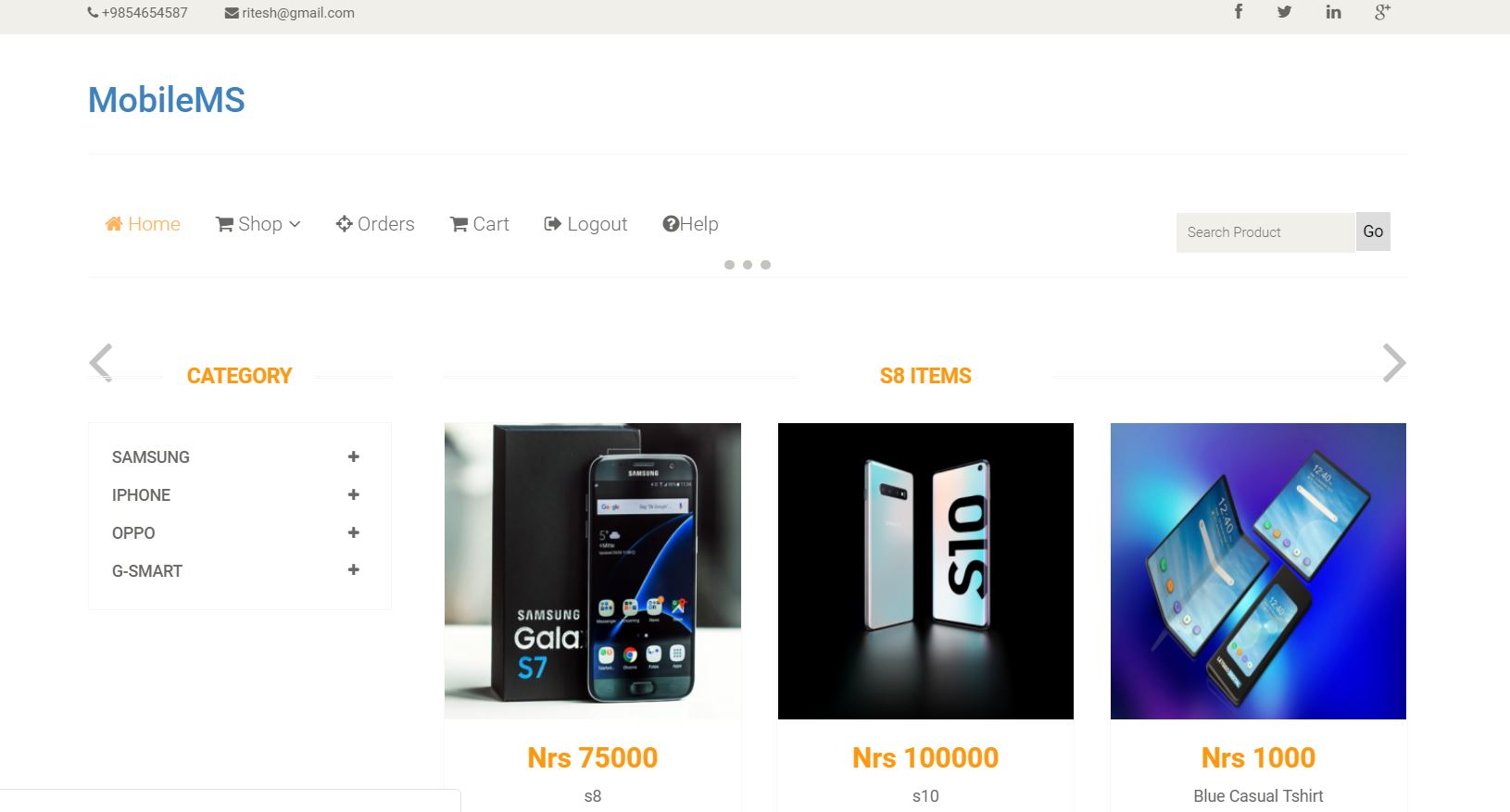


* The form is displayed below.
* Fill all the details in “New User Signup” to get registered for unregistered users.
* Or fill the details in “Login to your account” after for registered users.



