# Acknowledgement

Things do not happen overnight, I was able to complete the project because of the supports, guidance and efforts. So, I want to express my gratitude to all the supporters who supported me throughout the project and every effort has turned fruitfully into the completion of this project. It is possible only due to the supervision and supporters. So I’m extremely thankful to them.

Special gratitude goes to my respective teacher Mr. Kiran Rana for providing me the opportunities and giving his supports and guidance which made me possible in the completion of the project. I’m grateful to him for helping for the software analysis, design, implementation, testing and deployment.

I’m also extremely thankful and grateful to get the supports, efforts and guidance from all the staffs of Softwarica College of IT and E-Commerce that made me possible to complete the project successfully.

Last but not least, I am heartily thankful to the Lamb of God, my family and friends who gave the full supports and understand me unconditionally.

# Abstract

The product available in the stock is difficult task to manage for the business activities. As the stock management plays the vital role to run the business activities effectively and smoothly. The digital world has influence the most of the area. Likewise, with the help of technology it makes easier to manage the stock in the blink of eye.

To manage the stock, there people can visit the website and can insert, edit and delete the stock that are available in their warehouse. It helps to save the time as well as the cost. It makes easier to manage the stock.

This project helps to keep the record of stocks as it is the web based application. Not only for the business activities but also for the shops people can manage their stocks. This project includes all the requirements like use cases and its description, system architecture, model use, design, implementation as well as the testing. To do the web based application I use the PHP CodeIgniter to design the software.

Contents

[Acknowledgement 1](#_Toc7040789)

[Abstract 1](#_Toc7040790)

[List of figure 5](#_Toc7040791)

[Chapter 1 6](#_Toc7040792)

[Introduction 6](#_Toc7040793)

[Background of the system 7](#_Toc7040794)

[Justification 7](#_Toc7040795)

[Aims 7](#_Toc7040796)

[Objectives 7](#_Toc7040797)

[Overview of project design 8](#_Toc7040798)

[Chapter 2 10](#_Toc7040799)

[Analysis 10](#_Toc7040800)

[Introduction to Analysis 11](#_Toc7040801)

[Need for Analysis 11](#_Toc7040802)

[Object Oriented Analysis 11](#_Toc7040803)

[Pitfalls of my project 11](#_Toc7040804)

[Merits of my project 12](#_Toc7040805)

[Requirement 12](#_Toc7040806)

[Functional requirements 12](#_Toc7040807)

[Non-functional 14](#_Toc7040808)

[Requirement prioritization 15](#_Toc7040809)

[NLA 16](#_Toc7040810)

[Initial class diagram 17](#_Toc7040811)

[Use case diagram 18](#_Toc7040812)

[Justification 18](#_Toc7040813)

[Architecture 19](#_Toc7040814)

[Chapter 3 21](#_Toc7040815)

[Design 21](#_Toc7040816)

[Dynamic Modeling 22](#_Toc7040817)

[Structural Modeling 29](#_Toc7040818)

[Database Modeling 32](#_Toc7040819)

[Chapter 4 37](#_Toc7040820)

[Implementation 37](#_Toc7040821)

[Programming language used 38](#_Toc7040822)

[Chapter 5 39](#_Toc7040823)

[Testing 39](#_Toc7040824)

[Black box testing 40](#_Toc7040825)

[Unit testing 49](#_Toc7040826)

[Chapter 6 51](#_Toc7040827)

[Other project issues 51](#_Toc7040828)

[Risk management 52](#_Toc7040829)

[Configuration management 54](#_Toc7040830)

[Scheduling/Gantt chart 56](#_Toc7040831)

[Future work 58](#_Toc7040832)

[User manual 59](#_Toc7040833)

[Limitation 67](#_Toc7040834)

[Chapter 7 68](#_Toc7040835)

[Conclusion 68](#_Toc7040836)

[Chapter 8 70](#_Toc7040837)

[References and bibliography 70](#_Toc7040838)

[Chapter 9 72](#_Toc7040839)

[Appendix 72](#_Toc7040840)

[Code Snippets 73](#_Toc7040841)

[List of tables 76](#_Toc7040842)

[Database 76](#_Toc7040843)

[Controllers\Auth.php 77](#_Toc7040844)

[Controllers\Dashboard.php 78](#_Toc7040845)

[Controllers\Orders.php 79](#_Toc7040846)

[Controllers\Products.php 88](#_Toc7040847)

[Models\Model\_auth.php 93](#_Toc7040848)

[Models\Model\_orders.php 94](#_Toc7040849)

[Models\Model\_products.php 99](#_Toc7040850)

[Views\dashboard.php 101](#_Toc7040851)

[Views\login.php 102](#_Toc7040852)

[Views\orders\create.php 104](#_Toc7040853)

[Views\orders\edit.php 115](#_Toc7040854)

[Views\orders\index.php 127](#_Toc7040855)

[Views\templates\footer.php 131](#_Toc7040856)

[Views\templates\header.php 132](#_Toc7040857)

[Views\templates\header\_menu.php 134](#_Toc7040858)

[Views\templates\side\_menubar.php 135](#_Toc7040859)

# List of figure

[Figure 1: view of the system 8](#_Toc7040860)

[Figure 2: adding 9](#_Toc7040861)

[Figure 3: view of the available product 9](#_Toc7040862)

[Figure 4: bill of the product 10](#_Toc7040863)

[Figure 5: initial class diagram 17](#_Toc7040864)

[Figure 6: use case diagram of stock management system 18](#_Toc7040865)

[Figure 7: three tier architecture 20](#_Toc7040866)

[Figure 8: activity diagram of login 23](#_Toc7040867)

[Figure 9: activity diagram of product 24](#_Toc7040868)

[Figure 10: activity diagram of selling the product 25](#_Toc7040869)

[Figure 11: sequence diagram of login system of stock management system 26](#_Toc7040870)

[Figure 12: sequence diagram after entering the main page of the system 27](#_Toc7040871)

[Figure 13: sequence diagram of adding the product 28](#_Toc7040872)

[Figure 14: final class diagram of stock management system 30](#_Toc7040873)

[Figure 15: context diagram 31](#_Toc7040874)

[Figure 16: Er diagram of the stock management system 32](#_Toc7040875)

[Figure 17: login form of stock management system 34](#_Toc7040876)

[Figure 18: dashboard of the stock management system 35](#_Toc7040877)

[Figure 19: category form of stock management system 35](#_Toc7040878)

[Figure 20: adding product form 36](#_Toc7040879)

[Figure 21: testing valid email and password 49](#_Toc7040880)

[Figure 22: testing passed of email and password 50](#_Toc7040881)

[Figure 23: risks management 53](#_Toc7040882)

[Figure 24: configuration management in github 54](#_Toc7040883)

[Figure 25: configuration management 55](#_Toc7040884)

[Figure 26: schedule of the project 56](#_Toc7040885)

[Figure 27: Gantt chart of the project 57](#_Toc7040886)

# Chapter 1

## Introduction

## Background of the system

The stock management plays the vital role to run the business activities and helps to meet the goal of the business. It provides the better platform to make the business success. So, the stock management is one of the most required software for the business as well for the shops. This software helps to exchange the product and keep the record safe. It helps to manage the stock within the small period of time.

### Justification

Stock management helps to improve the quality of the product as well as save the time and cost. This software helps to edit the stocks and create the bill of the product as well. It manages the product in the easiest way and helps to understand how it works as well as run in the smooth way.

## Aims

The main aim of this project is to develop the stock management system which helps to keep the record of the stocks safe and manage the stocks properly as well as to run it smoothly. It keeps the details of the product and manages it within the short period of time.

So, it provides safety of stock keeps record and manage it properly as well as saves the time and cost which going to help to maintain the better quality of the product.

## Objectives

To meet the desired aims, following are the objectives of the project

* First objective is to collect the requirement of the project.
* To perform the installation that is required during designing the software.
* To design the database that makes it easier to build the software.
* To perform the planning so that project complete in the correct time period.
* To perform the testing so that the confirmation of the software.
* To perform the release that gives the feedback and demand of the project.

## Overview of project design

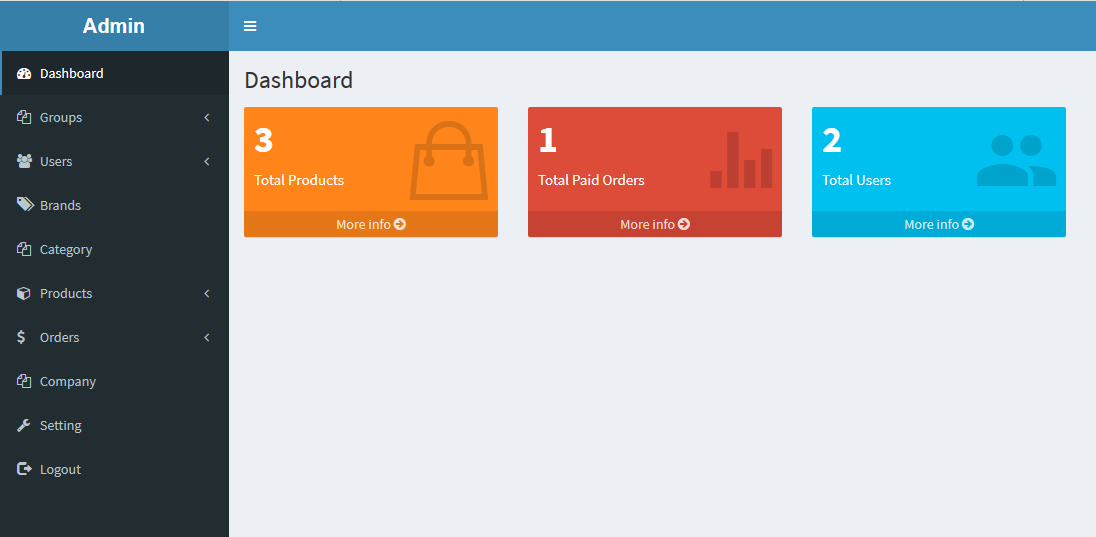


Figure : view of the system

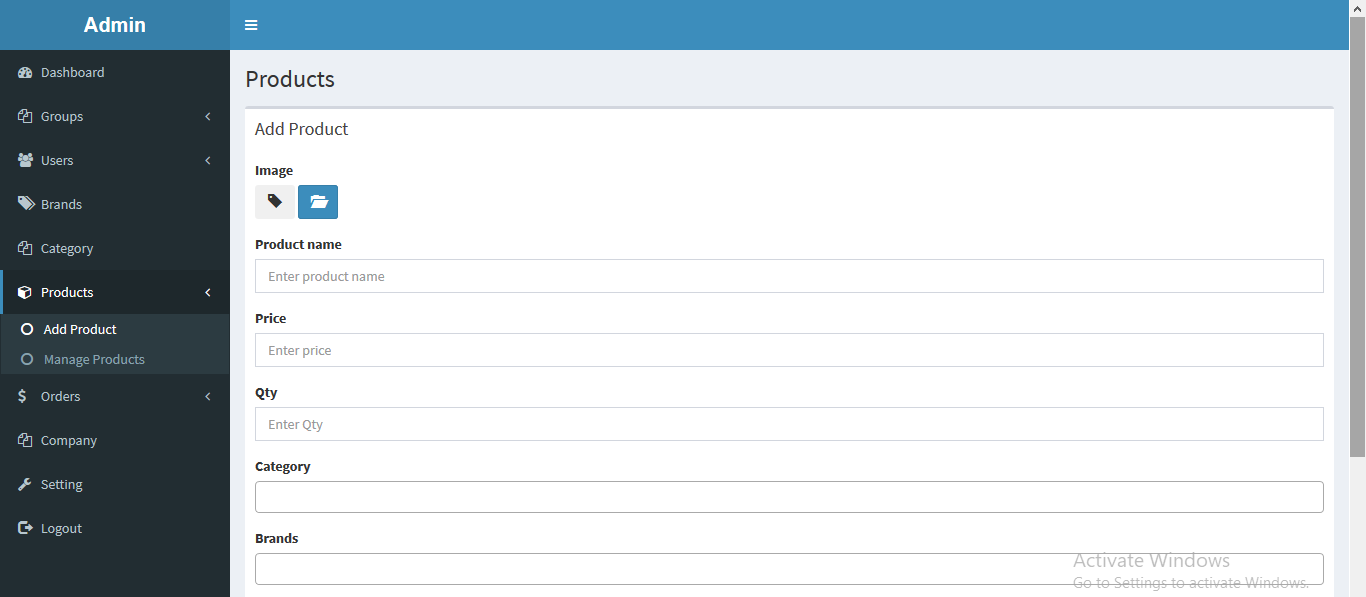
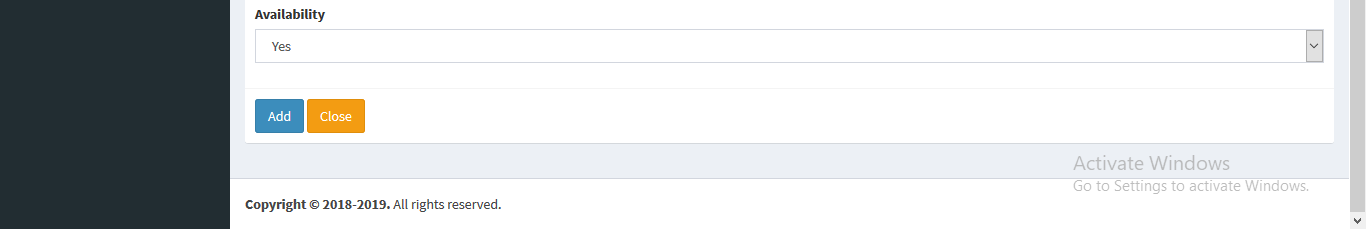
 

Figure : adding

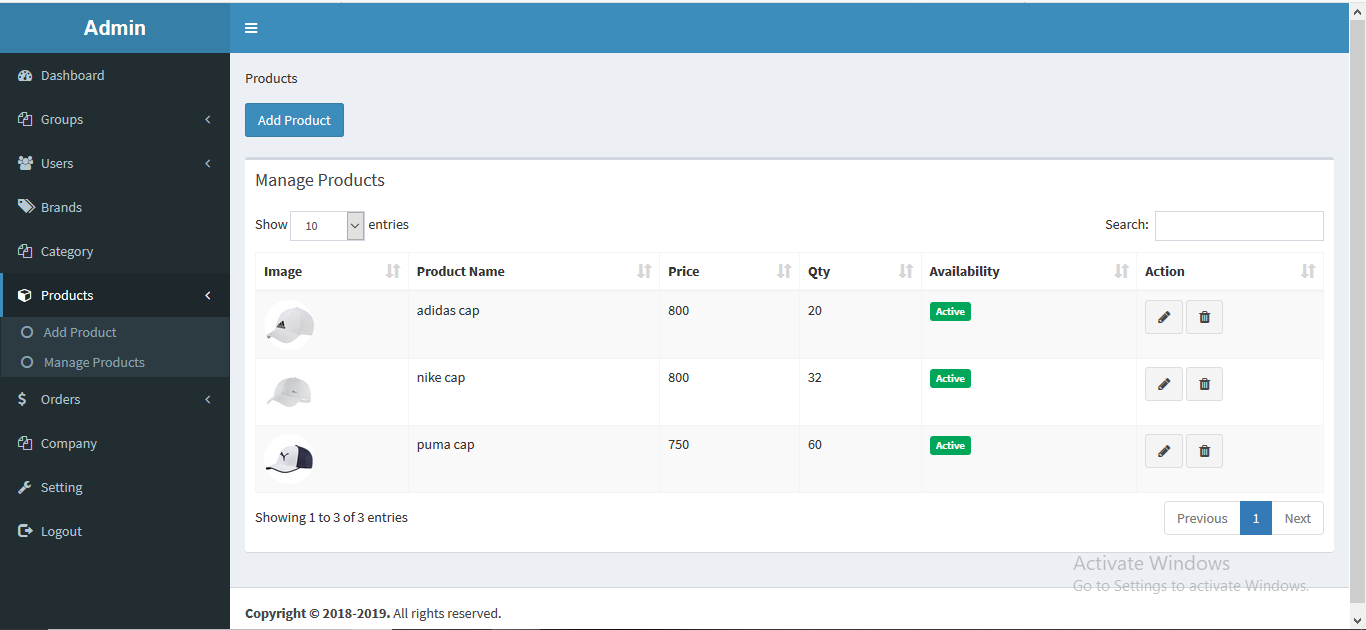


Figure : view of the available product

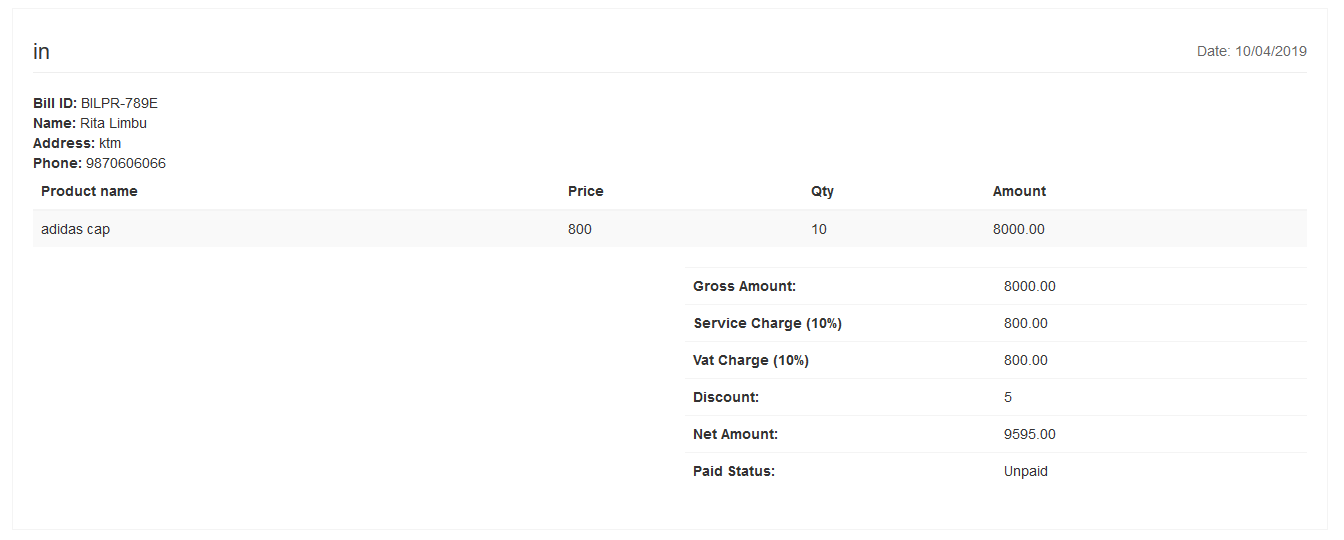


Figure : bill of the product

Here in the figure1 there is the view of the system where there admin has right to add, update and delete the users, groups, brands, category, products and orders. The figure2 show the add view of the product. There in the figure3 available products are shown and there is action field where admin can update and delete the products. The admin also can make the bill in this system. After receiving the order of the product from the user, the admin can produce the bill of that product.

# Chapter 2

## Analysis

## Introduction to Analysis

The word analysis comes from the Ancient Greek from ana-“up” and lysis-“loosening”. So, it helps to make the complex topic to smaller parts to get the best understanding of it. It helps to solve the problem as well as helps to make the decision to meet the requirements of the project.

## Need for Analysis

As for the software development, the analysis is required to make the right decision at the right time. The analysis in the software development is to collect the requirements and understand it as well as to formulate it.

## Object Oriented Analysis

It is one of the best technical approaches to analyze the system. It mainly focuses on the real world object that is important for the system. It helps to find the problem and requirements rather than a solution for the system development.

