ONLINE SHOPPING MANAGEMENT

ABSTRACT

We have witnessed the fall of a number of enterprises, in the wake of the global pandemic, covid-19. People, in multitudes, have lost the security and comfort of their jobs, and have been left to fend for themselves. But in these miserable times, we have seen a number of entrepreneurs rise up. Homemade masks, sweets and many other similar small businesses have taken birth.

In this project, we have created an Online Shopping website which provides various functionalities to certain actors. For example :

The admin and the seller can add categories and their corresponding products on the website. The admin can also look after the administration of the website.

The customers/normal users can browse the catalog of categories which contain the products of interest. The catalog consists of various categories and their corresponding products. The selected items are collected in a shopping cart. The shopping cart displays the total price of the items selected by the user.

In order to avail the above-mentioned shopping experience, primarily, the user has to get himself registered on the store. Wherein, the user will have to provide personal details such as first name, last name, contact no., email address, residential / office address, etc. Furthermore, set an appropriate username and password.

LIST OF CONTENTS

Chapter 1

INTRODUCTION

1.1 INTRODUCTION

"It is not just about ideas, it's also about making ideas happen"

We have witnessed the fall of a number of enterprises, in the wake of the global pandemic, covid-19. People, in multitudes, have lost the security and comfort of their jobs, and have been left to fend for themselves. But in these miserable times, we have seen a number of entrepreneurs rise up. Homemade masks, sweets and many other similar small businesses have taken birth. Also keeping in mind the Vocal For Local initiative his project aims to provide these new entrepreneurs with a platform to widen their reach. Also this pandemic has seen the rise of shopping via the internet. This project also aims to reach out to such people so that they can shop at the comfort of their home as well as support these cottage industries.

1.2 PURPOSE

Most of these enterprises are operating from people's homes. The majority of the entrepreneurs are women. In most cases these people do not have good ideas about technology and how they can use it to expand their business. In almost all cases, the production, management, book-keeping etc is being done by only one person. This project will take the burden of management, off the shoulders of the entrepreneurs. Their only job would be maintaining and improving production quality.

Chapter 2

LITERATURE SURVEY

2.1 Introduction to Database Management System :

DBMS software primarily functions as an interface between the end user and the database, simultaneously managing the data, the database engine, and the database schema in order to facilitate the organization and manipulation of data.

Though functions of DBMS vary greatly, general-purpose DBMS features and capabilities should include: a user accessible catalog describing metadata, DBMS library management system, data abstraction and independence, data security, logging and auditing of activity, support for concurrency and transactions, support for authorization of access, access support from remote locations, DBMS data recovery support in the event of damage, and enforcement of constraints to ensure the data follows certain rules.

Application of DBMS :

The application of database are many and varied; it can be divided into four major areas :

1. Hierarchical and network system
2. Flexibility with relational database
3. Object oriented application
4. Interchanging the data on the web for ecommerce

Display information

In this particular project, we have taken Html web page as a front end in order to display the information which is stored in the backend database. HTML stands for Hyper Text Markup Language. HTML describes the structure of web pages. Browsers do not display tags but use them to render the content of the page.

2.2 JSP :

Java Server Pages (JSP) is a technology that helps software developers create dynamically generated web pages on HTML,XML or other document types. Released in 1999 by Sun Microsystems, JSP is similar to PHP and ASP but it uses th eJava programming language . To deploy and run java server pages a compatible web server with servlet container such as Apache Tomcat or Jetty is required.

2.3 Hibernate :

Hibernate is a Java framework that simplifies the development of Java applications to interact with the database. It is an open source, lightweight, ORM (Object Relational Mapping) tool. Hibernate implements the specifications of JPA (Java Persistence API) for data persistence.

Advantages of Hibernate Framework :

Following are the advantages of hibernate framework:

1) Open Source and Lightweight

Hibernate framework is open source under the LGPL license and lightweight.

2) Fast Performance

The performance of hibernate framework is fast because cache is internally used in hibernate framework. There are two types of cache in hibernate framework: first level cache and second level cache. First level cache is enabled by default.

3) Database Independent Query

HQL (Hibernate Query Language) is the object-oriented version of SQL. It generates the database independent queries. So you don't need to write database specific queries. Before Hibernate, if the database is changed for the project, we need to change the SQL query as well, which leads to the maintenance problem.

4) Automatic Table Creation

Hibernate framework provides the facility to create the tables of the database automatically. So there is no need to create tables in the database manually.

5) Simplifies Complex Join

Fetching data from multiple tables is easy in hibernate framework.

6) Provides Query Statistics and Database Status

Hibernate supports Query cache and provides statistics about query and database status.