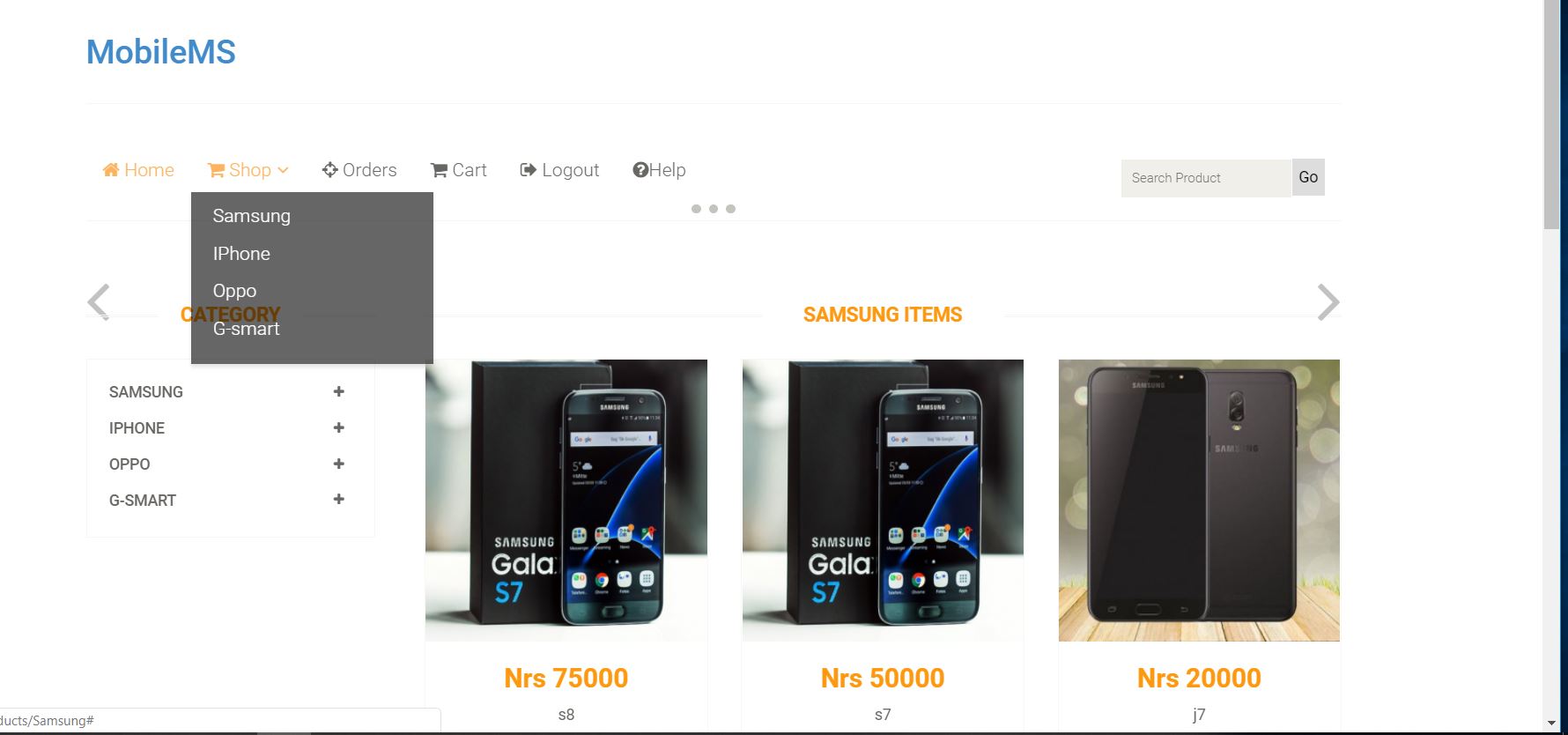
1. **HOMEPAGE**

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



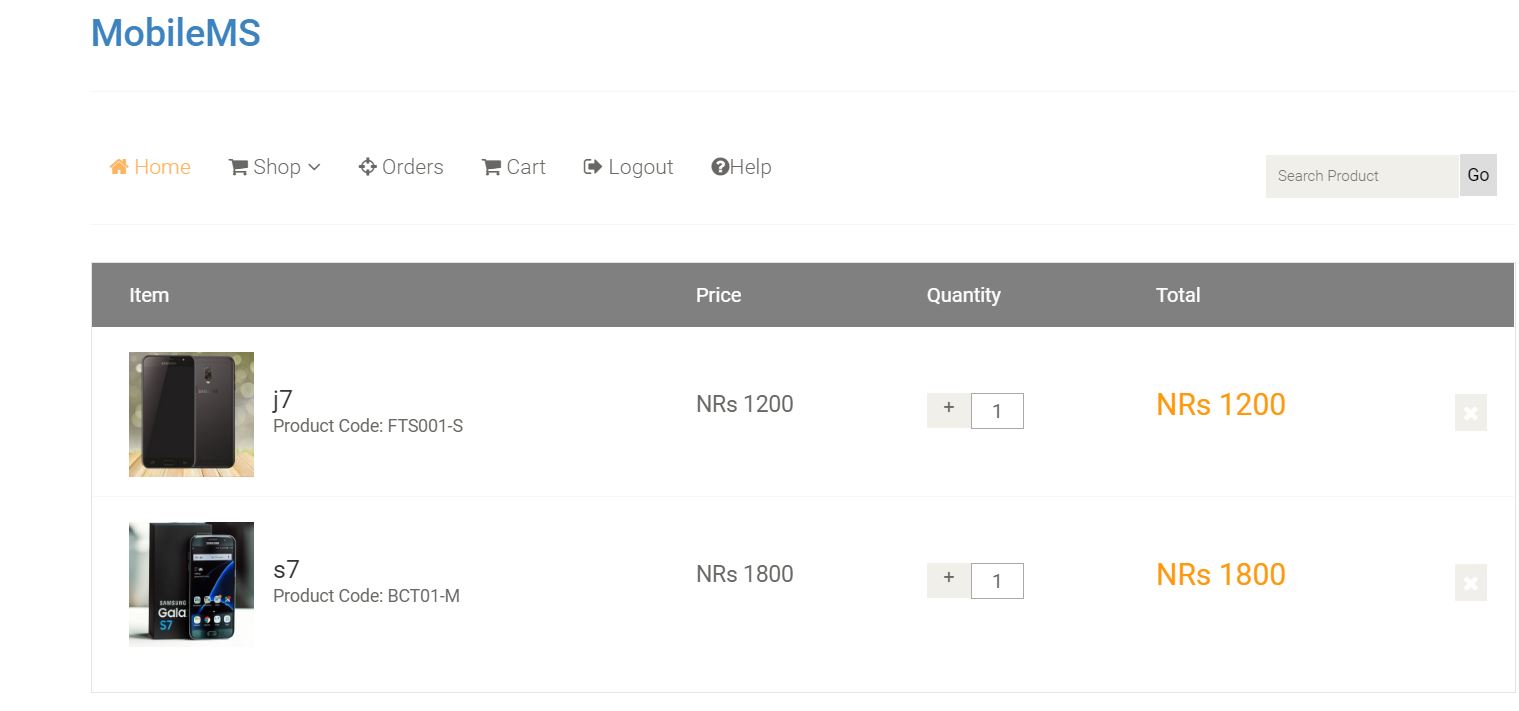
1. **SHOP**

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



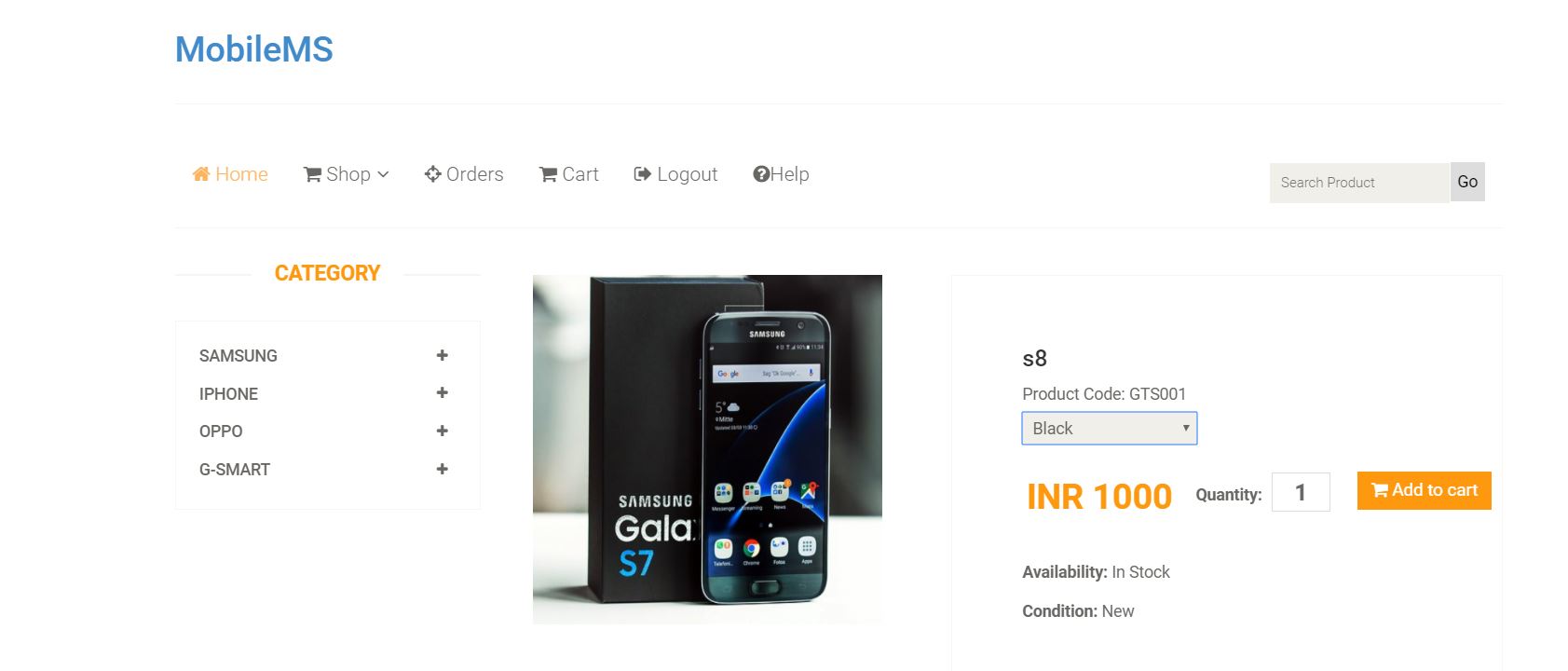
1. **CART**

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

****

1. **TO PURCHASE**

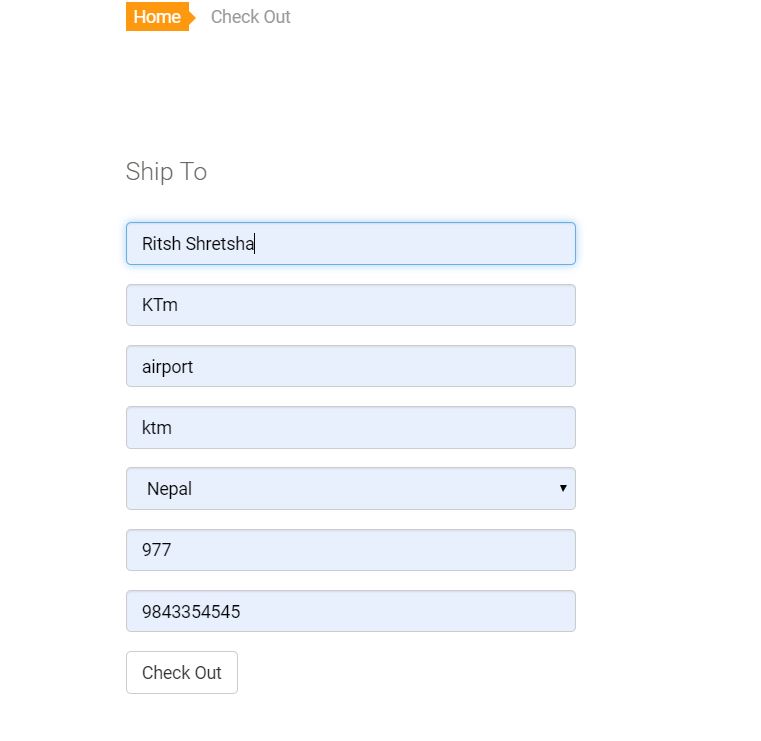
* Click on any item to purchase, where the price and the availability of the item will be displayed which is shown below.
* Item quantity can also be changed accordingly.
* Click on “Add to Cart” button, for further process.

****

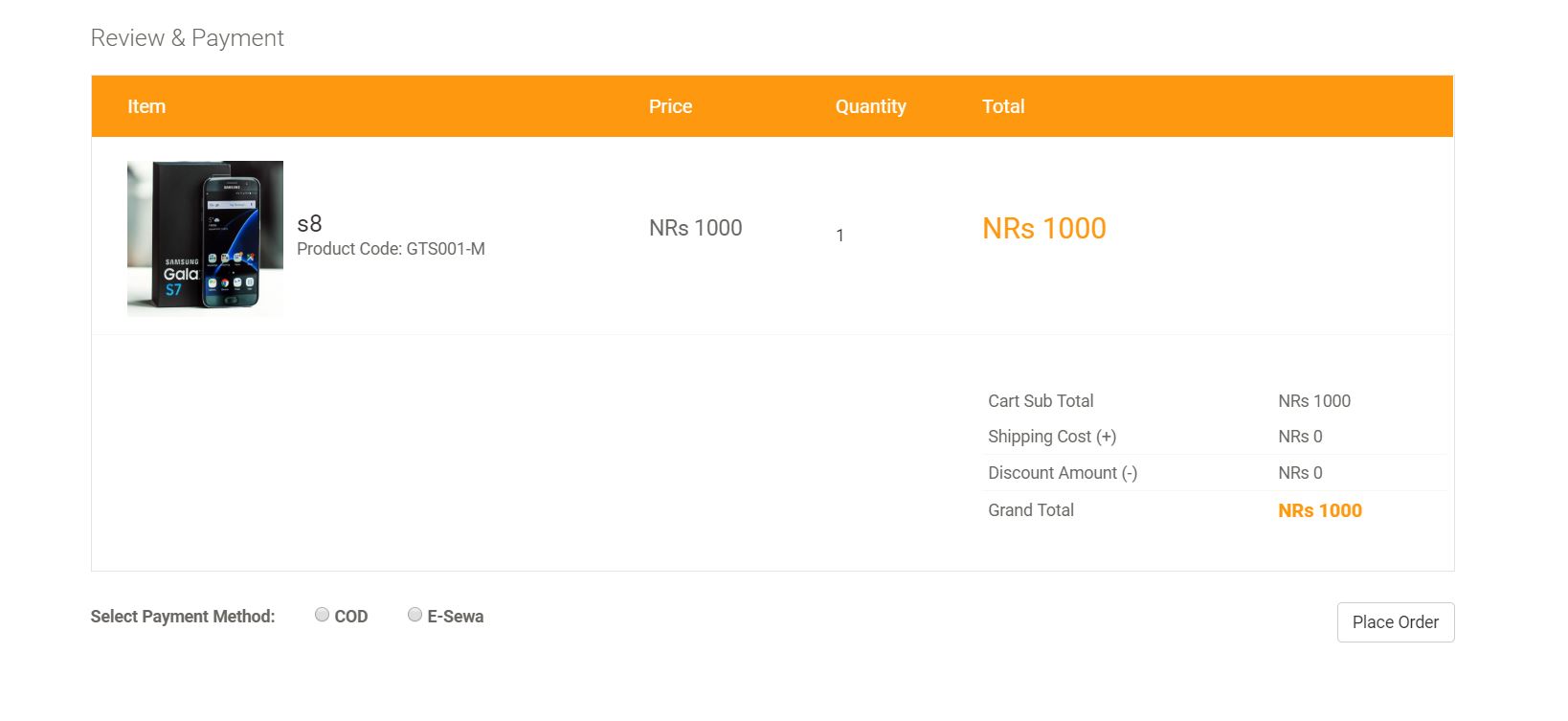
* The confirmation page will be displayed which is shown below.
* Click on “Check out” button, to confirm the purchase.

****

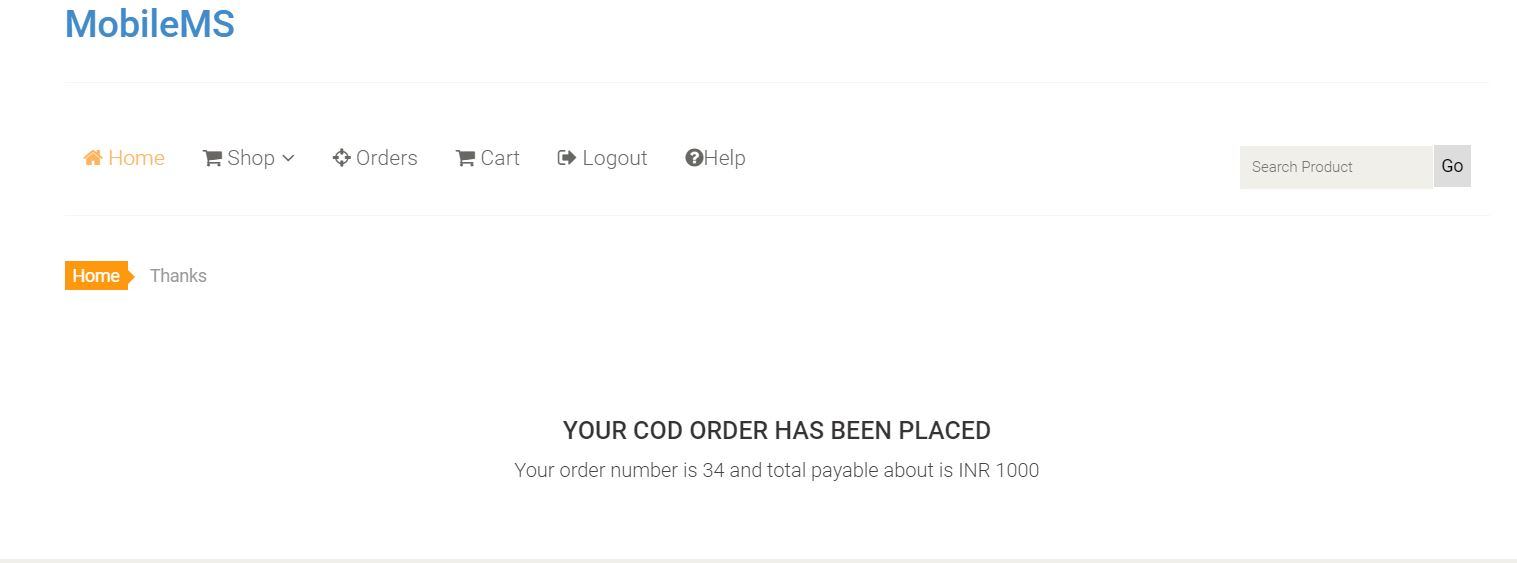
* Fill up all the required details, to get the ordered item, shown below.