## Pitfalls of my project

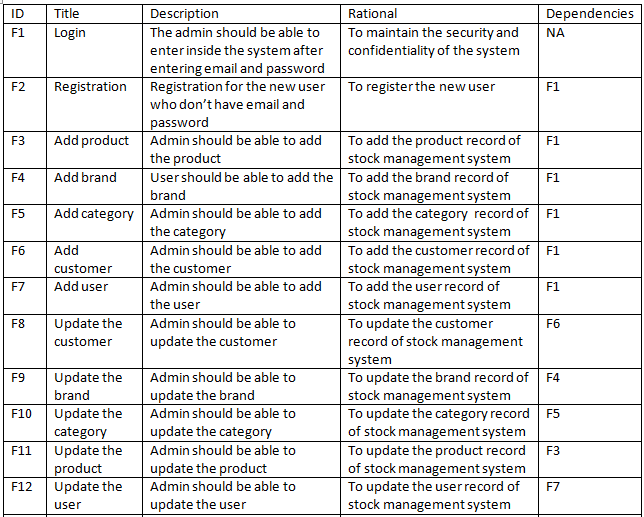
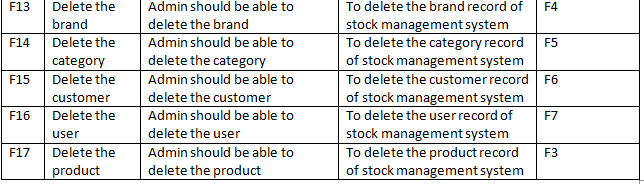
* As every project has its good and bad side. Here are the some pitfalls of my project:
* It is difficult to change or add to the first stage after entering another phase while developing the system.
* If changes need to be done then it is going to be costly and time consuming.

## Merits of my project

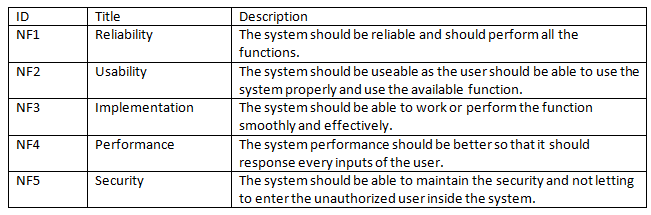
* The problems and error are identified before the stage is started while developing the system.
* There is clear about the requirement for the system as the requirement is investigated early.
* The system is likely to be very structural and it is easy to measure the progress.
* The system is done under the stages where without completion of one stage there is no step on another stage. So, the stages are completed smoothly and clearly.
* The system is easy to understand and easy to use.

## Requirement

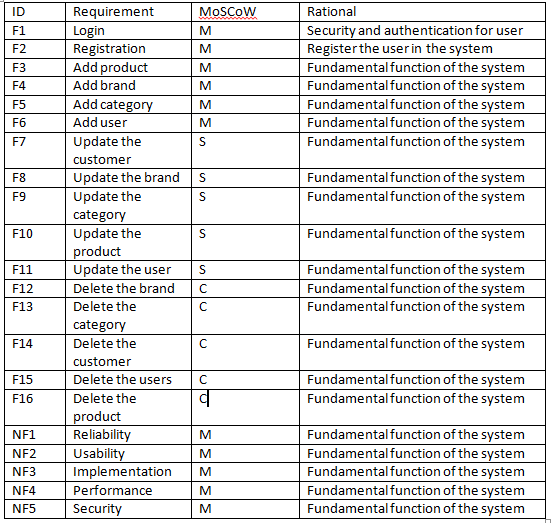
### Functional requirements

### Non-functional



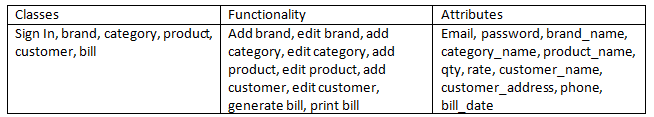
### Requirement prioritization



## NLA

It stands for the Natural Language Analysis. It is the artificial intelligence which mainly focuses on the interaction between the computer and human language. It helps to analyze the human language into the computer program form.

List



Reasons for selection

|  |  |
| --- | --- |
| Classes | Reasons |
| Sign in | To enter inside the system with the help of email and password. |
| Brand | To add, update and delete the brand of the stock. |
| Category | To add, update and delete the category of the stock. |
| Product | To add, update and delete the product of the stock. |
| Customer | To add, update and delete the customer. |
| Bill | To generate and print the bill of the product. |

## Initial class diagram

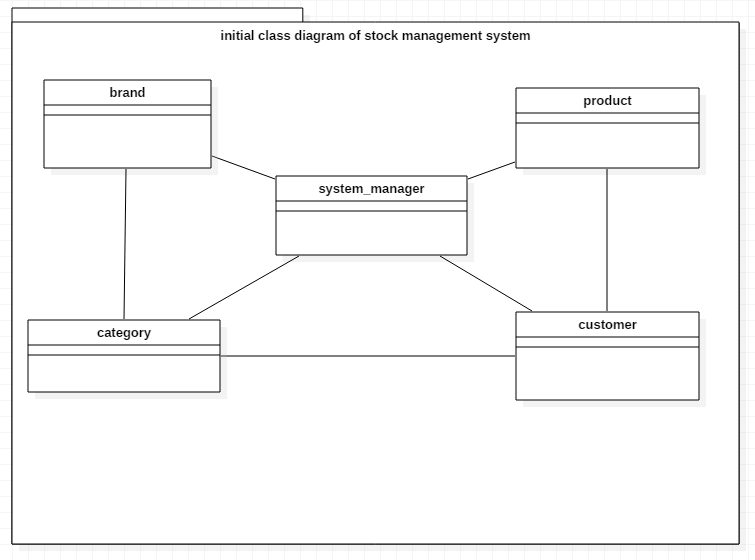


Figure : initial class diagram

## Use case diagram

It is the simple diagram which includes system, actor, use case and relationships. It represents the interaction of the user with the system and shows the relationships between the actor and the use cases. It helps to decrease the confusion as it gives clear information about the actor and their relationships with the use cases of the system.

### Justification

Use case diagram is very helpful to develop and design the system as it provides the clarity. Following are the reasons to choose the use case diagram for the system:

* Mainly it provides way to communicate complex ideas in a fairly basic way.
* It helps to easily understand the system as have proven an excellent bridge between software developers and end users.
* It helps to provide the function requirement of the system.
* It provides the easiness to understand the how system works and who are involve in the system.
* It helps to decrease the problems that can occur in the future.

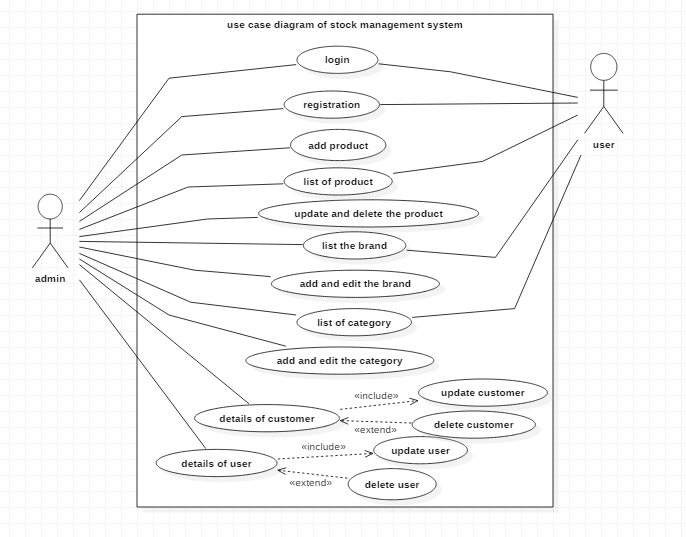


Figure : use case diagram of stock management system

* Here, in this use case diagram there are two actor admin and user.
* Firstly the admin can login in the system with valid email and password.
* The inside works is handle by the admin like insert, update and delete of product, brand, category, users and customer.
* The user can only manage the system that the admin allowed to.
* In this system the update should be done which is compulsory and the delete is in the hand of admin.
* If admin wants to delete then they can but don’t want to delete then also it is fine.

## Architecture

For the development of the project I used the 3-tier system architecture. It is divided into the three layers which are given down below:

1. Presentation layer

It is the first layer of the 3-tier architecture.

It is handles by the client system that transfer the data to the application layer.

1. Application layer

It is the second layer and also the middle layer of the 3-tier architecture that is handle by the application server.

It helps to move and process the data between the presentation and database layer.

1. Database layer

It is the third layer of the 3-tier architecture which is handling by the server system.

In this layer the data is stored and retrieved from a database or file system.

For example if the client wants to login the email then the client request to server and server request to the database. Likewise the database request from the server would send the data to server and server would forward it to the client. After this the email of the client would open. The 3-tier system architecture is given down below:

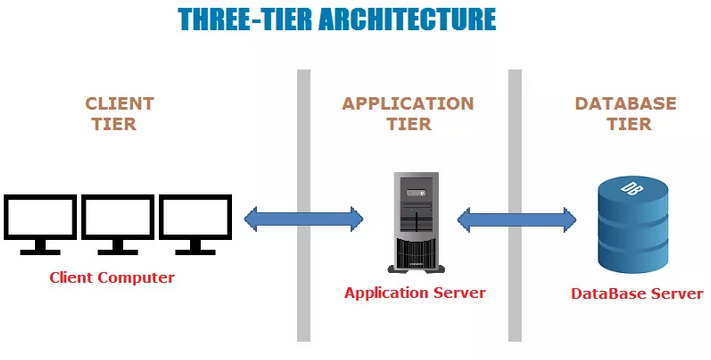


Figure : three tier architecture

# Chapter 3

## Design

### Dynamic Modeling

Dynamic modeling represents the behavior of the system. It helps to show the object interaction at the runtime. So, I used the activity and sequence diagram for the dynamic modeling of stock management system which is given down below:

#### Activity diagram

It is one of the important diagrams in UML that describe the dynamic aspect of the system. It is the diagram which is similar to flow chart that represents the flow from one activity to another activity. It is the behavioral UML diagram.

Justification

Activity diagrams are the user friendly diagrams which help to the flow of activity of the system. It shows the how the system behaves. So, following are the reasons to select the activity diagram:

* It provides the actual work flow behavior of the system.
* It describes the actual state of activities of a system by showing all the sequence of activities performed.
* It is use for analyzing an activity by describing what actions need to take place and when they should occur.
* It provides the easiness to understand the work flow of the system.
* It is mostly use diagram as it is user friendly.

The activity diagram of stock management system is given down below:

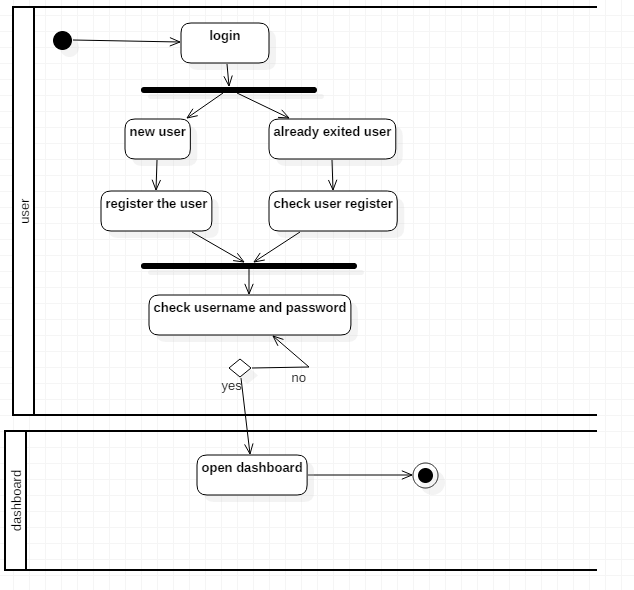


Figure : activity diagram of login

* The given activity diagram is of login system where the user can login.
* If it’s new user, then can register and login.
* If already exited then enter the email and password.
* In case of entering the right email and password, open the dashboard not then check the email and password.

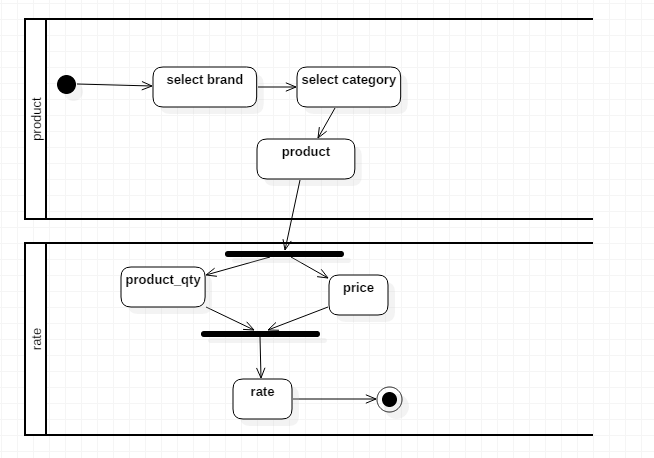


Figure : activity diagram of product

* Here, the admin can select the category, brand and enter the product.
* They can store the product by entering the product quantity and its price.

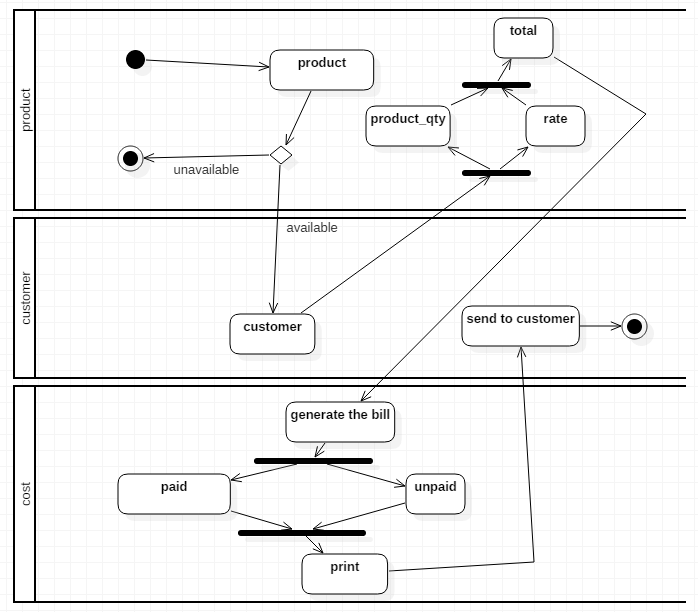


Figure : activity diagram of selling the product

* To provide the product to the buyer, select the category and brand of product.
* Check whether the product is available or not.
* If it’s available, then calculate the amount of the product.
* After calculation generate the bill for that product and send it to the customer.

#### Sequence diagram

Sequence diagram is the diagram which shows behave of the system. It has different components like actor, lifeline, objects, focus of control, message, etc. with the help of these components the sequence diagram is made which helps to interact with the objects with the aim of exchanging the message at the time.

Justification

Sequence diagram is one the most required diagram for the project as it helps to shows behaves of the system. So here are the reasons for selection of the sequence diagram:

* It shows the actual behavior of the system.
* It shows the message pass between the objects at a time.
* It shows the actual flow of the system.
* It provides the easier platform to understand the system.
* It is user friendly and easy to use as well.

The sequence diagrams for the stock management are given down below:

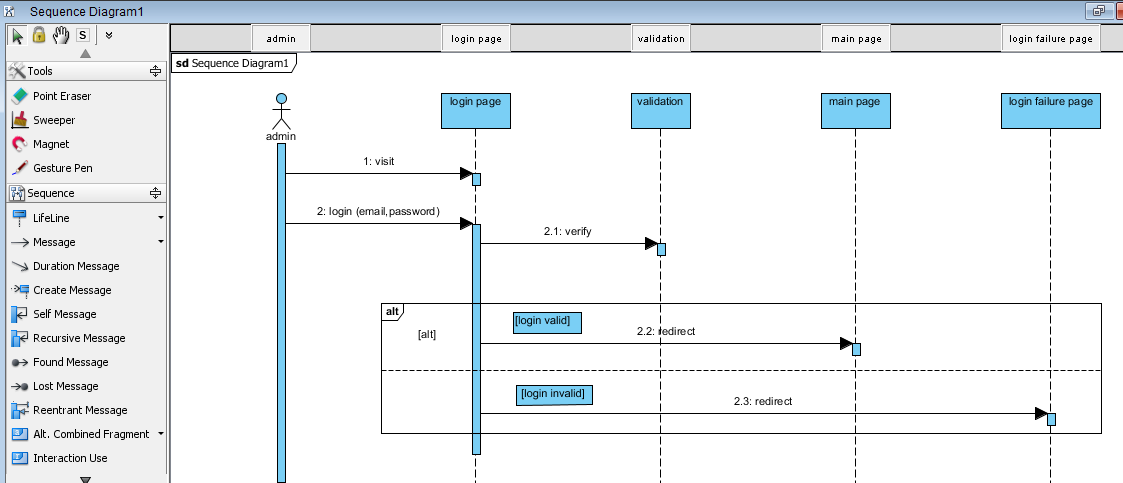


Figure : sequence diagram of login system of stock management system

Here in this diagram the admin is the actor. Login page, validation, main page and login failure page are the objects.in this sequence diagram I have used actor, object, lifeline, message, focus of control, text and the frame.

* Firstly the admin go to the login page.
* After visiting the login page, the admin login the system by entering the email and password.
* To check the correct email and password it will verify in the validation page.
* If it’s correct then the main page will open not then the login failure page will open.

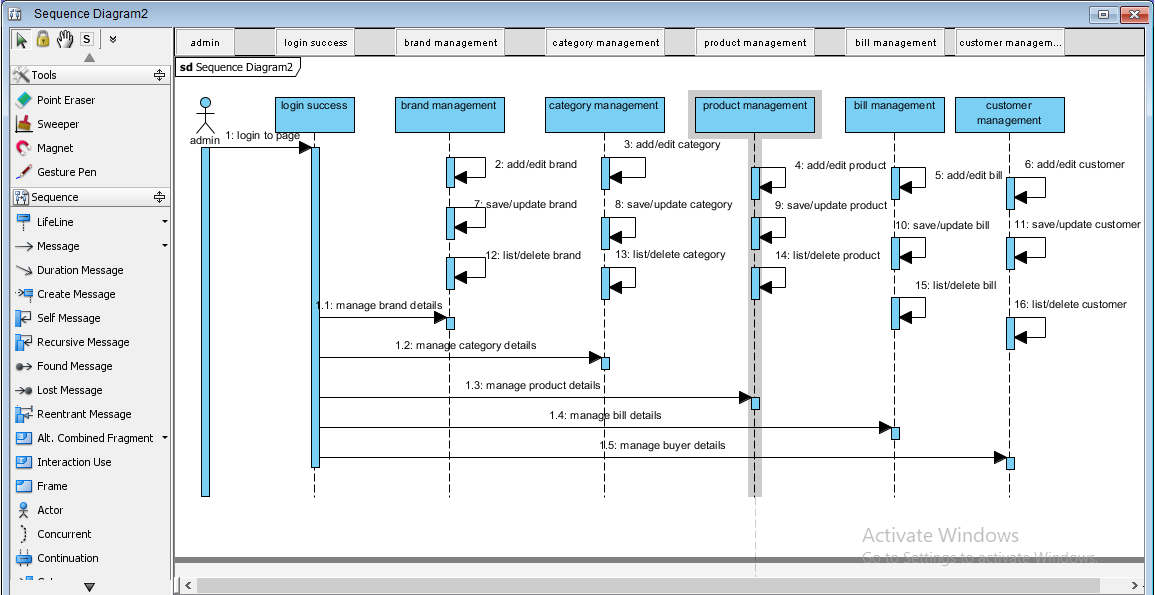


Figure : sequence diagram after entering the main page of the system

Here in this diagram also the admin is the actor. The login success, product management, brand management, category management, bill management and customer management are the objects of this sequence diagram.

* After entering the page the admin can manage all the details of the product, brand, category, bill and buyer.
* While in the product management object, the admin can add, edit and list the product.
* As well in the brand management object, the admin can add, edit and list the brand.
* Likewise in the category management, bill management and customer management, the admin can add, edit and list the category, bill and customer.