Chapter 3

PROPOSED METHODOLOGY

3.1 Actor Wise Functionalities :

Admin :

Keeping in mind that this website is for cottage industries and hence assuming that the number of sellers will be less the admin can perform the following functionalities

- Can view the total number of users
- Can view the total number of categories
- Can view the total number of products
- Can add the categories which can then be used to sort the products using categories and thus will be easier for the buyer to look for the products that he/she is buying
- Can add new products by filling in details like., product title, product description, product price, product discount, product quantity, product images (can only choose one picture but the project can be expanded as and when needed)and enter the category

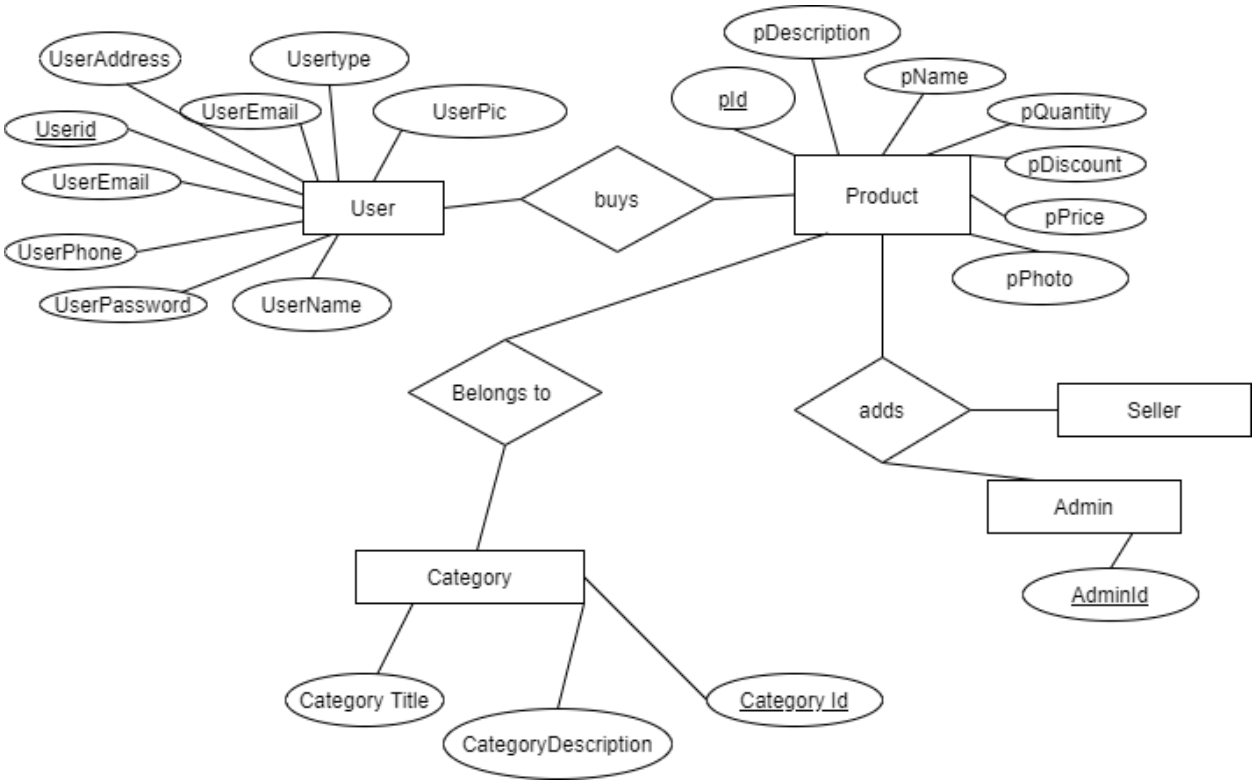
Buyer :

- Can login to the website using username and password
- Can register to the website using username, email, password, phone number and address
- Can purchase desired products from desired categories