****

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

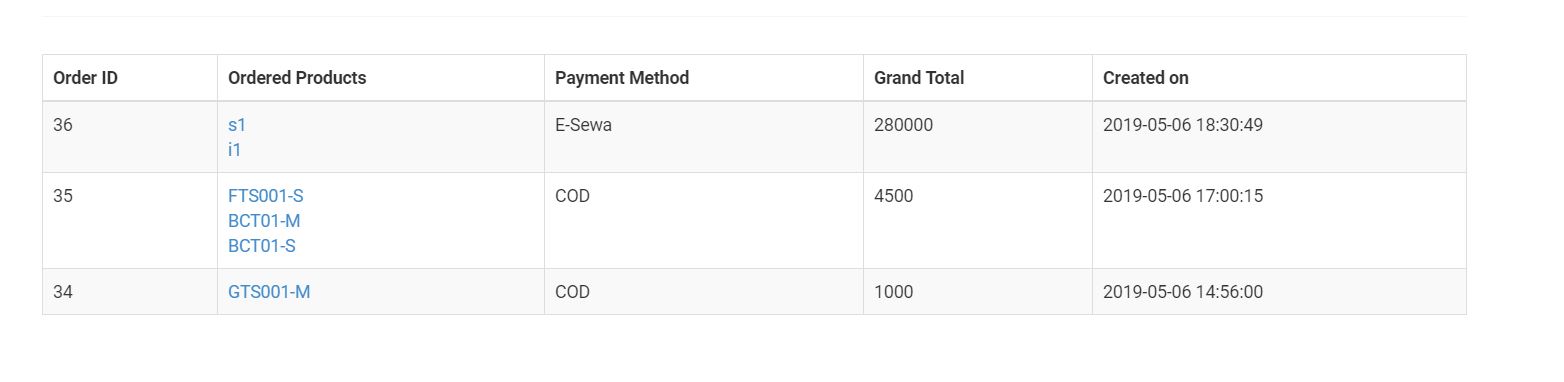
****

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

****

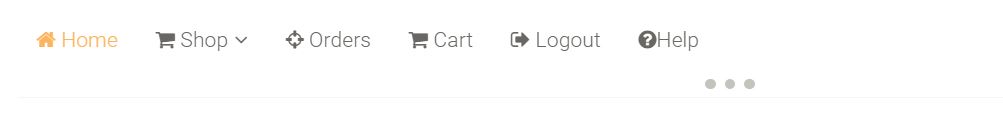
1. **ORDER**

* 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:***

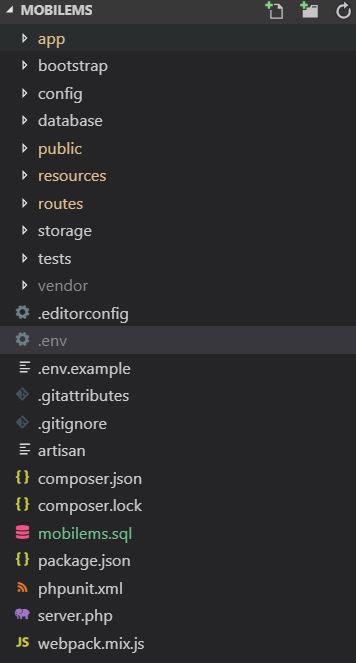


Figure 58: Laravel all files

***Database migrations:***

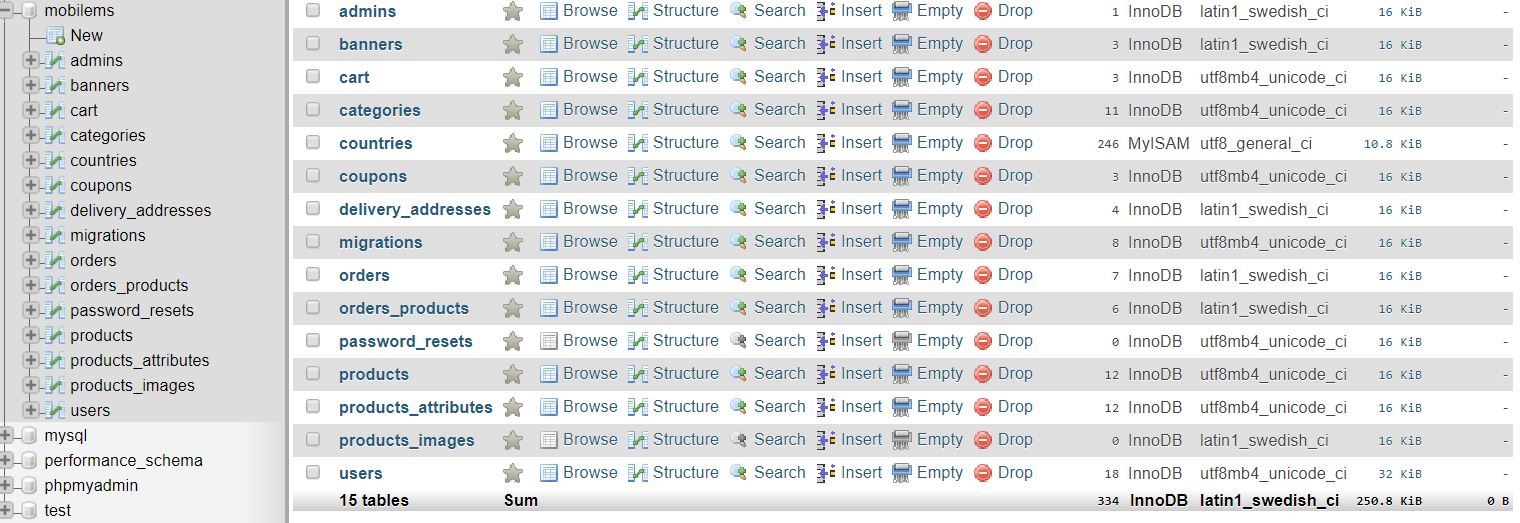


Figure 59: Database migrated in MYSQL

***Models on Laravel:***

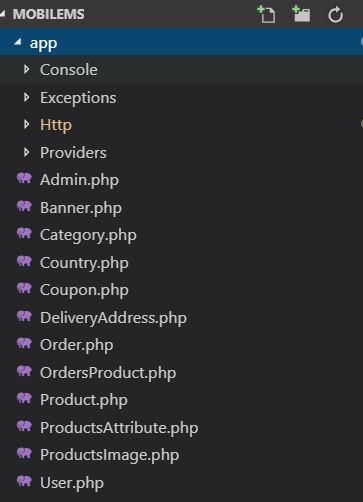
******

Figure 60: List of Models

***Controllers on Laravel:***

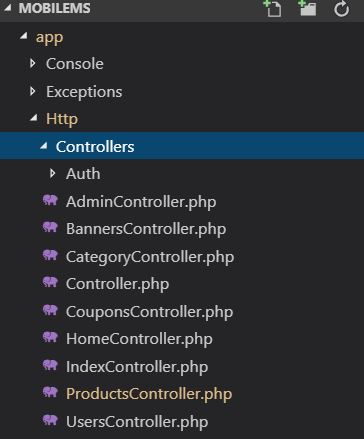
******

Figure 61: Controllers list