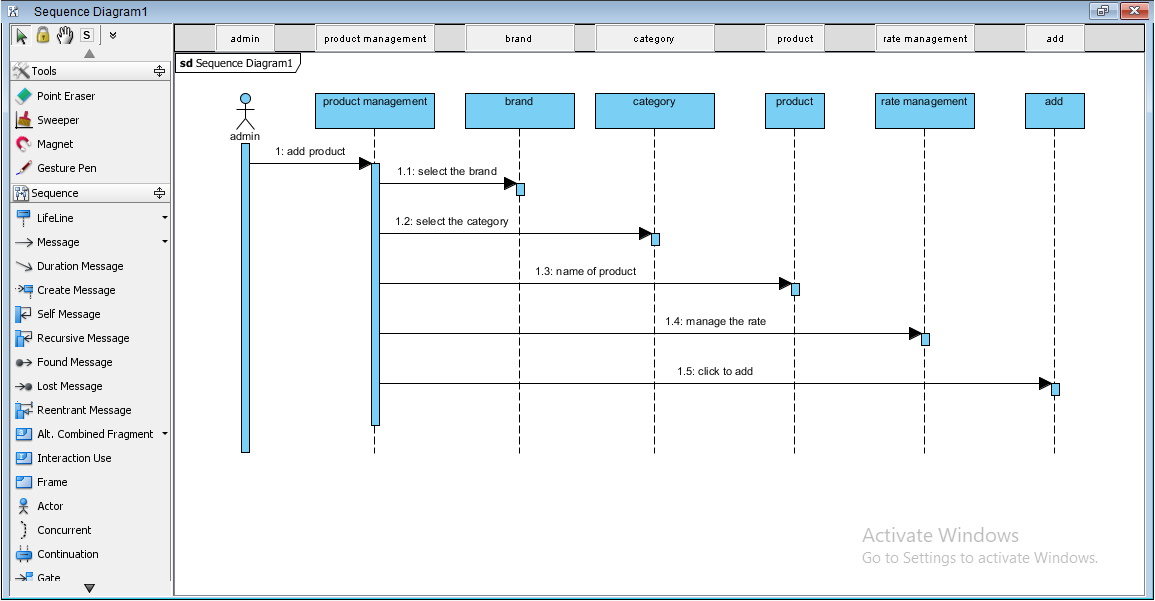


Figure : sequence diagram of adding the product

Here in this diagram the admin is the actor as well. The product management, category, brand, product, rate management and add are the objects of this sequence diagram.

* To add the products firstly visit in the product management object and select brand.
* Likewise select category, enter the name of product and manage the rate.
* Now add the product by clicking the add button.

### Structural Modeling

It represents the structure of the system. It doesn’t show the behavior of system as it is only related to structure of system. I have included the class and context diagram which is structural modeling for this project and that is given down below:

#### Class diagram

It is the graphical representation which is made up of different classes and relationship between them in the object oriented system. In the class diagram, the attributes and operation forms the complete class. It is use while systems design and development.

Justification

Class diagram provides the simple and effective way to design the system. So, I have also chosen the class diagram to design the system. Following are the reason to choose the class diagram.

* It provides the information about how the system is structured or design.
* It contains the relevant structural relations and data type.
* It provides the overall sketch of the system.
* It is flexible as it gives the clear structure of system.
* It is also mostly used diagram to design the system.

The class diagram of stock management system is given down below:

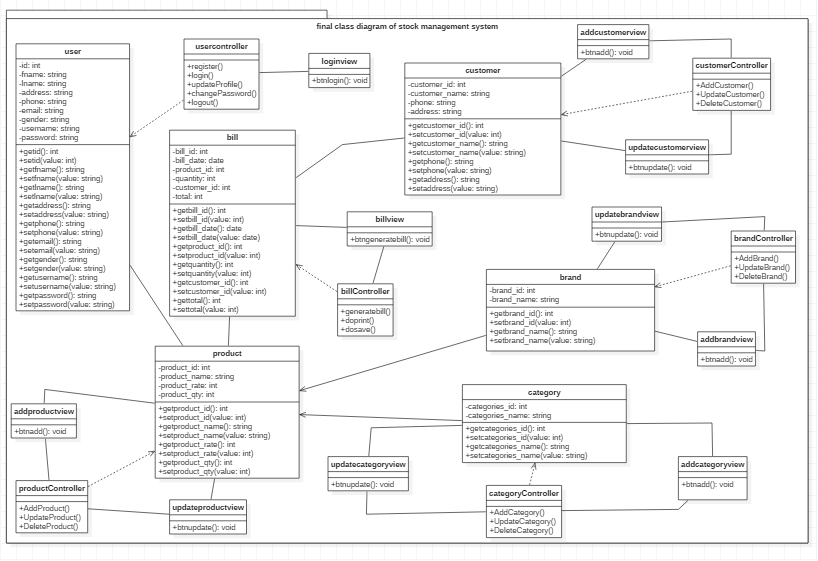


Figure : final class diagram of stock management system

Here, the classes that are involved in the class diagram are user, product, category, brand and customer. Here the admin login inside the system and manage the product. It manages the category and brand of the product. The customer can buy the product. The class diagram is made in the MVC pattern as shown in the figure above. In this class diagram, there is the attributes of classes and its operation are also given as well as the connection of classes.

#### Context diagram

It is the diagram which shows the high level view of a system. This diagram plays vital role while developing the system.

Justification

Following are reasons for selecting the context diagram to develop the system:

* The aim of this diagram is to mainly focus on the external factors while developing the system.
* It picturised by putting the system in the center and surrounded it by its interacting system, environment and activities.
* It shows how the information flows in the system.
* It is easy to draw as well as it can expand by adding.
* To draw it, there is no need of technical knowledge.

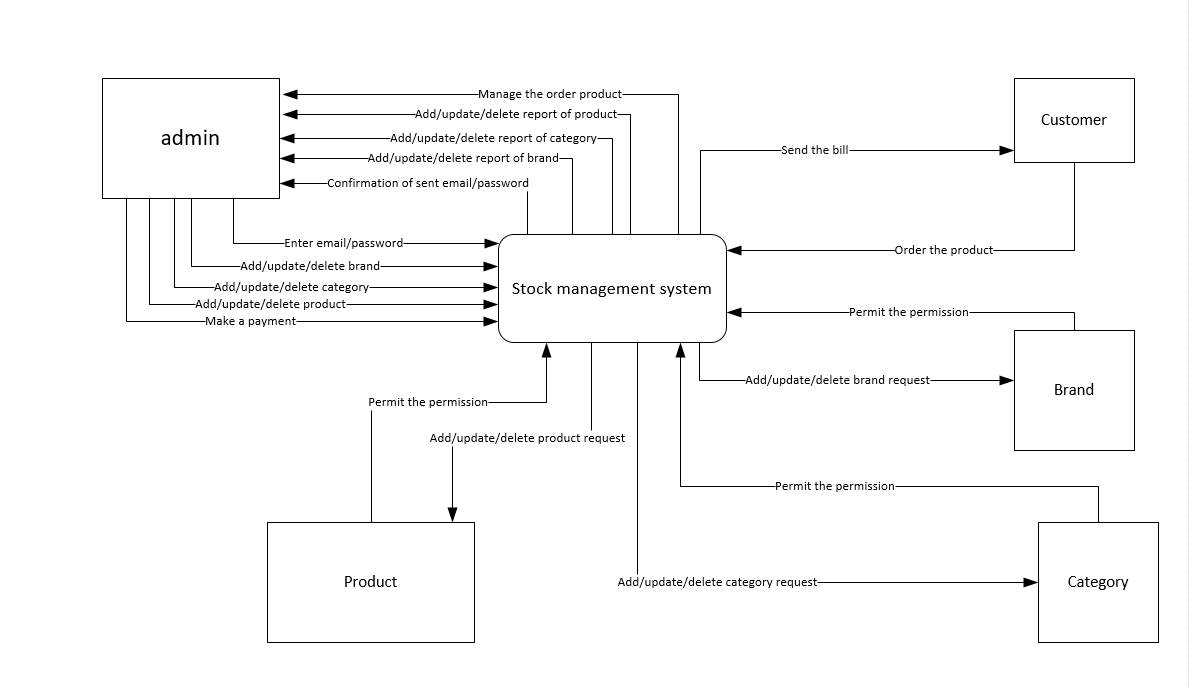


Figure : context diagram

* Here, the stock management system is in the center and surrounded by admin, product, customer, brand and category.
* Firstly the admin enter email/password and system gives the confirmation.
* Admin can add/update/delete the brand, category and product as system provide report as well.
* The customer can order the product and admin makes the payment and lastly send the bill of the product to the customer.

### Database Modeling

It shows the logical structure of database. It provides how the data can be organized, stored and manipulated. Some of the database modeling is given down below:

#### Er-diagram

It is the visual representation of the design which shows the relationships between the entities which is stored in the database. It provides simple way to understand the system and its work as it provides the details of the relationship of entity.

Justification

Following are the reasons to select the er-diagram for the system:

* Firstly it is highly flexible as it easily delivers the other relationships from the already existing ones.
* It is easy to understand as it acts as the blueprint for the database.
* It decreases the complication which can arise during the development.
* It provides the effective communication between the entities.
* It is very simple to understand.

Er-diagram of stock management system

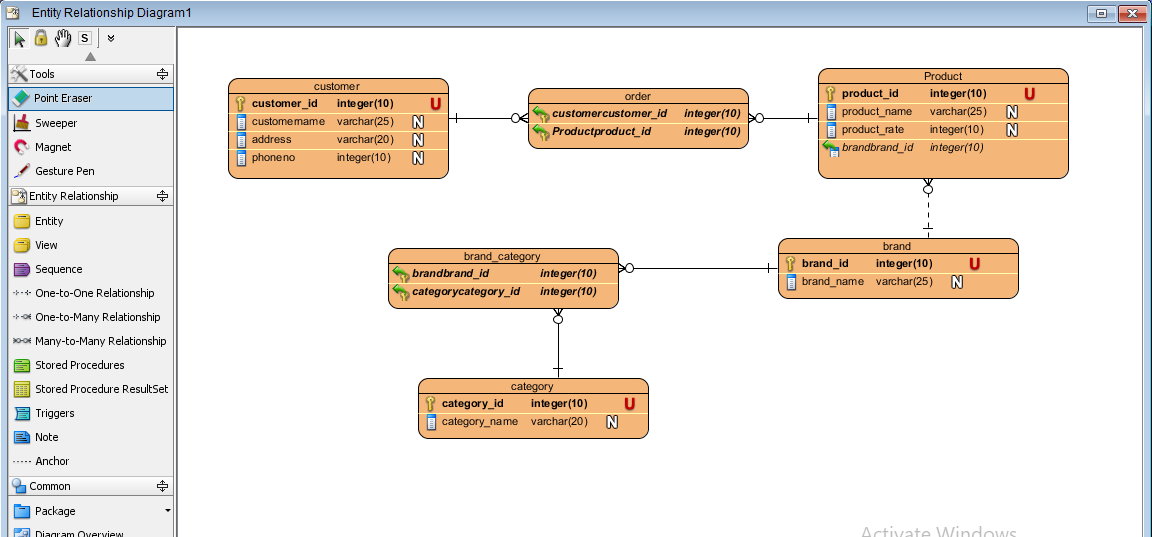


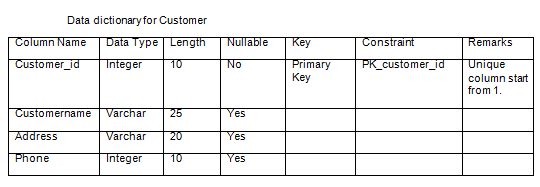
Figure : Er diagram of the stock management system

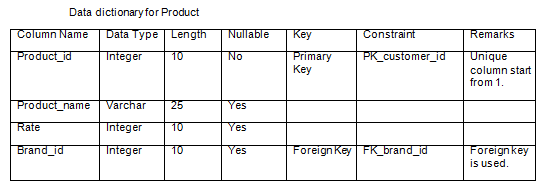
Here, there are six entities which are customer, order, product, brand, category and brand\_category. The product and customer forms the many to many relationships which is the order as show in the diagram. The brand and product also form the one to many relationships as the one brand can have the many products but one product is only of one brand. As well as the brand and category forms the many to many relationship which id the brand\_category as one brand has many category and one category has many brands as shown in the diagram.

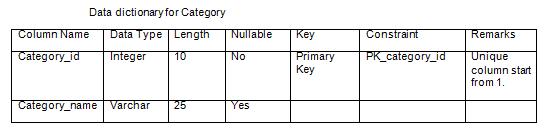
#### Data dictionary

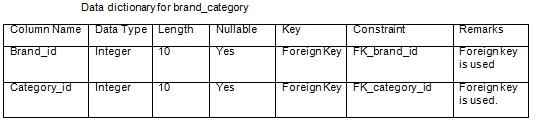
It is the set of collection of description of data object. It makes easier to understand the database of the system.

The data dictionary of this system is given down below:









#### UI modeling

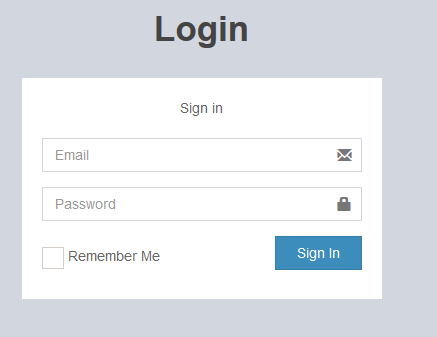


Figure : login form of stock management system

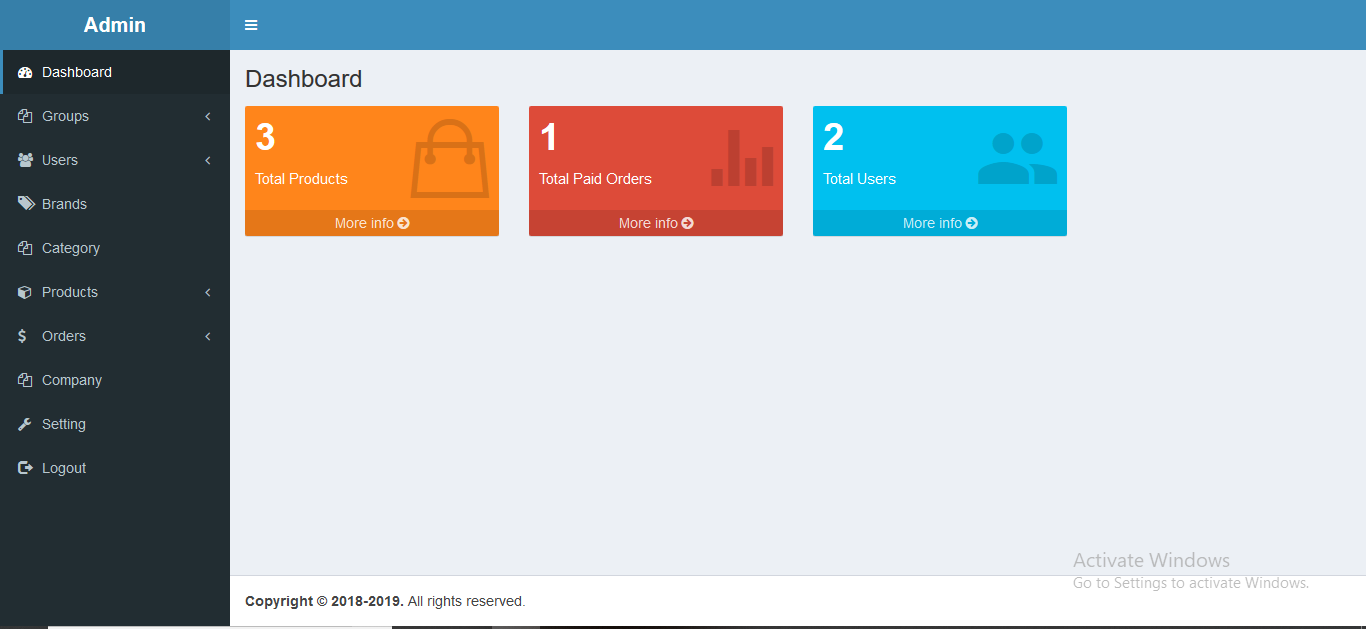


Figure : dashboard of the stock management system

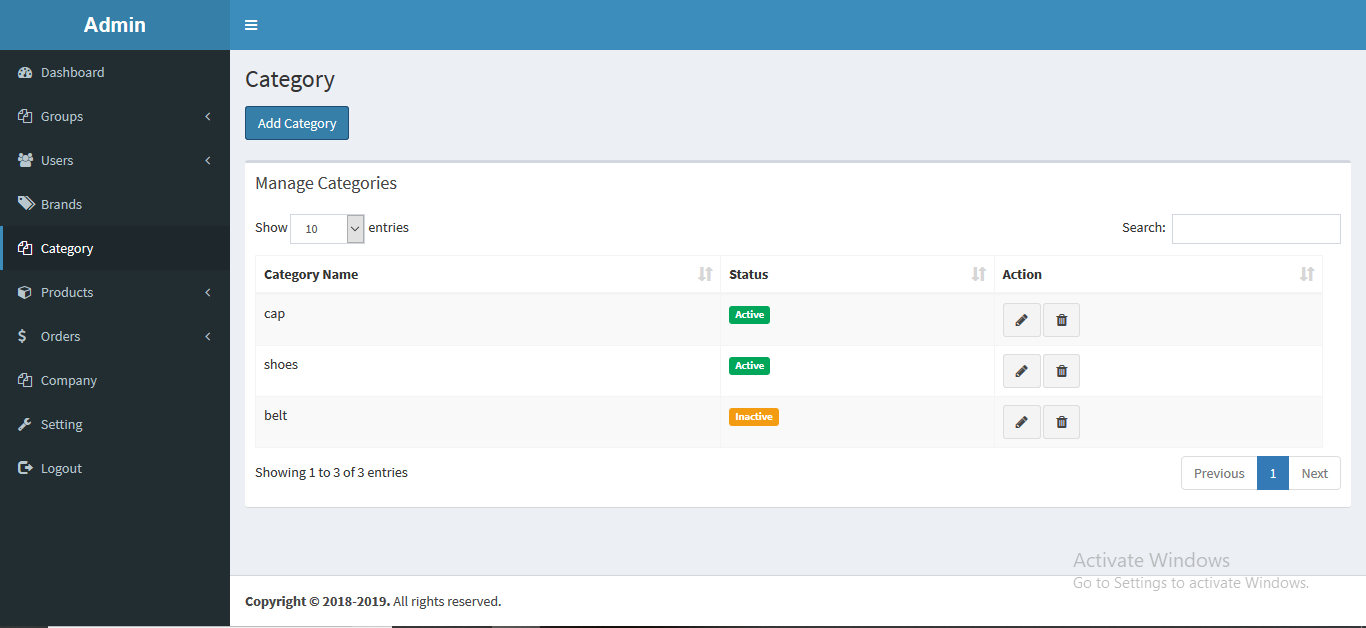
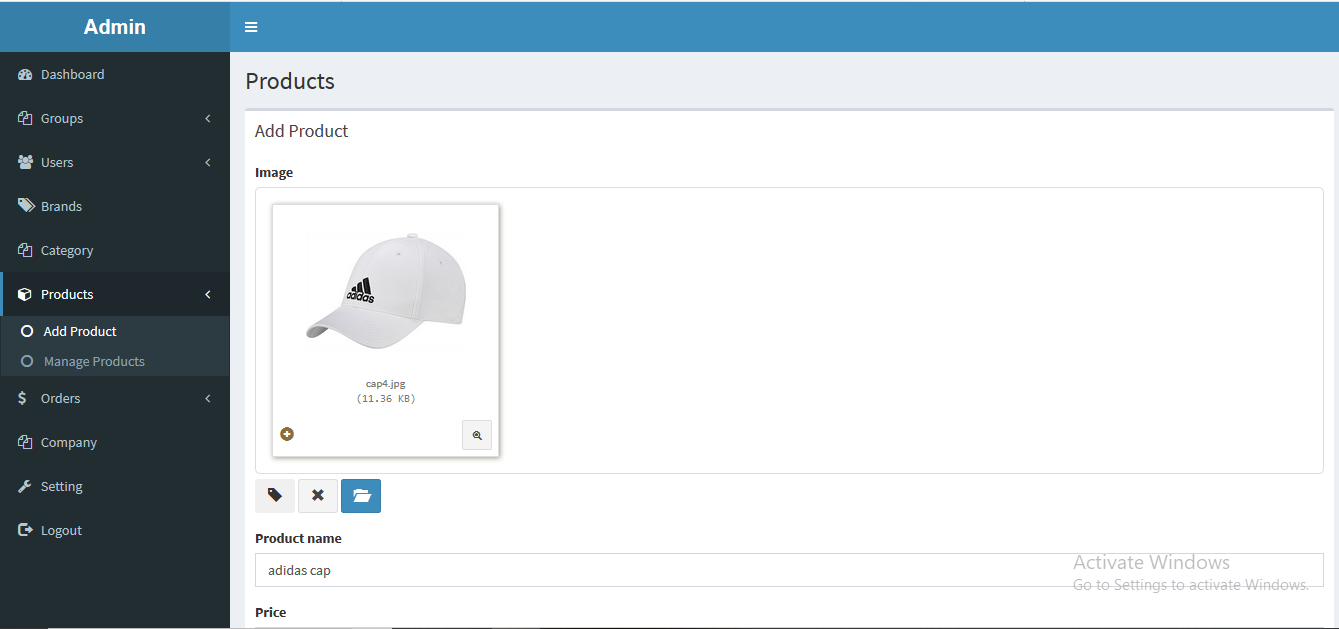


Figure : category form of stock management system



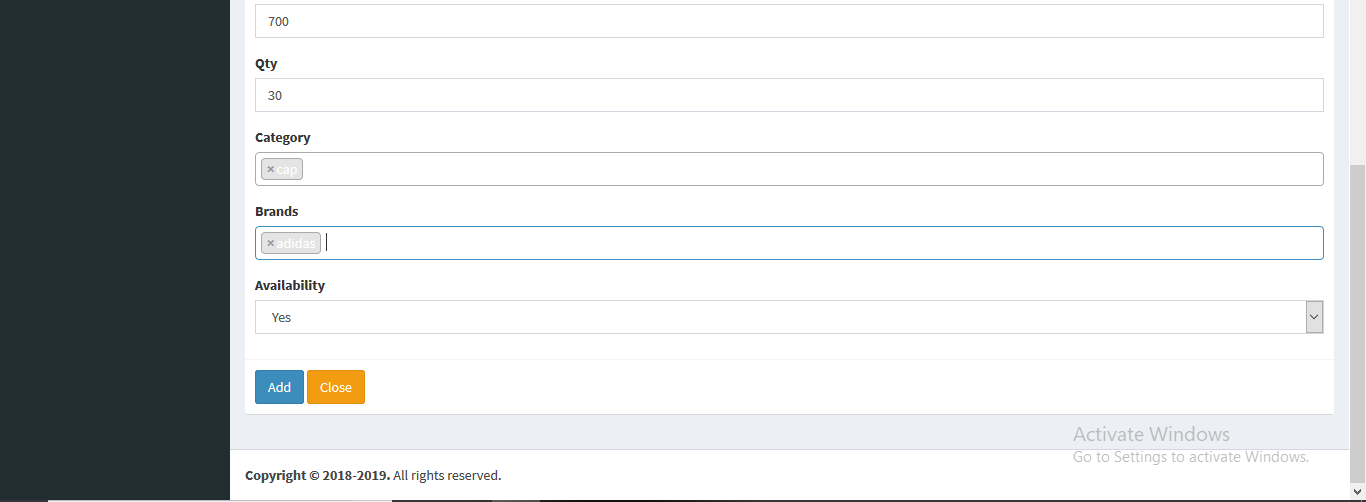


Figure : adding product form

# Chapter 4

## Implementation

### Programming language used

#### PHP

It stands for the Hypertext Preprocessor. It is the programming language which especially use for developing the dynamic web application. The php code embedded into the HTML code and also JavaScript as well as it can be used in combination with web framework. The database is stored in the MYSQL. So, I use php programming language to develop the web application (i.e. stock management system).

