# Advaita Lifestyle

**DISSERTATION**



**Submitted in partial fulfillment of the requirements for the award of the Degree of**

**Bachelor of Computer Application**

**at**

**ST ALOYSIUS COLLEGE (AUTONOMOUS)**

**Mangaluru – 575003**

**Department of Computer Science, Application & Animation**

**Shravan-174724**

**Sharath B C-174729**

**Sam Manasseh–174770**

**Work carried out at**

**Advaita lifestyle**

**238,1st floor, 4th Cross, 4th main,**

**Avalahalli BDA extension, BSK 3rd**

**Stage,Bengaluru-560085**

**during the academic year**

**2019-2020**

**Internal Guide External Guide**

MrYathish Ravindra Rao MrManoj Kumar

Department of BCA Advaita Lifestyle,

St Aloysius College (Autonomous) 238,1st floor, 4th Cross, ,Mangaluru – 575 003 Avalahalli, BDA extension, BSK 3rd Stage,Bengaluru-560085

 **ST ALOYSIUS COLLEGE (AUTONOMOUS),**

**Mangaluru – 575003**

**Department of Computer Science, Applications & Animation**

**Certificate for the approval of project report**

This is to certify that the following students of VI Semester BCA have satisfactorily completed the project “*Advaita Lifestyle ”* for the Bachelor of Computer Application prescribed by the college during the academic year 2019 – 20.

Shravan - 174724

Sharath B C - 174729

Sam Manasseh – 174770

**Project Guide:**

MrYathish Ravindra Rao Dr K Ravindra Swami

Department of BCA HOD, Department of BCA

St Aloysius College (Autonomous) St Aloysius College (Autonomous)

Mangaluru – 575 003 Mangaluru – 575 003

Rev. Dr PraveenMartis SJ Examiners:

Principal, 1.

St Aloysius College (Autonomous),

Mangaluru – 575003 2.



**ST ALOYSIUS COLLEGE (AUTONOMOUS),**

**Mangaluru – 575003**

**Department of Computer Science, Applications & Animation**

**Certificate for the approval of project report**

This is to certify that the SAM MANASSEH,174770. of VI Semester BCA has satisfactorily completed the project “ *Advaita Lifestyle”* for the Bachelor of Computer Application Prescribed by the college during the academic year 2017 – 18.

**Project Guide:**

MrYathish Ravindra Rao Dr K Ravindra Swami

Department of BCA, HOD, Department of BCA

St Aloysius College (Autonomous), St Aloysius College (Autonomous),

Mangaluru – 575 003 Mangaluru – 575 003

Rev. Dr Praveen Martis SJ Examiners:

Principal, 1.

St Aloysius College (Autonomous),

Mangaluru – 575003 2.

**DECLARATION**

We hereby declare that this project work entitled “*Advaita lifestyle ”* has been prepared by us during the academic year 2019 – 20 under the guidance of *Mr.Yathish Ravindra Rao*, Department of Computer Science, Application & Animation, St Aloysius College(autonomous), Mangaluru in the partial fulfillment of BCA degree prescribed by the College.

We also declare that this project is the outcome of our own efforts, that it has not been submitted to any other university for the award of any degree and diploma.

Team Members Signature

1. Shravan - 174724
2. Sharath B C – 174729
3. Sam Manasseh – 174770

**DECLARATION**

I hereby declare that this project work entitled “*Advaita lifestyle”* has been prepared by me during the year 2019 – 2020 under the guidance of Mr.Yathish Ravindra Rao, Department of Computer Science, Application and Animation, St Aloysius College(Autonomous) in the partial fulfillment of BCA degree prescribed by the college.

I also declare that this project is the outcome of my own effort, that it has not been submitted to any other university for the award of any degree and diploma.

Student Name : SAM MANASSEH Signature

Register Number: 174770

**ACKNOLWEDGEMENT**

We present our project entitled “ADVAITA LIFESTYLE” with great pleasure and satisfaction. We would like to thank the people who have supported us during the course of our project .We indeed indebted To Principal Rev. Dr Praveen Martis SJ. To Mr Ashok M. Prasad (Dean) Department of Computer Science, Application & Animation, Dr K Ravindra Swami, HOD, Department of Computer Science, Application & Animation, To Yathish Ravindra Rao Department of Computer Science, Application & Animation,To Manoj Kumar advata life style ,Bangalore

**Team Members**

1. Shravan - 174724
2. Sharath B C – 174729
3. Sam Manasseh – 174770

**ACKNOLWEDGEMENT**

I present my project entitled “ADVAITA LIFESTYLE” with great pleasure and satisfaction. I would like to thank the people who have supported us during the course of my project .We indeed indebted To Principal Rev. Dr Praveen Martis SJ. To Mr Ashok M. Prasad (Dean) Department of Computer Science, Application & Animation, Dr K Ravindra Swami, HOD, Department of Computer Science, Application & Animation, To Yathish Ravindra Rao Department of Computer Science, Application & Animation,To Manoj Kumar advata life style ,Bangalore

Name: SAM MANASSEH

Reg. No:174770

**TABLE OF CONTENTS**

1. **INTRODUCTION 1**

1.1 OVERVIEW OF THE PROJECT 2

1.2 OBJECTIVE OF THE PROJECT 2

1.3 PROJECT CATEGORY 2

1.4 TOOLS AND PLATFORM TO BE USED 2

1.5 OVERVIEW OF THE TECHNOLOGIES USED 2

1.5.1 Hardware Requirements 2

1.5.2 Software Requirements 2

1.6 ORGANIZATION PROFILE 3

1.7 STRUCTURE OF THE PROGRAM 3

1.8 STATEMENT OF THE PROBLEM 3

1. **SOFTWARE REQUIREMENTS SPECIFICATION 4**

2.1 INTRODUCTION 5

2.1.1 Purpose 5

2.1.2 Scope of the project 5

2.1.3 Intended audience and Reading Suggestions 5

2.1.4 Definitions, Acronyms and Abbreviations 5

2.1.5 References 5

2.1.6 Overview 5

2.2 OVERALL DESCRIPTION 6

2.2.1 Product Perspective 6

2.2.2 Product Features 6

2.2.3 User Characteristics 7

2.3 OPERATING ENVIRONMENT 7

2.3.1 Design and Implementation Constraints 7

2.3.2General Constraints 7

2.3.3 Assumptions and Dependencies 7

2.4 SPECIFIC REQUIREMENTS 8

2.4.1 External Interface Requirements 8

2.4.1.1User Interface 8

2.4.1.2 Hardware Interface 8

2.4.1.3 Software Interface 8

2.4.1.4 Communication Interface 9

2.4.2 Functional Requirements 9

2.4.3 Performance Requirements 9

2.4.4 Design Constraints 9

2.4.5 Other Requirements 9

1. **SYSTEM ANALYSIS AND DESIGN 10**

3.1 INTRODUCTION 11

3.2 DATA FLOW DIAGRAM 11

3.3 DATA BASE DESIGN 16

3.3.1 Entity-Relationship Diagram 17

3.3.2 Table Relationship 23

3.3.3 Table Description 24

3.4SYSTEM DESIGN IMPLEMENTATION 26

3.4.1 Use case 26

3.4.2 Activity Diagram 27

3.4.3 Sequence Diagram 28

3.5 USER INTERFACE DESIGN 30

1. **TESTING 40**

4.1 INTRODUCTION 41

4.2 TESTING OBJECTIVE 42

4.2.1 Unit Testing 43

4.2.2 IntegrationTesting 43

4.3 TEST CASES 43

4.3.1 User Login 43

4.3.2 User Registetration 44

4.3.3 Product Detail 45

4.3.4 Shopping Cart 45

4.3.5 My order 46

4.3.6 Payment 46

4.3.7 Admin login 47

4.3.8 Admin dash board 48

4.3.9 Product 49

4.3.10 Product Category 49

4.3.11 View Order 50

4.3.12 View Customer 51

4.3.13 Coupon 51

4.3.14 View Coupon 52

4.3.15 Send Coupon 52

**5. SYSTEM SECURITY** 54

5.1 INTRODUCTION 54

5.2 SOFTWARE SECURITY 54

**6 CONCLUSION** 56

**7. FUTURE ENCHACHMENT**  58

**8.BIBLOGRAPY**  60

**INTRODUCTION**

**1 INTRODUCTION**

**1.1 OVERVIEW OF THE PROJECT**

Advaita lifestyle project documentations contains the functional, non-functional requirement of the software, system design and implementation are also added in this document. The document contains the testing report, system security report of the software

**1.2 OBJECTIVE OF THE PROJECT**

Our project is a website for buying the herbal products like Soaps, Shampoos, Oils, Cream etc online. User can view the product price, description etc and can add those products to the cart, buy the product online and user also can add a coupon code to avail the special offer. Admin can manage shopping system by adding products, view the products bought by user, manage the stock etc. User can also add the buying products to the cart and buy them.

**1.3 PROJECT CATEGORY**

The project category is Web-based.

* 1. **TOOLS AND PLATFORM TO BE USED**

Operating system:windows 10

Editor: sublime text 3

Xampp with php version 7.2 and apache 2

**1.5 OVERVIEW OF THE TECHNOLOGIES USED**

**1.5.1Hardware Requirements**

Ram:3gb or above

Processor:Clock Cycle 1Ghz

**1.5.2Software Requirements**

Operating system: windows 7 or above

PHP version 7.2

Server:apache 2

Web browser: chrome, fire fox, internet explore

**1.6 ORGANIZATION PROFILE**

A L (Advaita Lifestyle) is consumer goods company which is established in 2018, for high quality products which are safe for human health and the environment.

Modern day consumerism is full of unfortunate dualities like ‘business interests v/s human well-being’ and ‘human desires v/s planet ecosystem’. Consumers constantly seek low prices which creates a demand force for cheaper goods. Businesses in an attempt to maximize profit, supply to this demand by producing goods made of cheaper ingredients. This game of supply-demand is blind to the adverse effects on human health and the environment.

We, at A L advocate for a balanced approach to deal with these duality. The word ‘advaita’ translates to ‘non-dual’. We promote a non-dual lifestyle. We aim to produce high quality affordable consumer goods with harmless ingredients and we chose to educate consumers about adverse effects of low quality ingredients to battle the supply-demand market forces instead of surrendering to the demand for cheaper goods. That’s how we live lead our non-dual lifestyle, AL – Advaita Lifestyle.

**1.7 STRUCTURE OF THE PROGRAM**

Admin module: The admin is responsible for uploading the database. He can add the products and make any kind of modifications. And, he can mage the orders of the customer

Customer module: The customer can login or register to the system . He can add product to cart, use coupon code to purchase item .

Product module: Product can be added to the database based on category. admin can change the price or can delete the item from the database.

Shopping module: Customer can add the product to cart, proced to checkout, can use coupon code to get additional discount, then the payment can be done online.

Order module: this module contains order details such as order date,address of order to be dispatched, customer details, payment details, order status weather it pending or delivered, payment status etc.

**1.8 STATEMENT OF THE PROBLEM**

**SOFTWARE REQUIREMENTS SPECIFICATION**

**2 SOFTWARE REQUIREMENTS SPECIFICATION**

**2.1 INTRODUCTION**

**2.1.1 Purpose**

Defining and describing the functions and specifications of the Advaita life style online store is the primary goal of this Software Requirements Specification (SRS). This Software Requirements Specification illustrates, in clear terms, the system’s primary uses and required functionality as specified by our customer.

**2.1.2 Scope of the project**

The software system being produced is called Advaita life style online store. It is being produced for a customer interested in selling herbal product via the Internet. This system is designed to “provide automation support” for the process of placing herbal product for sale on the Internet and facilitating the actual sale. This system is available for all using Internet. The system will be run on a web browser such as chrome,fire fox,internet explorer etc

**2.1.3 Intended Audience and Reading Suggestions**

The intended readers of the documents are developers of the site, testers,website owners,managers,coordinators.

Suggested reading: Fundamentals of software engineering

**2.1.4 Definitions, Acronyms, Abbreviation**

* SRS: software requirement specification
* System: advaita life style store
* User: admin and customer of organization
* Product : item sold by the organization
* He/she/them : consumers of organization

**2.1.5 References**

<http://en.wikipedia.org/wiki/Software_Requirements_Specification>

**2.1.6 Overview**

This documentation contain the functional and non functional other requirements of the system.

**2.2 OVERALL DESCRIPTION**

**2.2.1 Product Perspective**

The website must be available to anyone using the Internet and as such must work correctly in all the browser. As stated by the customer, there are no hardware or software requirements beyond these including, but not limited to, memory or specific software packages that need to be utilized nor software packages that need not be utilized.

**2.2.2 Product Features**

Advaita Lifestyle store will provide a number of features ; each is listed below.

* Maintain data associated with the inventory (a collection of products)

A product has a category and price

The inventory also keep track of the stock/quantity of each item

* Maintain records for many customers

A customer can be either a member or non-member.

A customer has a username, password (no restrictions),email address (no restrictions), and postal address(unverified.)

Anyone may sign up for a customer account.

* Allow any customer to become a member.
* Show a listing of available item
* Allow customers and Admin to log in and out of the system.

Users (both customers and the manager) will be logged out if inactive after browser is closed .

* Shopping cart

Anyone is able to add one or more item to the shopping cart.

The shopping cart can allow multiple number of any item.

* Allow Admin to update stock details

Allow manager to change any item's price

Allow manager to view transaction logs

Payment methods

Payments can be done cash on delivery

Online transaction is available

**2.2.3 User characteristics**

The user classes are as follow

**Login as customer**

Browse

Add to cart

Add address

Buy item

Payment mode

**Login as Admin**

Check Inventory

Add new item

View customer

Data base management

The advaita life style store user is simply anyone that has access to the Internet and a web browser in the computer and hand held device. It is assumed that the user is familiar enough with a computer to operate the browser, keyboard and mouse and is capable of browsing to, from and within simple websites

**2.3 OPERATING ENVIRONMENT**

Advaita life style system that shall operate in both windows and android operating system. The website will run in any browser and in any version.

**2.3.1 Design and implementation constraints**

System shall be built using a standard web page development tools.

The computer must be equipped with web browser such as google chrome. System must be stored in such a way that allows the client easy access to it.

Response time for loading the system should take no longer than few minutes

A general knowledge of basic computers and handheld devices skills are required to use the system.

**2.3.2 General constraints**

The admin can view the customer side designs but customer are not permited to access admin side.

**2.3.3 Assumptions and dependencies**

Client:

We have assumed that most of places in country are accessible internet .The users of advaita life style products use the official website of the brand.

Provider:

We have assumed that the advaita life style will be running on a properly working web server and database system with an Internet connection that allows this system to perform all communications with clients.

**2.4 SPECIFIC REQUIREMENTS**

**2.4.1 External interface requirements**

**2.4.1.1 User interface**

The user interface for the software shall be compatible to any browser such as chrome, fire fox etc by which user can access to the system

The user interface shall be implemented using any tool like html, css, javascript.

User interface is required for the following module customer

* Login
* Home page
* Cart
* Shoping
* Payment

User interface for Admin are following

* Login
* Add/send coupans
* Add/remove product
* View customer
* View invoice

**2.4.1.2 Hardware interfaces**

Since the website must run over the internet, all the hardware shall require to connect to the internet will be hardware interface for the system.For computer the internet connection will be provided by modem and Wifi network.For mobile devices cellular network or wifi.

**2.4.1.3 Software interfaces**

The e system shall communicate with the configarator to identify all the available components to configure the product.

The system shall be verisign like software which shall allow the user to complete the secured transaction .this usually shall be the third party software system which is widely used for internet transaction.

**2.4.1.4 Communications interfaces**

The advaita life style shall use the HTTP protocol for communication over the internet and for the internet communication will be through TCP/IP protocol suite

**2.4.2 Functional requirements**

Admin module: The admin is responsible for uploading the database. He can add the products and make any kind of modifications. And, he can mage the orders of the customer

Customer module: The customer can login or register to the system . He can add product to cart, use coupon code to purchase item .

Product module: Product can be added to the database based on category. admin can change the price or can delete the item from the database.

Shopping module: Customer can add the product to cart, proced to checkout, can use coupon code to get additional discount, then the payment can be done online.

**2.4.3 Performance requirements**

The system shall be based on the web that has to be run from web server. The system shall take initial load time depending on the internet connection strength which also depends on the media from which the system is run .

The performance shall depend upon hardware component of the client/customer

**2.4.4 Design Constraints**

System shall be built using a standard web page development tools.

System must be stored in such a way that allows the client easy access to it.

Response time for loading the system should take no longer than few minutes

**2.4.5 Other Requirements**

Null

**SYSTEM ANALYSIS AND DESIGN**

**3 SYSTEM ANALYSIS AND DESIGN**

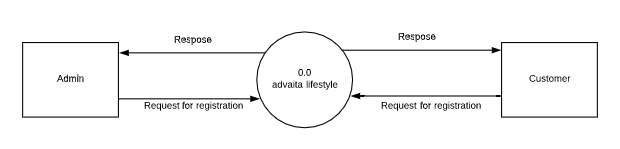
**3.1 INTRODUCTION**

System design is a process through which requirements are translated into a representation of the software. In detail design we specify how the modules in the system interact with each other. Detailed specification is given by explaining in natural language for the modules is supposed to do.

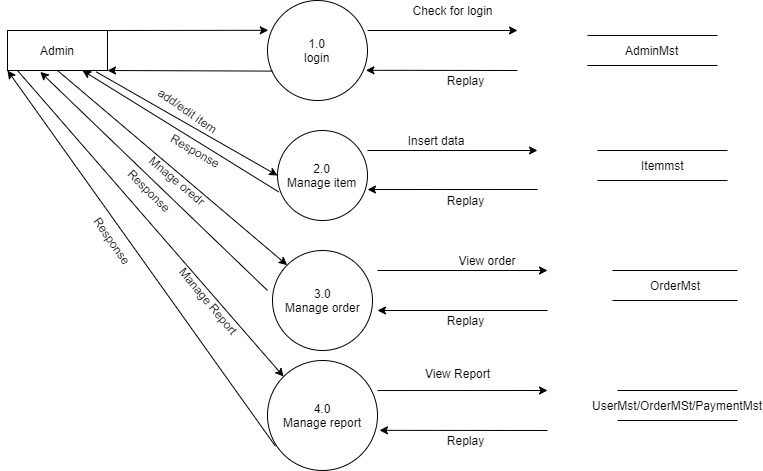
Detail design specification describes the features of the system. Detail design is the refinement of the system design that essentially expands system design to contain more detail description of the processing logic of the components and data structure such that design can be easily implemented.

**3.2 DATA FLOW DIAGRAM**

Context Flow Diagram

****

Level 1 DFD of Admin side



Level 2 DFD Admin Side (2.0**)**



Level 2 DF DFD Admin Side (3.0**)**



Level 2 DFD Admin Side (4.0**)**

****

**User side DFD**

****

**3.3 DATA BASE DESIGN**

The word “database”, used to describe everything from a single set of data, to a complex set of tools, such as SQL server, and a whole lot in between. The term data model to mean the conceptual description of the problem space. This includes the definition of entities, their attributes, and the entity constraints. The data model also includes a description of the relationships between entities and any constraints on those relationships. It is the translation of the conceptual model into a physical representation, which shall be implemented using a database management system.

The DBMS provides flexibility in the data storage and retrieval of data and production of information. It determines what type of data, is needed and how they are processed, and the operating system of the computer is responsible for placing the

data on the storage devices. A schema defines the databases and the sub schema defines the portion of the database of the database that a specific program will use.

To retrieve data from the database:

* The application program determines what data is needed and communicates the need to the DBMS.
* The DBMS system determines that the data requested are in fact stored in the database.
* The data must be defined in the sub schema.
* The DBMS system instructs the OS to locate and retrieve the data from the specific location on the disk.
* The copy of the data is given to the operating system for processing.
* A database, must be created before it can be used.

**3.3.1 Entity-Relationship Diagram**

ER diagram stands for an Entity-Relationship diagram. It is a high-level data model. This model is used to define the data elements and relationship for a specified system.

It develops a conceptual design for the database. It also develops a very simple and easy to design view of data.

In ER diagram, the database structure is portrayed as a diagram called an entity-relationship diagram.

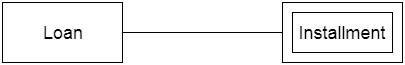
**Notations used in ER diagram:**

1. **Entity:**

An entity may be any object, class, person or place. In the ER diagram, an entity can be represented as rectangles.

**Weak Entity**

An entity that depends on another entity called a weak entity. The weak entity doesn't contain any key attribute of its own. The weak entity is represented by a double rectangle.