***Laravel views:***

******

Figure 62: Views list

***Routes:***

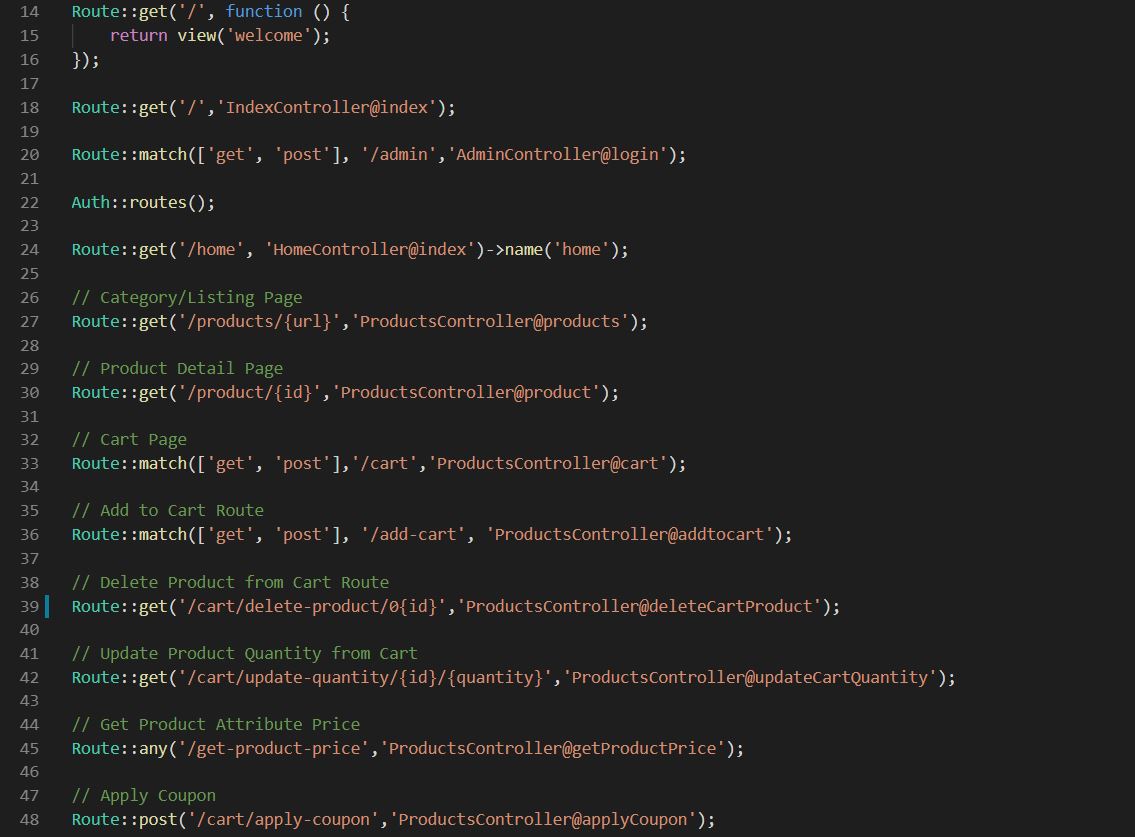
******

Figure 63: Routes

***UI Design***

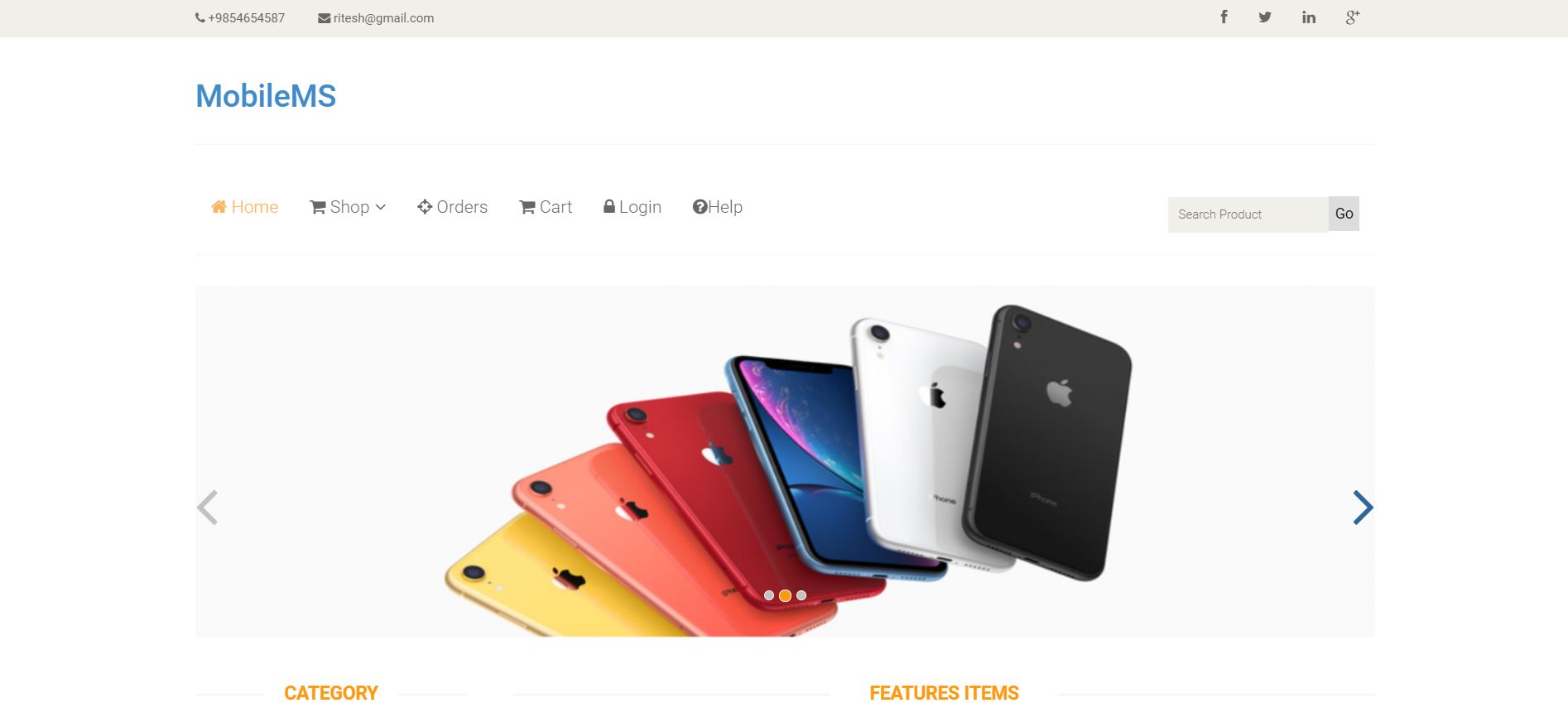
******

Figure 64: Home page

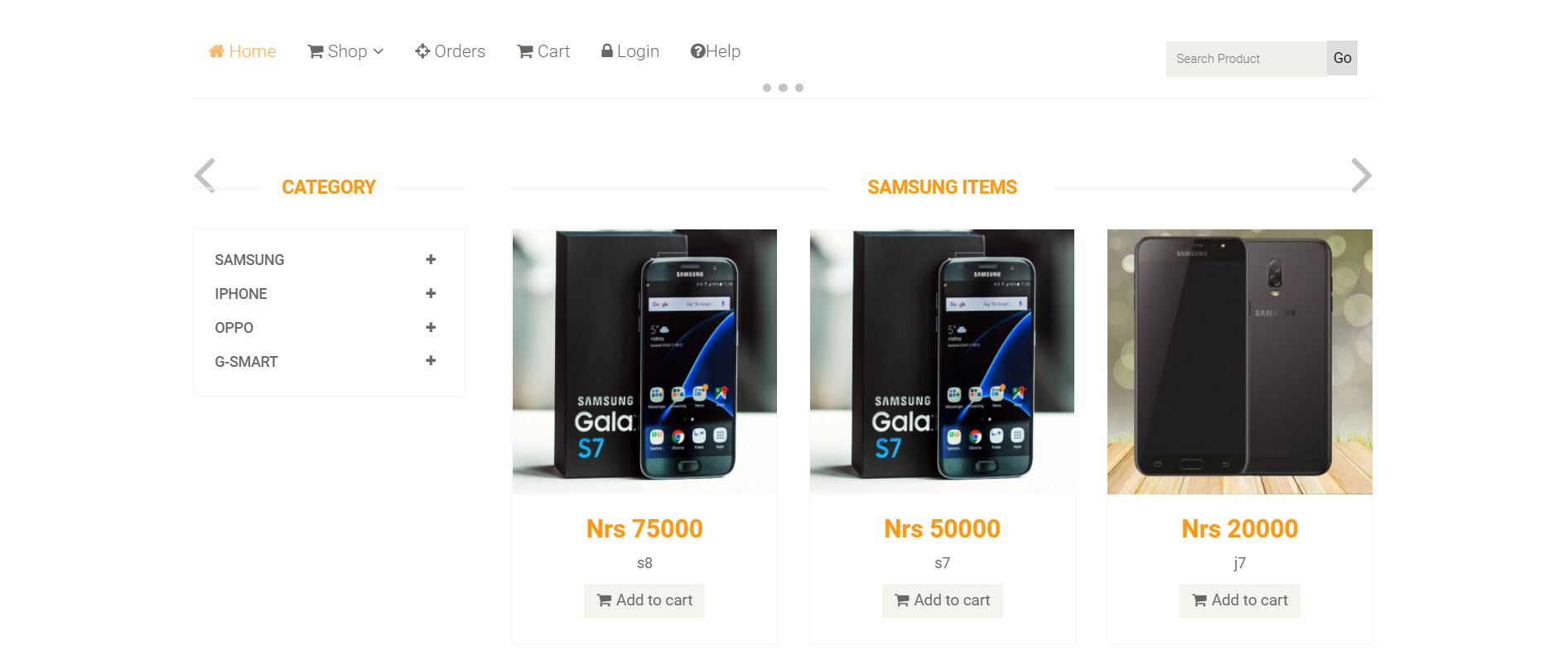
******

Figure 65:Gallery



Figure 66: Cart page

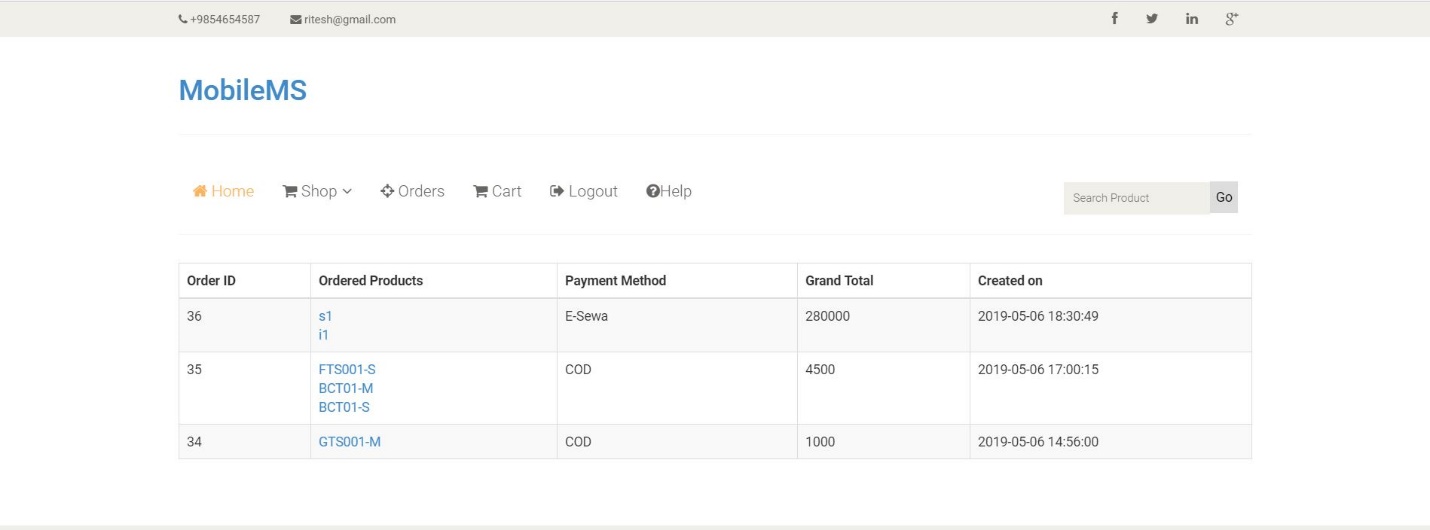


Figure 67:Order Page

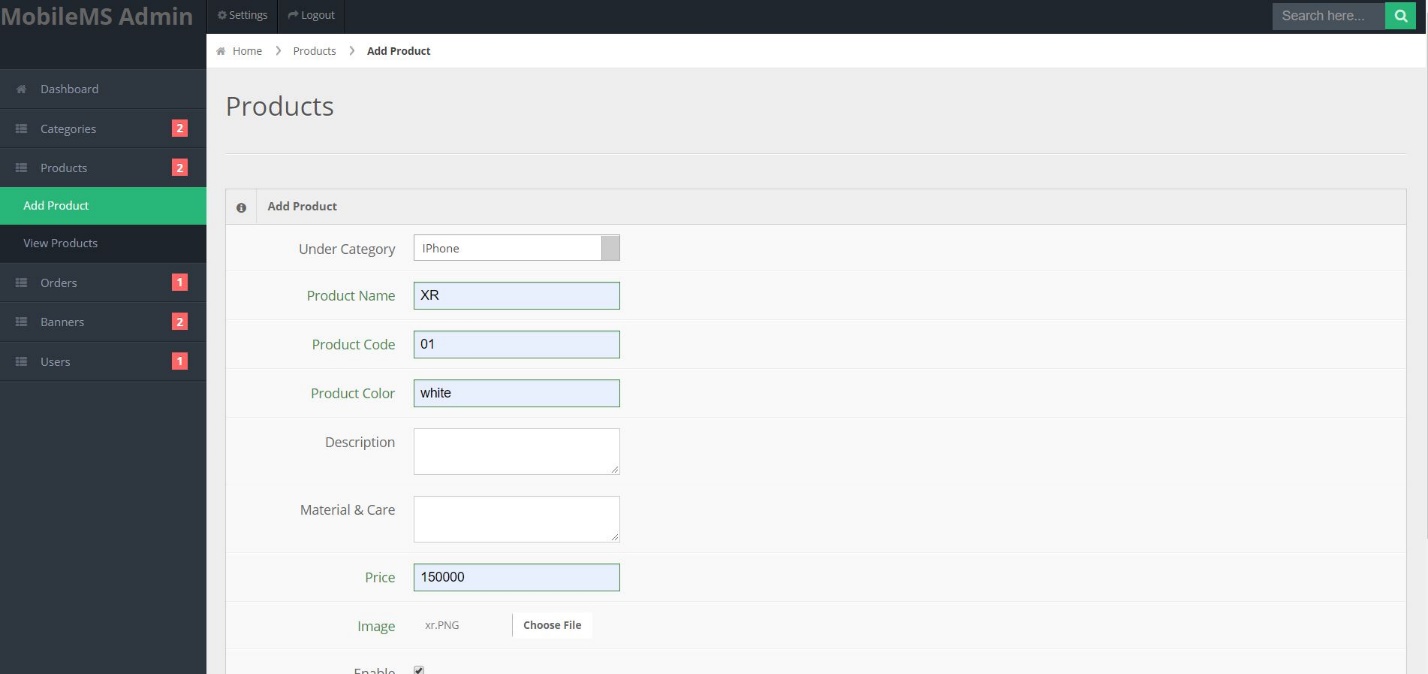


Figure 68:Add product(admin) interface

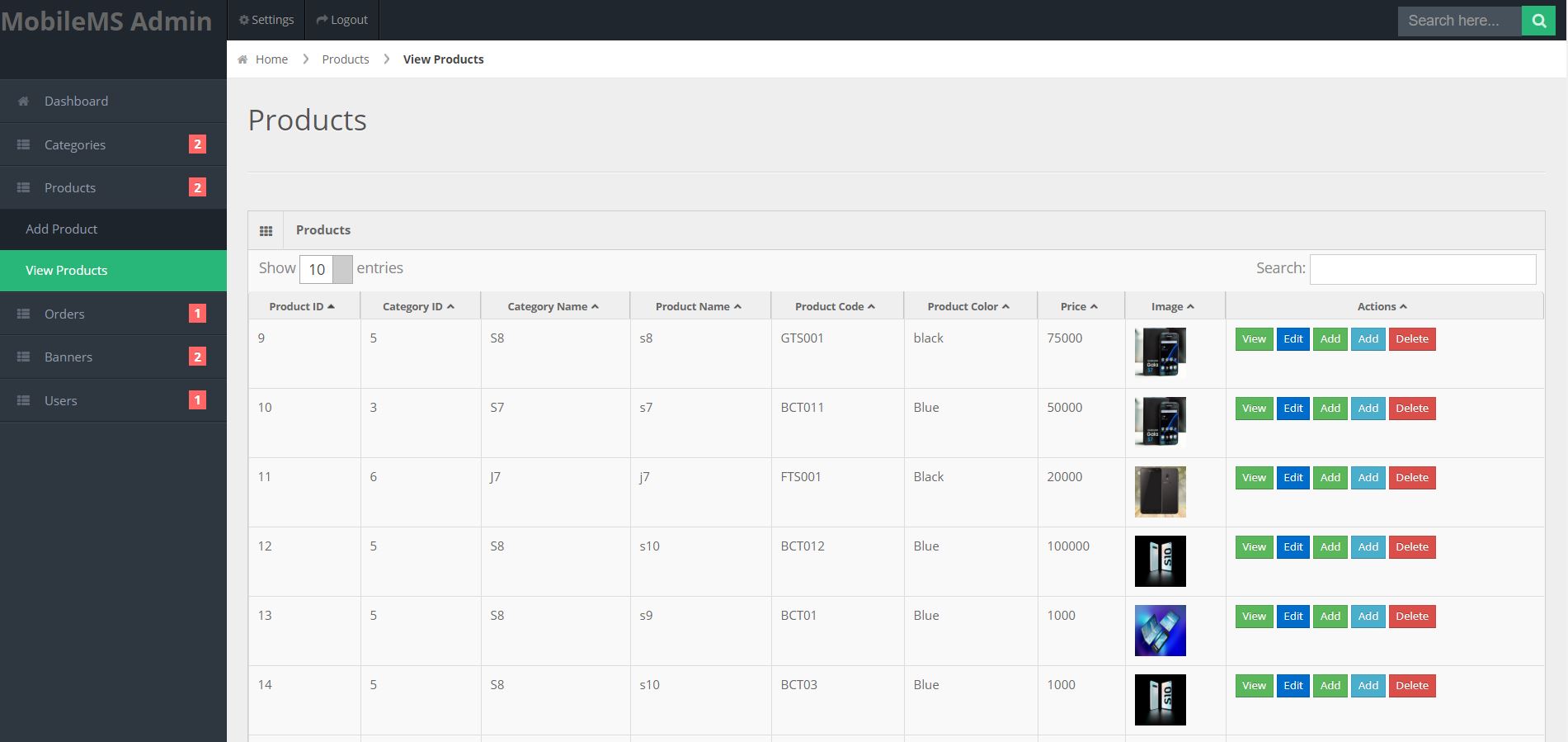


Figure 69:Added product interface

***Users Model:***

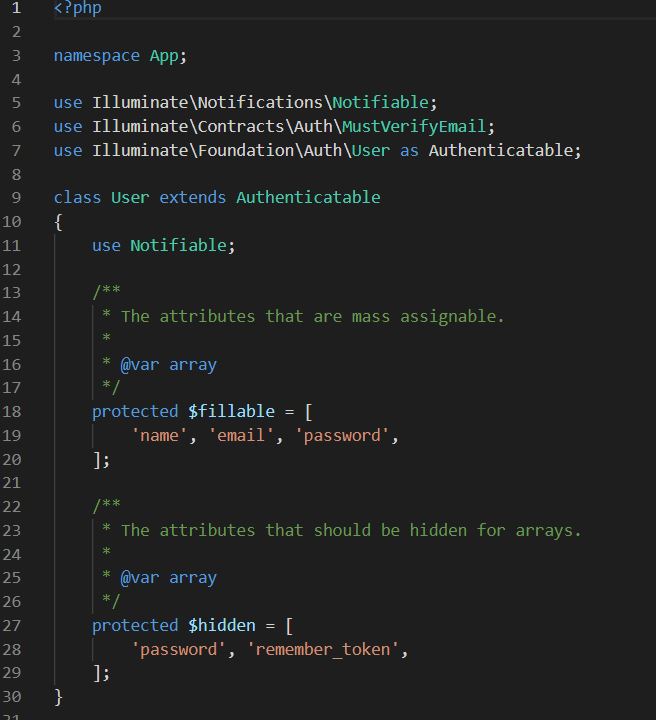
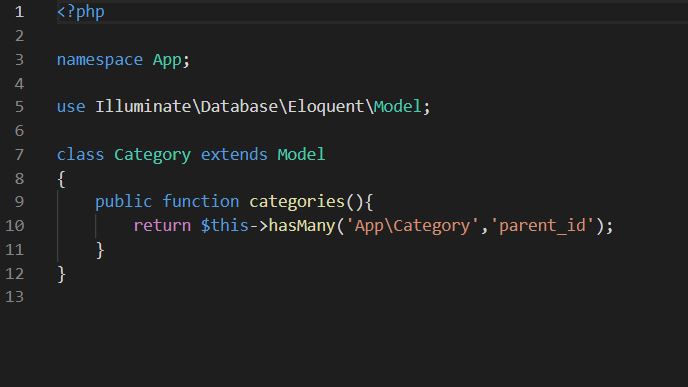
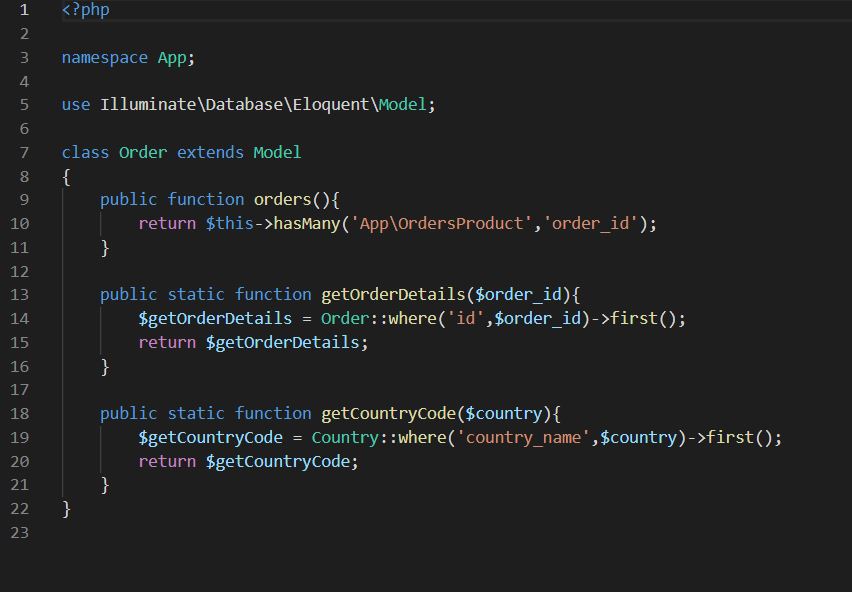
****