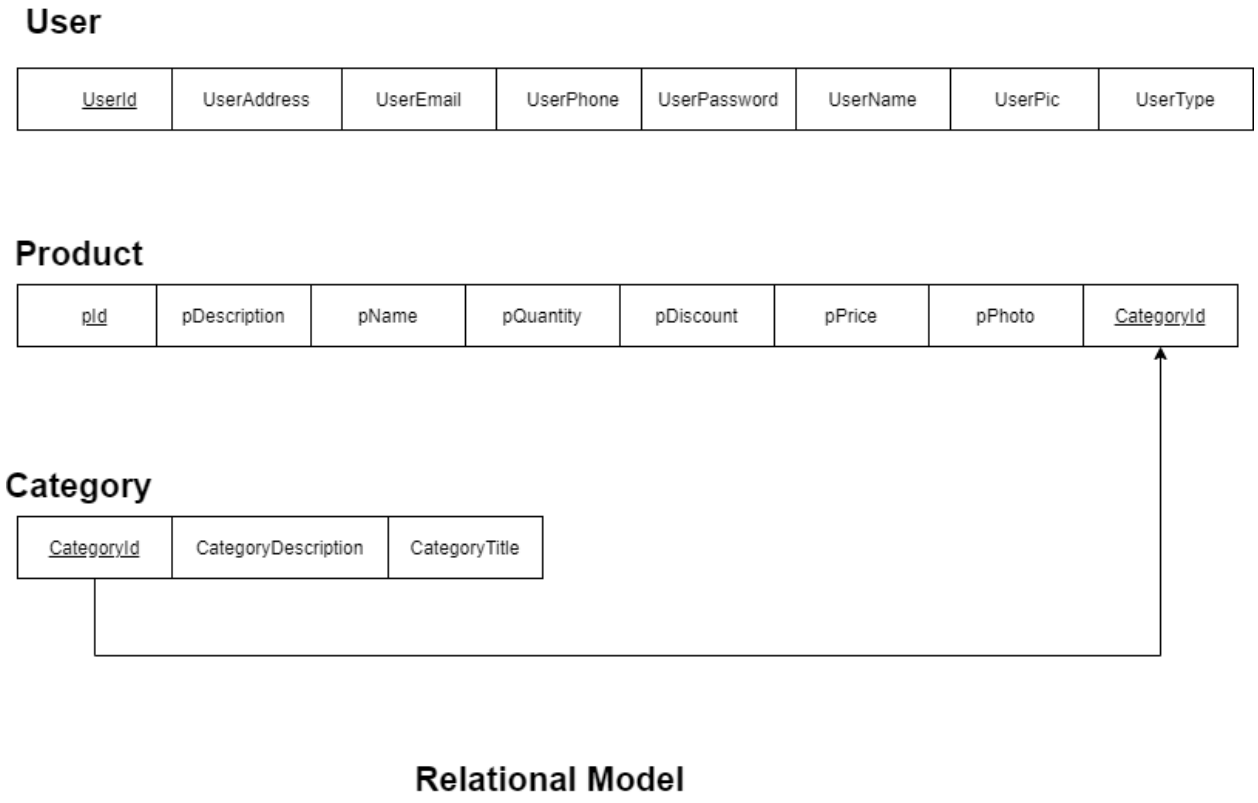
Seller :

- Can add new categories to the website.
- Can add new products to the categories specified.

3.2 ER Diagram :



3.3 Relational Model :



Relational Model

Chapter 4

HARDWARE

4. REQUIREMENT ANALYSIS

4.1 Requirement Specifications :

The requirement for the efficient working of the project are as follows

4.1.1 Front End:

1. HTML , CSS, JS
2. Bootstrap

4.1.2 Backend:

1. Java
2. JSP, Servlets
3. Hibernate

4.1.3 Other requirements :

1. Java IDE(Eclipse)
2. MySQL Workbench
3. Tomcat Server

Chapter 5

IMPLEMENTATION

5.1 Database

User :

user_id	userAddress	userEmail	user_name	user_password	user_phone	user_pic	user_type
1	Mumbai	jess@gmail.com	Jessica	jess	9876543210	default.jpg	admin
2	Delhi	sakshi@gmail.com	Sakshi	sakshi	9768540321	default.jpg	normal
3	Agra	liny@gmail.com	Liny	liny	9087563412	default.jpg	normal
4	Kolkata	ames@gmail.com	Ameaza	ames	9908765432	default.jpg	normal
5	Delhi	joe@gmail.com	Joe	joe	9911223344	default.jpg	normal
6	Mumbai	sam@gmail.com	Sam	sam	9087654321	default.jpg	normal
8	Mumbai	dan@gmail.com	Dan	dan	9606543210	default.jpg	seller

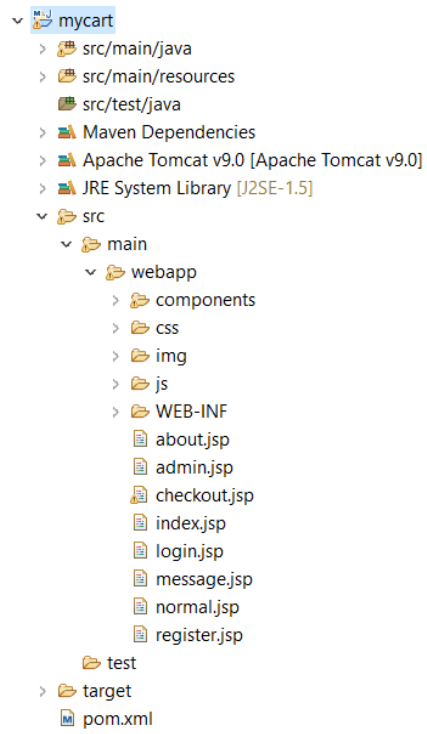
Category :