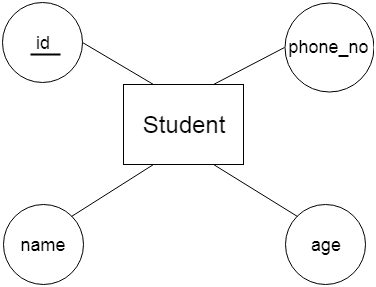


1. **Attribute**

The attribute is used to describe the property of an entity. Eclipse is used to represent an attribute.

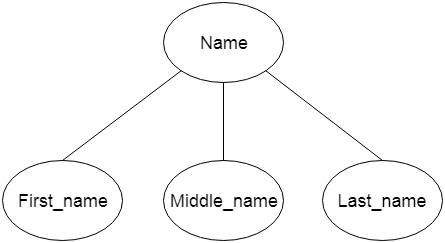
1. **Key Attribute**

The key attribute is used to represent the main characteristics of an entity. It represents a primary key. The key attribute is represented by an ellipse with the text underlined.



**b. Composite Attribute**

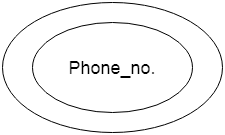
An attribute that composed of many other attributes is known as a composite attribute. The composite attribute is represented by an ellipse, and those ellipses are connected with an ellipse.



**c.Multivalued Attribute**

An attribute can have more than one value. These attributes are known as a multivalued attribute. The double oval is used to represent multivalued attribute.

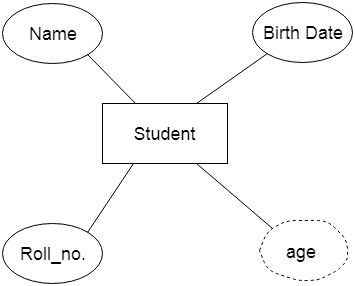
**For example,** a student can have more than one phone number.



**d. Derived Attribute**

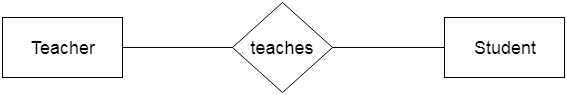
An attribute that can be derived from other attribute is known as a derived attribute. It can be represented by a dashed ellipse.

**For example,** A person's age changes over time and can be derived from another attribute like Date of birth.



**3) Relationship**

A relationship is used to describe the relation between entities. Diamond or rhombus is used to represent the relationship.

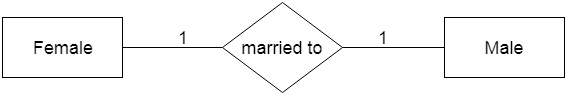


**Types of relationship are as follows:**

**a. One-to-One Relationship**

When only one instance of an entity is associated with the relationship, then it is known as one to one relationship.

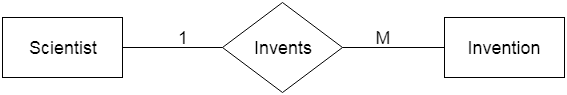
**For example,** A female can marry to one male, and a male can marry to one female.



**b. One-to-many relationship**

When only one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then this is known as a one-to-many relationship.

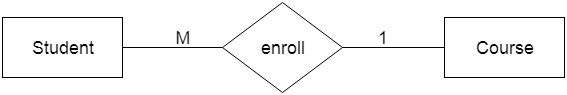
**For example,** Scientist can invent many inventions, but the invention is done by the only specific scientist.



**c. Many-to-one relationship**

When more than one instance of the entity on the left, and only one instance of an entity on the right associates with the relationship then it is known as a many-to-one relationship.

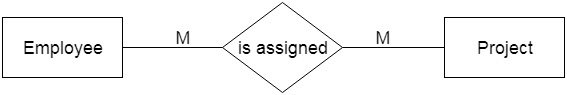
**For example,** Student enrolls for only one course, but a course can have many students.



**d. Many-to-many relationship**

When more than one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then it is known as a many-to-many relationship.

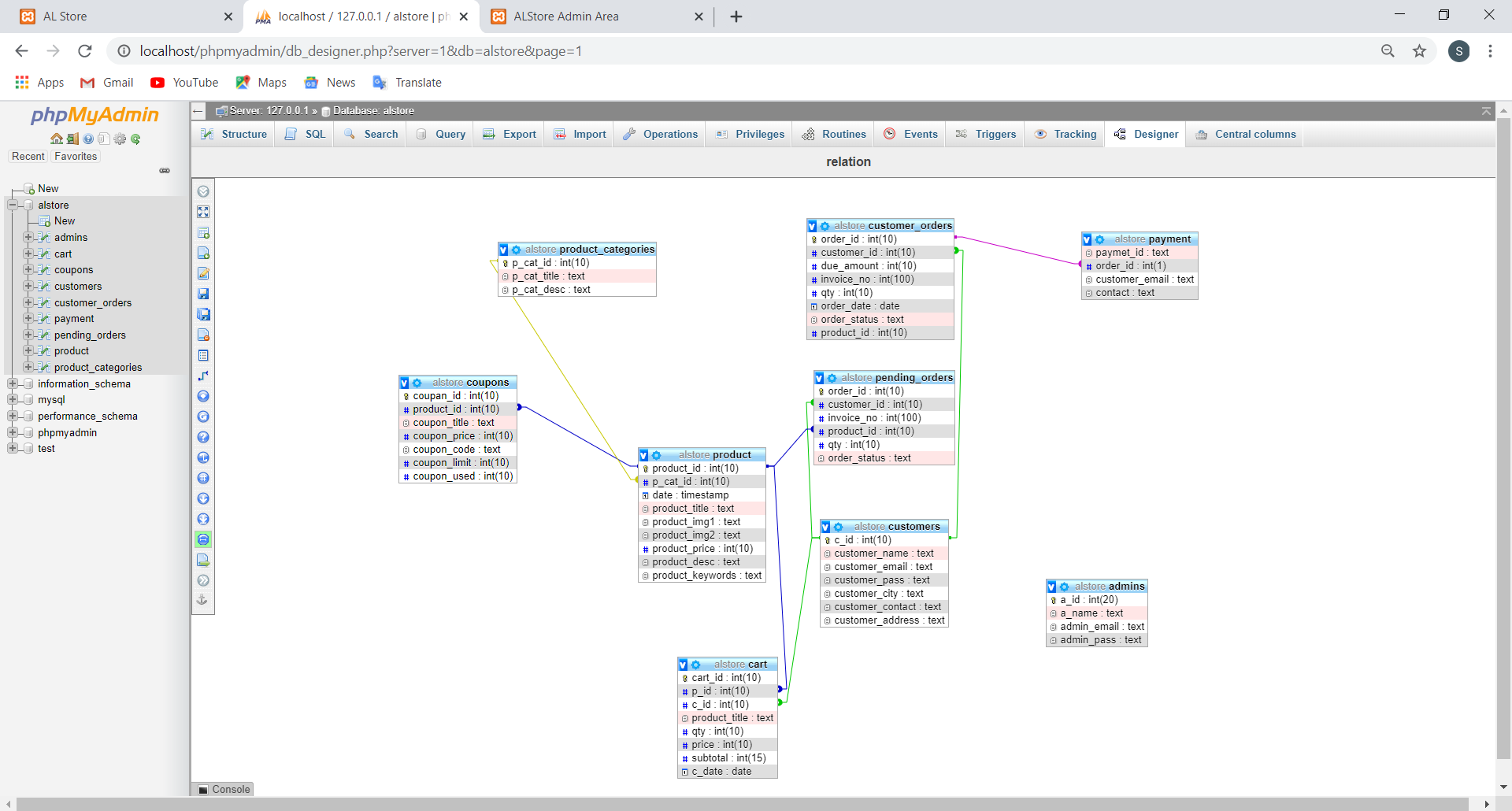
**For example,** Employee can assign by many projects and project can have many employees.



**ER diagram**

****

**3.3.2 Table Relationship**

****

**3.3.3 Table Description**

**Product Category:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column\_name** | **Type** | **Size** | **Notnull** | **Key** |
| Category\_id | INTEGER | 10 | Notnull | Primarykey |
| Category\_name | VARCHAR | 20 | Notnull |  |
| Category\_description | VARCHAR | 50 | Notnull |  |

**Cart :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column\_name** | **Type** | **Size** | **Notnull** | **Key** |
| Prod\_id | INTEGER | 10 | Notnull | Foreignkey |
| Qty | INTEGER | 10 | Notnull |  |
| Unit\_price | INTEGER | 20 | Notnull |  |
| Total\_price | INTEGER | 20 | Notnull |  |

**Product:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column\_name** | **Type** | **Size** | **Notnull** | **Key** |
| Prod\_id | INTEGER | 10 | Notnull | Primarykey |
| Prod\_name | VARCHAR | 20 | Notnull |  |
| Prod\_cat\_id | INTEGER | 10 | Notnull | Foreignkey |
| Price | INTEGER | 20 | Notnull |  |
| Prod\_key | VARCHAR | 20 | Notnull |  |
| Prod\_desc | VARCHAR | 50 | Notnull |  |

**Customer :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column\_name** | **Type** | **Size** | **Notnull** | **Key** |
| Cust\_id | INTEGER | 10 | Notnull | Primarykey |
| Name | VARCHAR | 20 | Notnull |  |
| Phone\_no | INTEGER | 20 | Notnull |  |
| Email | VARCHAR | 20 | Notnull |  |
| Password | VARCHAR | 20 | Notnull |  |
| City | VARCHAR | 20 | Notnull |  |
| Country | VARCHAR | 20 | Notnull |  |
| Address | VARCHAR | 50 | Notnull |  |

**Coupon :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column\_name** | **Type** | **Size** | **Notnull** | **Key** |
| Copon id | INTEGER | 10 | nottnull | Primarykey |
| Product\_id | INTEGER | 10 | nottnull | Foreignkey |
| Coupon \_titel | VARCHAR | 20 | Notnull |  |
| Coupon \_limit | INTEGER | 10 | nottnull |  |
| Coupon \_price | INTEGER | 10 | nottnull |  |
| Coupon \_used | INTEGER | 10 | nottnull |  |
| Coupon \_code | VARCHAR | 20 | Notnull |  |

**Customer Order:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column\_name** | **Type** | **Size** | **Notnull** | **Key** |
| Order\_id | INTEGER | 10 | nottnull | Primarykey |
| Customer \_id | INTEGER | 10 | nottnull | Foreignkey |
| Due\_ammount | INTEGER | 10 | nottnull |  |
| Invoice\_no | INTEGER | 10 | nottnull |  |
| qty | INTEGER | 10 | nottnull |  |
| Order\_date | date |  | nottnull |  |
| Order status | VARCHAR | 20 | nottnull |  |
| Product id | INTEGER | 10 | nottnull | Foreignkey |

**Payment:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column\_name** | **Type** | **Size** | **Notnull** | **Key** |
| Payment\_id | VARCHAR | 20 | nottnull |  |
| Ordered | INTEGER | 10 | nottnull | Foreignkey |
| Customer\_email | VARCHAR | 20 | nottnull |  |
| Customer\_contact | VARCHAR | 20 | nottnull |  |

**Admin:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column\_name** | **Type** | **Size** | **Notnull** | **Key** |
| A\_id | INTEGER | 10 | nottnull | Primarykey |
| A\_name | VARCHAR | 20 | nottnull |  |
| admin\_email | VARCHAR | 20 | nottnull |  |
| A\_pass | VARCHAR | 20 | nottnull |  |

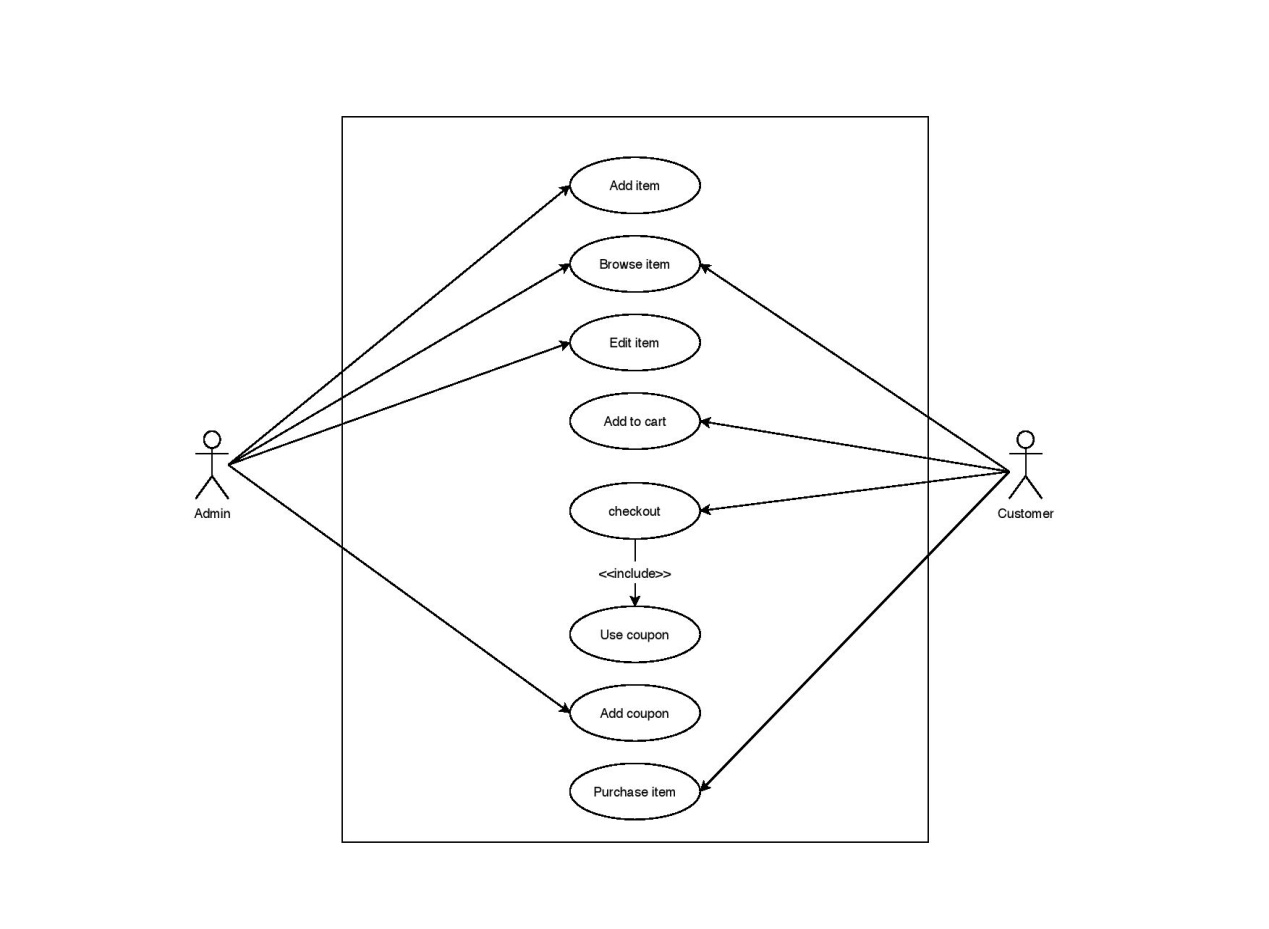
**3.4 SYSTEM DESIGN IMPLEMENTATION**

**3.4.1 Use case**

An **actor** is a direct external user of a system - an object or set of objects that communicates directly with the system but that is not part of the system. Each actor

represents those objects that behave in a way toward the system. ADMIN and CUSTOMER are different actors of Advaita Lifestyle .

The various interactions of actors with a system are quantized into use cases. A **use case** is a coherent piece of functionality that a system can provide by interacting with actors. Each use case involves one or more actors as well as the system itself.