Figure 70: User Model Code

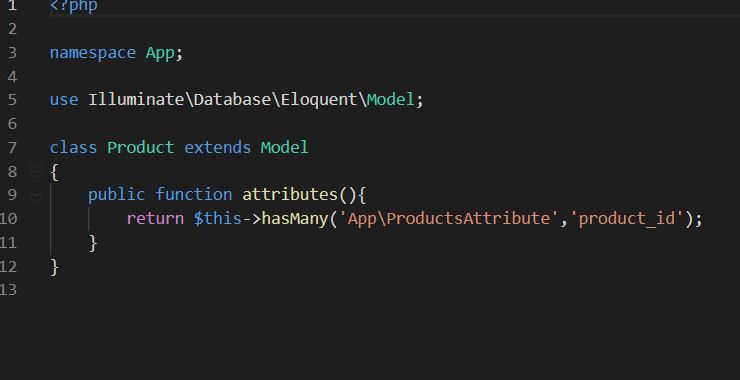
Categories



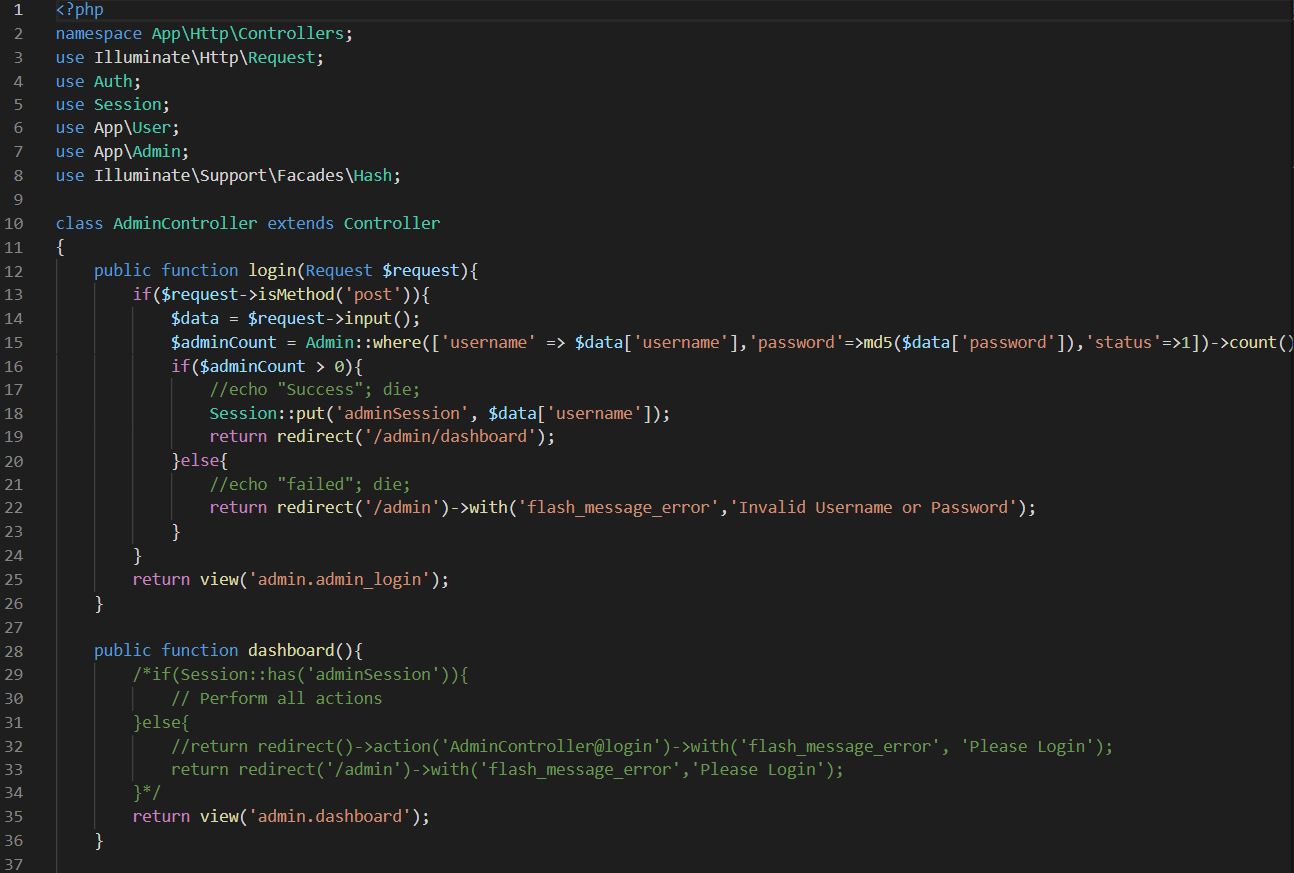
Order



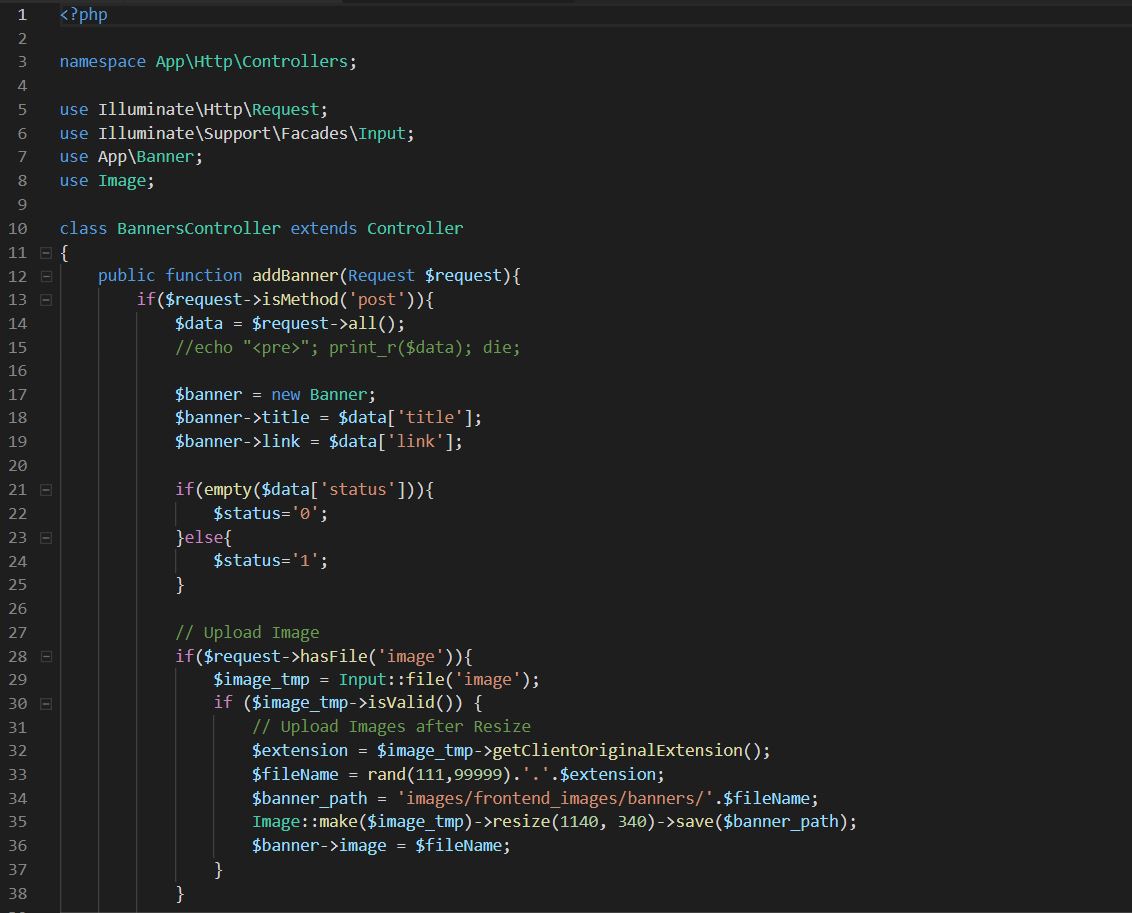
Attributes



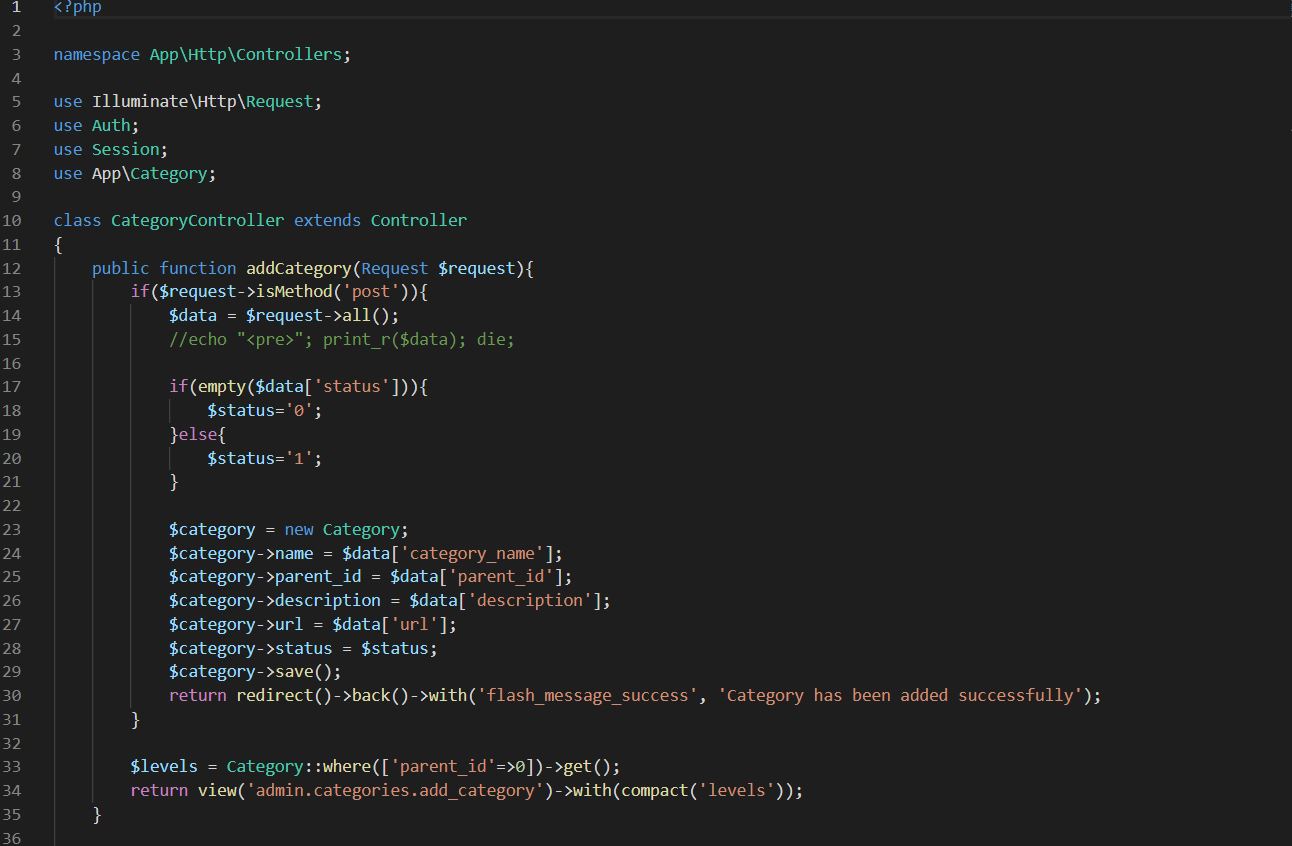