Following are the reasons for using php programming language

* It is simple to use and easy to understand.
* It is also free and fast as well which helps to develop web application.
* The making of dynamic web application is possible with the help of HTML as well.

#### CodeIgniter

To develop this project I use the CodeIgniter framework. CI is one of the best php framework uses to develop web application with advanced features. It helps to develop the project faster in compare to other framework and it is also reliable. It is based on the model view controller (MVC) development pattern.

Model

Model classes contain functions that help to retrieve, insert and update information in database.

View

View is the web page or what the user sees but in CodeIgniter a view also can be a page fragment like a header or footer.

Controller

Controller serves between model and view and generates a web page.

Following are the reasons for the selection of CodeIgniter framework

* It is easy to migrate from one server to another server.
* It is easy to understand and also based on model view controller (MVC) pattern.
* It provides the facility of adding the custom codes which help to change the configuration.

So, with the help of PHP programming language and CodeIgniter framework the stock management system is develop. As the project is not developing in the team, it is easy and fast to develop the project in the PHP CodeIgniter framework.

# Chapter 5

## Testing

### Black box testing

Black box testing is one the testing method for the software in which the functional part of the software is tested. It doesn’t test the internal part of the system. It helps to test only the behavior of the system that is why it is also known as the behavioral testing.

Following are the reasons for selection of black box testing

* It is easy to test and simple to use.
* No need of technical knowledge as little knowledge is enough to perform the black box testing.
* It is simple to understand as well.

The black box testing of stock management system is given down below:

Test cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test\_id** | **Description** | **Test value** | **Expected result** | **Actual result** | **Yes or NO** |
| Test\_001 | Testing: empty value in the login form | Email=  Password= | Message box with email field is required  Password field is required. | The email field is required  The password field is required. | Yes |
| Test\_002 | Testing: inserting the invalid password in login form. | [Email=admin@gmail.com](mailto:Email=admin@gmail.com)  Password=dh123f | Message box with incorrect email/password | Incorrect email/password | Yes |
| Test\_003 | Testing: inserting the invalid email and password | [email=gita@gmail.com](mailto:email=gita@gmail.com)  Password=nvdfkfenej | Message box with email does not exist. | Email does not exits | Yes |
| Test\_004 | Testing: inserting the email but leaving the password box empty | [Email=](mailto:Email=admin@admin.com)[admin@gmail.com](mailto:admin@gmail.com)  Password= | Message box with password field is required | Password field is required | Yes |
| Test\_005 | Testing: inserting the password but leaving the email box empty. | Email=  Password=12345678 | Message box with email field is required | Email field is required | Yes |
| Test\_006 | Testing: inserting the correct email and password | [Email=](mailto:Email=admin@gmailcom)[admin@gmail.com](mailto:admin@gmail.com)  password=12345678 | Should open the dashboard | Opening the dashboard | Yes |
| Test\_007 | Testing: changing the password with length is less than 8 characters | Password=123456  Confirm password=123456 | Message box with password field must be at least 8 characters in length | The password field must be at least 8 characters in length | Yes |
| Test\_008 | Testing: changing the password which length is more than 8 characters | Password=1234567890  Confirm password=1234567890 | Message box with successfully updated. | Successfully updated. | Yes |
| Test\_009 | Testing: changing the email | [Email=mandira@admin.com](mailto:Email=mandira@admin.com) | Message box with successfully updated. | Successfully updated. | Yes |
| Test\_0010 | Testing: changing the username | Username=mandy | Message box with successfully updated. | Successfully updated. | Yes |
| Test\_oo11 | Testing: inserting the brand | Brand=Gucci | Message box with successfully added | Successfully added | Yes |
| Test\_0012 | Testing: updating the brand | Brand=Gucci  Status=inactive | Message box with successfully updated | Successfully updated. | Yes |

Test log

|  |  |
| --- | --- |
| Test\_id | Output |
| Test\_001 |  |

|  |  |
| --- | --- |
| Test\_id | Output |
| Test\_002 |  |

|  |  |
| --- | --- |
| Test\_id | Output |
| Test\_003 |  |

|  |  |
| --- | --- |
| Test\_id | Output |
| Test\_004 |  |

|  |  |
| --- | --- |
| Test\_id | Output |
| Test\_005 |  |

|  |  |
| --- | --- |
| Test\_id | Output |
| Test\_006 |  |

|  |  |
| --- | --- |
| Test\_id | Output |
| Test\_007 |  |

|  |  |
| --- | --- |
| Test\_id | Output |
| Test\_008 |  |

|  |  |
| --- | --- |
| Test\_id | Output |
| Test\_009 |  |

|  |  |
| --- | --- |
| Test\_id | Output |
| Test\_0010 |  |

|  |  |
| --- | --- |
| Test\_id | Output |
| Test\_0011 |  |

|  |  |
| --- | --- |
| Test\_id | Output |
| Test\_0012 | E:\SMS\testing\test18.PNG |

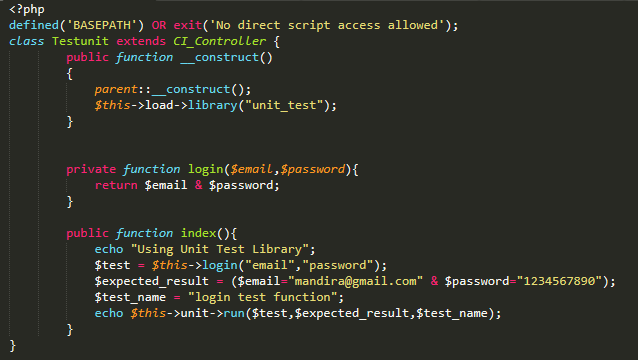
### Unit testing

Unit testing is one of the best testing which is used to test the software. It helps to test the internal parts of the system or application. Not any one can use it to test because it needs the knowledge about the programming and coding.

Following are reasons for the selection of unit testing

* Only the knowledgeable of coding and programming can perform the testing.
* It tests the internal parts of the system.
* Not any one can perform unit testing so it is important testing for the software development.

Test case



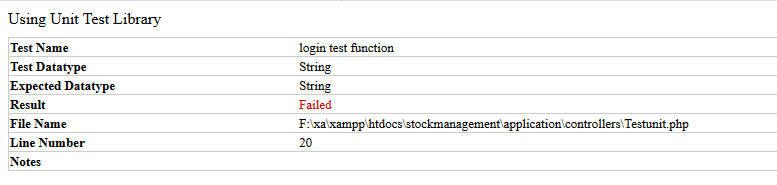
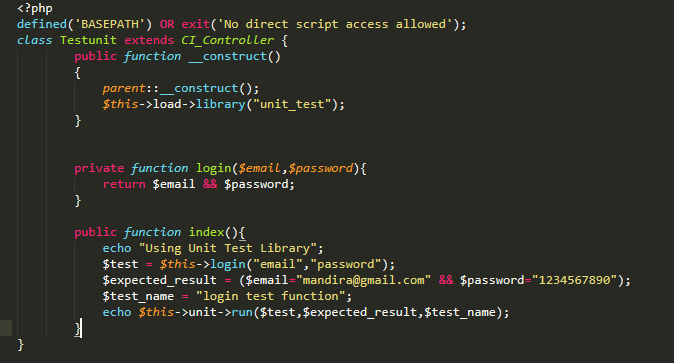


Figure : testing valid email and password



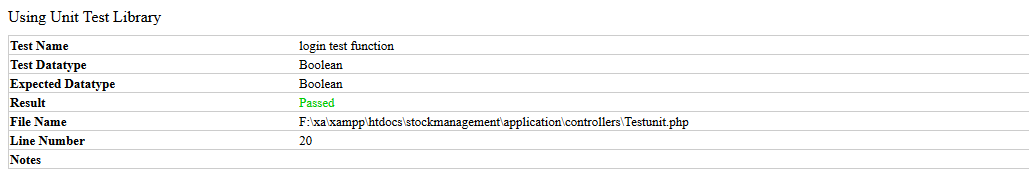
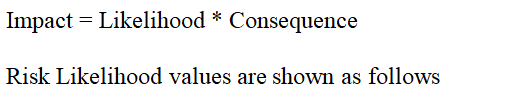


Figure : testing passed of email and password

# Chapter 6

## Other project issues

### Risk management



Risk management is the process of managing the risk or the problems that arise in the present and in the future. It helps to manage the risk that appears in the system and also helps to reduce it. It helps to find out the risk in the system and helps to give the solutions that can arise in the present as well as in the near future. The risk and its solution for this project are given down below in the table:

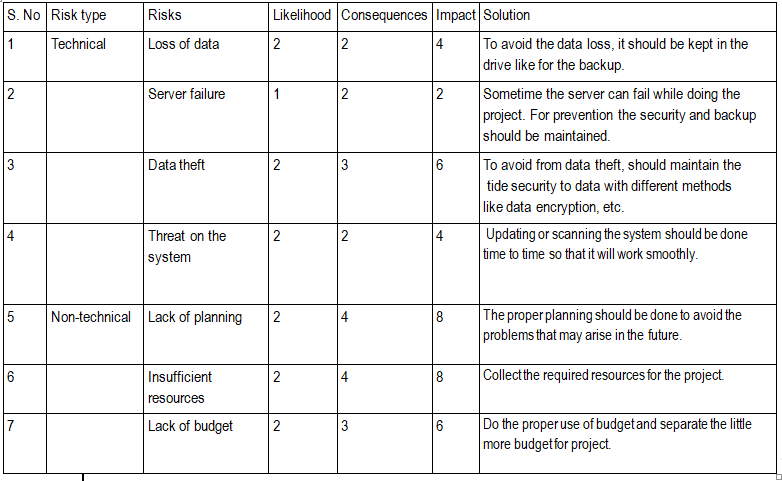


Figure : risks management

### Configuration management

Configuration management is the management which helps to keep the record safe and helps to get the correct and relevant document for everyone. If any problem occur and any changes need to done then with the help of configuration management we can add, edit and modify the files. It also helps to know that which items need to be configured. I used the github for the configuration management. The diagram of configuration management of this project is given down below:

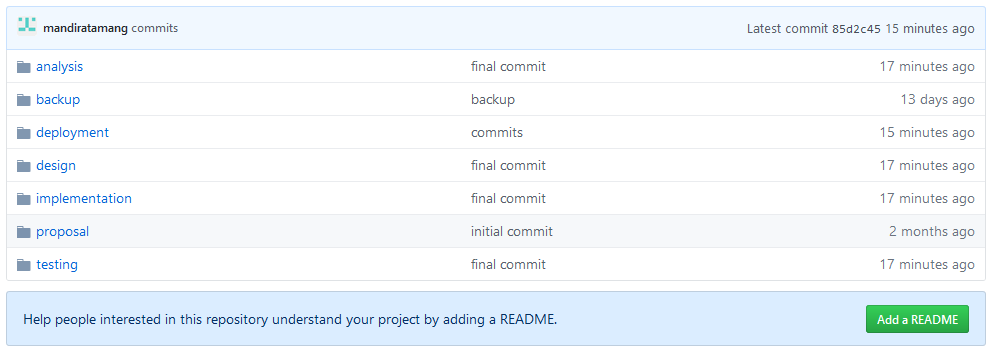


Figure : configuration management in github

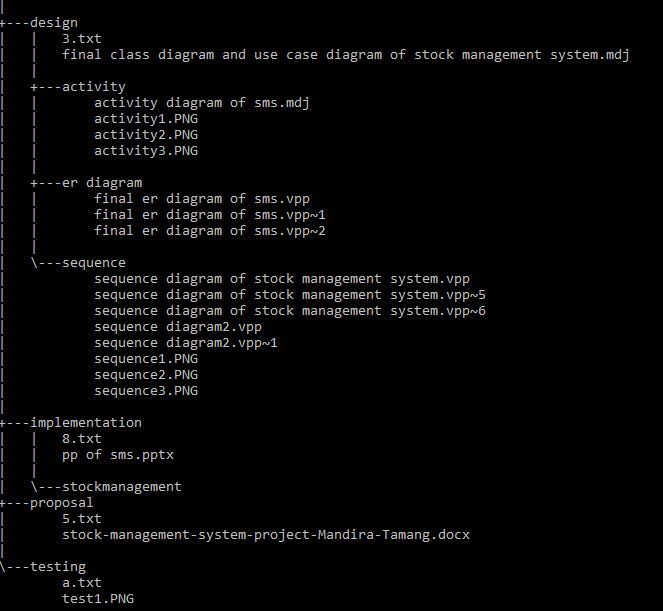
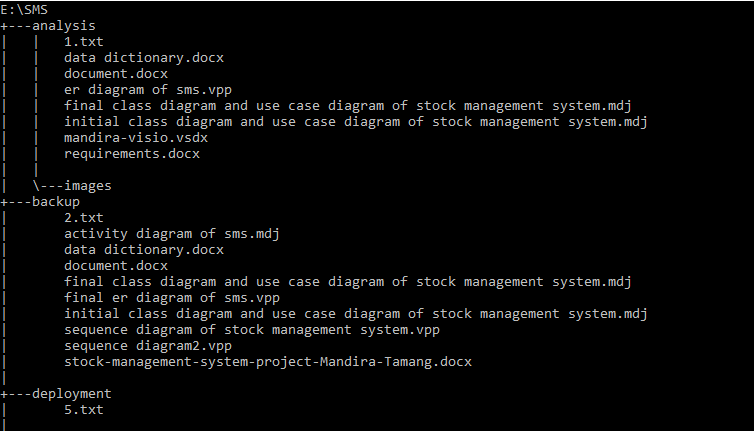


Figure : configuration management

### Scheduling/Gantt chart

The scheduling helps to set the start date and the end date of the project. It helps to complete the project on time and helps to manage the project smoothly. The schedule of this project is show down below:

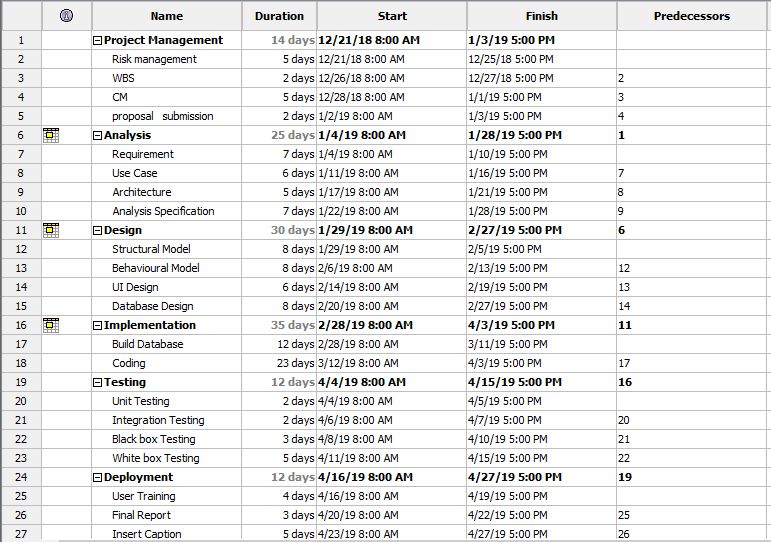


Figure : schedule of the project

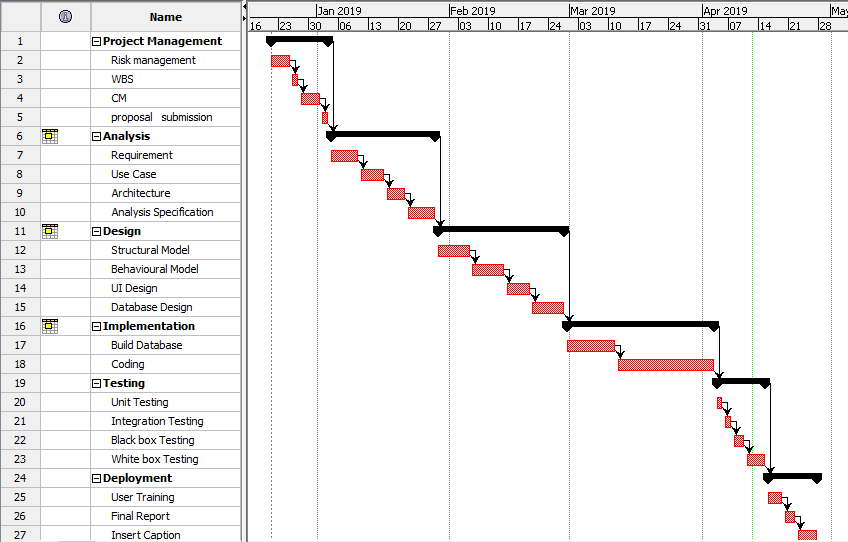


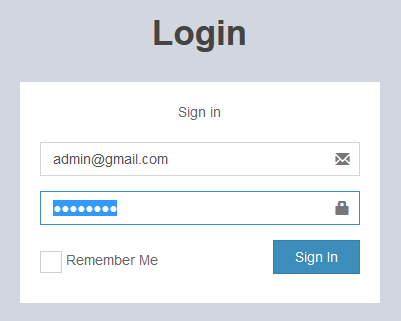
Figure : Gantt chart of the project

### Future work

* If the admin wants to keep the staff and manage the stock then they can allow and also can select the feature that they want to allow to the staff in the future.
* This project is not only implemented in the business but also for the small shops as it manage the stock.
* The admin can absolutely manage the details of the stock as they can delete it or edit it.
* If the admin has the knowledge about the coding of the project then they can modify it as it is not the online website.
* The calendar can add in the future.
* The flow chart of the sell product can add in the future.