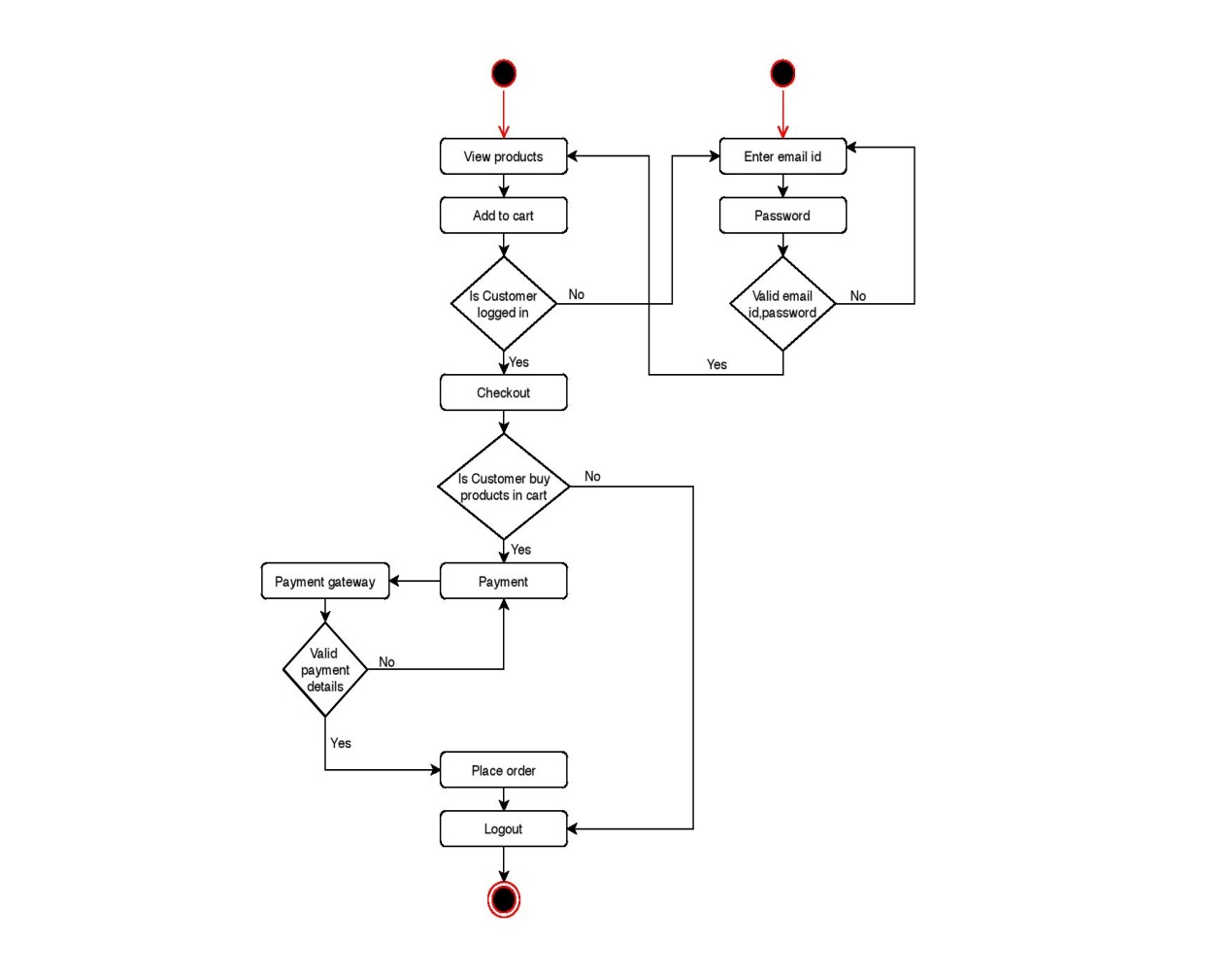
****

**3.4.2 Activity Diagram**

Admin activity diagram

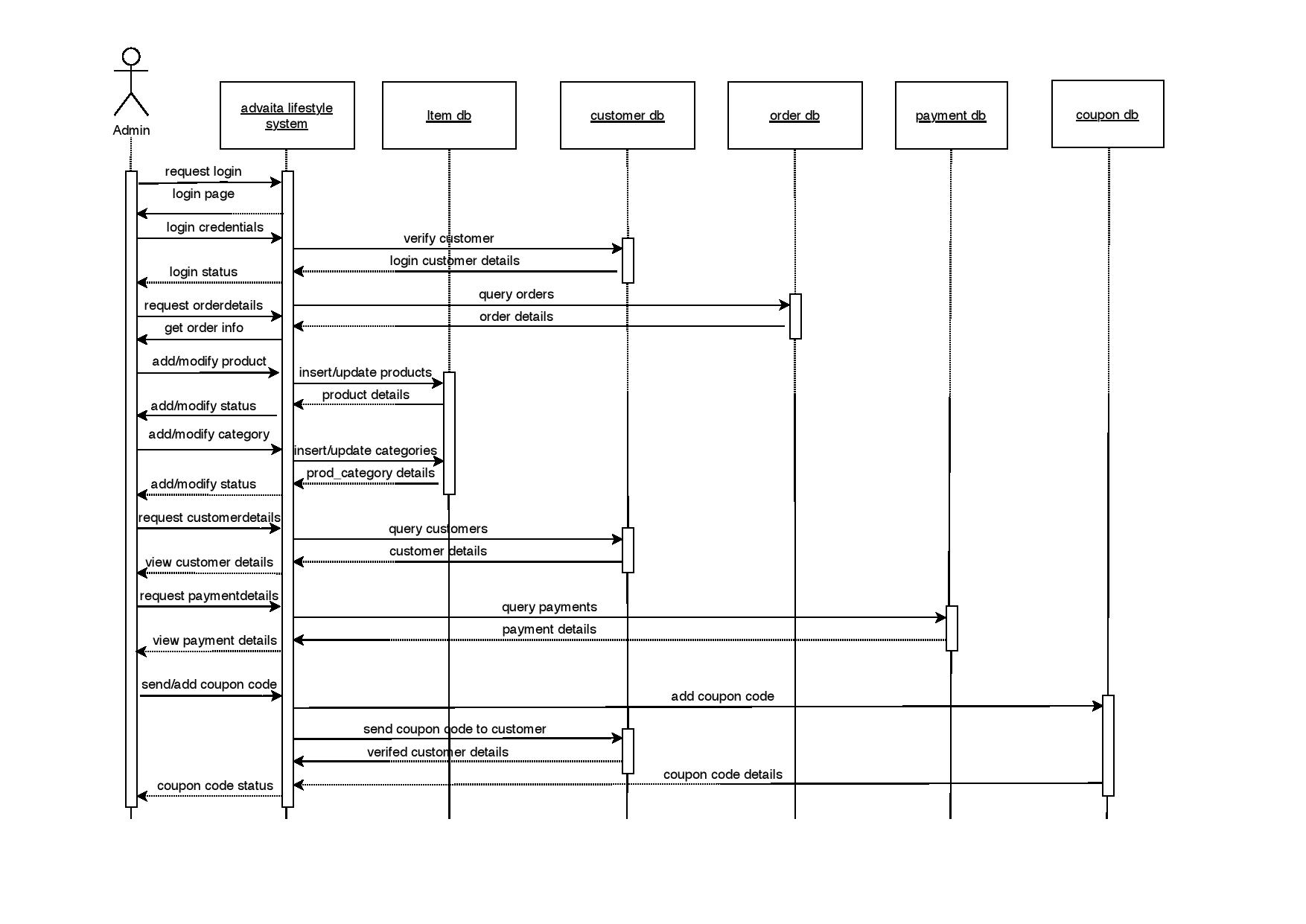
****

Customer activity diagram



**3.4.3 Sequence Diagram**

Admin sequence diagram

****

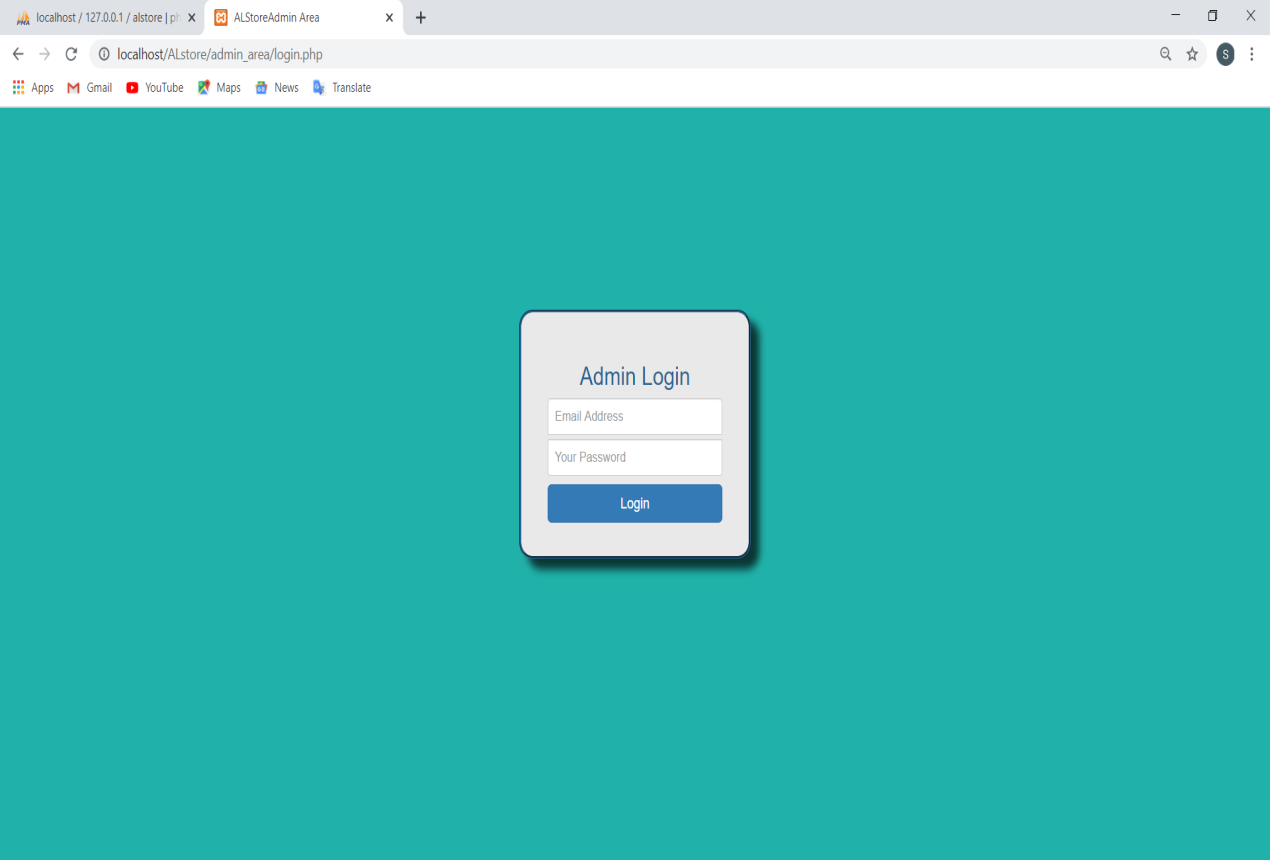
Customer sequence diagram



**3.5 USER INTERFACE DESIGN**

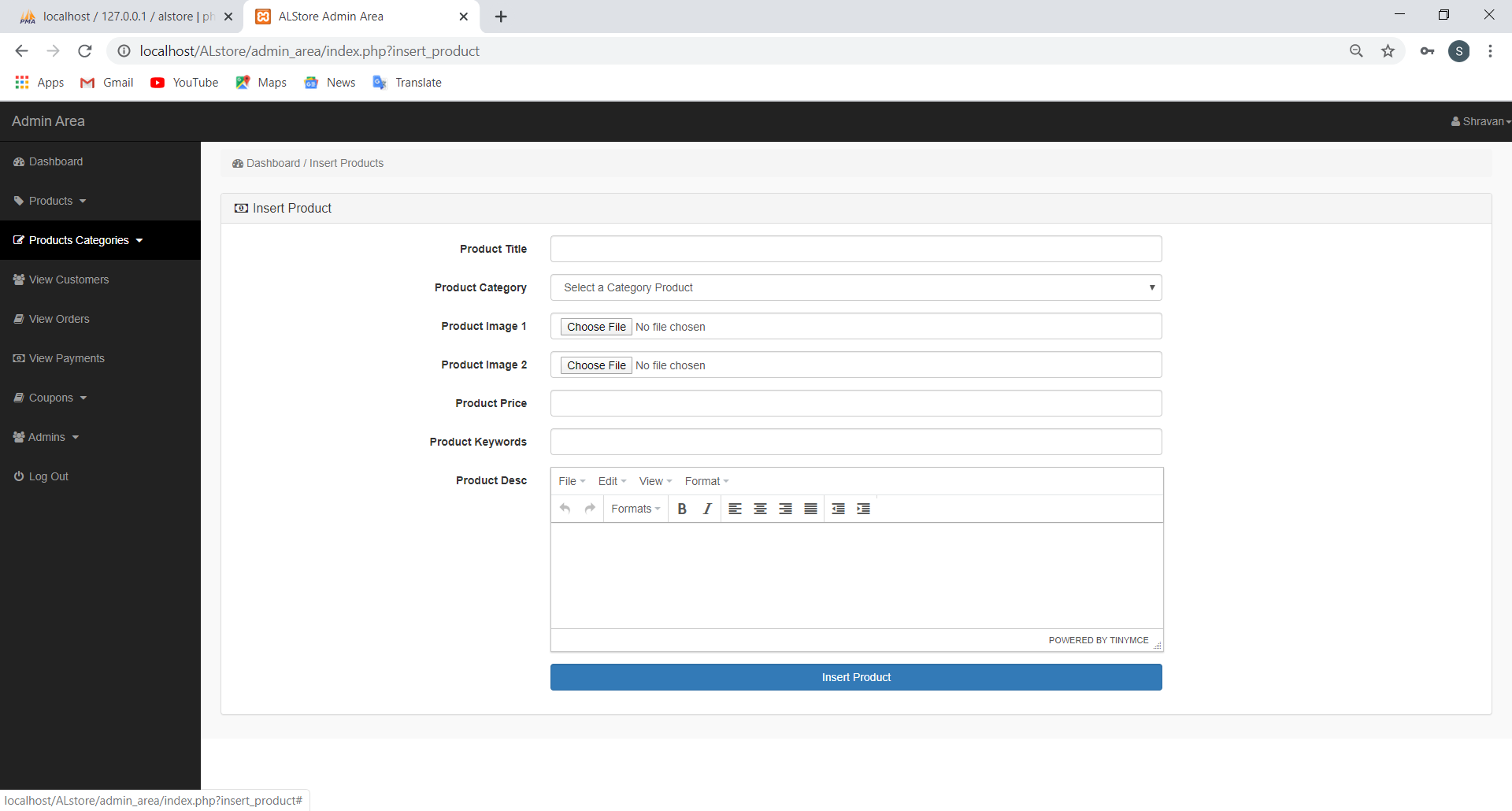
**3.5.1 Admin user interface**

Admin login

****

|  |  |  |
| --- | --- | --- |
| **Field name** | **Type** | **Description** |
| Email adress | Textbox | Field to enter email id |
| Password | Textbox | Field to enter password |
| Login | Button | Login to admin page |

Inserting product

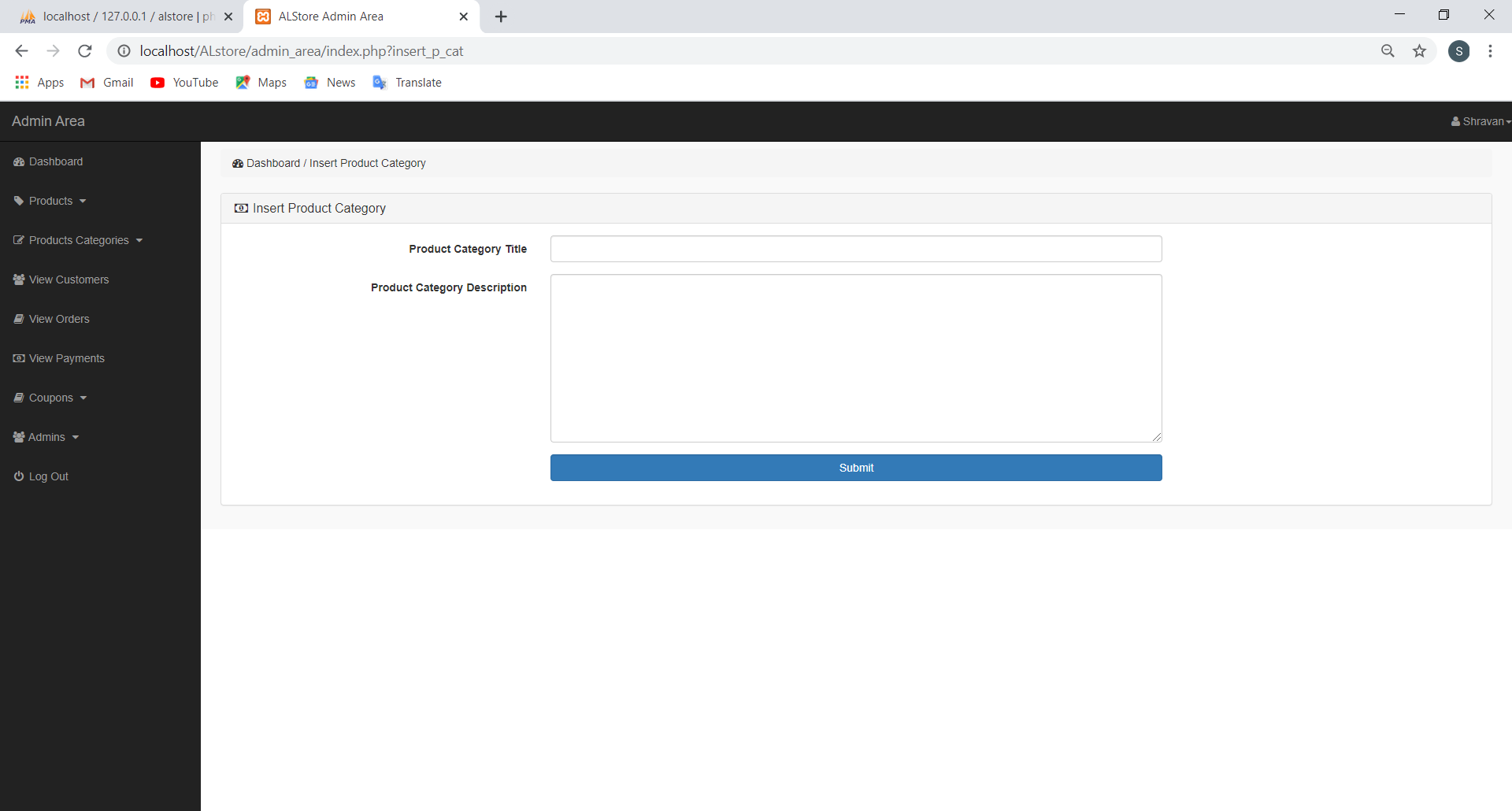


|  |  |  |
| --- | --- | --- |
| **Field name** | **Type** | **Description** |
| Product titel | Textbox | Field to enter the name of the product |
| Product category | Dropdown Box | Field to slect the catagory |
| Product image 1 | File | Image1 of the product |
| Product image 2 | File | Image 2 of the product |
| Product price | Number | Price of the product |
| Product keyword | Textbox | Keyword of the product |
| Product description | Textbox | Desrcption of product |
| Insert item | Button | To insert to product table |

View products

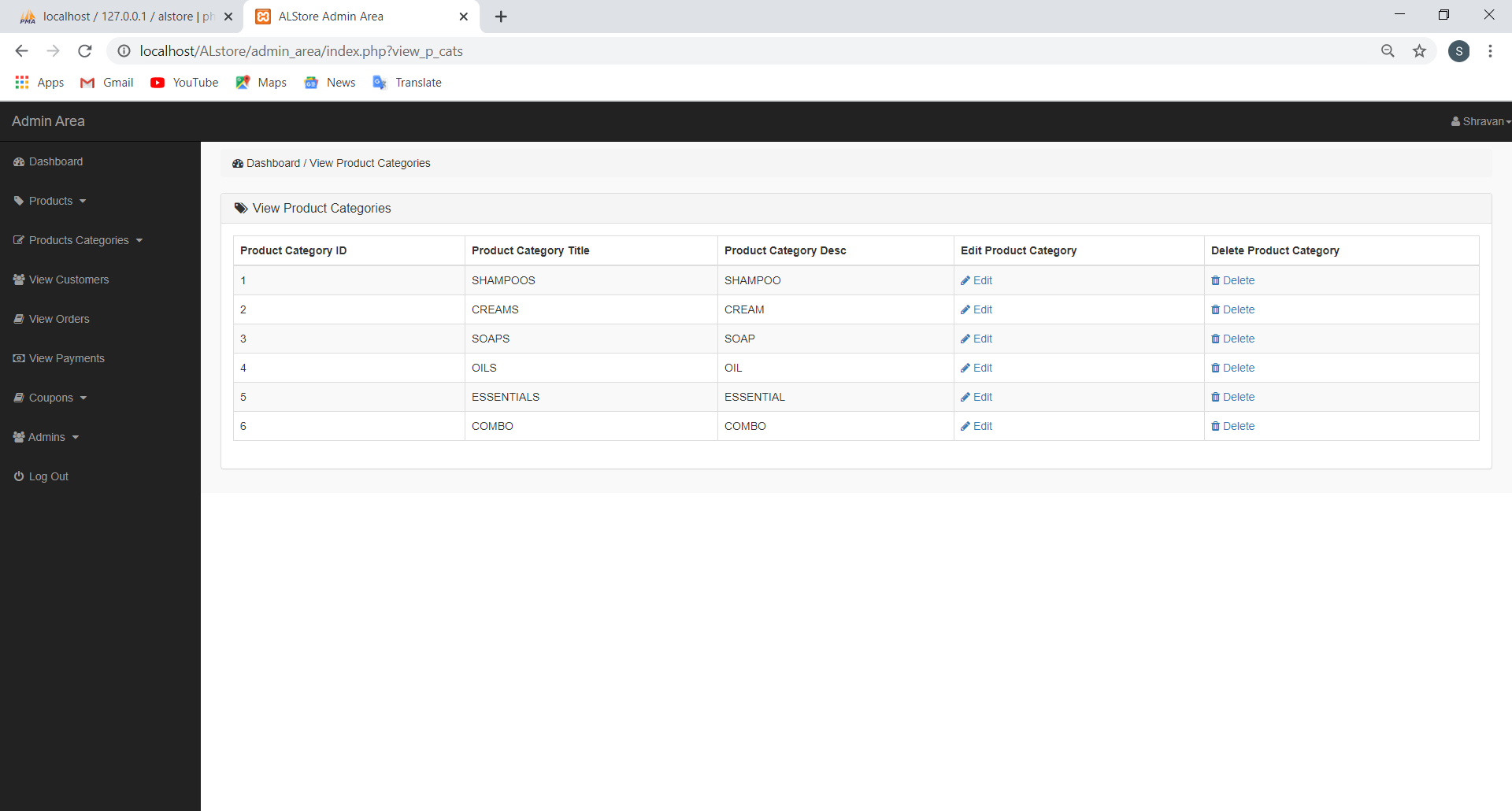
****

Inserting product catagory

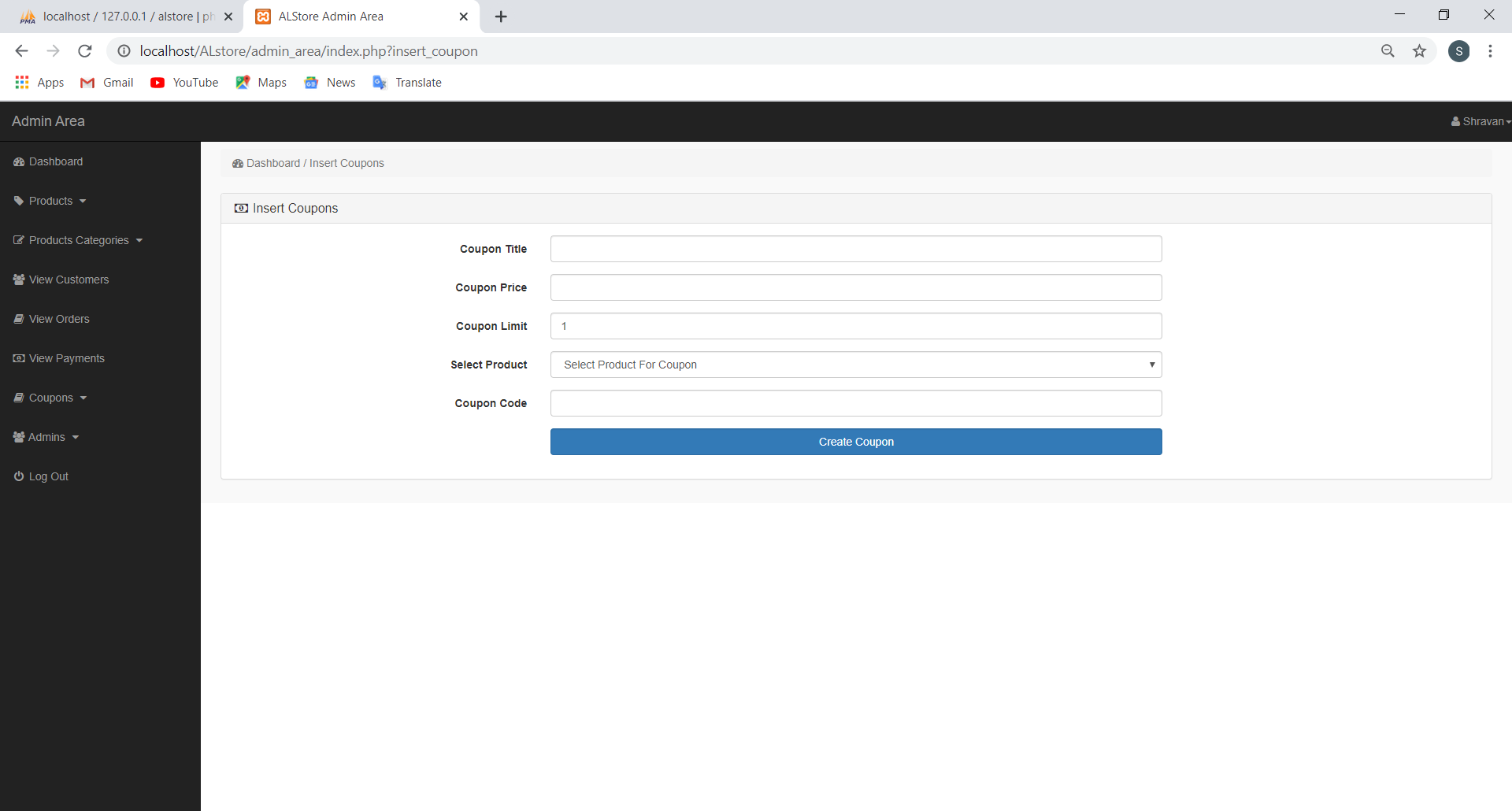
****

|  |  |  |
| --- | --- | --- |
| **Field name** | **Type** | **Description** |
| Product category title | Textbox | Name of the category |
| Productcategorydescription | Textbox | Description of category |
| submit | submit | Add to category table |