Admin Controller



Banner controller

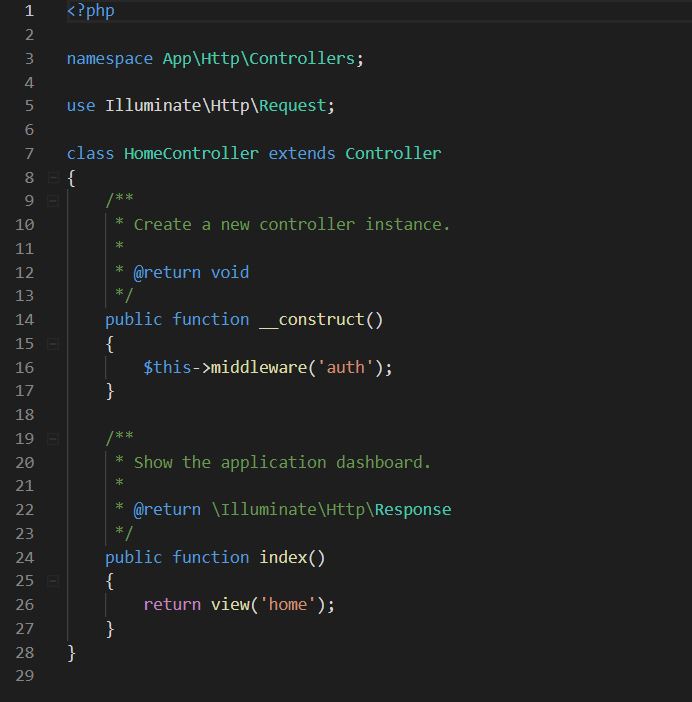


**Category Controller**

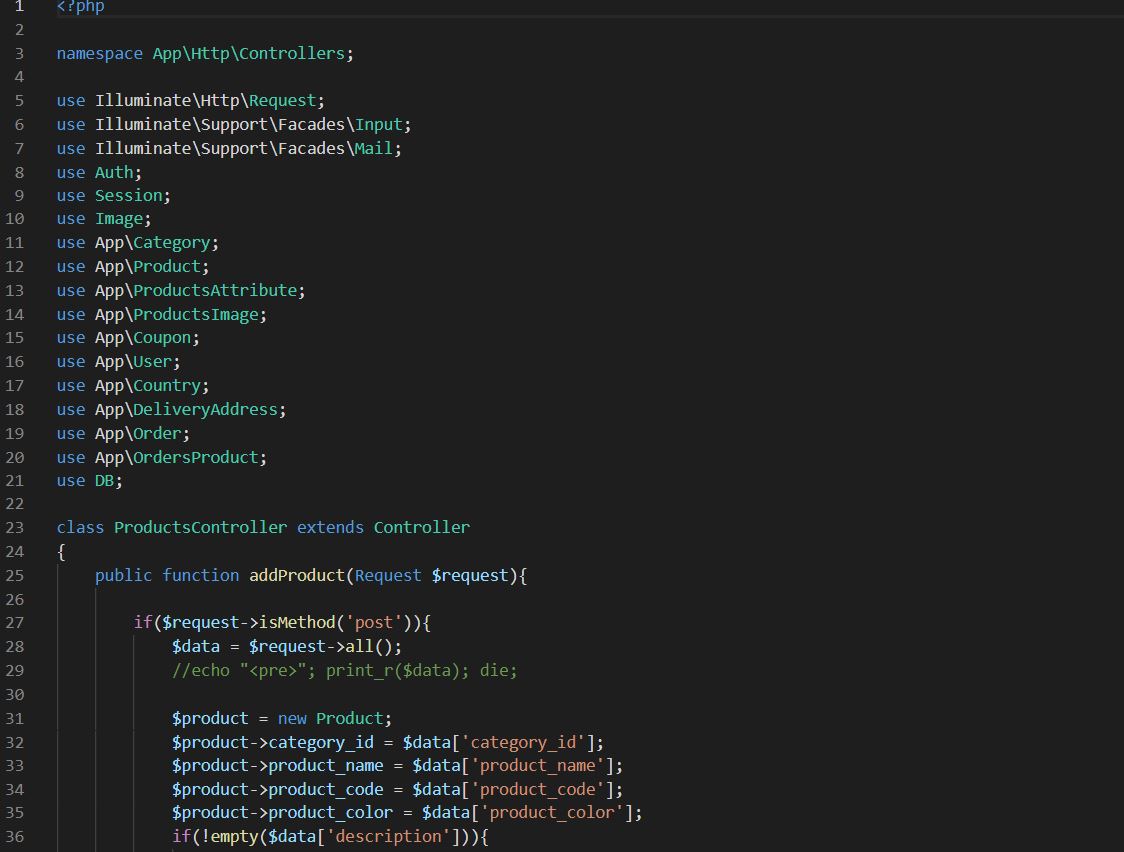


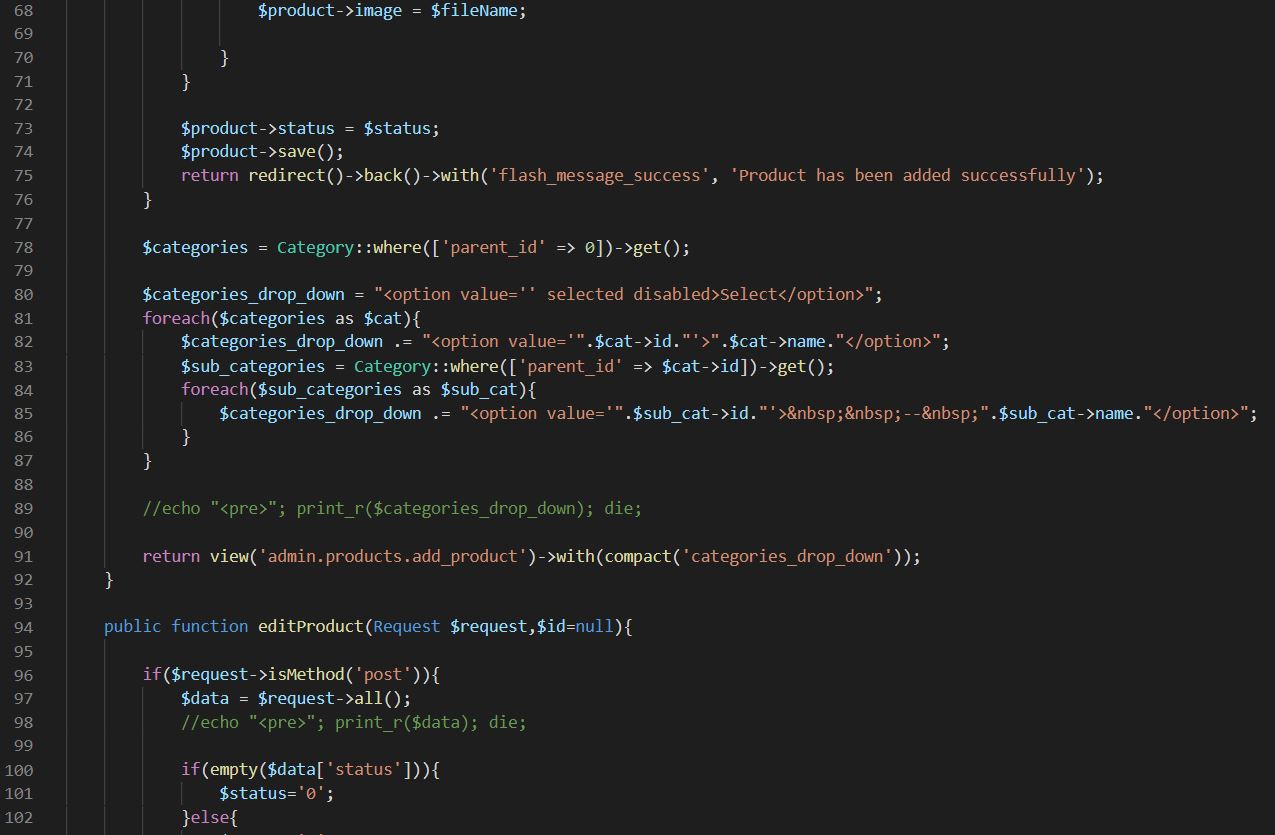
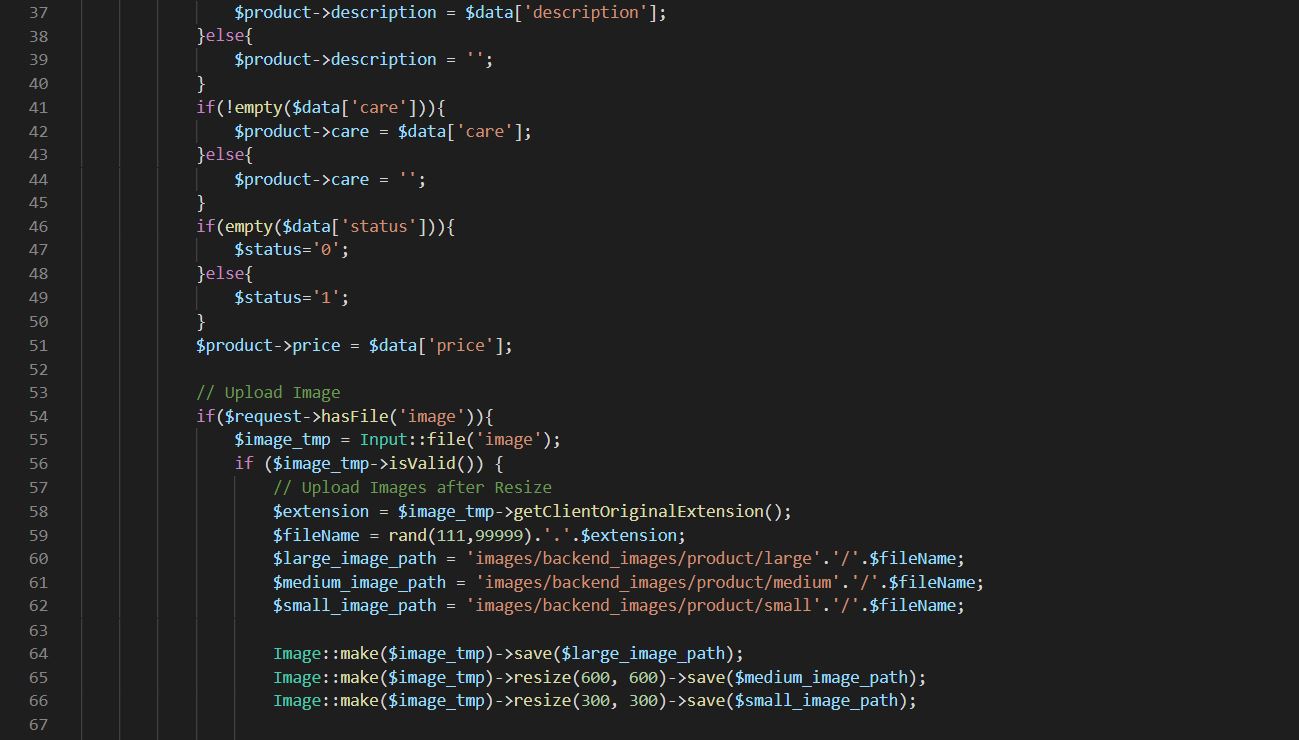