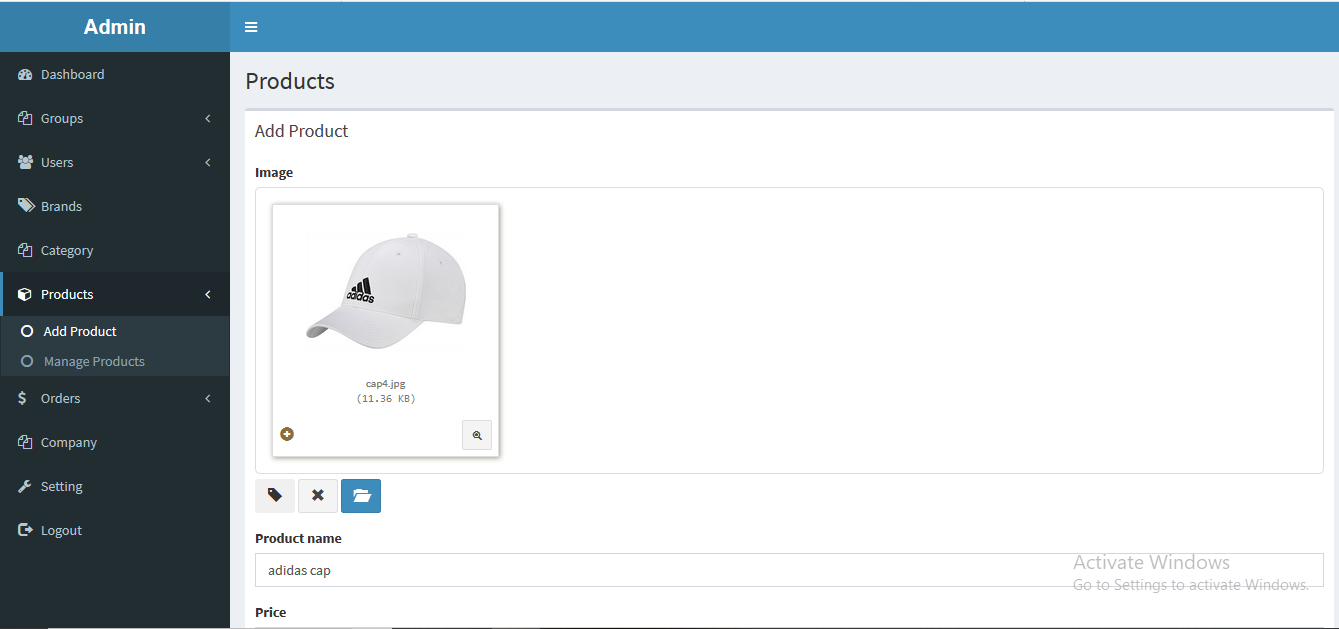
### User manual

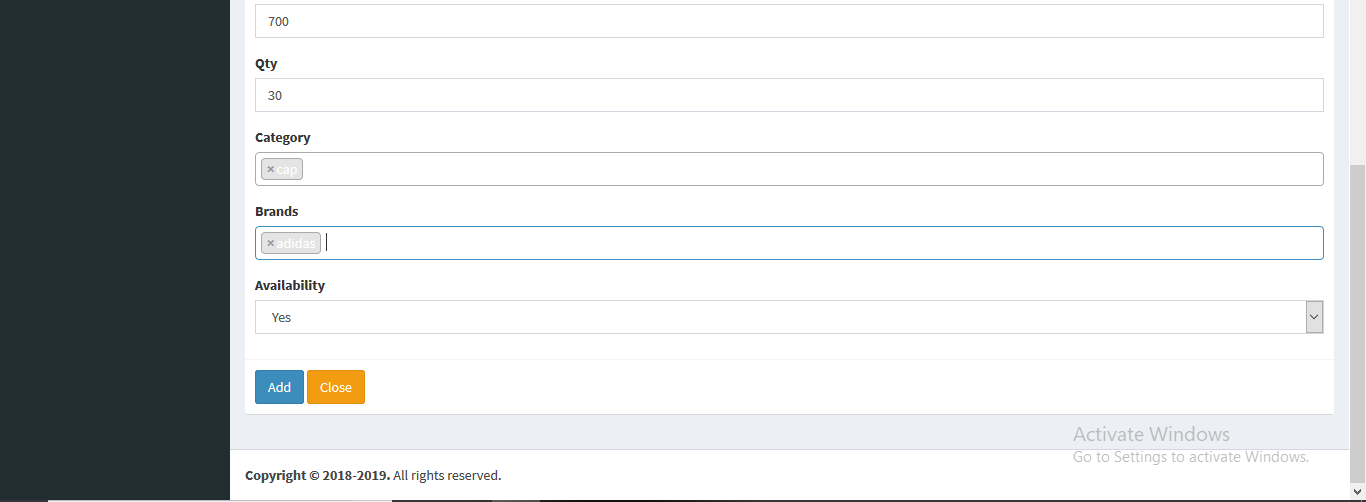
#### Login in



* Firstly login page appears where admin should enter the valid email and password.
* After entering the valid email and password, click the sign in button.

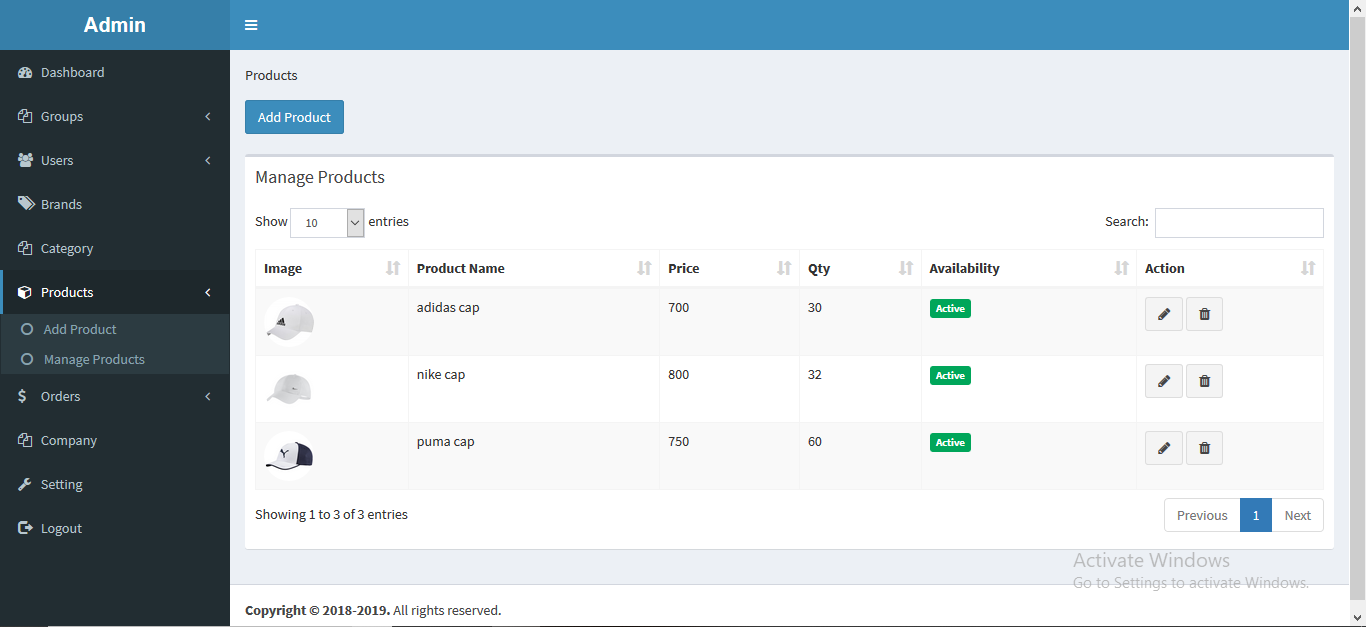
#### Adding new product





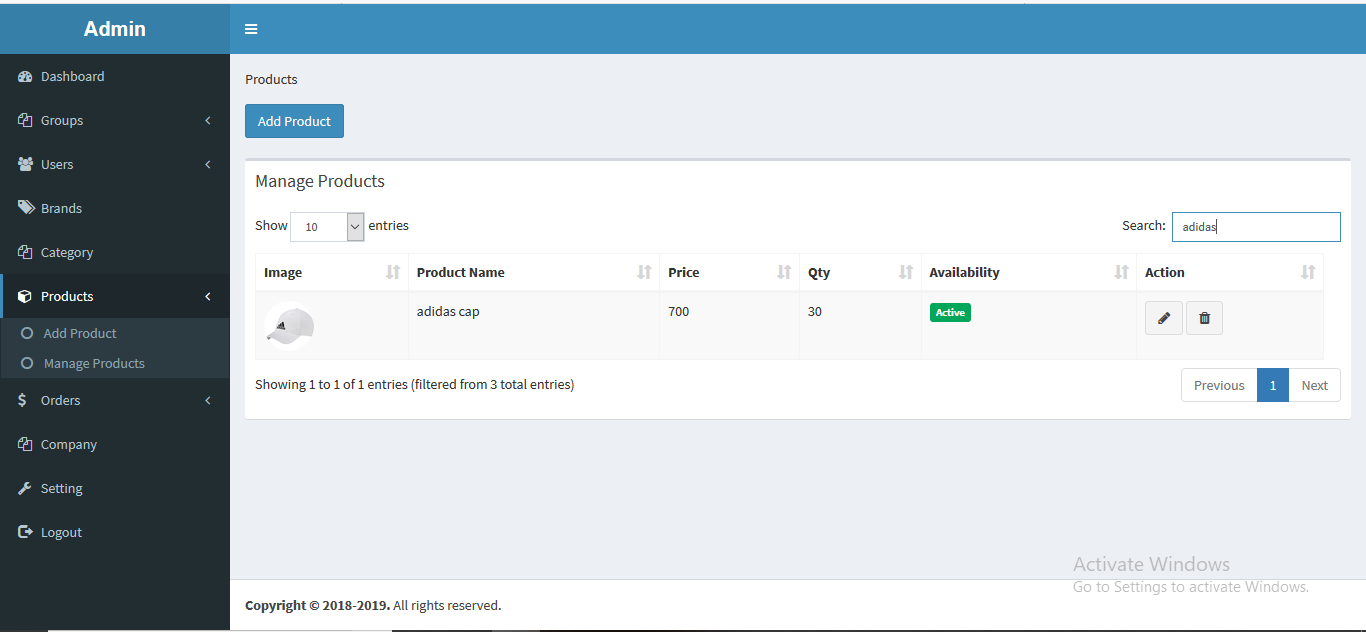
* After entering the valid email and password, the dashboard appears where there is product in the side-menu bar as given the figure above.
* Add product and manage products appears after clicking the products.
* Click the add product and input the product data.
* To add it click the add button as shown in the above figure.

#### Displaying all the products



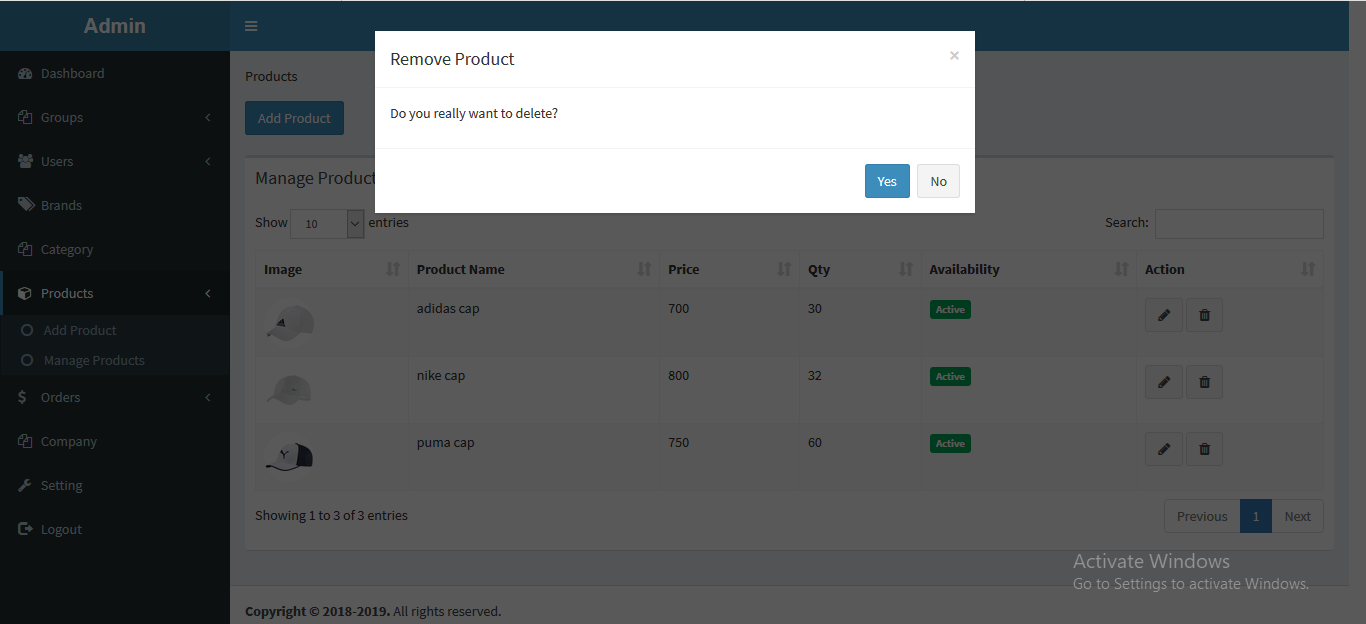
* Enter the valid email and password.
* Click the product on the dashboard.
* To view the added product, click the manage products after clicking the product.
* Inside the manage products, there display all the products that are added which is given in the figure above.

#### Searching and sorting result-product



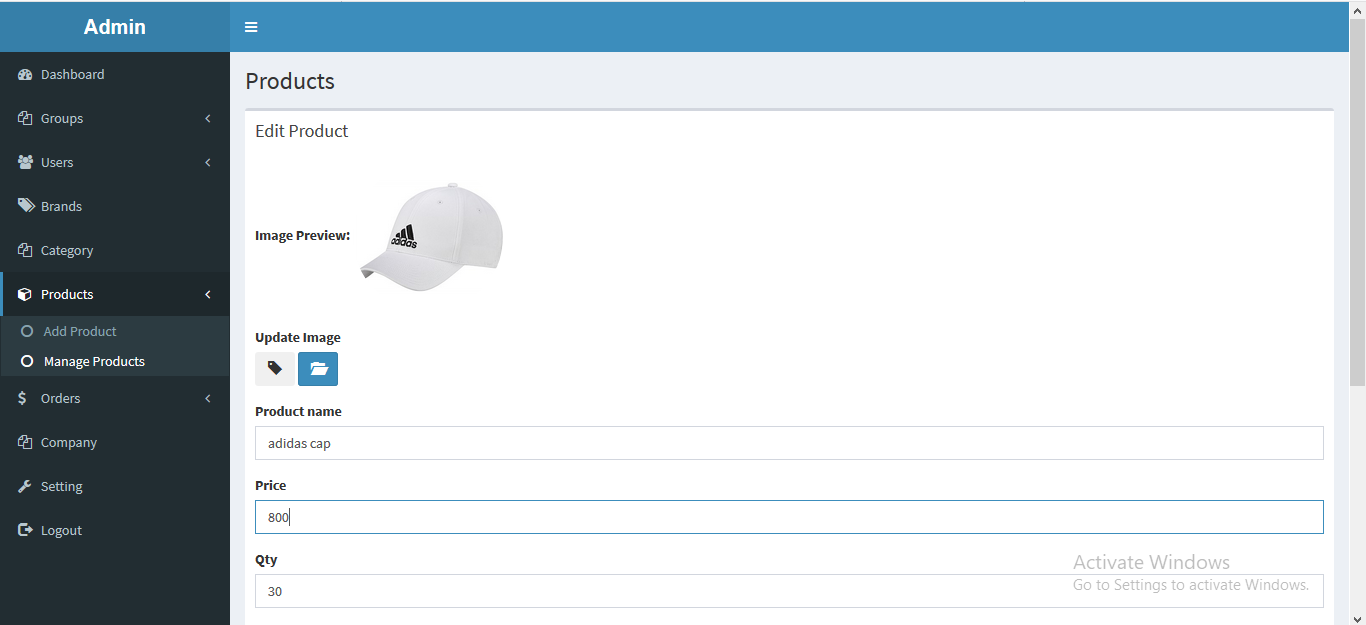
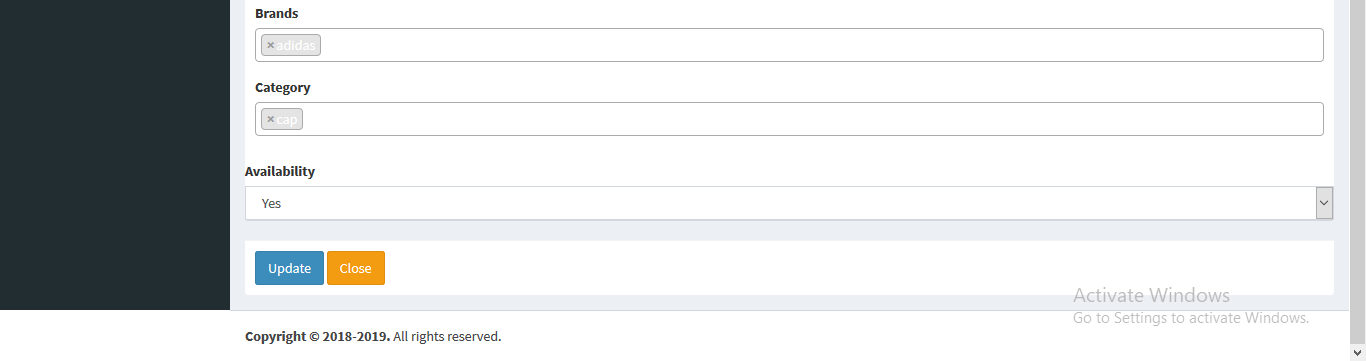
* Enter the valid email and password in login form.
* Click the product on the dashboard.
* Click manage products on drop down menu and there display all the products.
* On the top right side, there is the search box and enter the search string.
* Appears the search result as show in the figure above.

#### Deleting the product



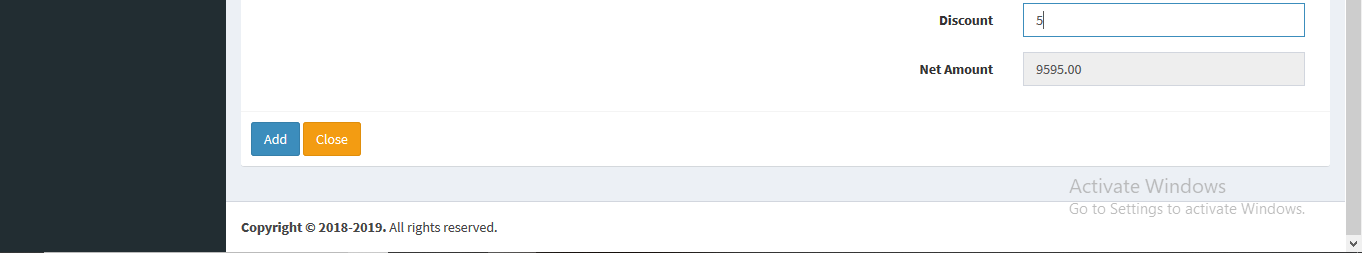
* Enter the valid email and password.
* Click product on the dashboard.
* Click the manage products on dropdown menu.
* There display all the products and click on the delete symbol which is in the action field.
* Here appears the message box with ‘do you really want to delete?’
* Click on the yes button to delete the product.