View products category

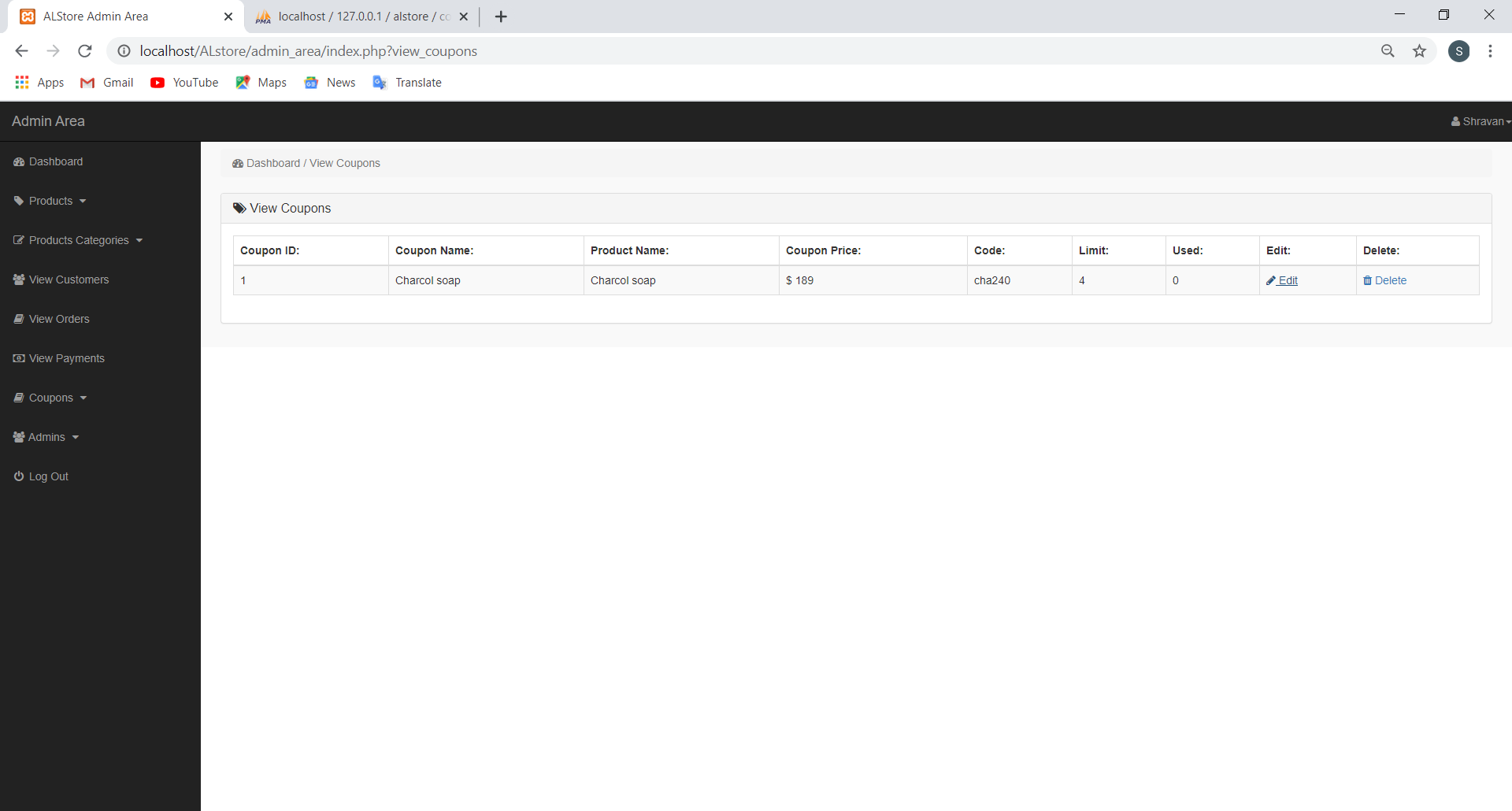
****

Inserting cuopan

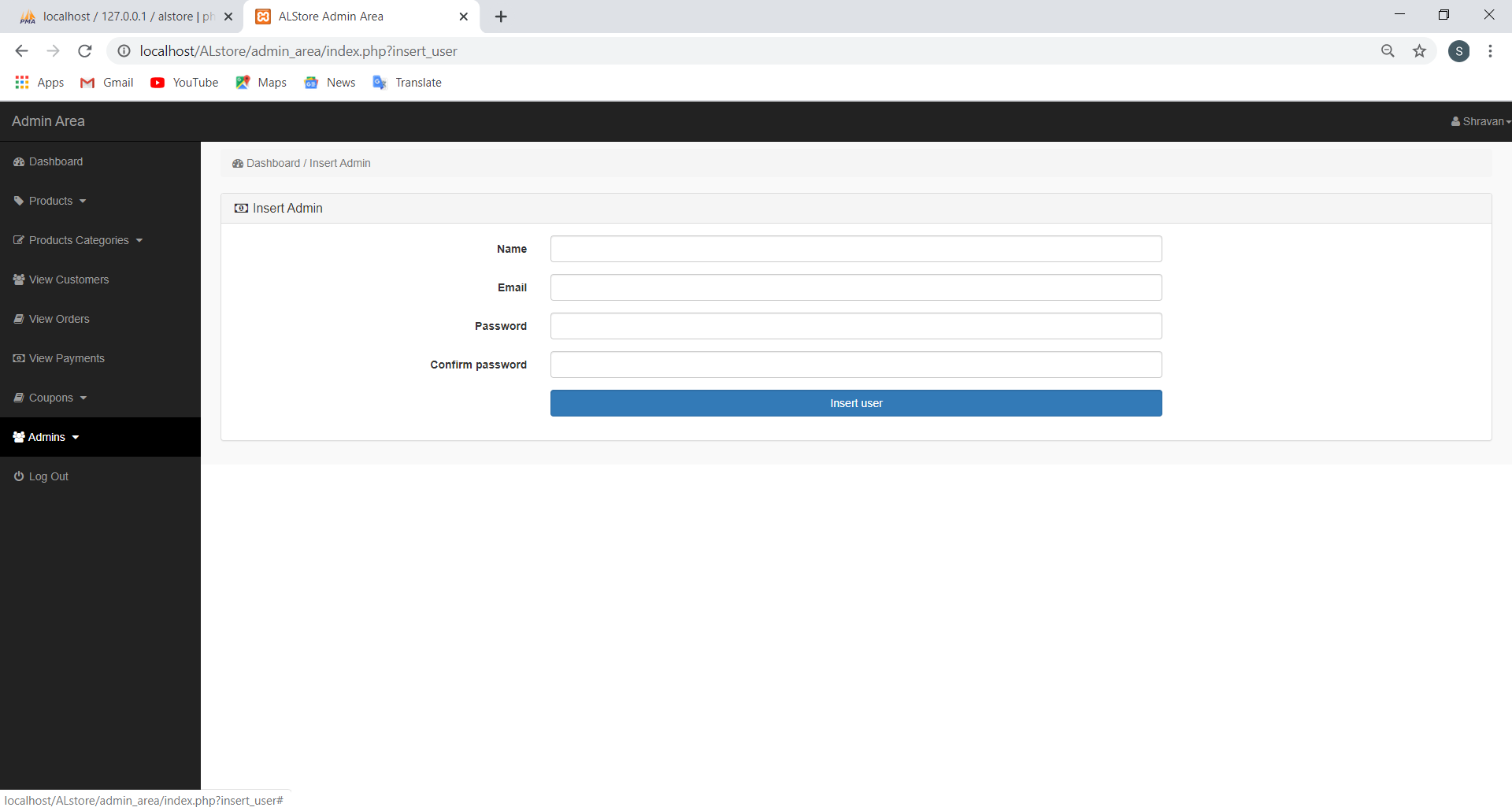


|  |  |  |
| --- | --- | --- |
| **Field name** | **Type** | **Description** |
| Coupon title | Textbox | Name of the coupon |
| Coupon price | Number | Value of coupan |
| Coupon limit | Number | No of coupon |
| Select product | Dropdown Box | Product to apply discount |
| Coupon code | Textbox | Code to get discount |
| Create Coupon | Button | Button to add to table |

View Coupons

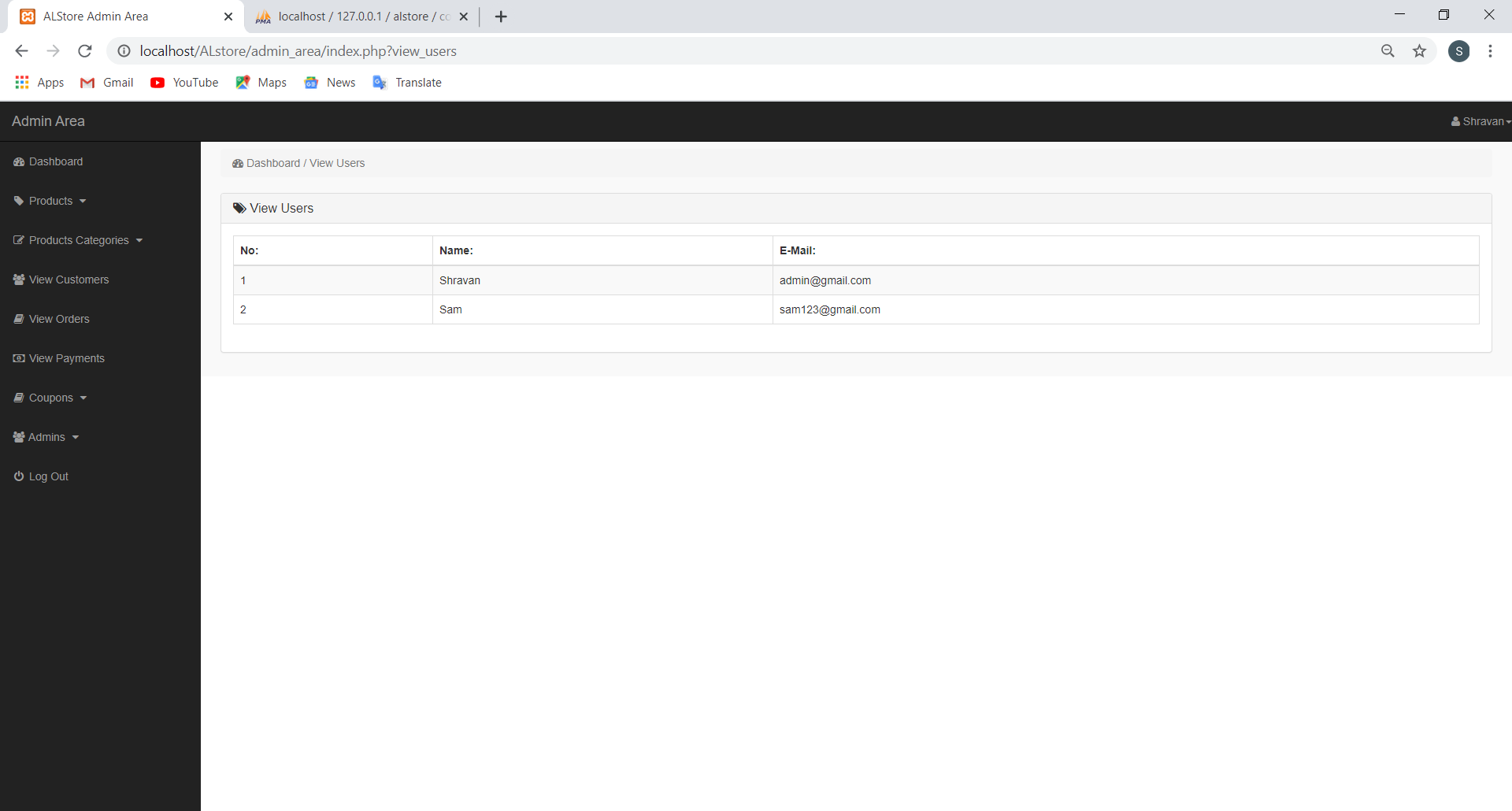


Insert new admin

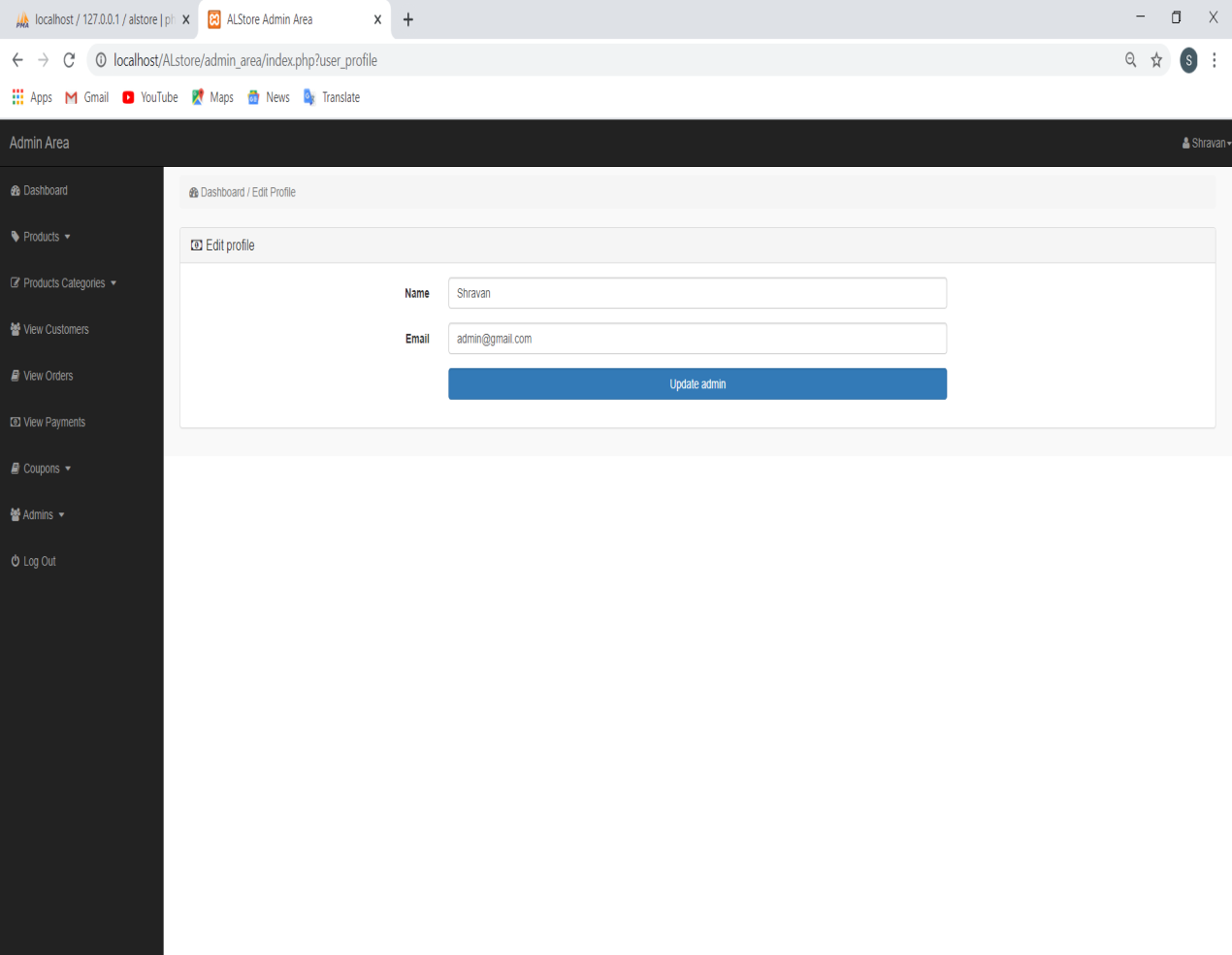


|  |  |  |
| --- | --- | --- |
| **Field name** | **Type** | **Description** |
| Name | Textbox | Name of the admin |
| Email | Textbox | Email address of admin |
| Password | Textbox | New password |
| Confirm password | Textbox | Confirm password |
| Insert user | Button | Insert to adim table |