**Home Controller**

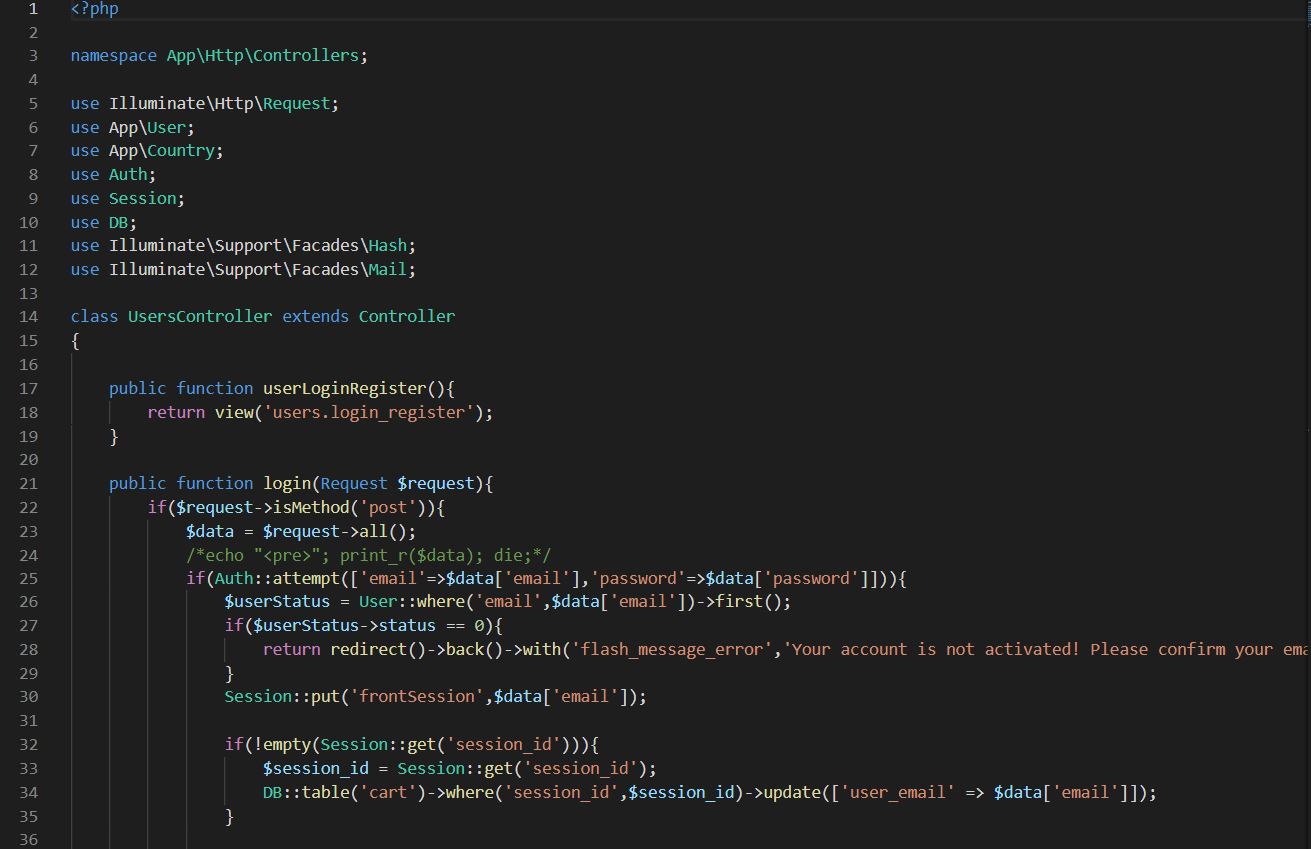


**Product Controller**

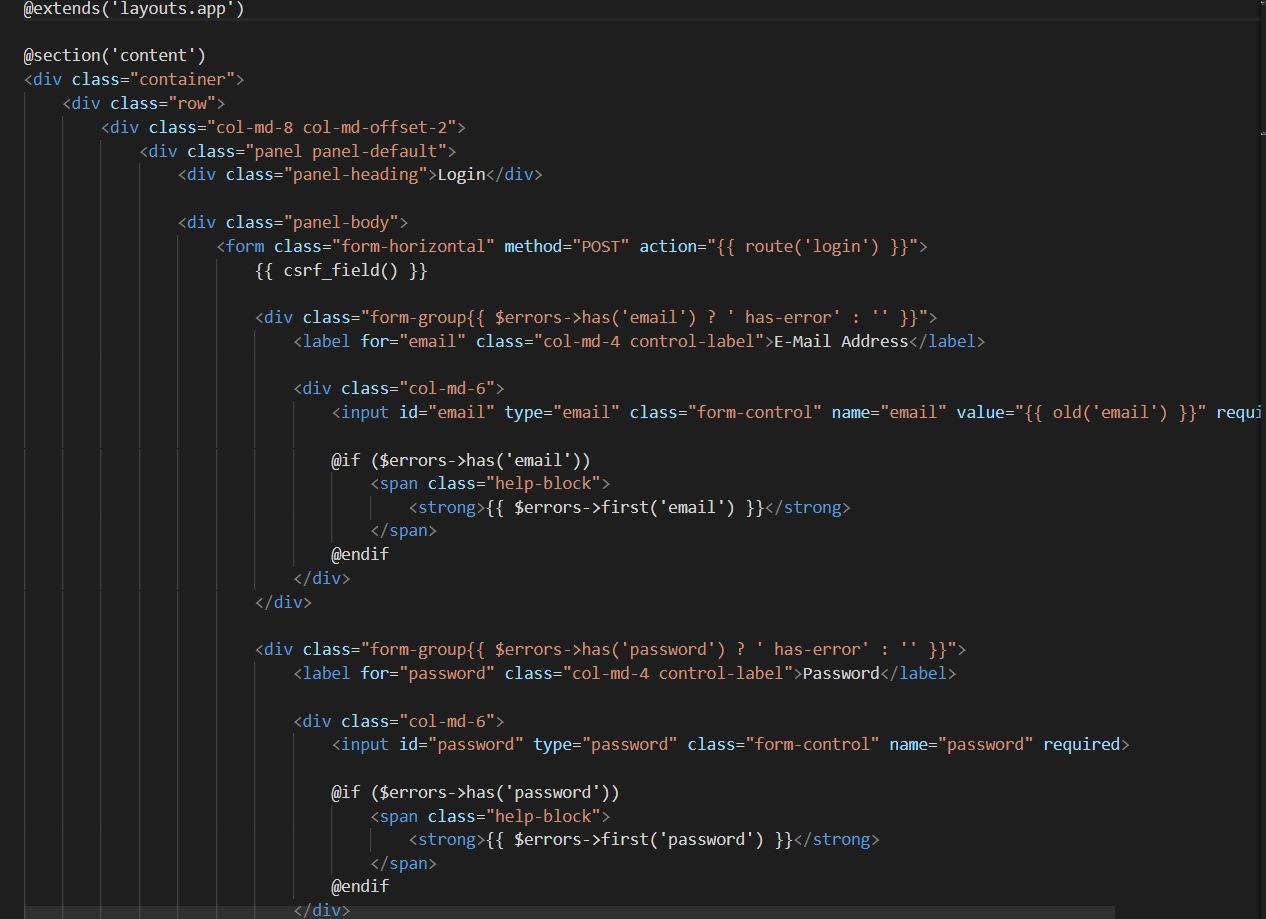
******

******

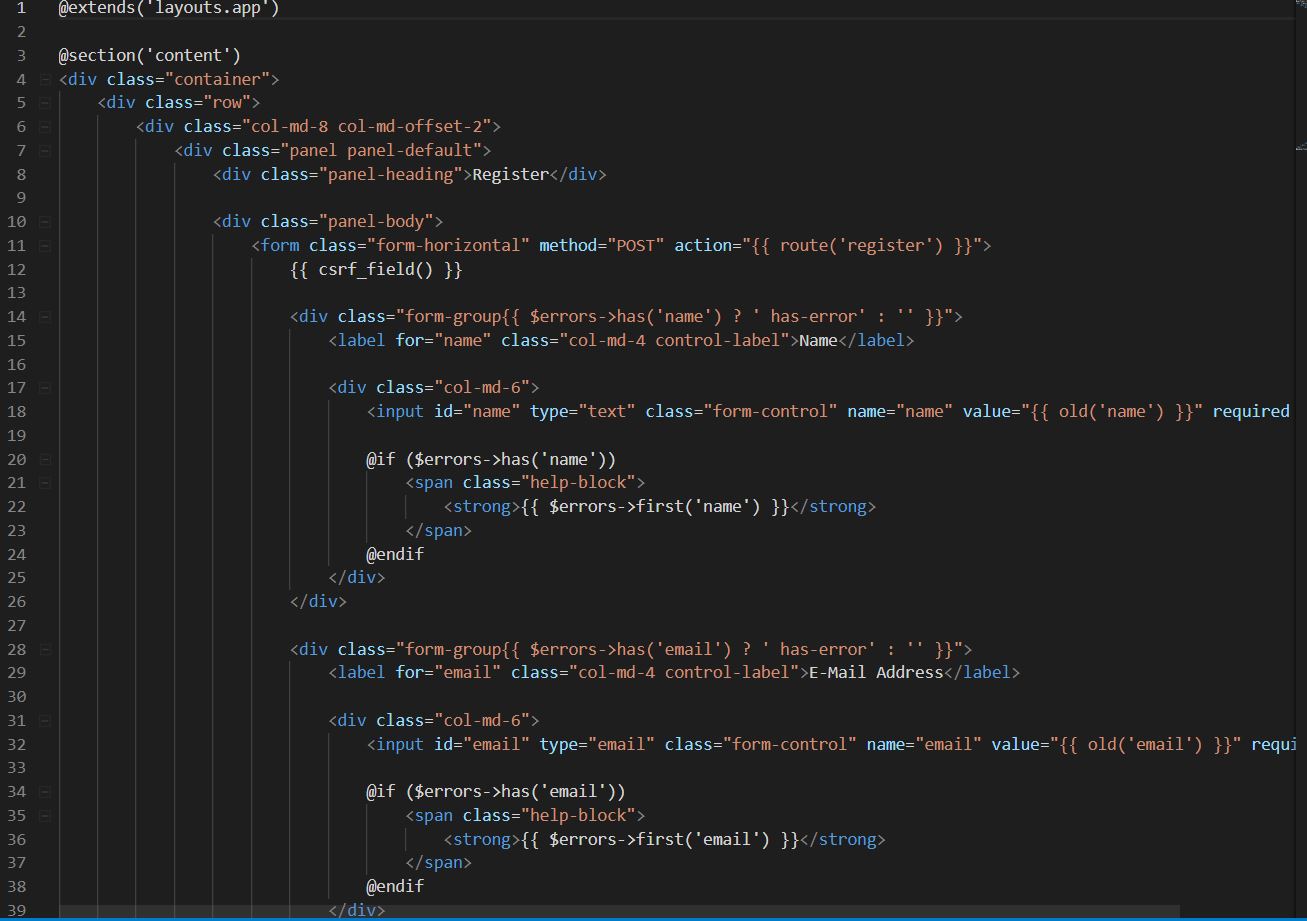
***User Controller***

******

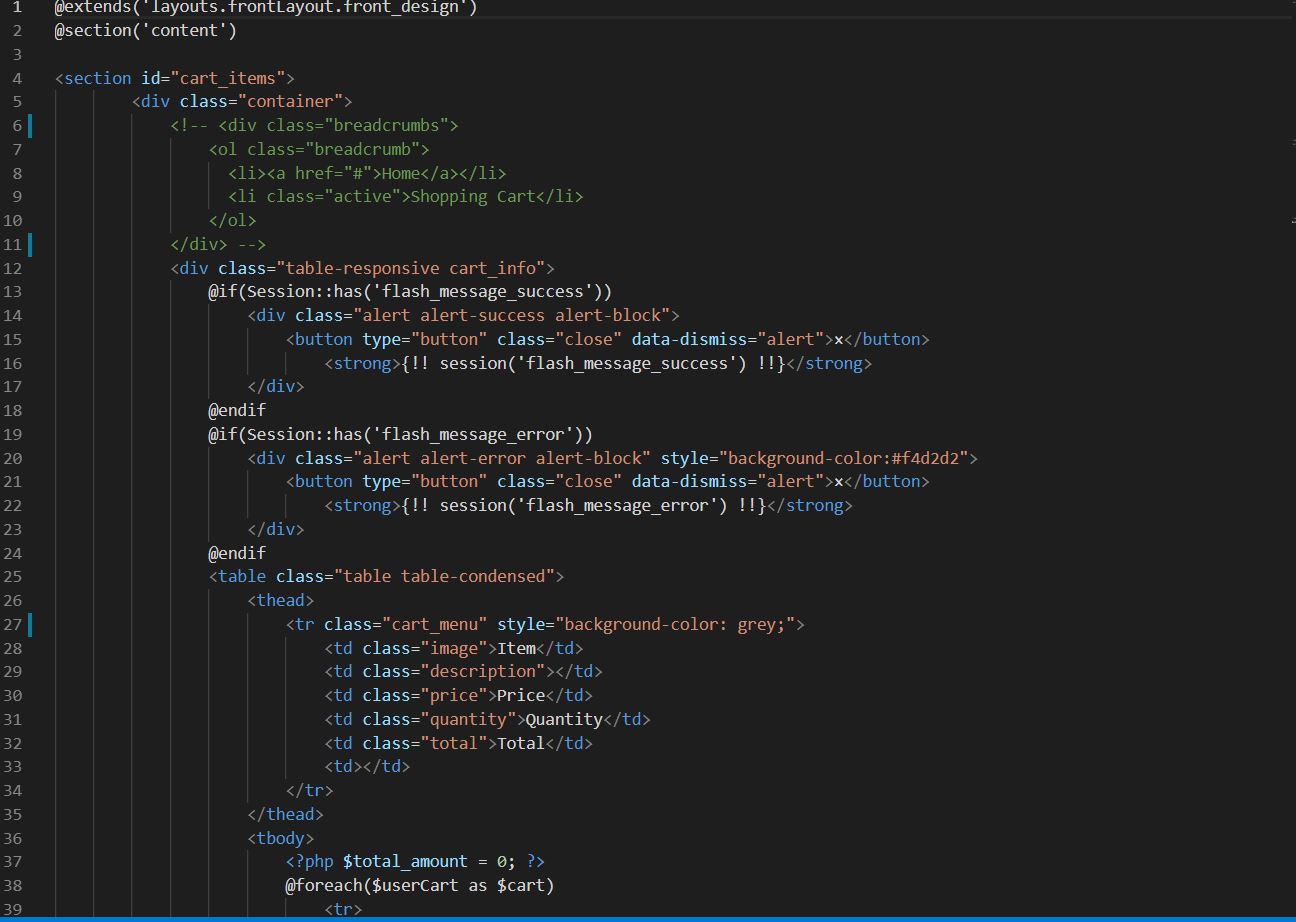