#### Updating the product

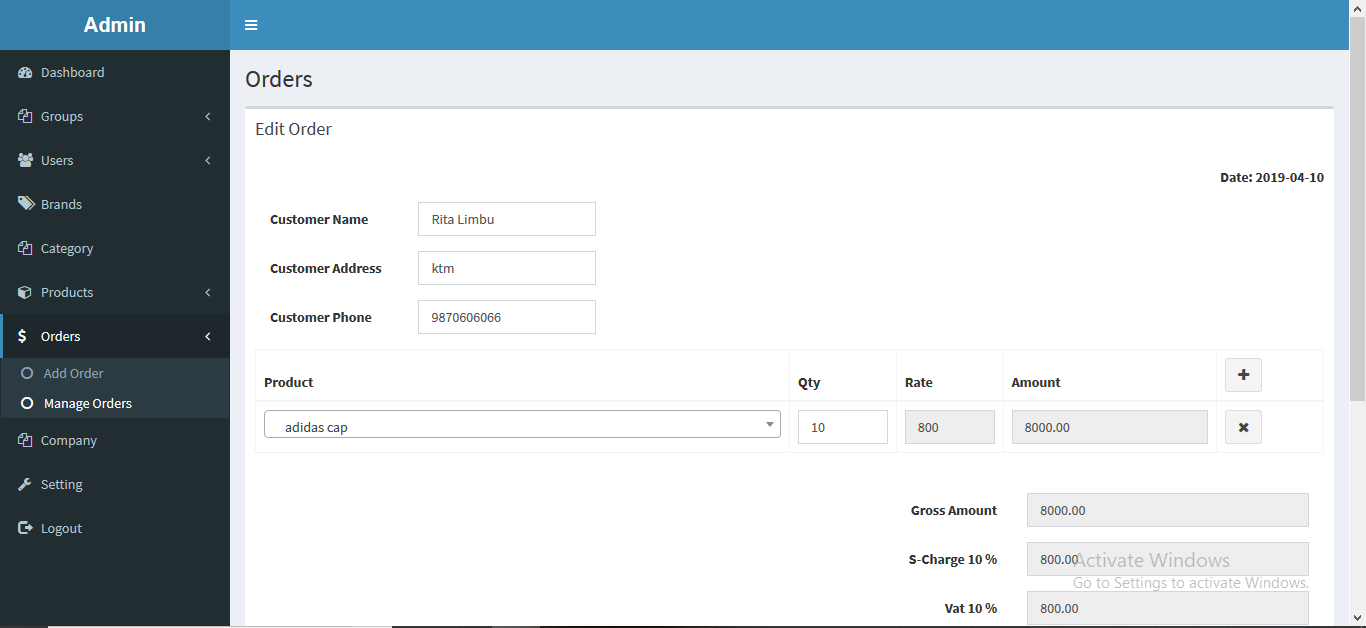
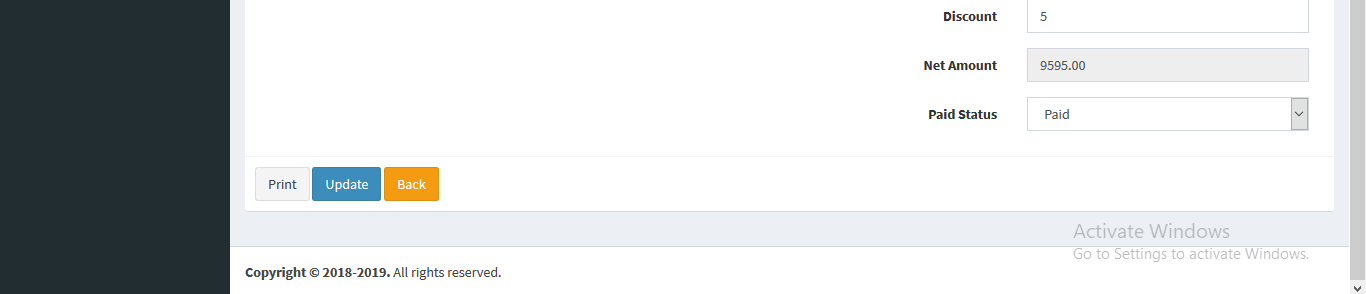
* Enter the valid email and password.
* Click product on the dashboard.
* Click the manage products on dropdown menu.
* There display all the products and click on the edit symbol which is in the action field.
* Update the record of the product and click the update button which is given in the figure above.

#### Adding the order

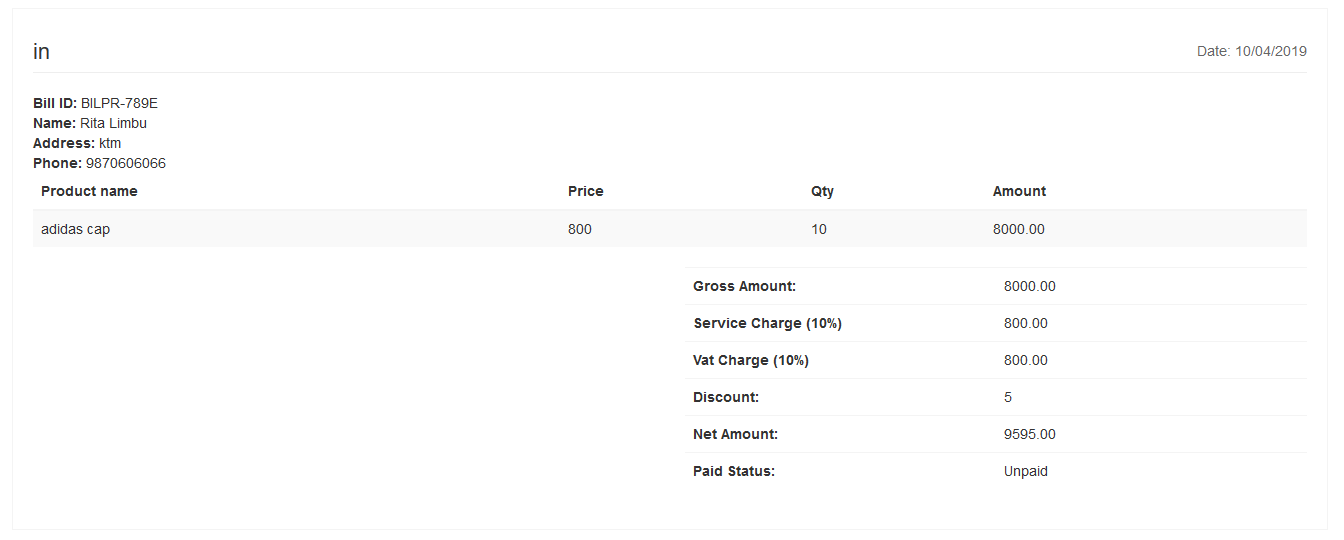
* Enter the valid email and password.
* Click orders on the dashboard.
* Click the add orders on dropdown menu.
* Enter the order.
* Click the add button to add it.

#### Print the order

* Enter the valid email and password.
* Click orders on the dashboard.
* Click the manage orders on dropdown menu.
* There display all the orders and click on the edit symbol which is in the action field.
* Click the print button to print it.

#### Bill of the product



* Enter the valid email and password.
* Click orders on the dashboard.
* Click the manage orders on dropdown menu.
* There display all the orders and click on the edit symbol which is in the action field.
* Click the print button to print it.
* After clicking the print here is the bill of the product which is shown in the figure above.

### Limitation

Not all the projects are perfect as they carry some flaws. The stock management system is the web based project. It is made up of using different method, diagrams and time box. As it is made by using the waterfall model so changes in any stage is impossible. The stage should complete it completely without any problem because after entering another stage the change is like impossible. If want to change then firstly should complete all the stages and start from the top again. So, it’s going to be costly and time consuming. The admin only has right to run the system and outsider cannot run it. Outsider can run it but needs the permission of admin.

# Chapter 7

## Conclusion

Finally the project is completed for the stock management system. The project includes the introduction, analysis, design, implementation, testing and other project issues. In the introduction, there is the background of the system, aims, and objectives and over view of the project. It is describe in the details in the chapter-1.

As well in the analysis, requirement of the project is defined. In this phase there is also the requirement prioritization which is made with the help of MoSoCoW prioritization. There are also the function and non-function requirements. NLA, Use case diagram, architecture and initial class diagram are also in it which is in the chapter-2.

In the design also there are the diagrams which are the sequence diagram and activity diagram of dynamic modeling, class diagram and context diagram of the structural diagram and er-diagram and data dictionary of database modeling of this project. There is the clear description of these diagrams with justification for choosing it which is in the chapter-3.

In this way in the implementation, there is the implementation of project in the programming language. So, I used the php programming language with the CodeIgniter framework. It is clearly described in the chapter-4.

Likewise in the testing, there is the test of the project by using the different type of the testing. I used the black box and the unit box testing to test the system. The tests are in the chapter-5.

Lastly in the other project issues, there is the schedule/Gantt chart of this project. Future work, risk management and configuration management are also included which is in the chapter-6.The references and appendix containing all the test results, code and user manual are in the chapter-8 and 9.

# Chapter 8

## References and bibliography

<https://www.tutorialspoint.com>

<https://www.laravel-vuejs.com>

<https://www.lucidchart.com>

<https://www.quora.com>

<http://benefitof.net>

<http://www.tutorialspoint.com>

[https://www.carajaclasses.com](https://www.youtube.com/redirect?v=ibEd2Tajrk8&event=video_description&redir_token=YGKZjX2ybdw8y704-Pgz62u5USZ8MTU0NjYwOTUwN0AxNTQ2NTIzMTA3&q=https%3A%2F%2Fwww.carajaclasses.com)

[http://www.lynda.com/Business-Project..](https://www.youtube.com/redirect?event=video_description&redir_token=uJnc85vm6Jgm56lsQ61FdHGmKod8MTU0NjYxMTMyN0AxNTQ2NTI0OTI3&q=http%3A%2F%2Fwww.lynda.com%2FBusiness-Project-Management-tutorials%2FProject-Management-Fundamentals%2F80780-2.html%3Futm_medium%3Dviral%26utm_source%3Dyoutube%26utm_campaign%3Dvideoupload-80780-0302&v=CQ_QfrClfR4)