View admins



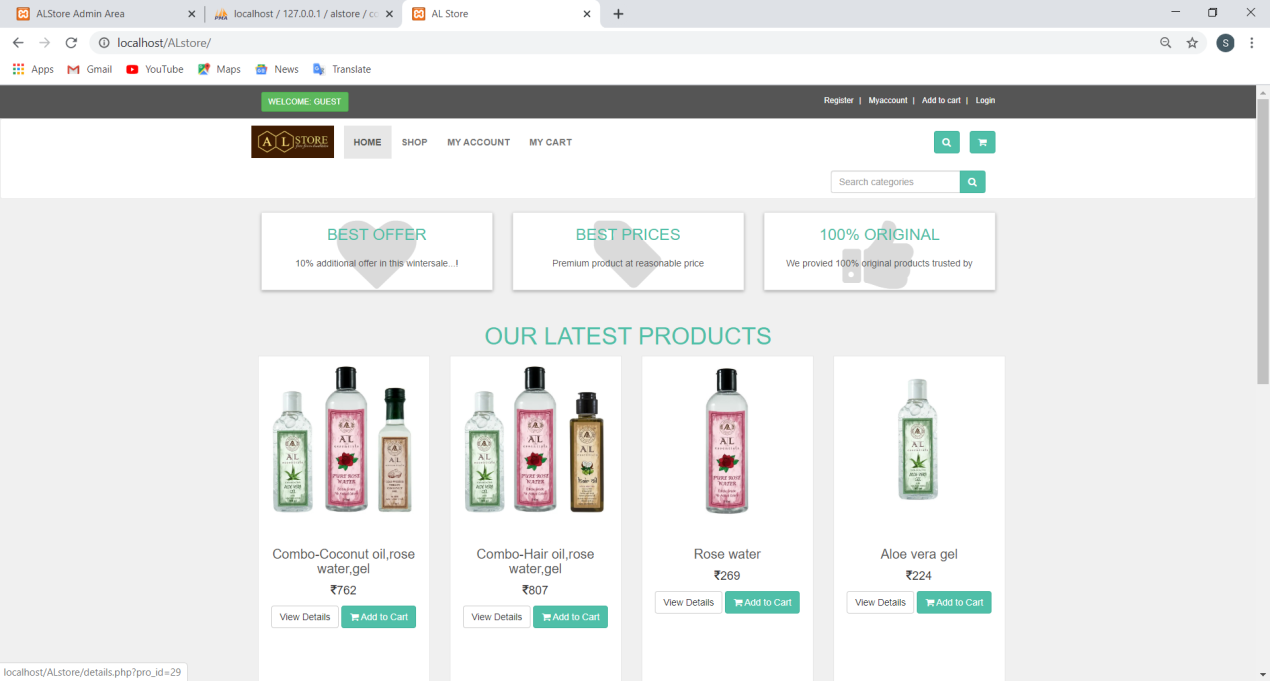
Upadate admin info



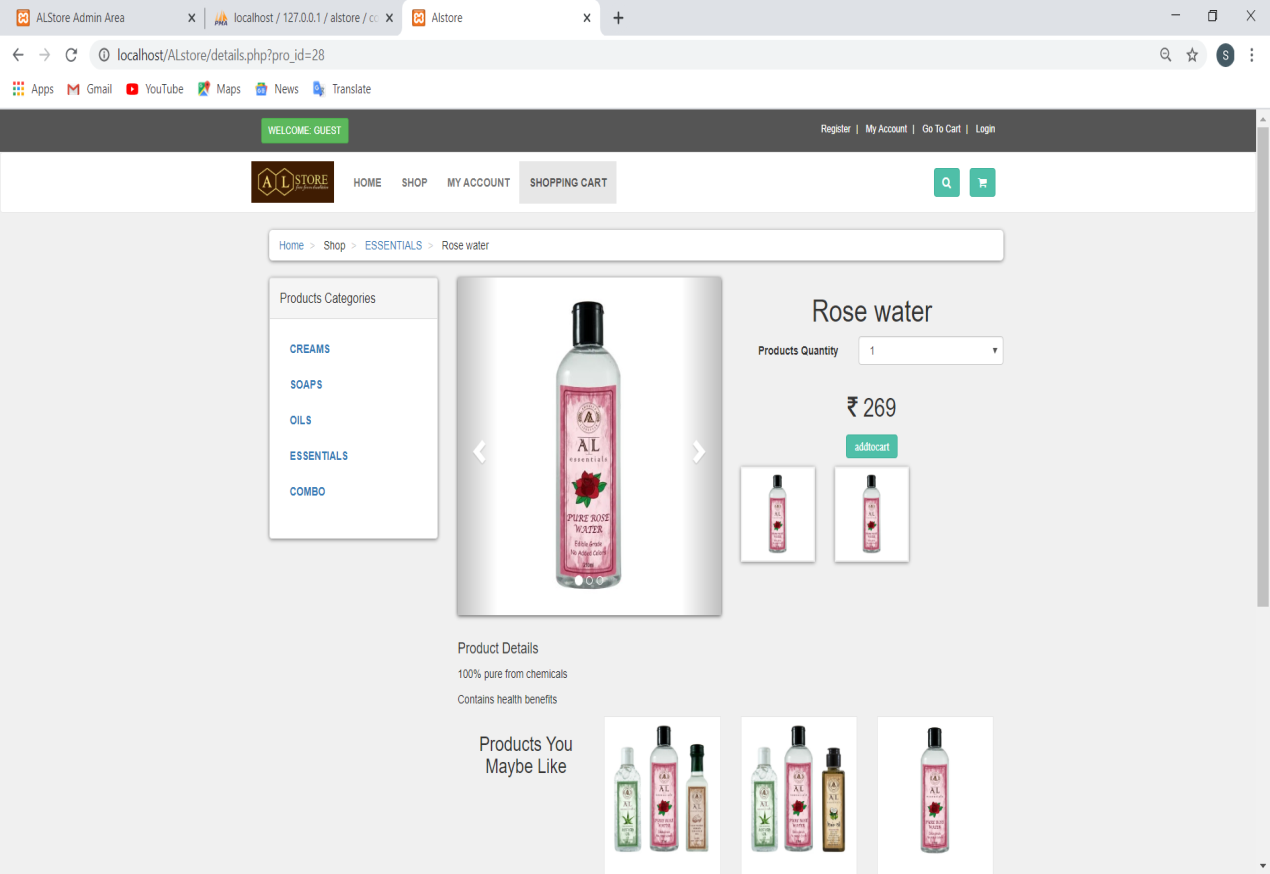
|  |  |  |
| --- | --- | --- |
| **Field name** | **Type** | **Description** |
| Name | Textbox | Name of the admin |
| Email | Textbox | New email address |
| Update admin | Button | Button to change |

**3.5.2 Customer user interface**

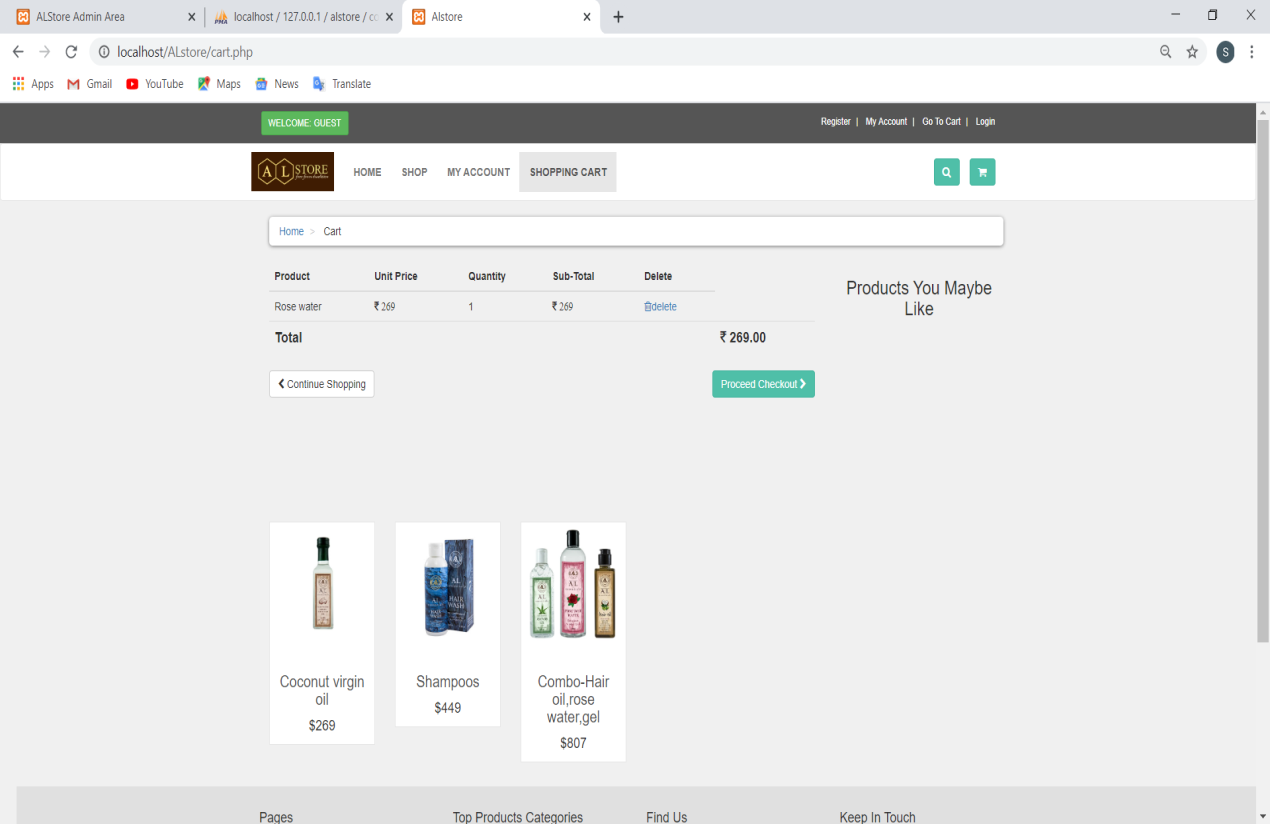
Home page



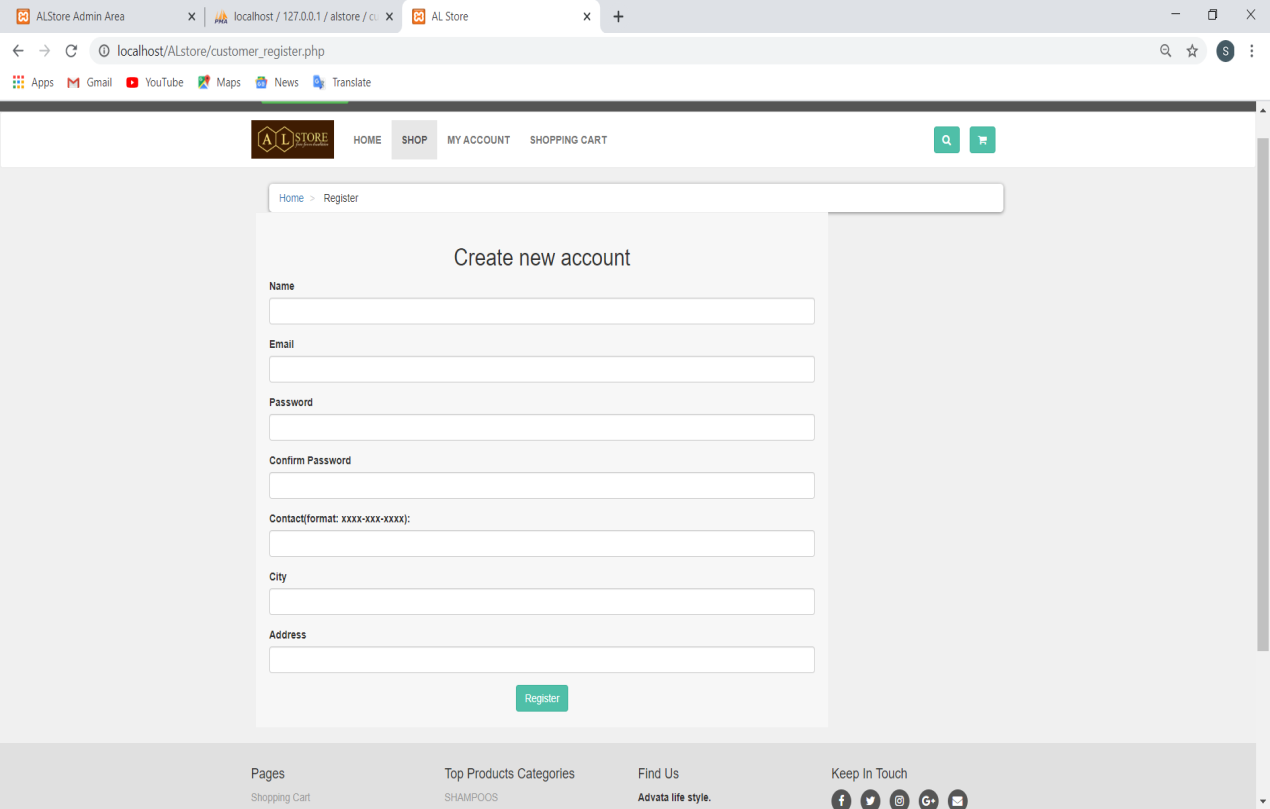
Details of product



Shopping Cart

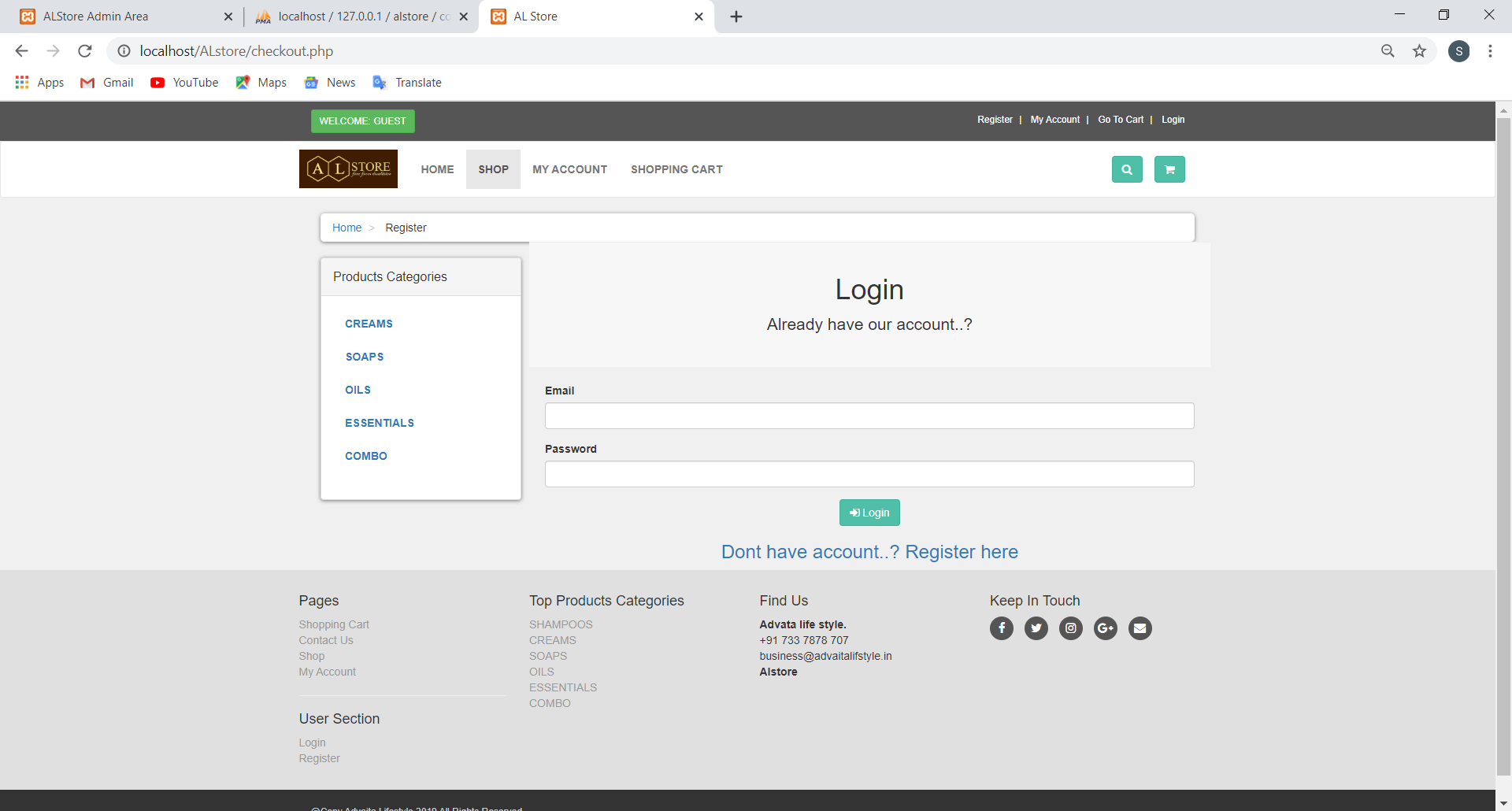


Customer Registration Form



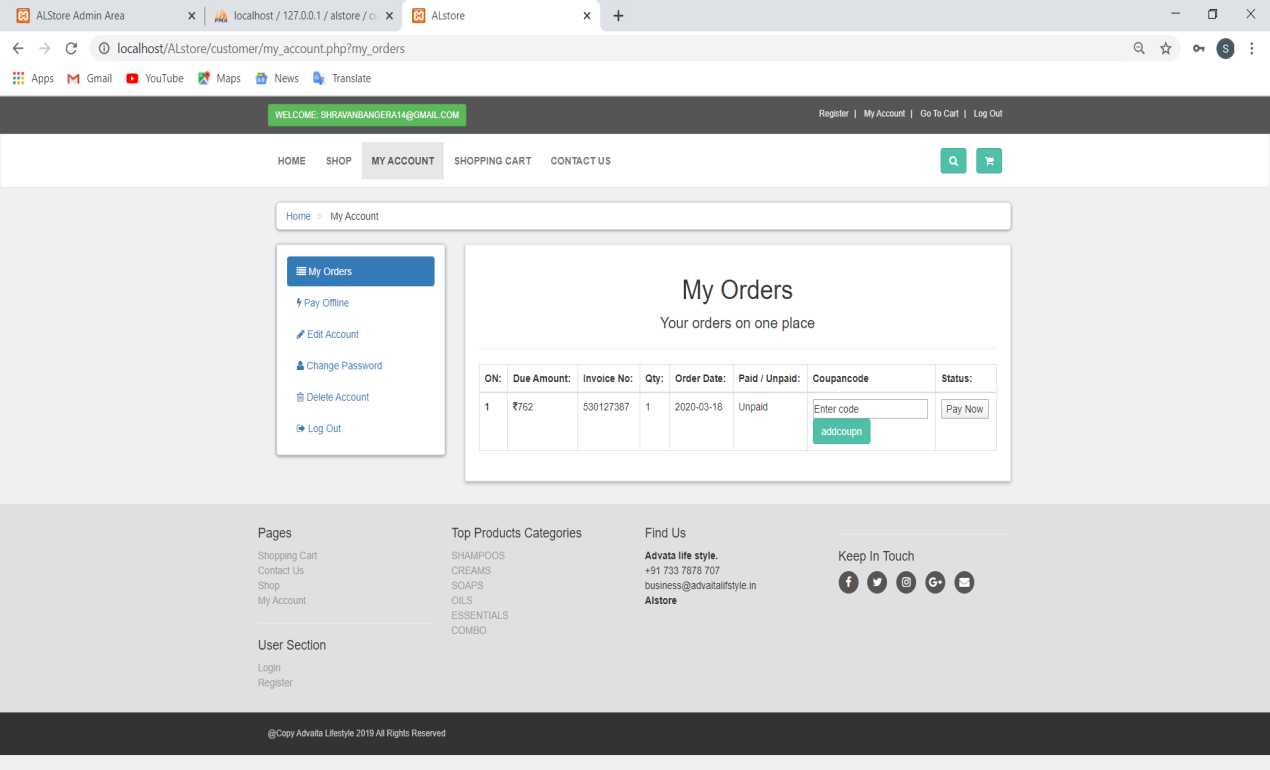
|  |  |  |
| --- | --- | --- |
| **Field name** | **Type** | **Description** |
| Name | Textbox | Name of the customer |
| Email | Textbox | Email id of cutomer |
| password | Textbox | password |
| Confirm password | Textbox | To Confirm password |
| contact | Textbox | Contact of customer |
| city | Textbox | City of customer |
| address | Textbox | Address of customer |
| Register | Button | To create the account |

Customer Login

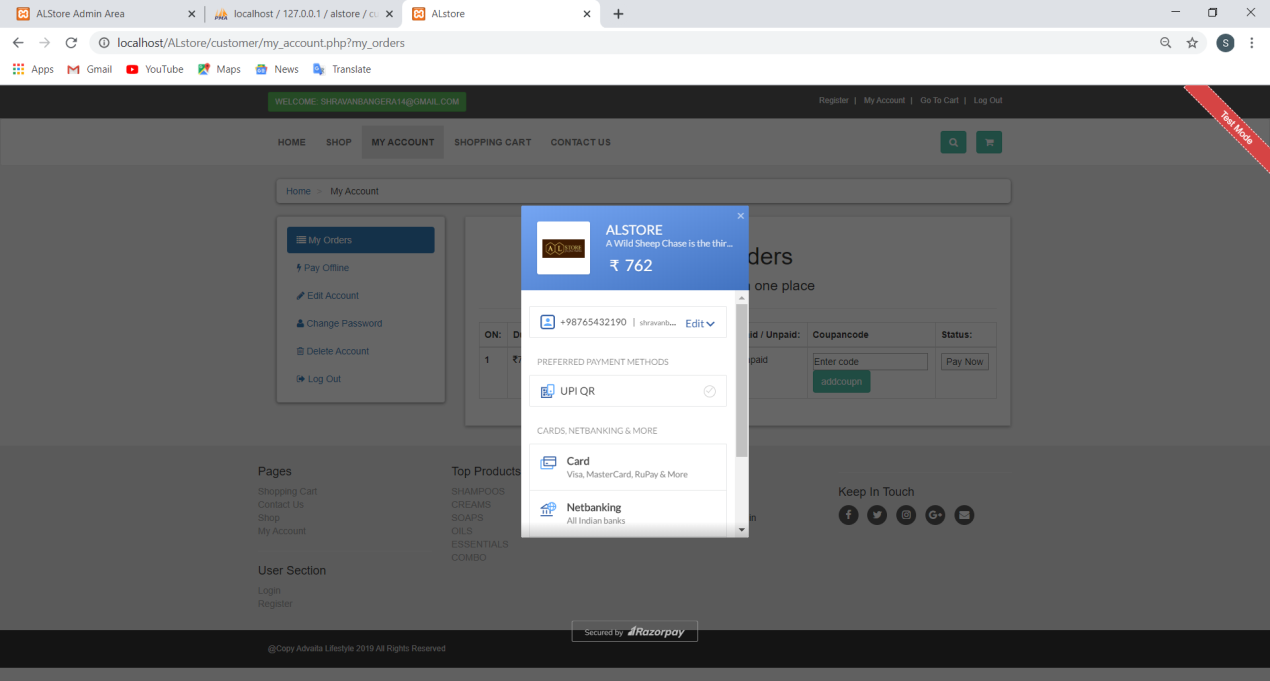


|  |  |  |
| --- | --- | --- |
| **Field name** | **Type** | **Description** |
| Email | Textbox | Email id of customer |
| password | Textbox | Password of the account |
| login | Button | To login |

Customer Order



Payment



**TESTING**

**4 TESTING**

**4.1 INTRODUCTION**

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Testing has been defined as the process of analysing a software item to detect the differences between existing and required conditions and to evaluate the features of the software item. Software testing is the process used to assess the quality of computer software.