***Login Blade***

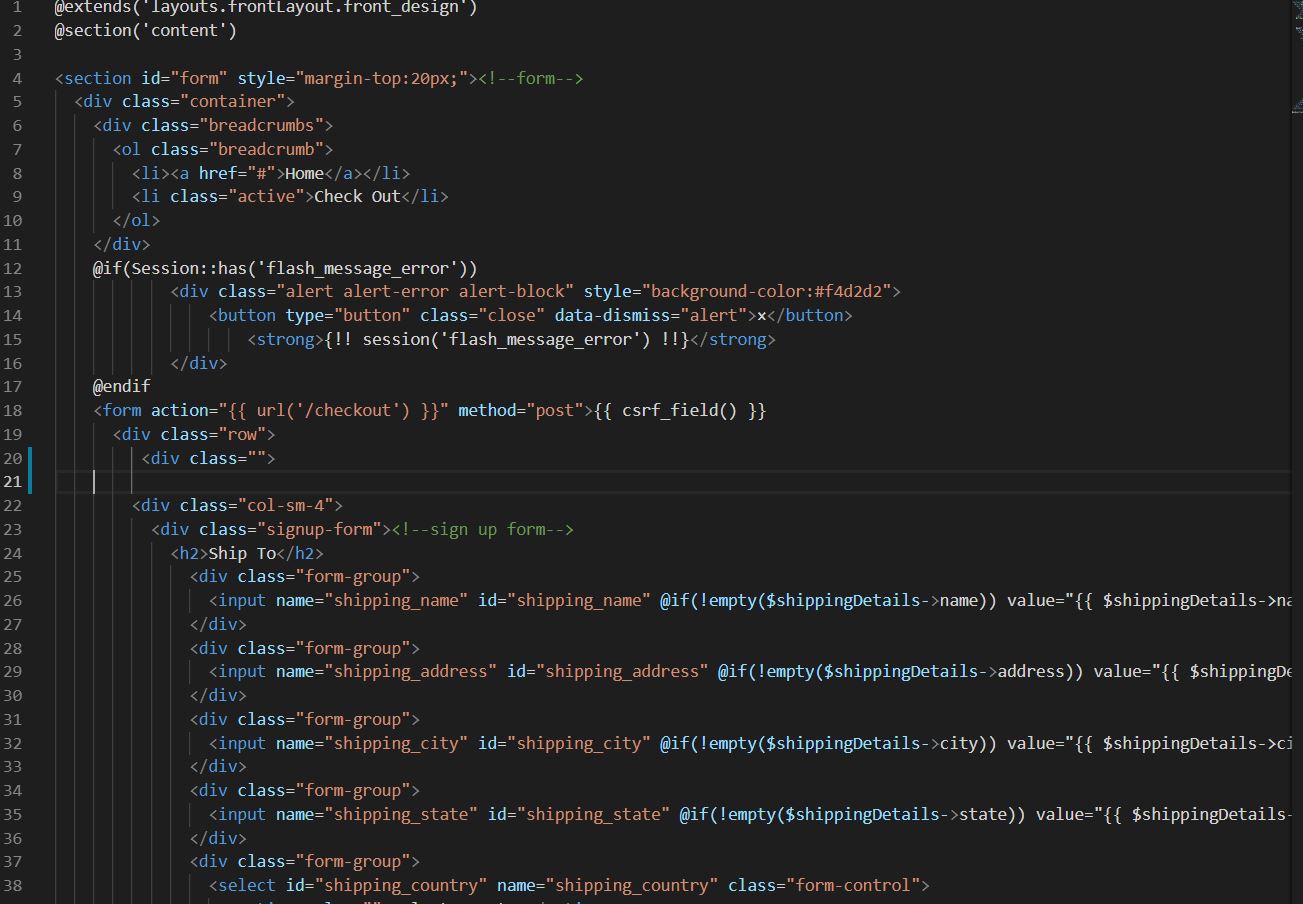
******

***Register Blade***

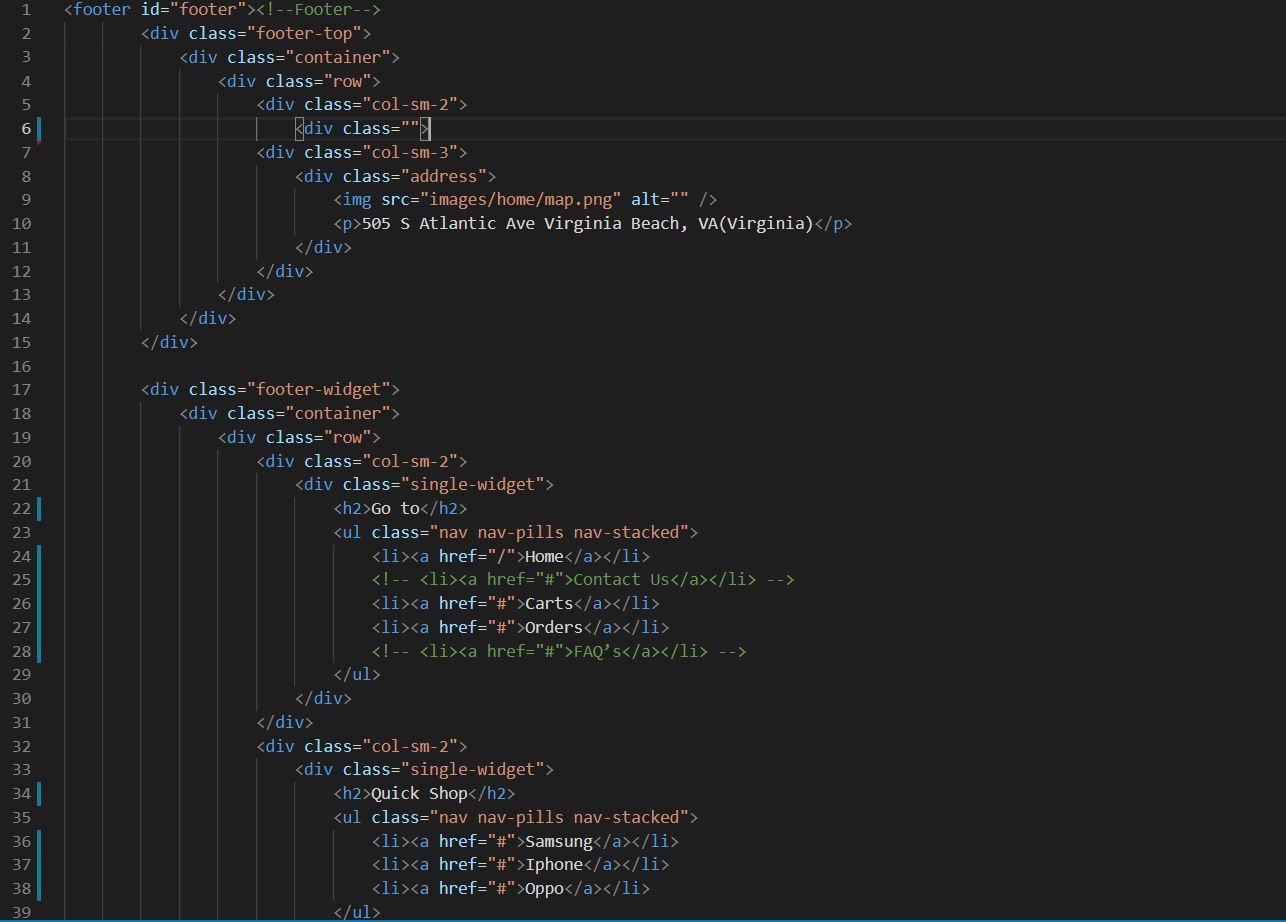
******

***Cart Blade***

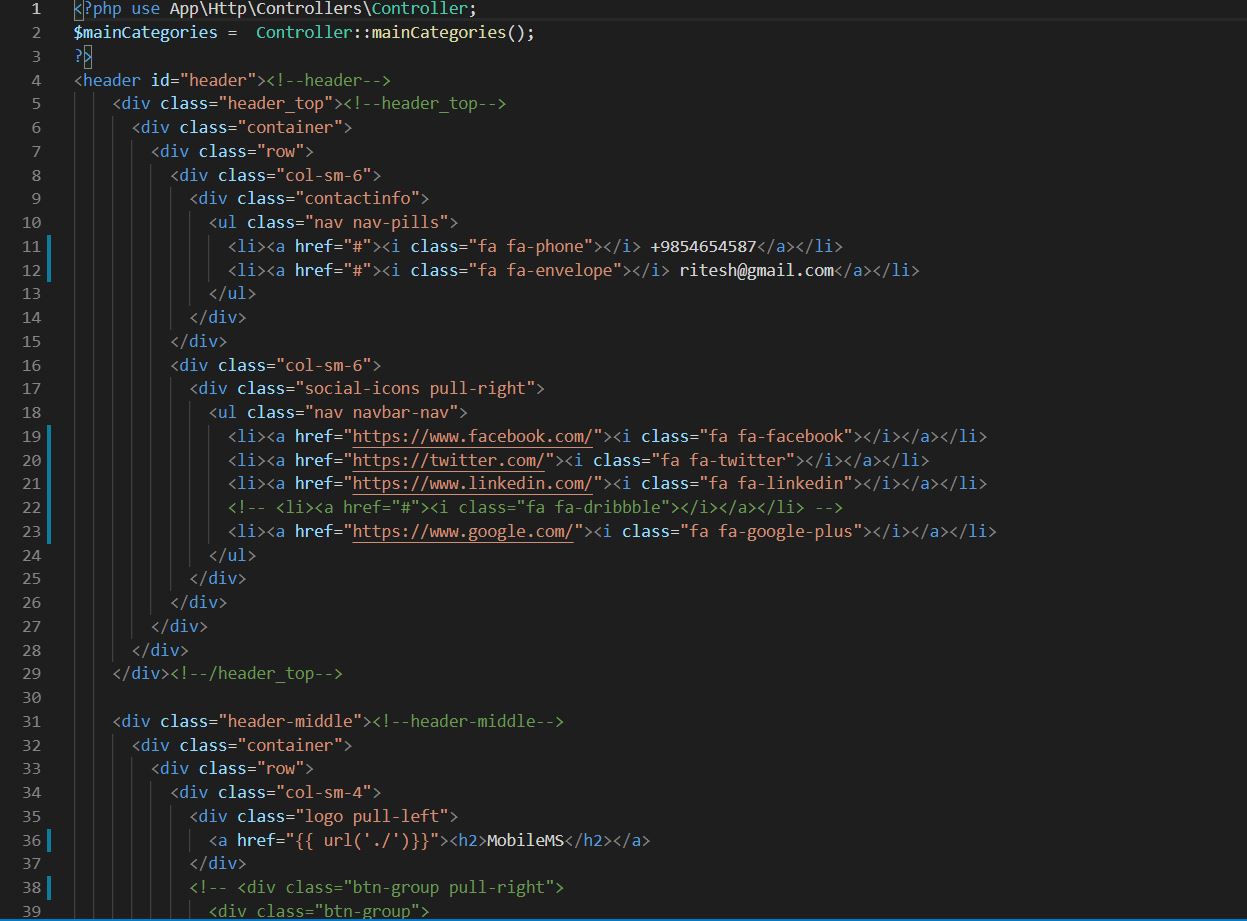
******

***Checkout Blade***

***Footer Blade***

******

***Header Blade***

******

**The End**