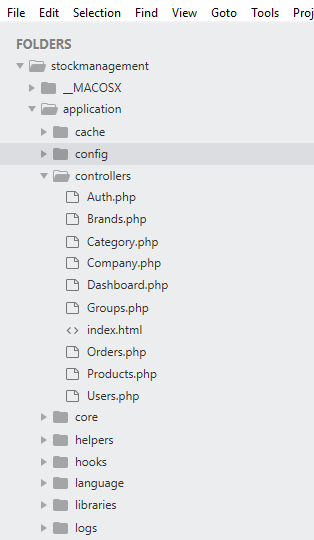
<https://en.wikipedia.org>

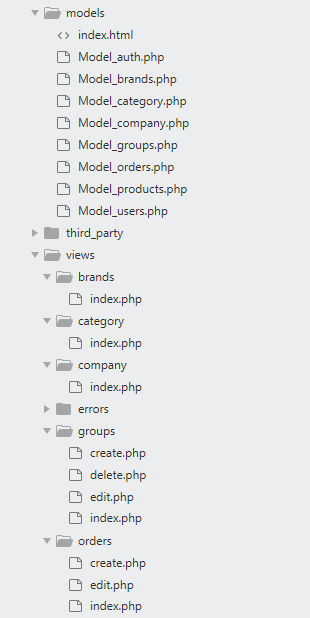
# Chapter 9

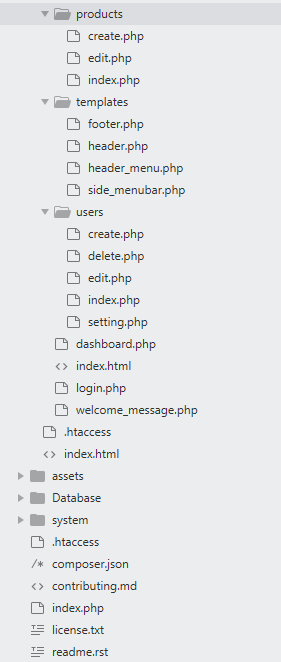
## Appendix

### Code Snippets

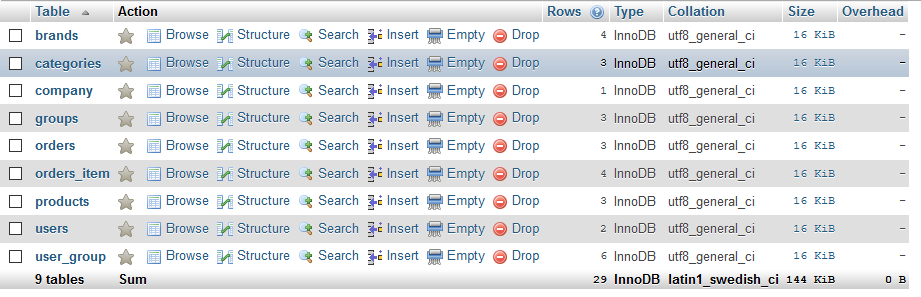
#### Source code directory structure







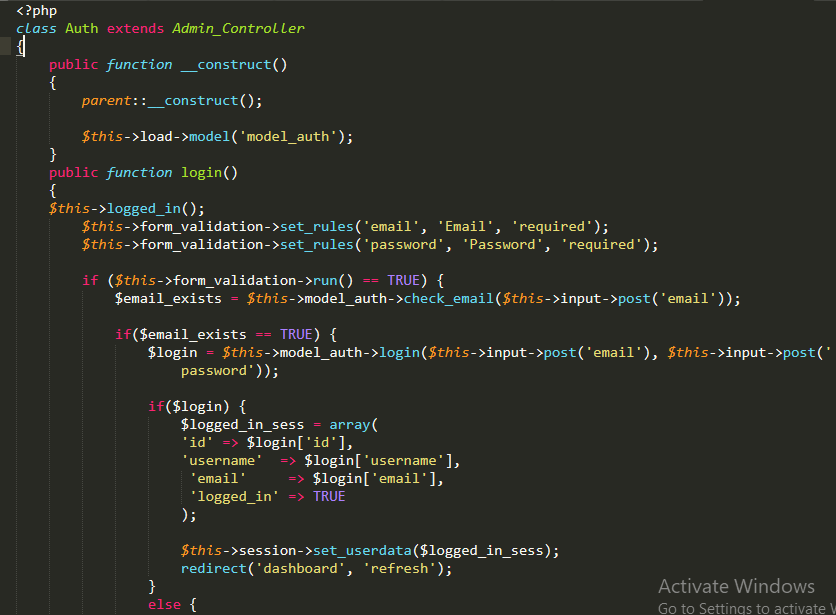
### List of tables



### Database



### Controllers\Auth.php

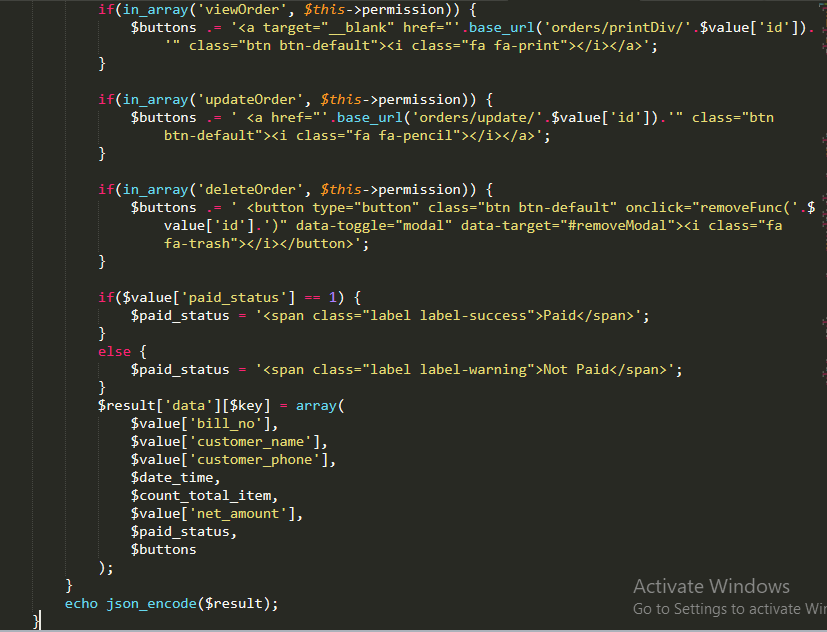


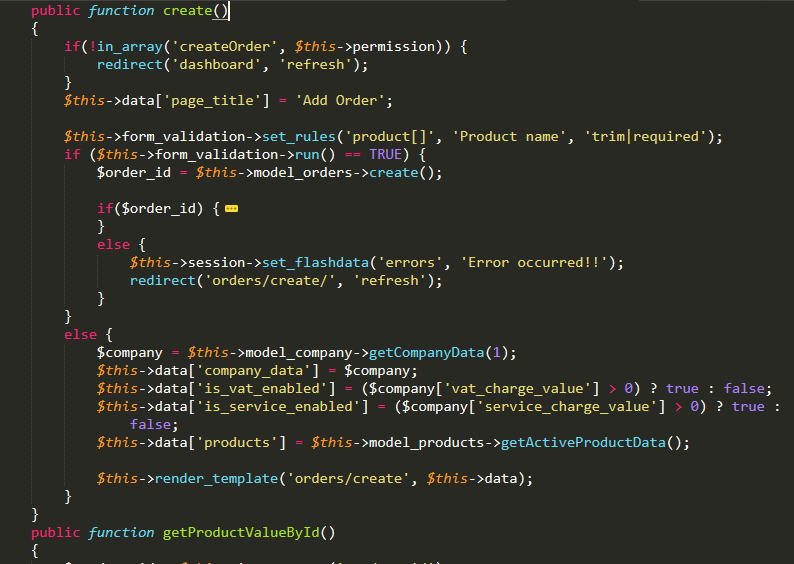
### Controllers\Dashboard.php



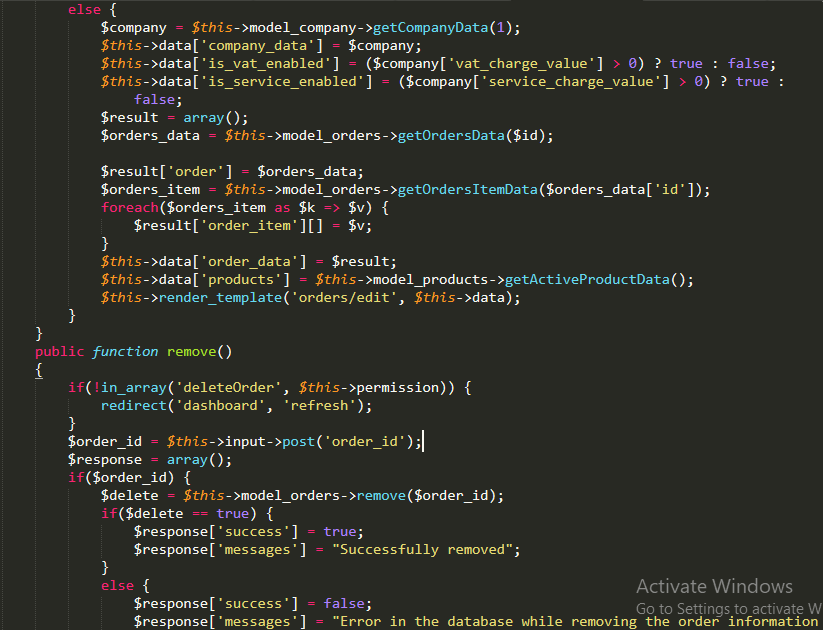
### Controllers\Orders.php



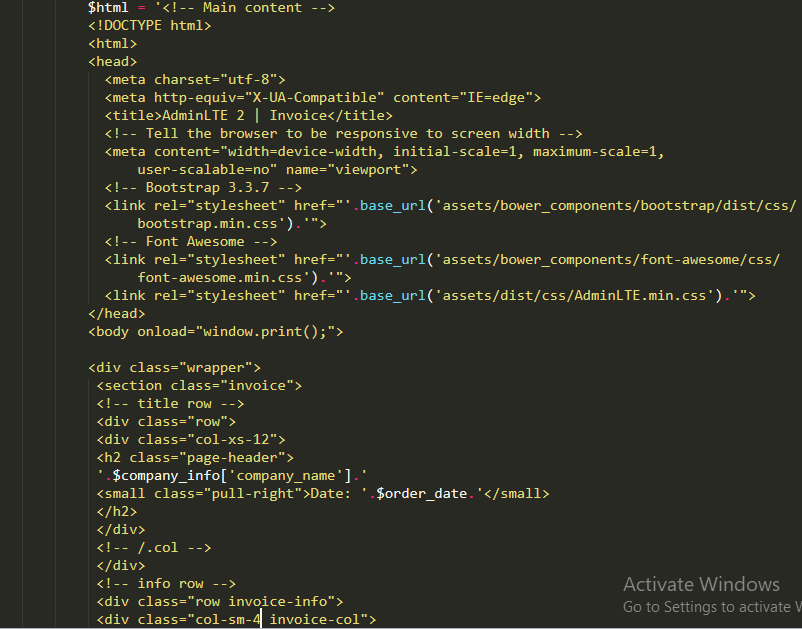




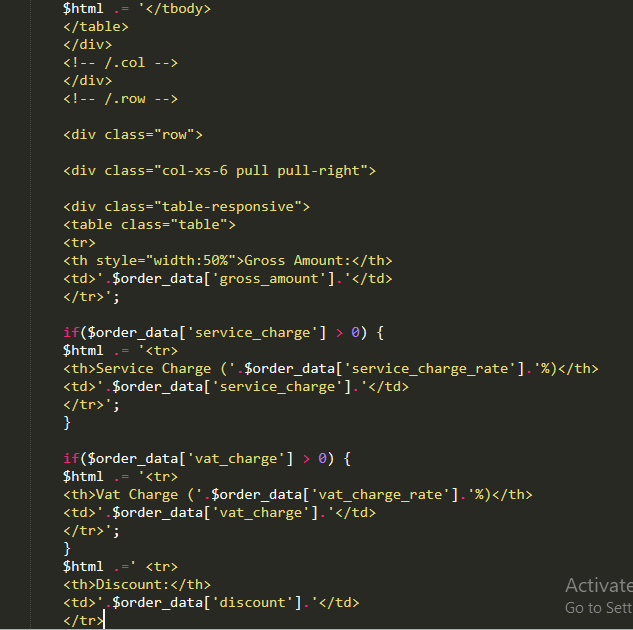




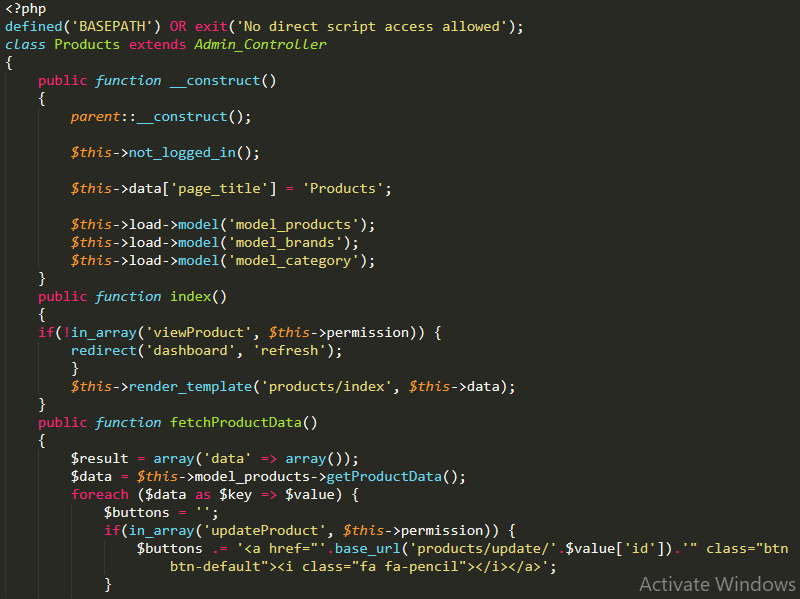




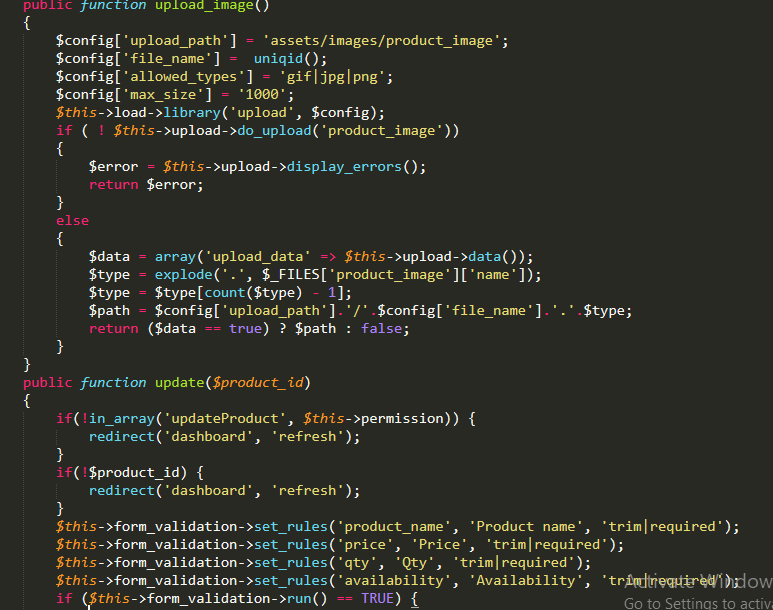




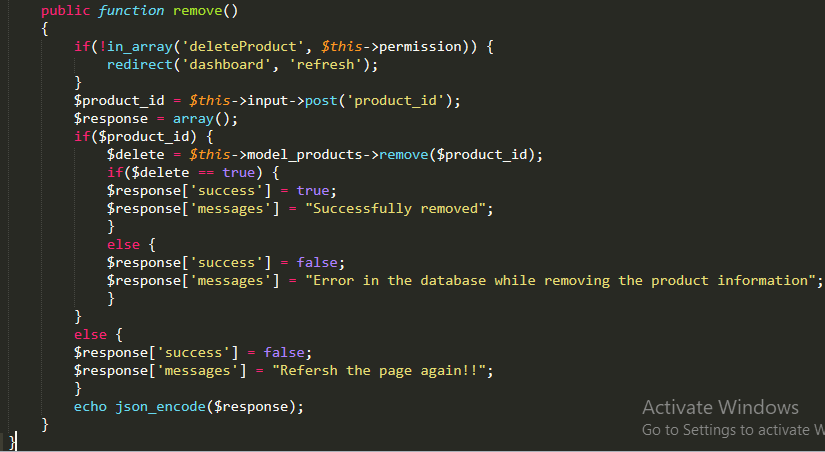
### Controllers\Products.php







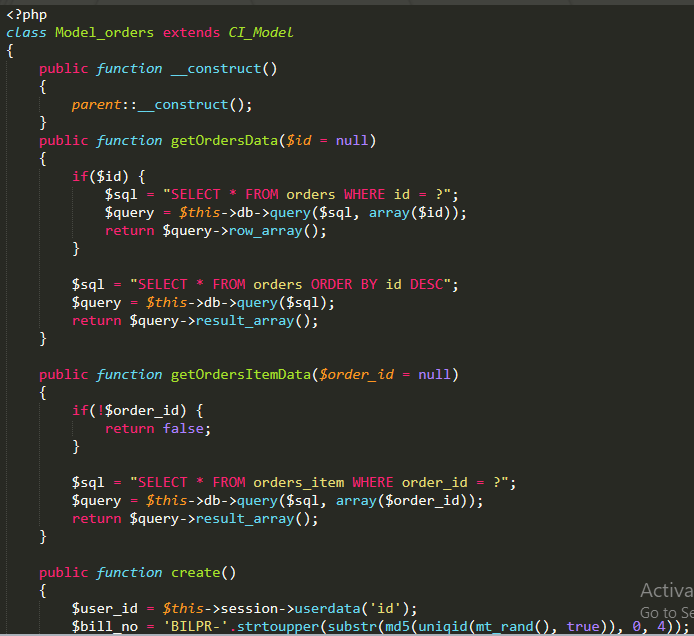


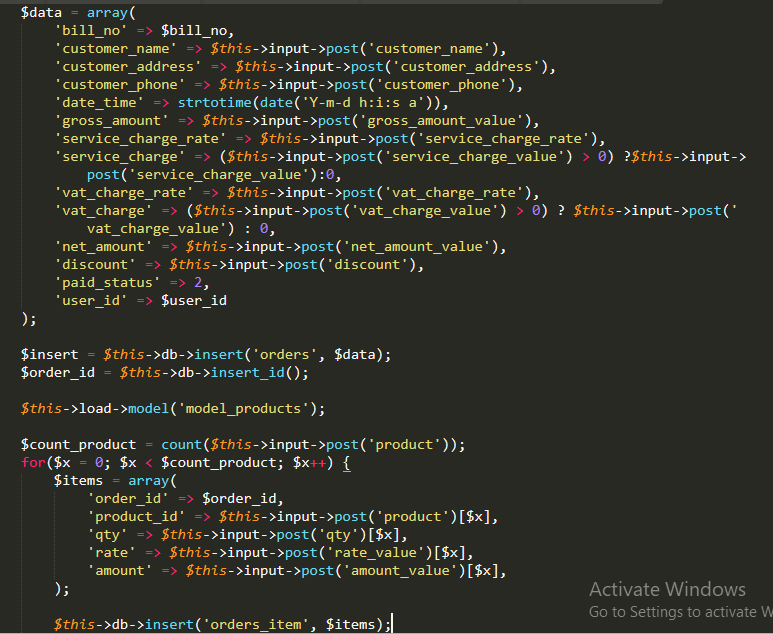


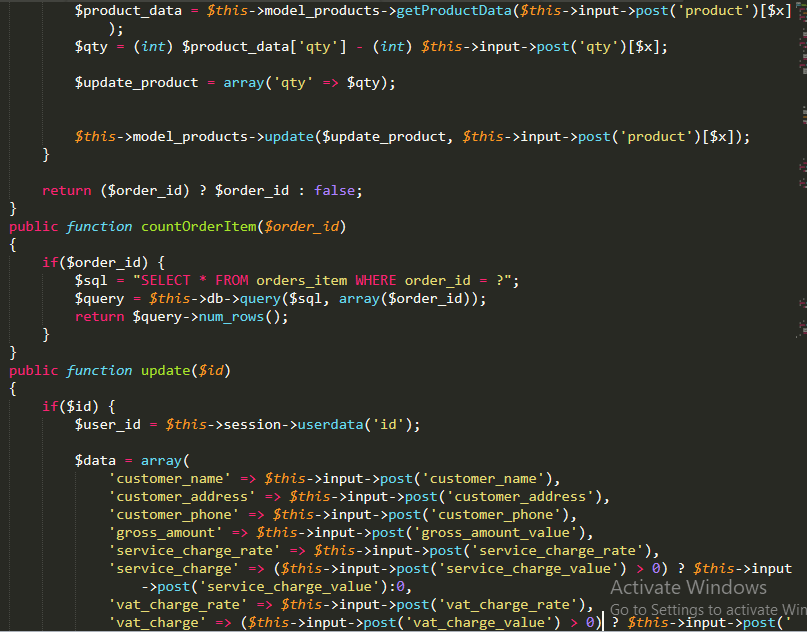
### Models\Model\_auth.php



### Models\Model\_orders.php





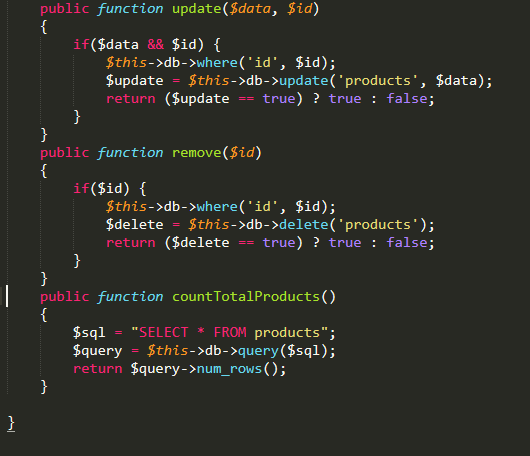




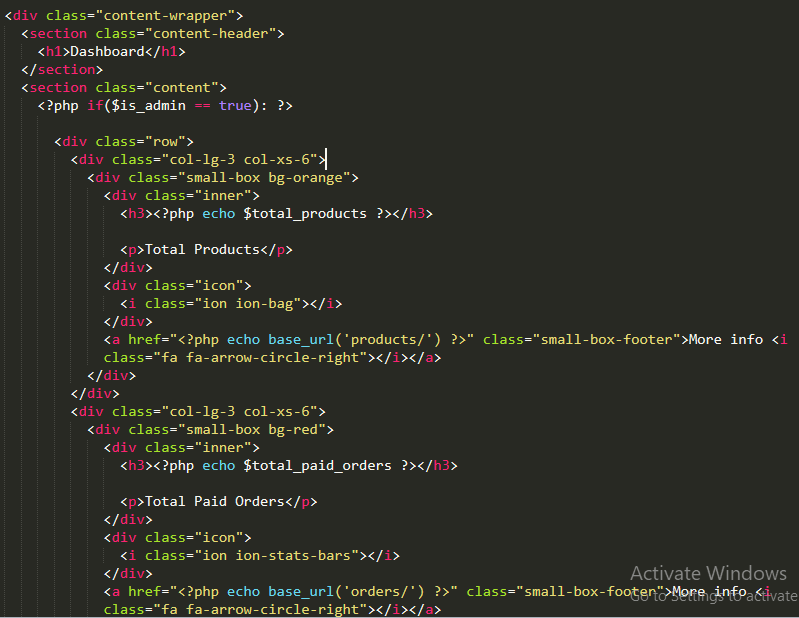


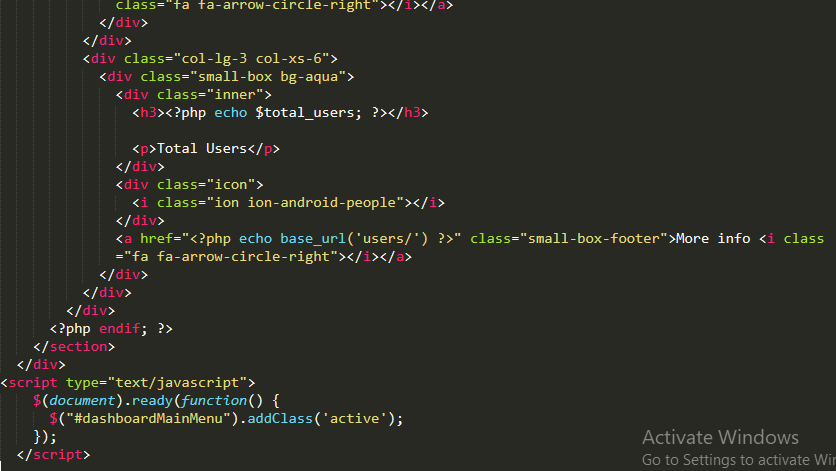
### Models\Model\_products.php





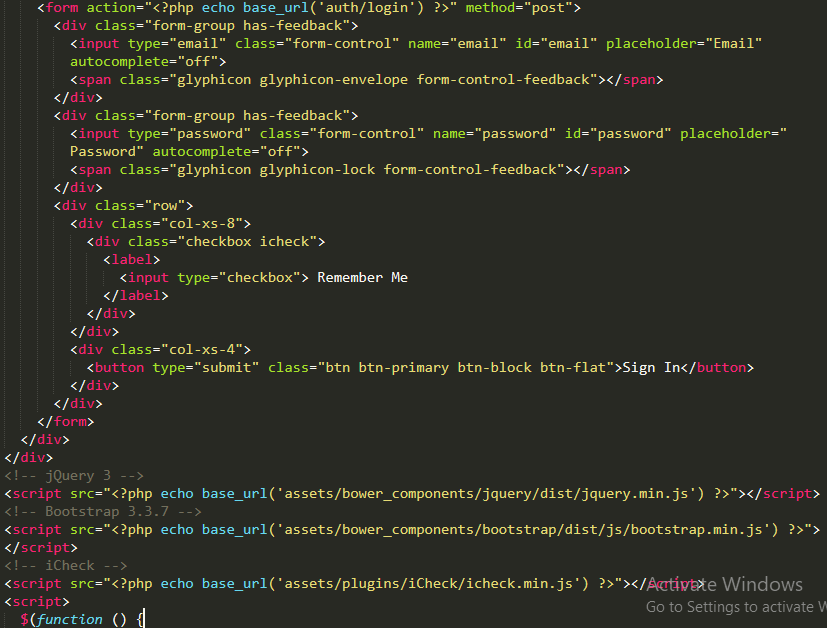