It involves operation of a system or application under controlled conditions and evaluating the results. The controlled conditions should include both normal and abnormal conditions. Testing should intentionally attempt to make things go wrong to determine if things happen when they should. It is oriented to ‘detection’.

Software testing has three main purposes

* The verification process confirms that the software meets its technical specifications. A “specification” is a description of a function in terms of a measurable output value given a specific input value under specific preconditions.
* The validation process confirms that the software meets the business requirements.
* A defect is a variance between the expected and actual result. The defect’s ultimate source may be traced to a fault introduced in the specification, design, or development phases. Not all the defects will necessarily result in failures.

There are two types of software testing:

* Black box testing-internal system design is not considered in this type of testing. test is based on requirements and functionality.
* White box testing-this testing is based on knowledge of threw internal logic of an application’s code. Also known as glass box testing. Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths and conditions.

A test case is a software testing document, which consists of event, action, input, output, expected result and actual result. Clinically defined a test case is an input and an expected result. This can be pragmatic as ‘for condition x your derived result is y’, whereas other test cases described in more detail the input scenario and what results might be expected. It can occasionally be a series of steps but one with expected results or expected outcome. A test case should also contain a place for the actual result.

White box testing is applicable at the unit, integration and system levels of the software testing process.

**4.2 TESTING OBJECTIVE**

* Finding defects which may get created by the programmer while developing the software
* Gaining confidence in and providing information about the level of quality
* To prevent defects
* To make sure that the end results meets the business and user requirements.
* To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specification

**4.2.1 Unit Testing**

Unit tests are written from a programmer’s perspective. They ensure that a particular method of a class successfully performs a set of specific tasks. each test confirms that a method produces the expected output when given a known input. a series of stand-alone tests are conducted during unit testing. Each test examines an individual component that is new or has been modified. Unit tests focus on functionality and reliability. Unit testing is done in a test environment prior to system integration. If a defect is discovered at unit test, the severity of the defect will dictate whether or not it will be fixed before the module is approved.

**4.2.2 Integration Testing**

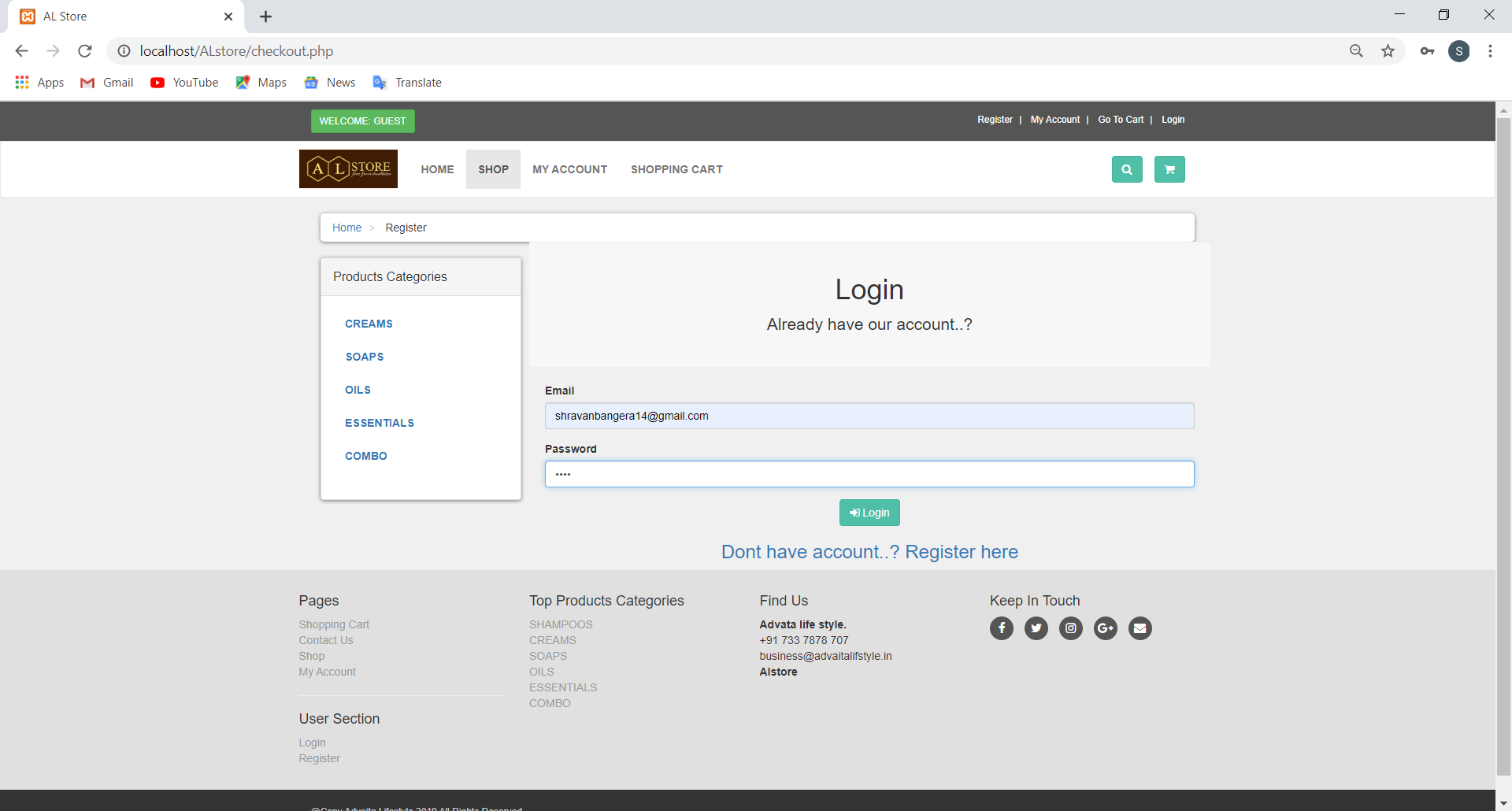
Integration testing examines all the components and modules that are new,changed,affected by a change, or needed to form a complete system. Integration testing requires involvement of other systems and interfaces with other applications, including those owned by an outside vendor, external partners, or the customer.

it is the phase of software testing in which individual software modules are combined and tested as a group. it follows unit testing and precedes system testing. integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

Therefore, without adequate testing, there is a greater risk that an application will inadequately deliver what was expected by the users or that the final product will have problems such that the users will eventually abandon it out of frustration. In either case, time and money are lost and the credibility and reputation of both the developers and the software is damaged. More formal, rigorous testing will go far to reducing the risk that either of these scenarios occurs.

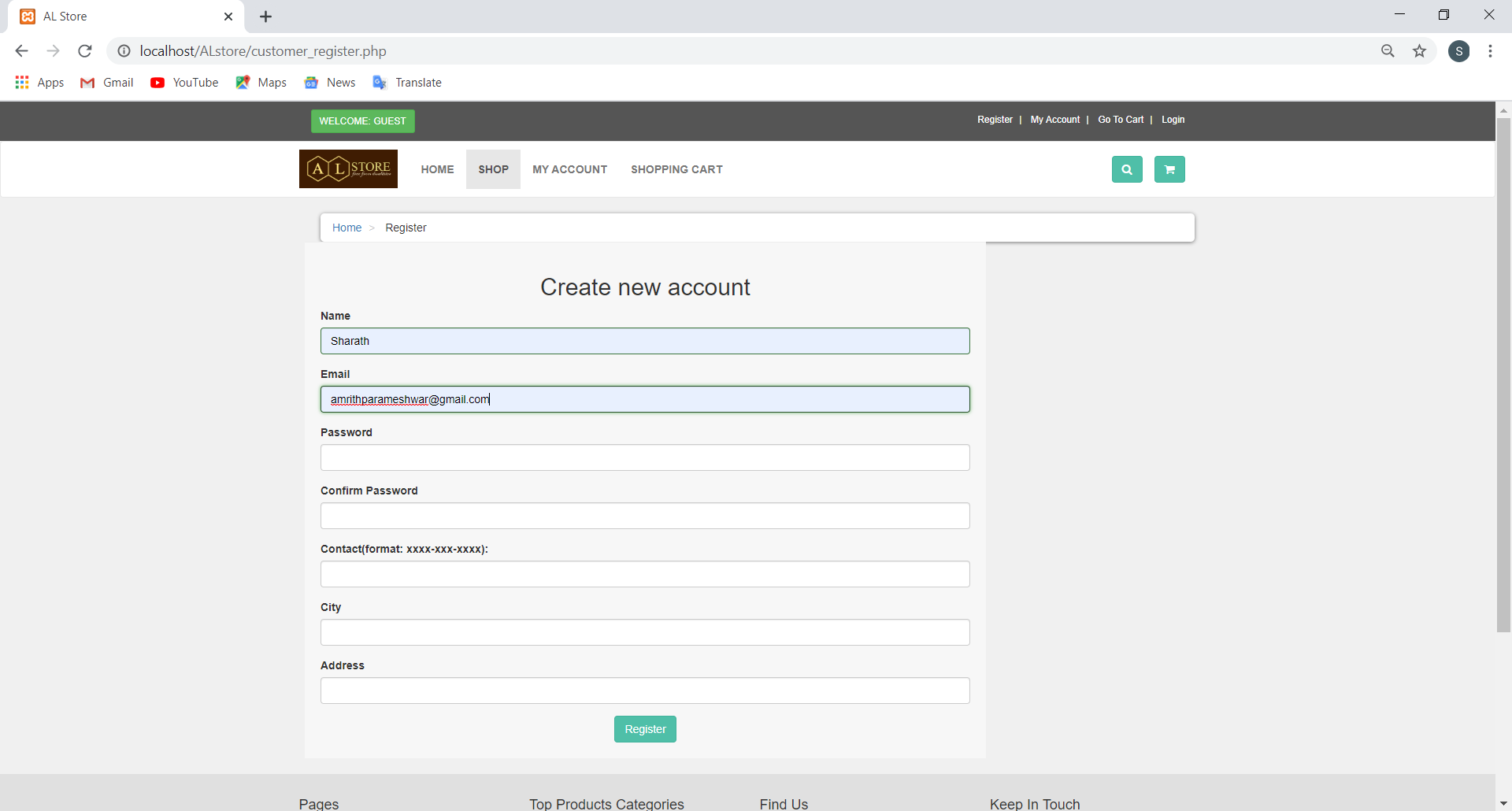
**4.3 TEST CASES**

**4.3.1 User login**



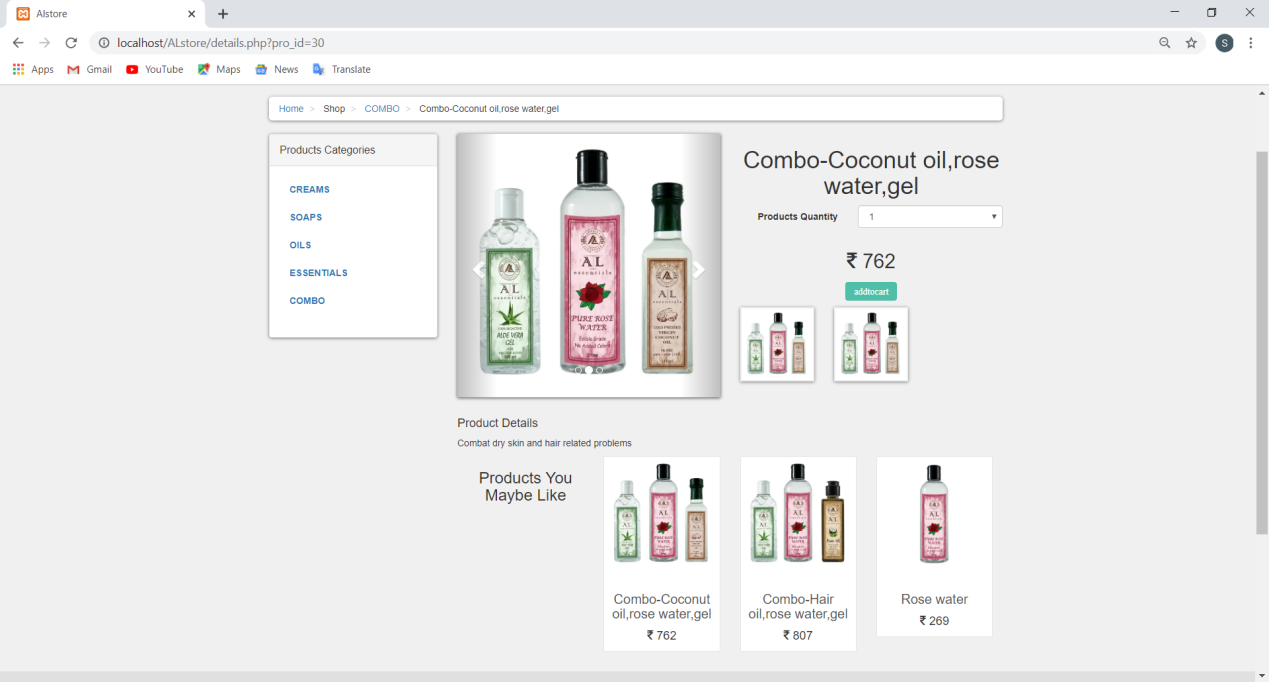
|  |  |  |  |
| --- | --- | --- | --- |
| **Sl.No** | **User Action** | **Expected Result** | **Test Result** |
| 1 | On load of the form | Displays the form | Pass |
| 2 | if User leaves the fields empty or enters invalid details and clicks login in button | Appears a message with an error message for respective field | Pass |
| 2 | if User leaves the fields empty or enters invalid details and clicks login in button | Appears a message with an error message for respective field | Pass |
| 3 | On click of Login button | If the user leaves any field empty or enters invalid detail an error message will appear otherwise allows the user to login to the system | Pass |

**4.3.2 User Registration**



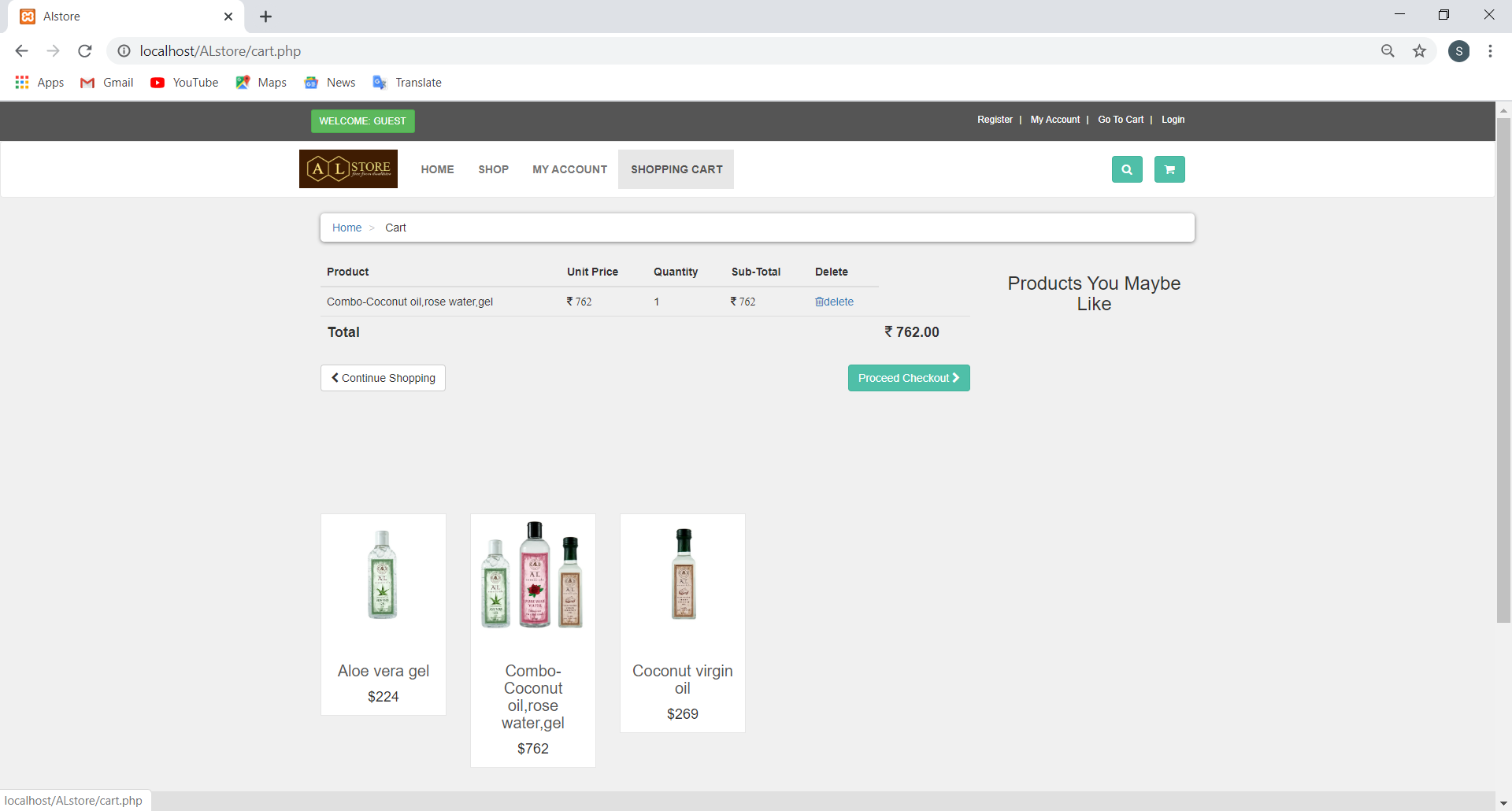
|  |  |  |  |
| --- | --- | --- | --- |
| **Sl.No** | **User Action** | **Expected Result** | **Test Result** |
| 1 | On load of the form | Displays the form | Pass |
| 2 | if User leaves the fields empty or enters invalid details and clicks register in button | Appears a message with an error message for respective field | Pass |
| 3 | On click of register button | If the user leaves any field empty or enters invalid detail an error message will appear  otherwise allows the user to Register to the system to the system | Pass |

**4.3.3 Product details**

****

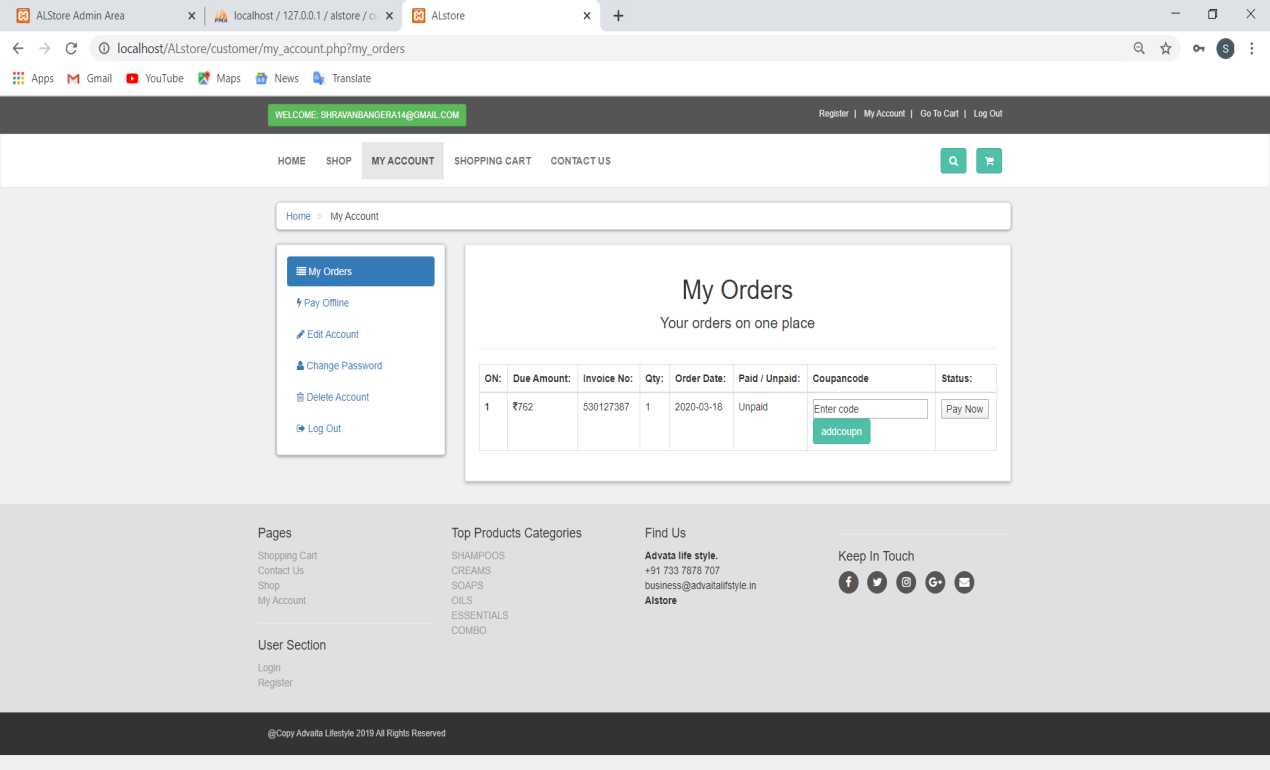
|  |  |  |  |
| --- | --- | --- | --- |
| **Sl.No** | **User Action** | **Expected Result** | **Test Result** |
| 1 | On click on the product image | shows the details of product | pass |
| 2 | On click of addtocart button | item added to cart | pass |

**4.3.4 Shopping cart**

****

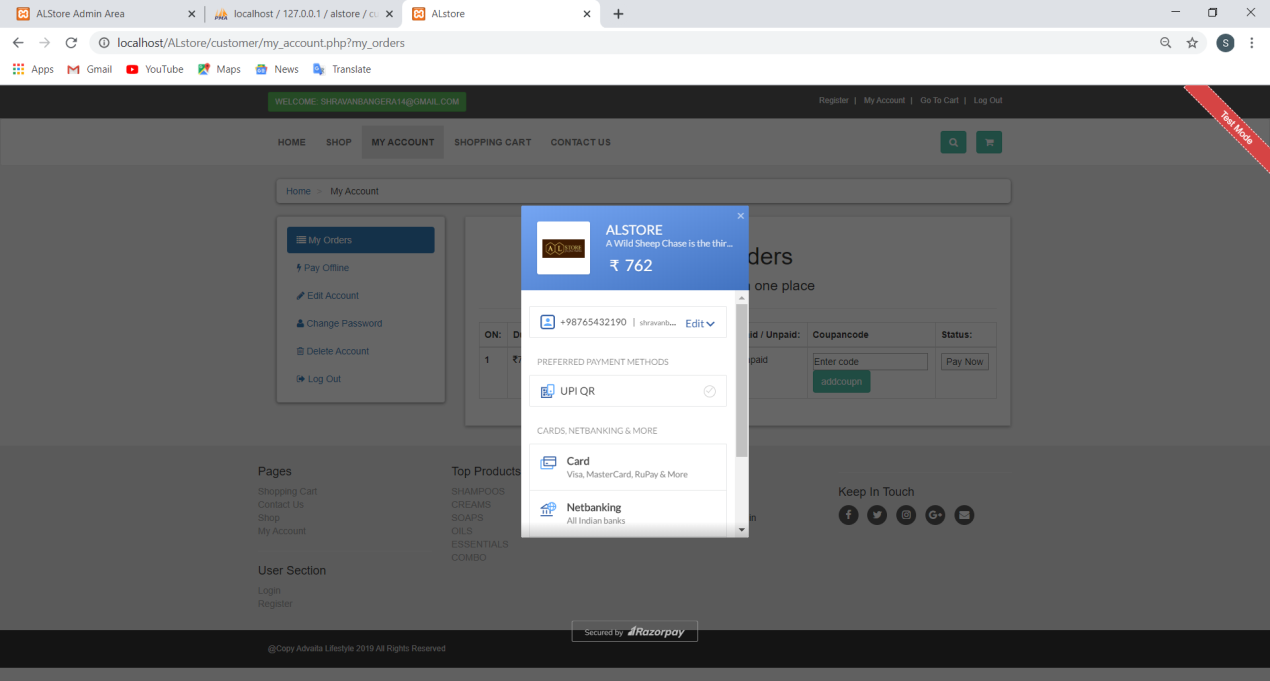
|  |  |  |  |
| --- | --- | --- | --- |
| **Sl.No** | **User Action** | **Expected Result** | **Test Result** |
| 1 | On load | shopping cart appears | pass |
| 2 | If no item in cart shows cart is empty |  | pass |
| 3 | If item present in cart shows item price quantity and total delete option and proced to checkout option |  | pass |

**4.3.5 My orders**

****

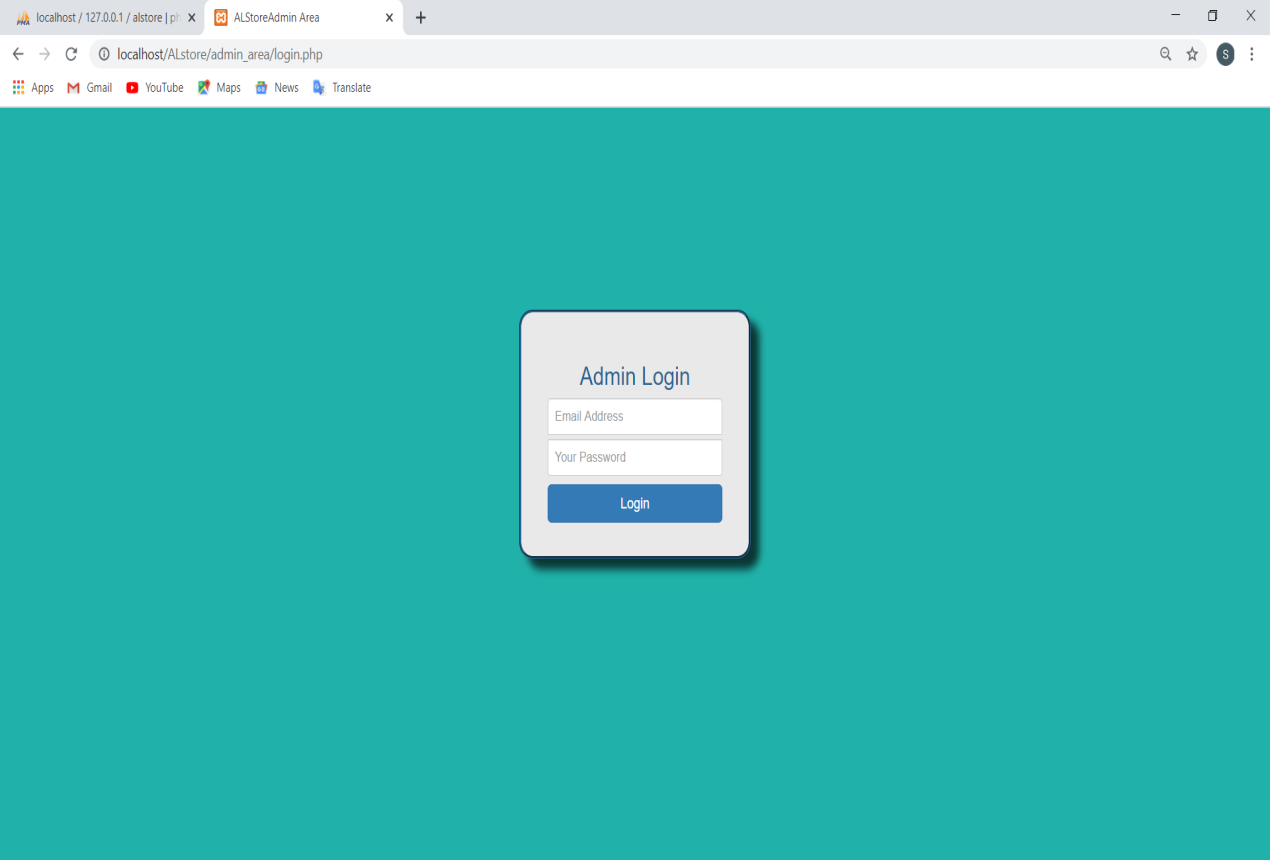
|  |  |  |  |
| --- | --- | --- | --- |
| **Sl.No** | **User Action** | **Expected Result** | **Test Result** |
| 1 | On form load | shows the customer order if the payment not done shows pay now option | pass |
| 2 | Shows the coupon code entry option | if code entered is valied reduce the price else gives error message | pass |

**4.3.6 Payment**

****

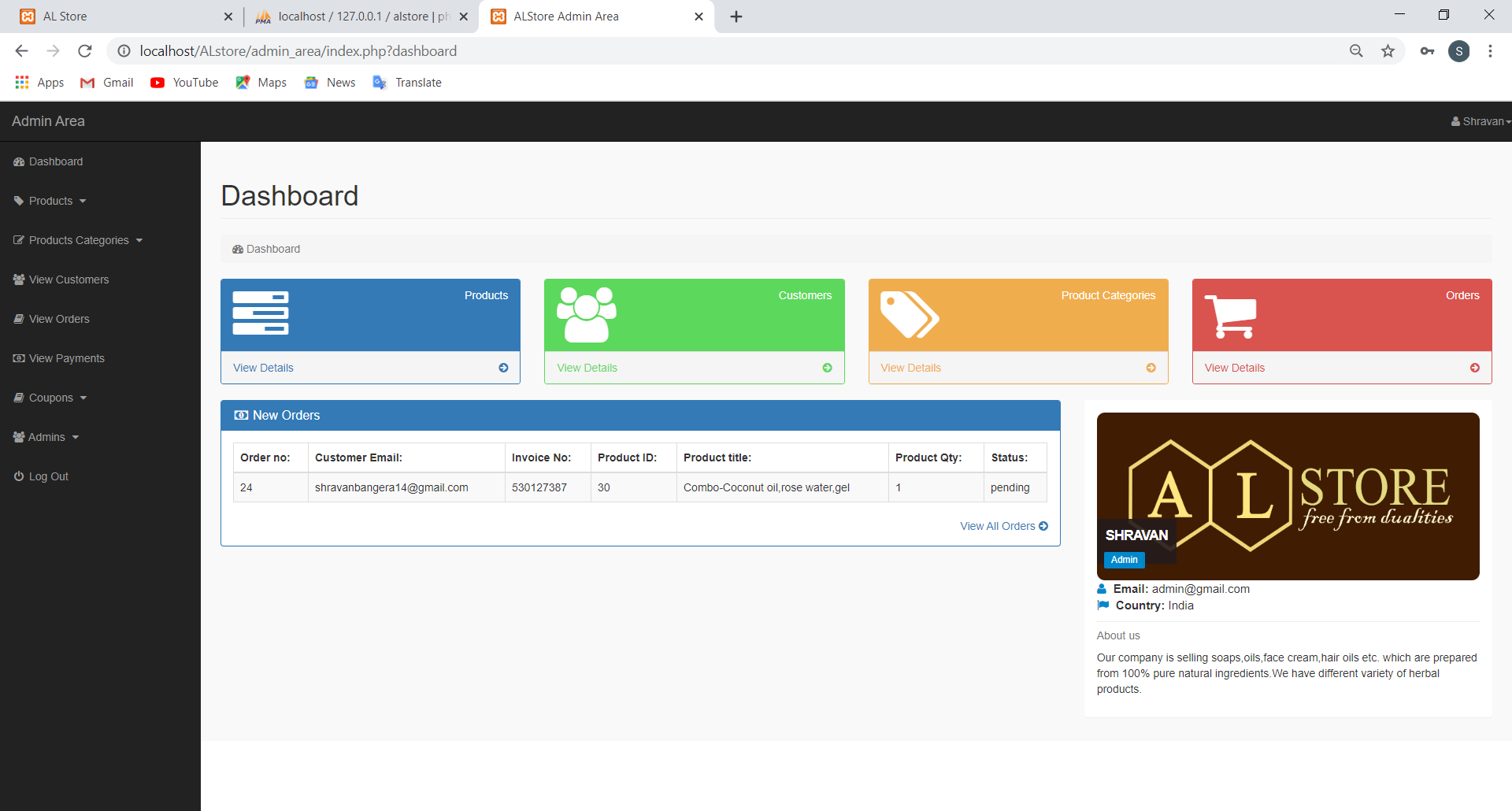
|  |  |  |  |
| --- | --- | --- | --- |
| **Sl.No** | **User Action** | **Expected Result** | **Test Result** |
| 1 | On click pay now button | shows the payment option | pass |
| 2 | Payment | success shows next page else my oder page | pass |

**4.3.7 Admin login**



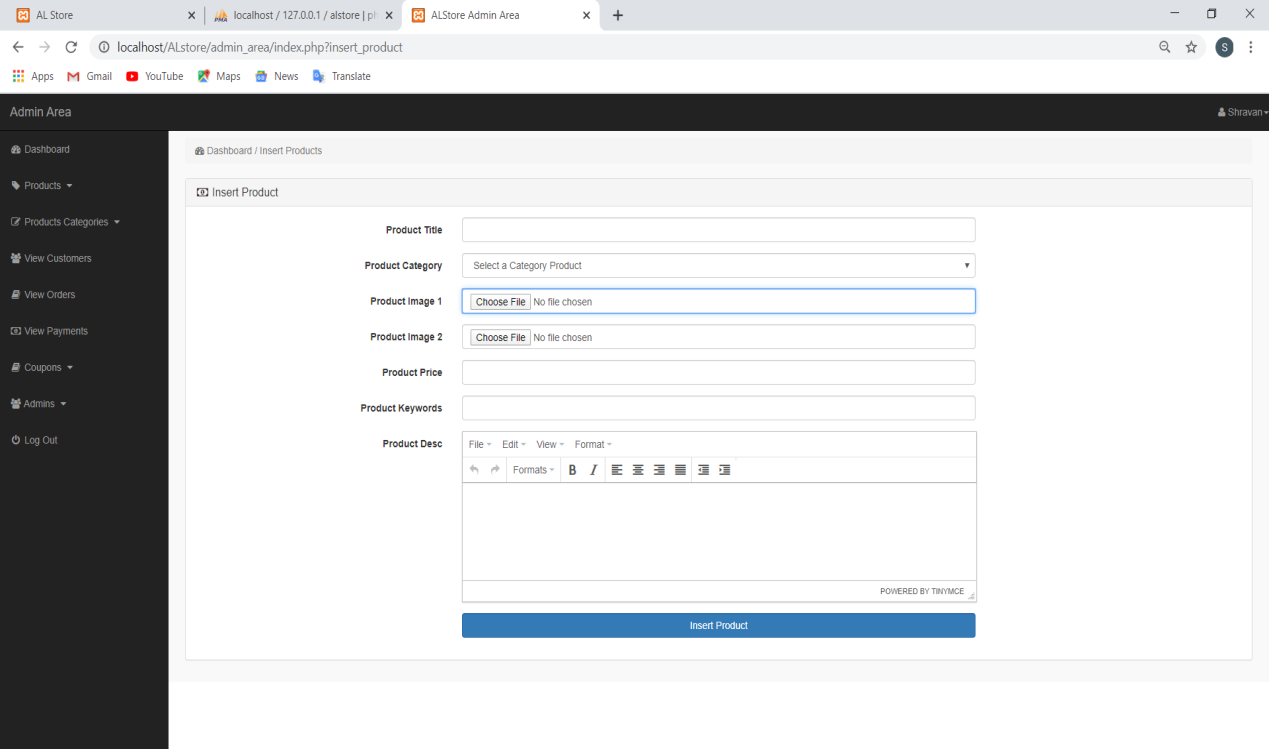
|  |  |  |  |
| --- | --- | --- | --- |
| **Sl.No** | **User Action** | **Expected Result** | **Test Result** |
| 1 | On load of the form | Displays the form | Pass |
| 2 | if User leaves the fields empty or enters invalid details and clicks login in button | Appears a message with an error message for respective field | Pass |
| 3 | On click of Login button | If the user leaves any field empty or enters invalid detail an error message will appear otherwise allows the user to login to the system | Pass |

**4.3.8 admin dash board**



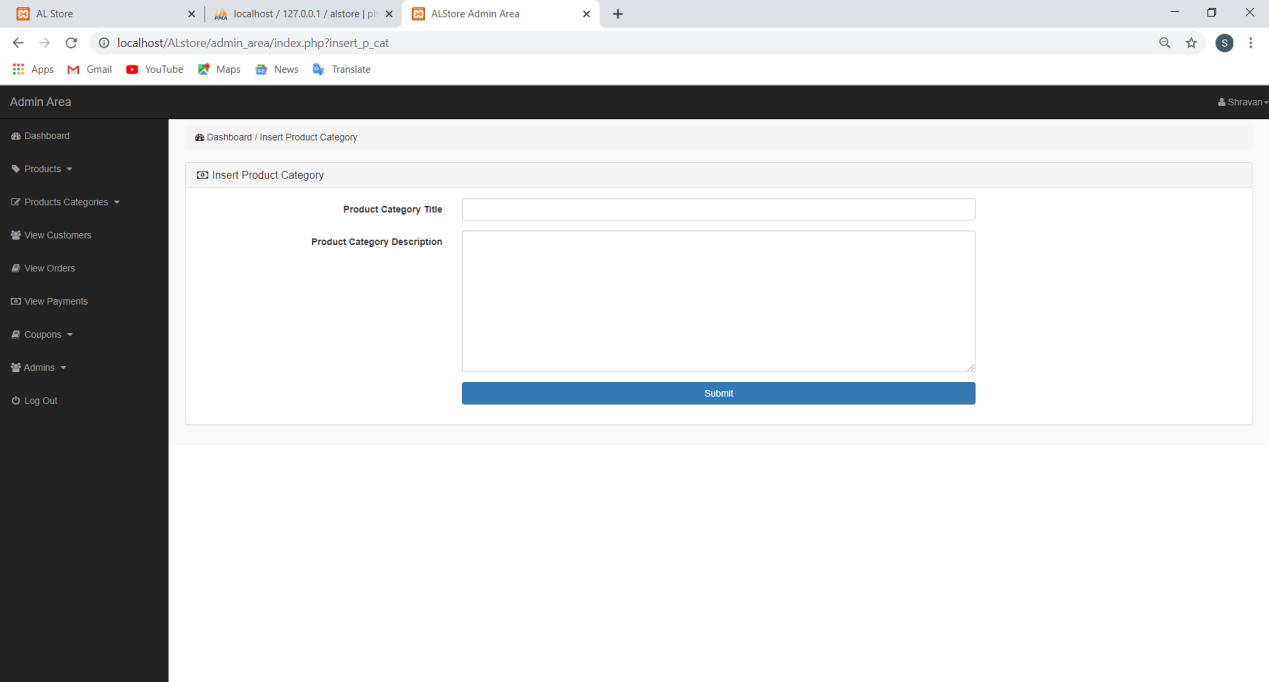
|  |  |  |  |
| --- | --- | --- | --- |
| **Sl.No** | **User Action** | **Expected Result** | **Test result** |
| 1 | Onload after login | shows admin dash board | pass |