### Views\dashboard.php

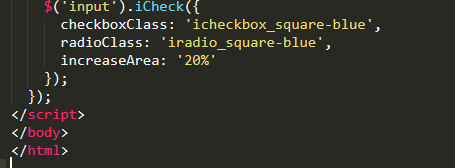




### Views\login.php

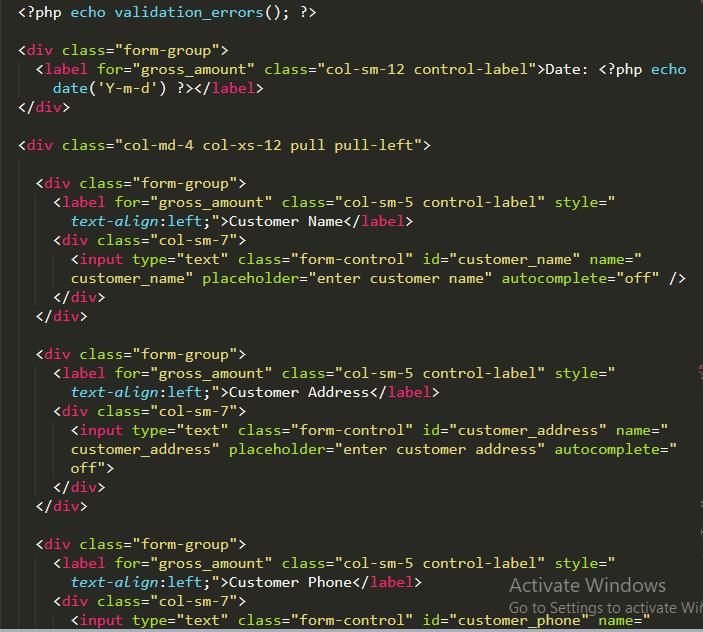


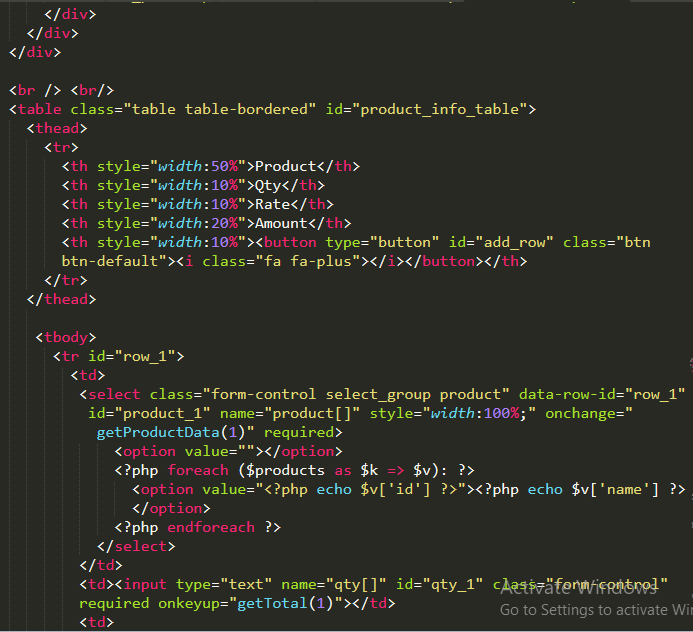


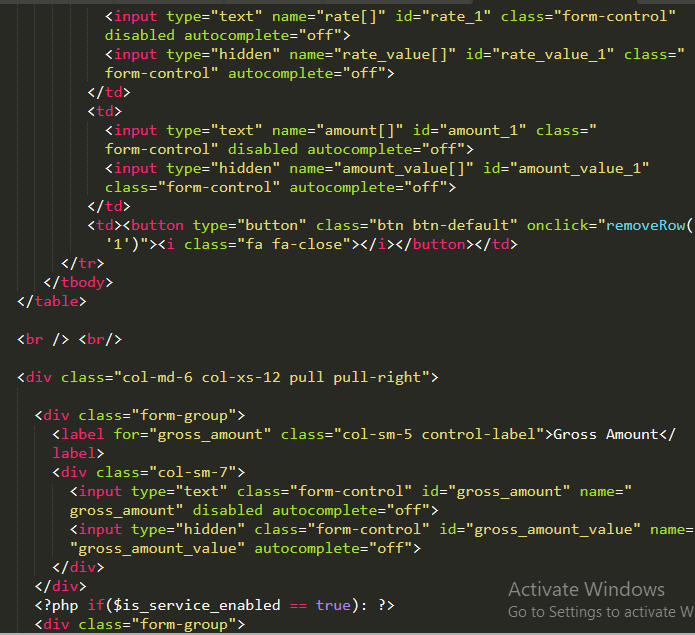


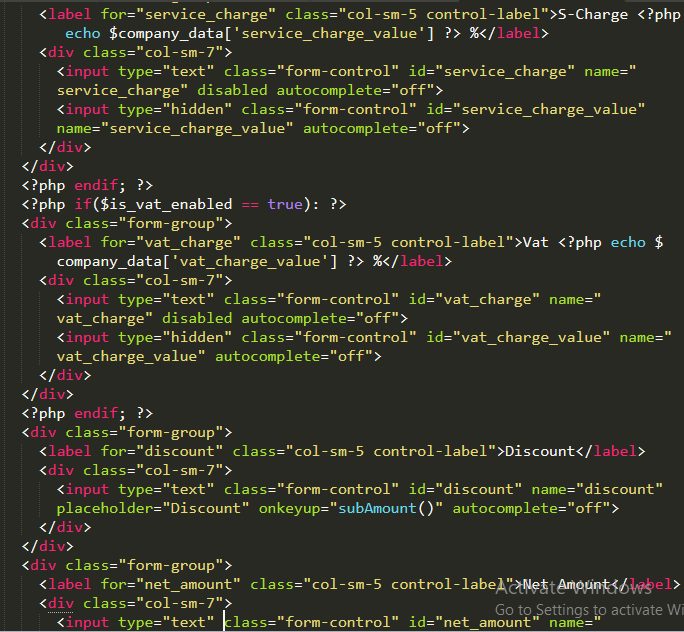
### Views\orders\create.php



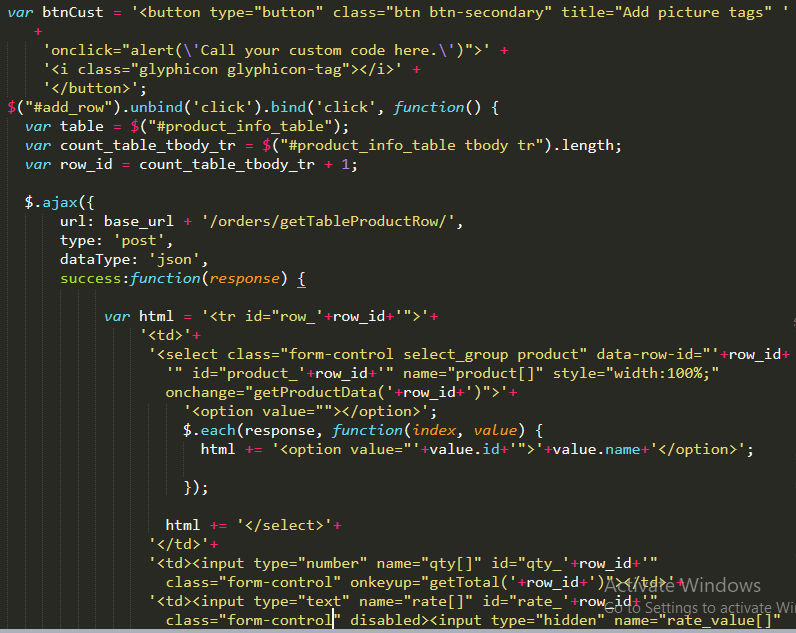




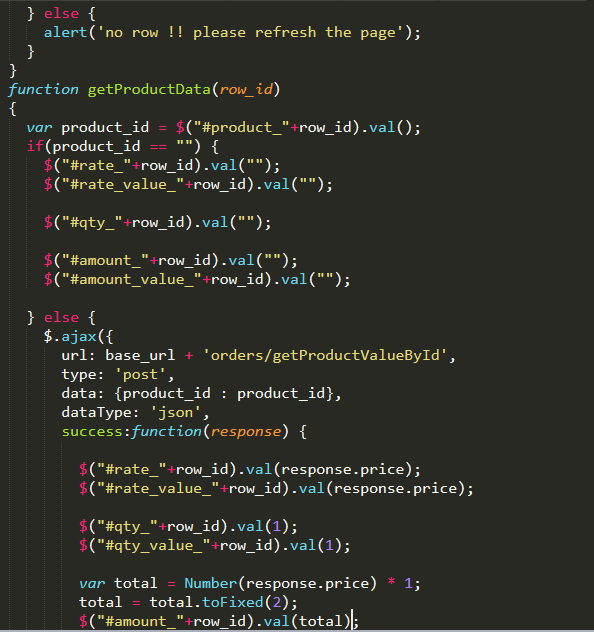


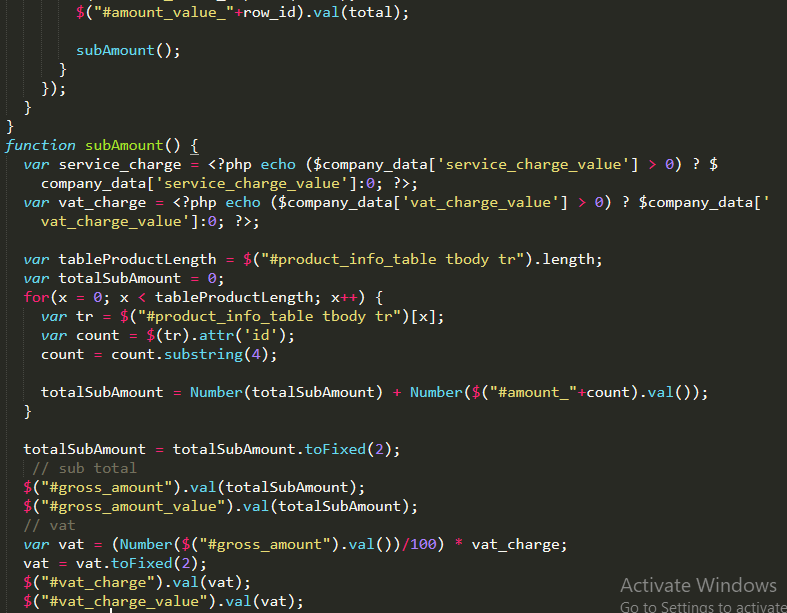








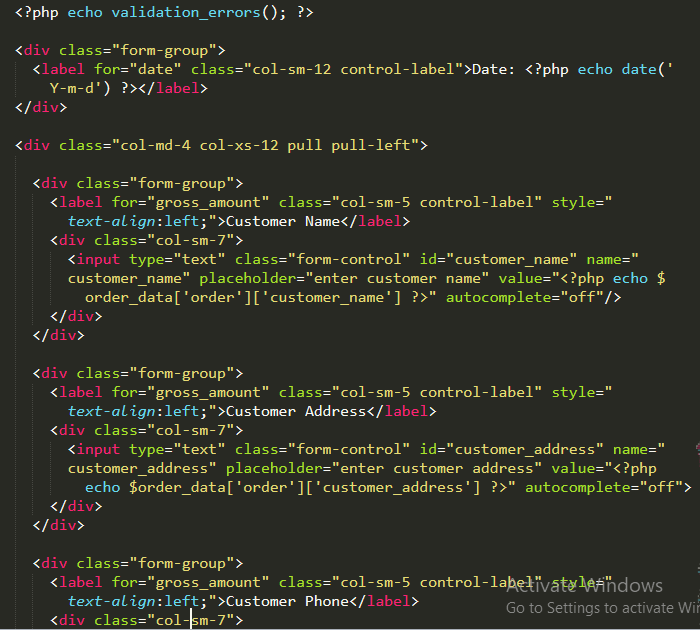






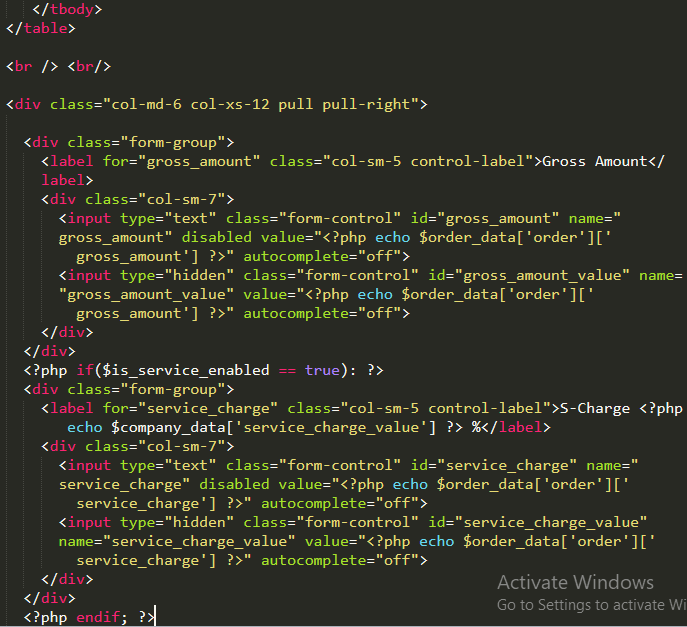
### Views\orders\edit.php



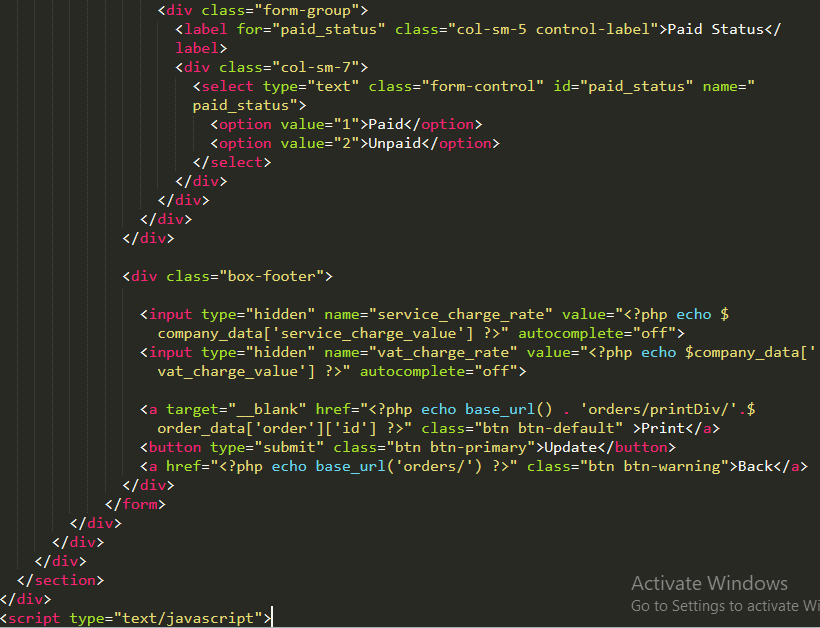








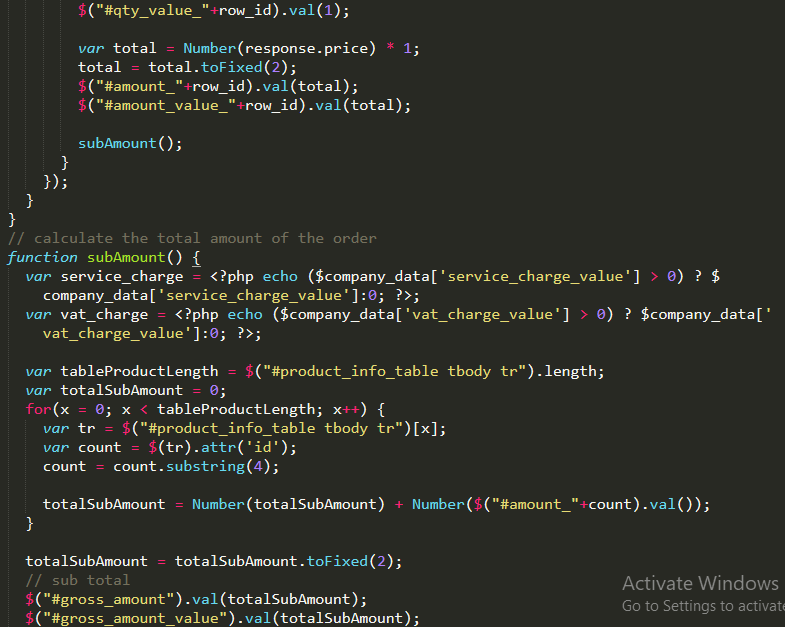




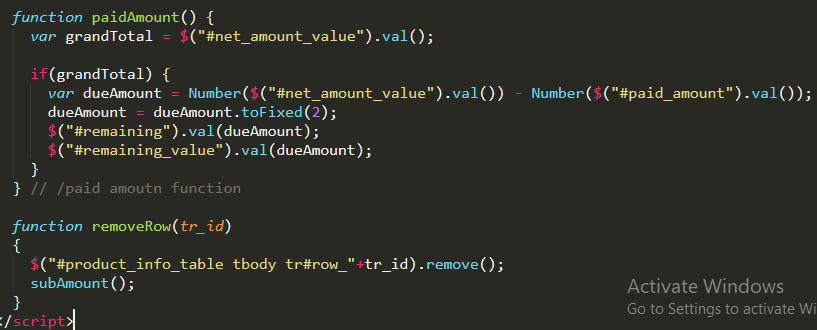






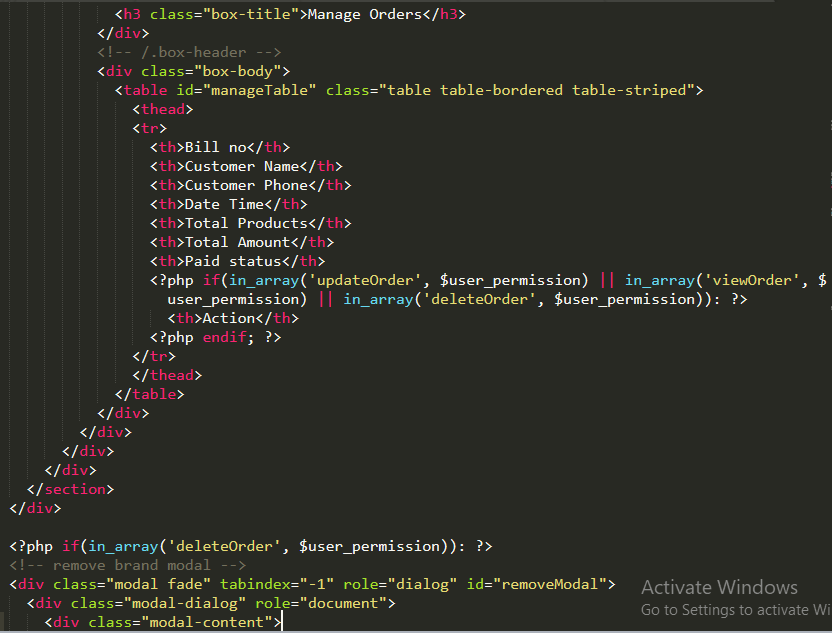






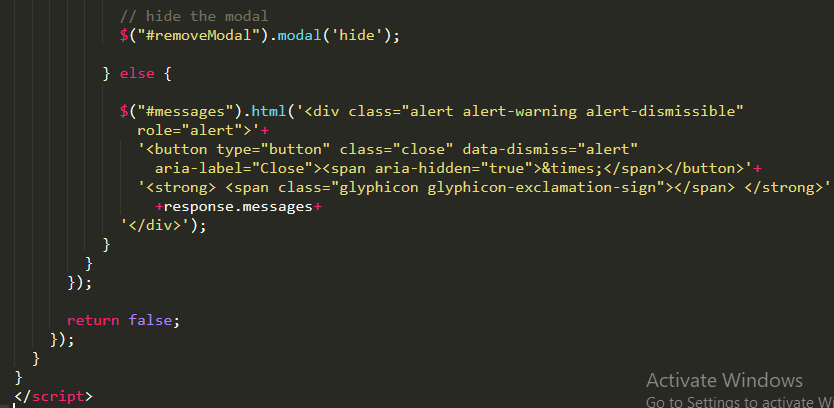
### Views\orders\index.php



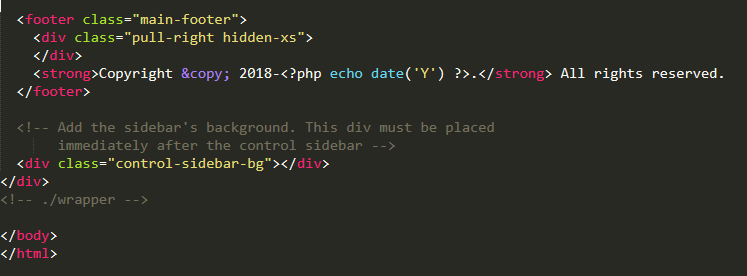




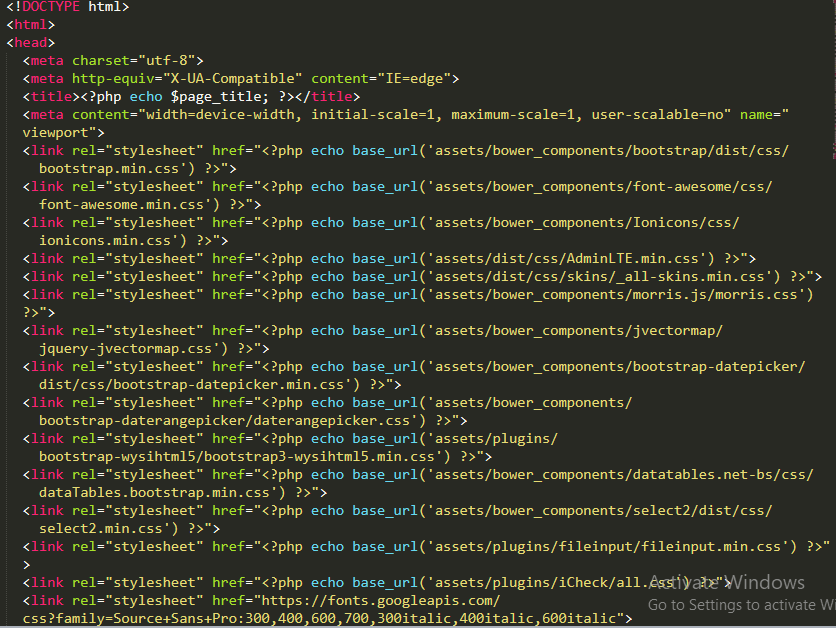


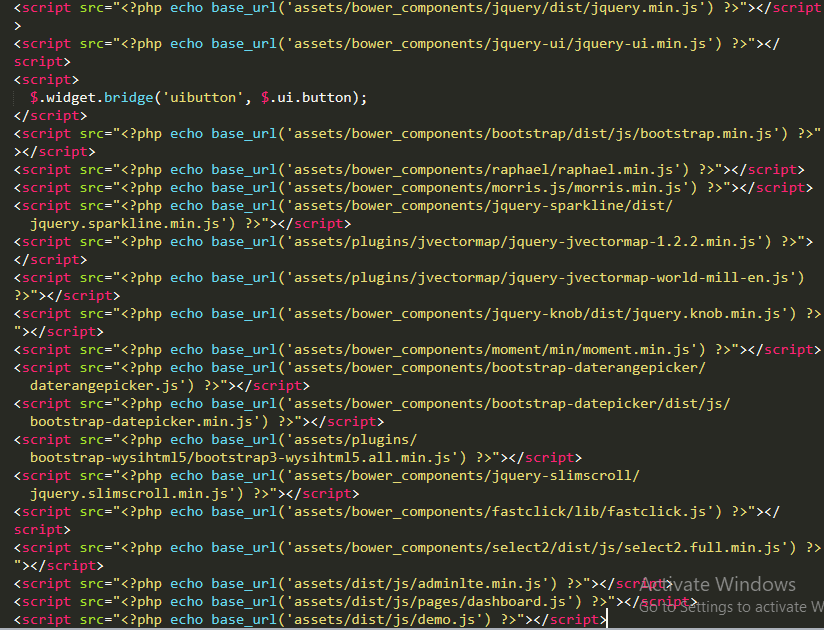


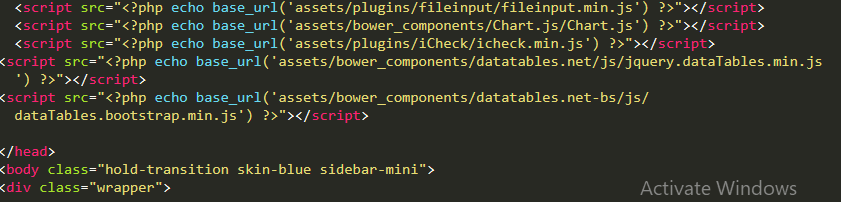
### Views\templates\footer.php



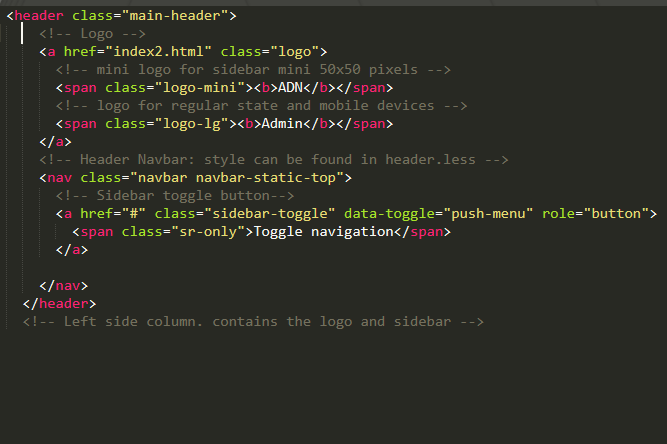
### Views\templates\header.php







### Views\templates\header\_menu.php



### Views\templates\side\_menubar.php