**4.3.9 Product**

****

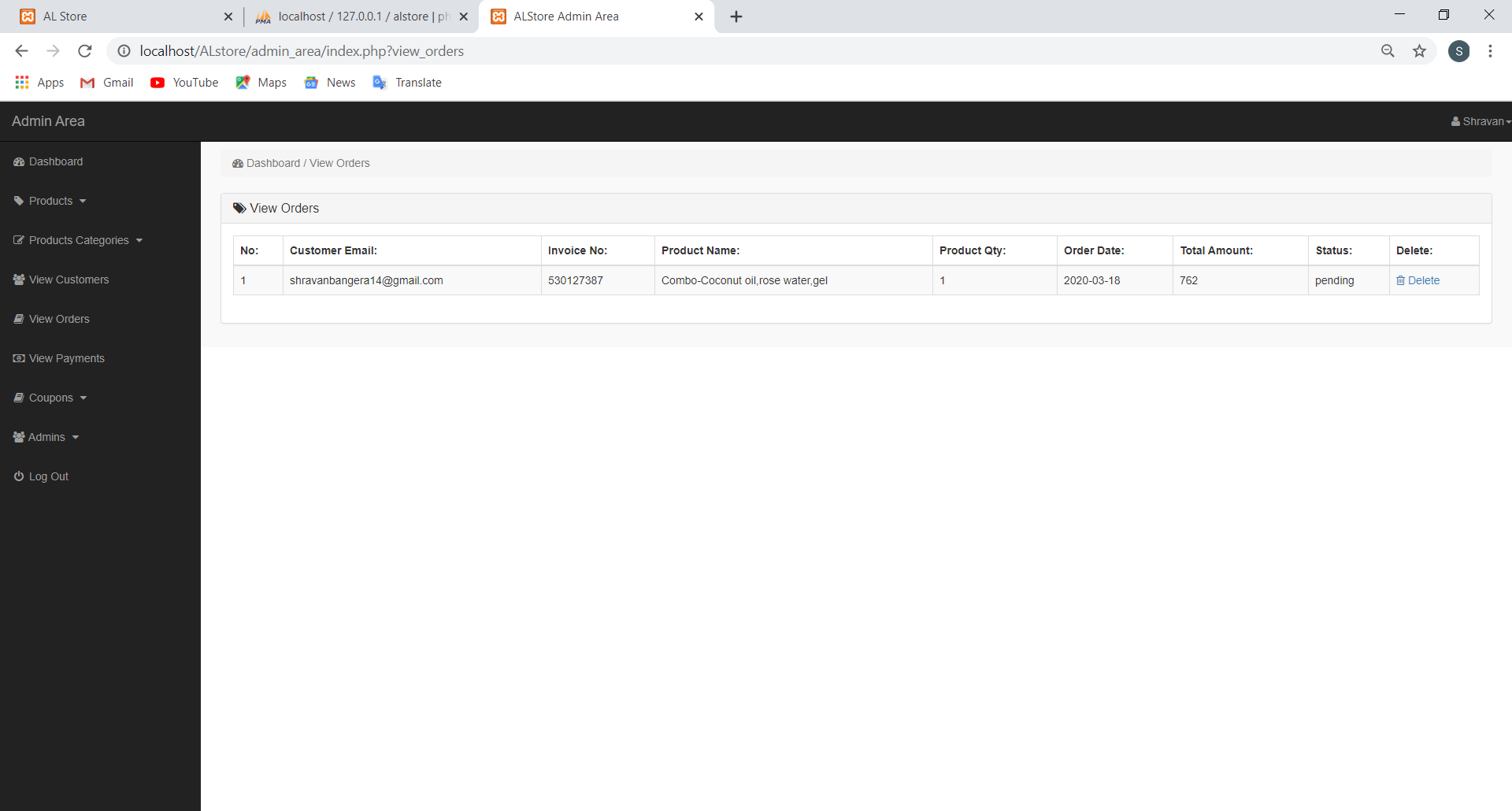
|  |  |  |  |
| --- | --- | --- | --- |
| **Sl.No** | **User Action** | **Expected Result** | **Test Result** |
| 1 | On load of the form | Displays the form to insert | pass |
| 2 | if User leaves the fields empty or enters invalid details and clicks insert button | Appears a message box with an error message for respective field | pass |
| 3 | If all details true and insert button clicked | Item added to database | pass |

**4.3.10 Product category**



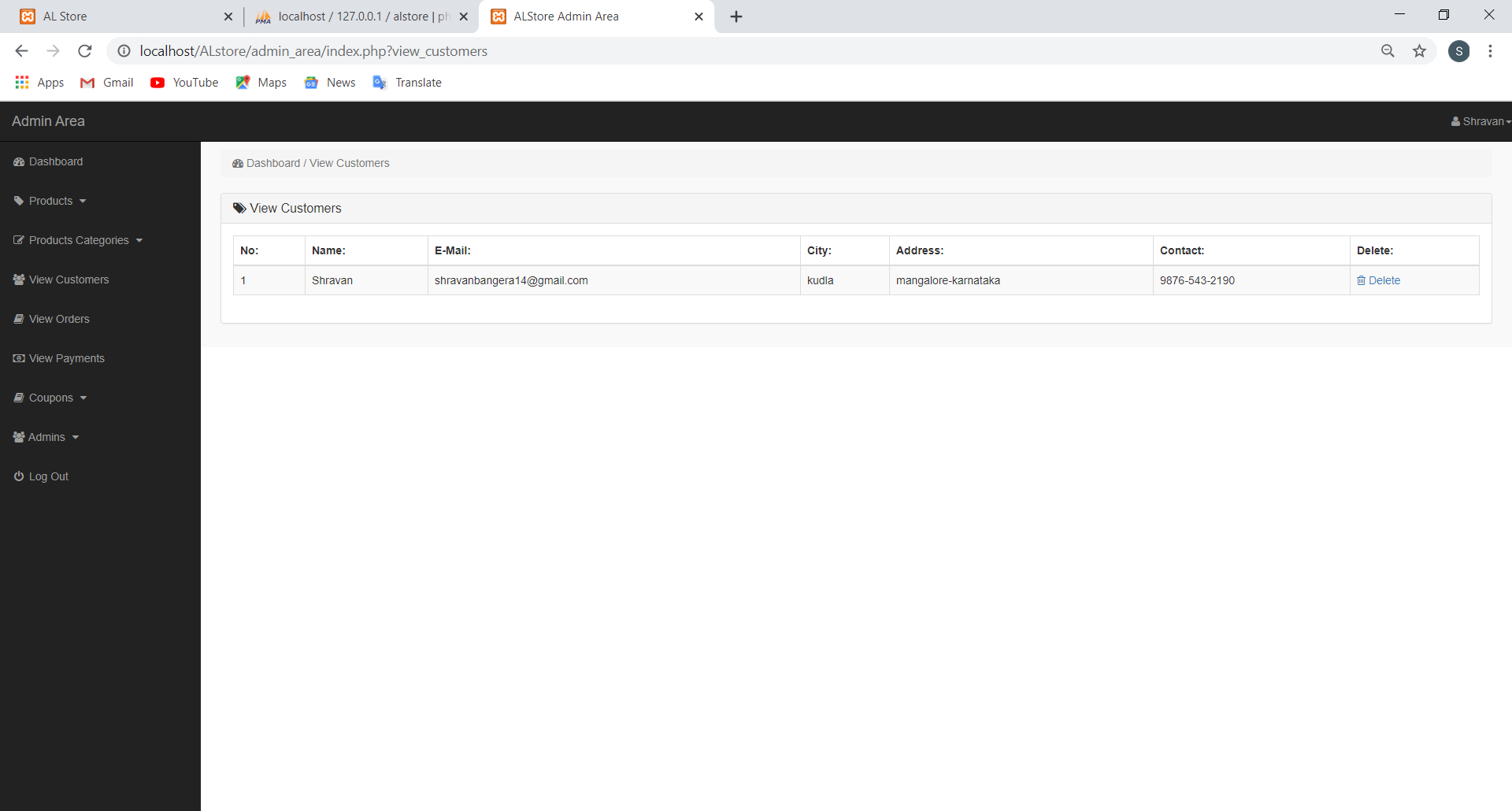
|  |  |  |  |
| --- | --- | --- | --- |
| **Sl.No** | **User Action** | **Expected Result** | **Test Result** |
| 1 | On load of the form | Displays the form to insert | pass |
| 2 | if User leaves the fields empty or enters invalid details and clicks insert button | Appears a message box with an error message for respective field | pass |
| 3 | If all details true and insert button clicked | category added to database | pass |

**4.3.11 View orders**



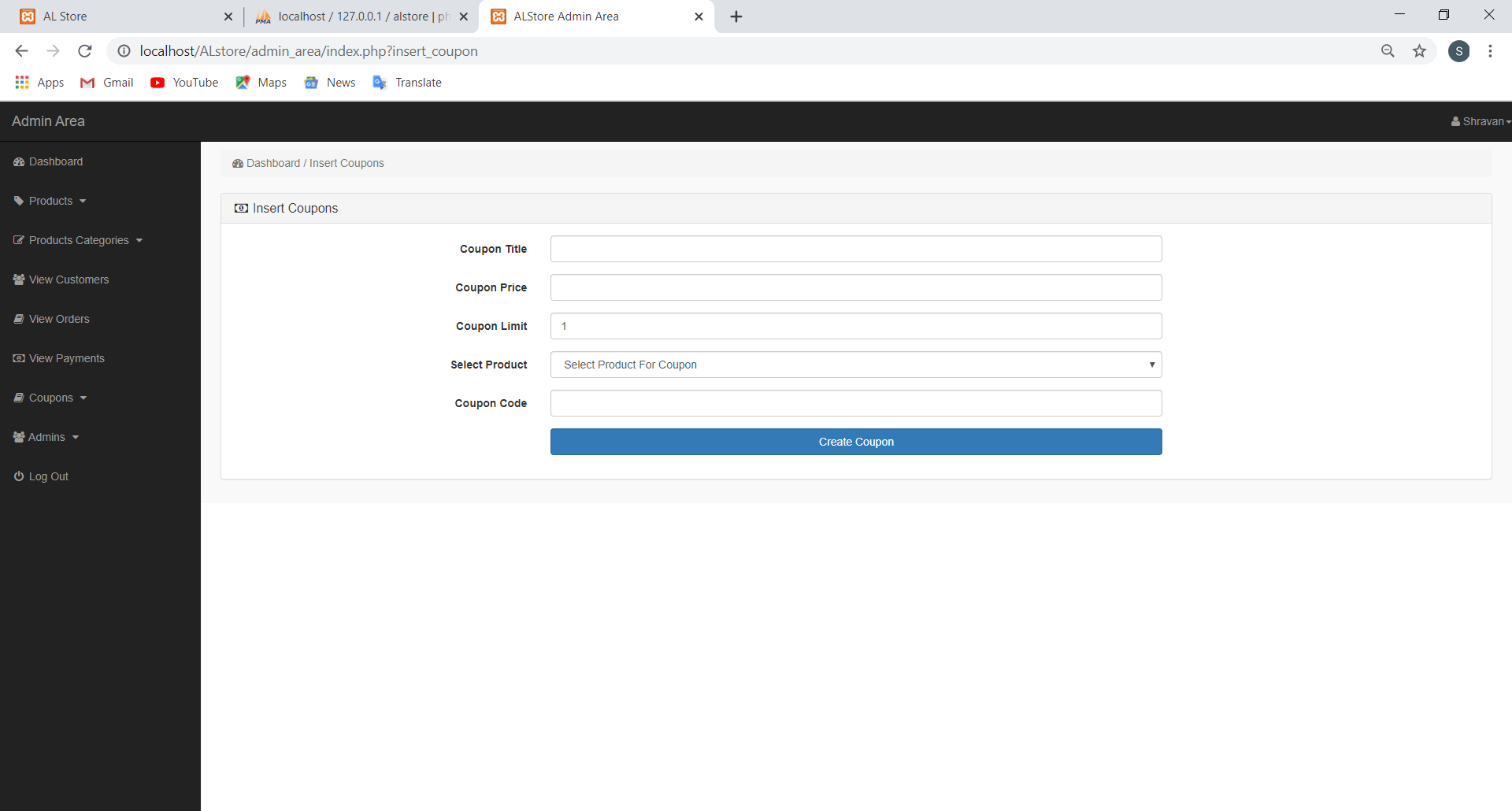
|  |  |  |  |
| --- | --- | --- | --- |
| **Sl.No** | **User Action** | **Expected Result** | **Test Result** |
| 1 | Clicks on view oder | Loads page with all order details | pass |

**4.3.12 View customer**

****

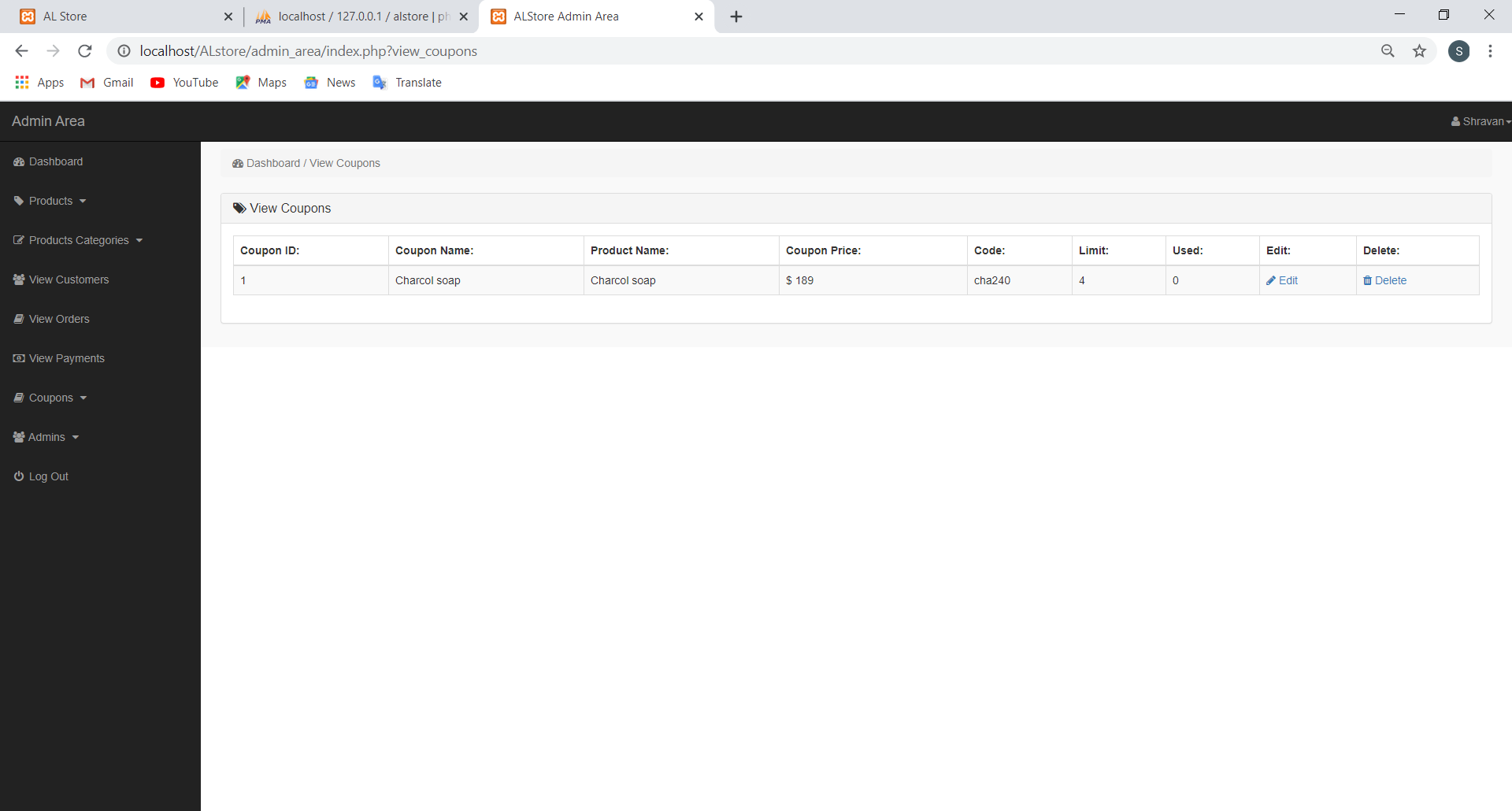
|  |  |  |  |
| --- | --- | --- | --- |
| **Sl.No** | **User Action** | **Expected Result** | **Test Result** |
| 1 | Clicks on view customer | Loads page with all customer details | pass |

**4.3.13 Coupon**

****

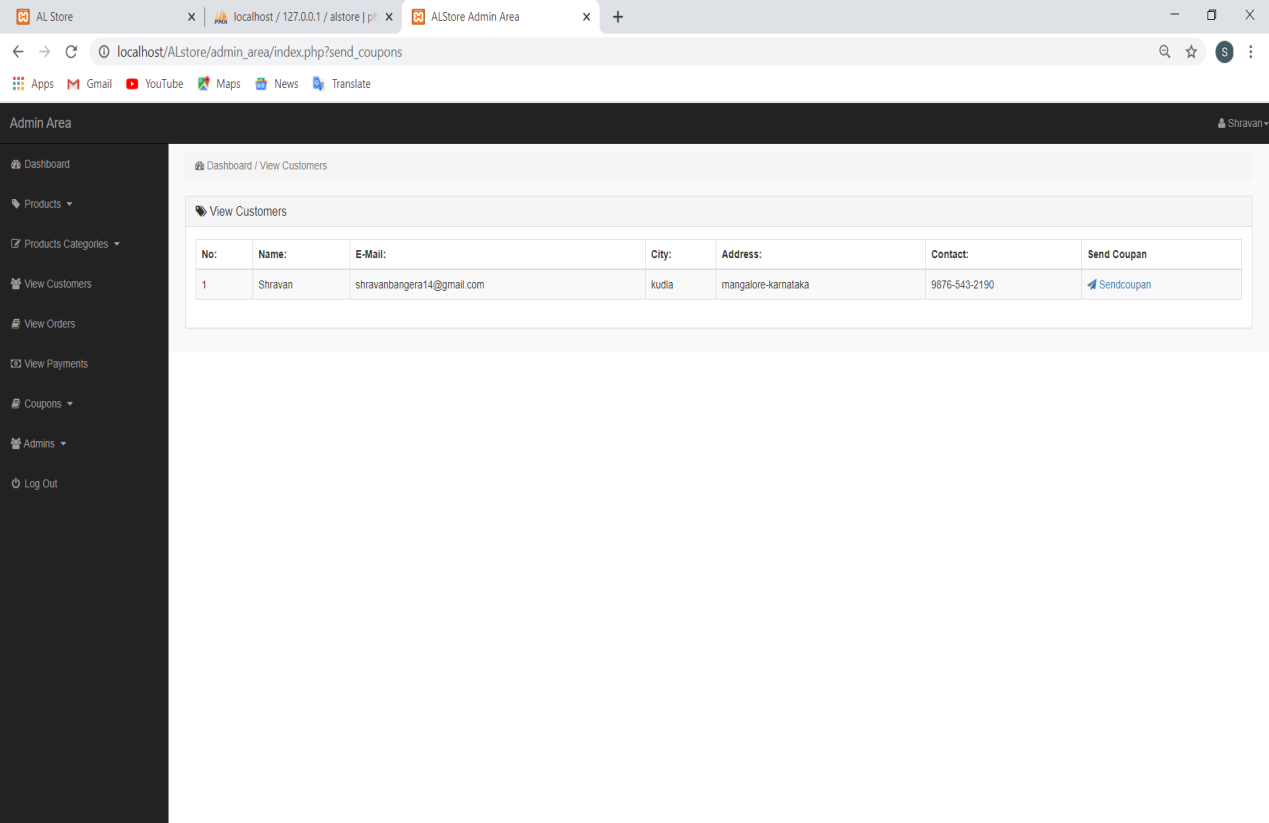
|  |  |  |  |
| --- | --- | --- | --- |
| **Sl.No** | **User Action** | **Expected Result** | **Test Result** |
| 1 | On load of the form | Displays the form to insert | pass |
| 2 | if User leaves the fields empty or enters invalid details and clicks insert button | Appears a message box with an error message for respective field | pass |
| 3 | If all details true and insert button clicked | category added to database | pass |

**4.3.14 View coupon**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl.No** | **User Action** | **Expected Result** | **Test Result** |
| 1 | Clicks on view coupon | Loads page with all coupon details | pass |

**4.3.15 Send coupon**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl.No** | **User Action** | **Expected Result** | **Test Result** |
| 1 | Clicks on send coupon | Loads page with all coupon details | pass |
| 2 | Clicks on send button | Coupon mailed to customer | pass |

**SYSTEM SECURITY**

**5. SYSTEM SECURITY**

**5.1 INTRODUCTION**

System Security encompasses many different techniques and methods which ensure safety from threats and attacks.

**5.2 SOFTWARE SECURITY**

The proposed website is a secured website for the ADVAITA LIFESTYLE. There are two different categories of users they are admin and customer. Depending upon the category of user the access rights are decided. It means if the user is an admin then he can be able to add, delete and edit the records in data base. Authorized customer can browse inventory, buy item. Admin can access the full site.

**CONCLUSION**

**6. CONCLUSION**

The Project **“Advaita Lifestyle”** was successfully tested and implemented under windows 10 and most of the Web Browsers including Google Chrome, Mozilla, Internet Explorer 8.0 etc. This Website has been developed to satisfy the needs of the customer.

The Website developed is more users friendly and interactive. The Website is found to be really beneficial for the concerned aspects. The performance of the software is provided efficiently. This software has been tested with all possible sample data and was found to function efficiently. Website Developed is futuristic and secure. The users may add any modules in future without affecting the current Project.

**FUTURE ENHANCEMENTS**

**7. FUTURE ENHANCEMENTS**

Additional features can be added easily based on requirements.

An android application can be built.

**BIBLIOGRAPHY**

**8. BIBLIOGRAPHY**

Websites Referred

<http://en.wikipedia.org/wiki/Software_Requirements_Specification>.

<https://www.smartdraw.com/entity-relationship-diagram/>

<https://www.javatpoint.com/software-engineering-data-flow-diagrams>

<https://jqueryvalidation.org/documentation/>

<https://razorpay.com/docs/payment-gateway/web-integration/>