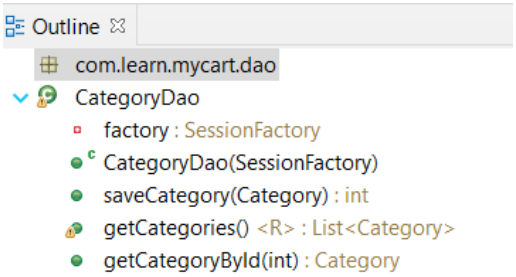
categoryId	categoryDescription	categoryTitle
8	This category contains hand-made masks.	Masks
9	This category contains home-made cakes.	Cakes
10	This category contains home-garden flowers.	Flowers
11	This category contains hand-made jewellery.	Jewellery
12	This category contains home-made pickles.	Pickles
13	This category contains clothes	Clothes
NULL	NULL	NULL

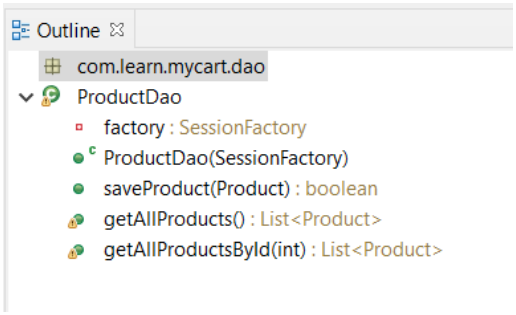
Product :

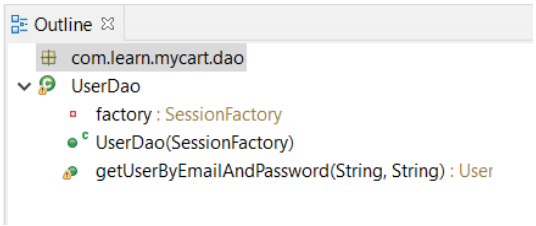
pId	pDescription	pDiscount	pName	pPhoto	pPrice	pQuantity	category_categoryId
1	Men Pink Printed 3-Ply Reusable Cloth Mask	10	RARE RABBIT	mask.png	200	1	8
2	Men Pink Printed 3-Ply Reusable Cloth Mask	10	RARE RABBIT	mask.png	200	1	8
3	Half Kg	10	Fudge Brownie Cake	cake.png	600	1	9
4	Arrangement of 20 Red and Pink Roses in White...	10	Red N Pink Rose Bouquet	flowers.png	900	1	10
6	Bunch of 10 Pink Roses	10	Pinky Promise Rose Bouquet	flowers(2).png	499	1	10
5	This pendant in sparkling zircon features the mo...	5	Silver Zircon Pendant with Link Chain	jewellery.png	999	1	11
7	This is a home-made pickle made with boneless ...	5	HOMEMADE CHICKEN PICKLE	pickle.png	200	1	12
8	Denim jacket	5	Fashion Classic Stitching Denim Jacket	clothes.png	500	1	13
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

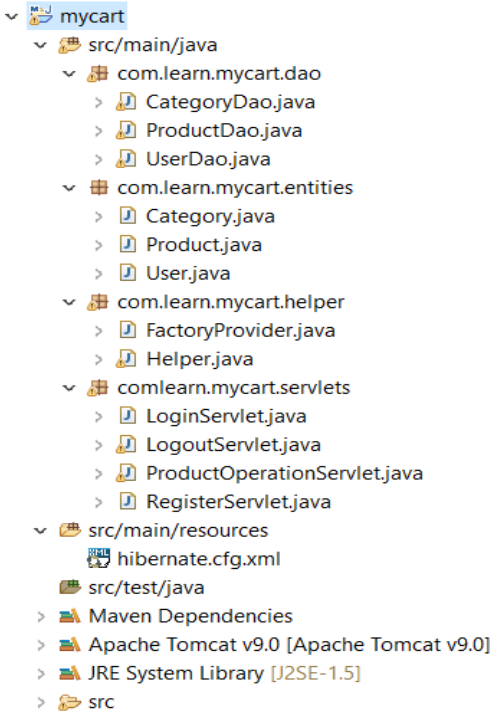
5.2 Project Analysis



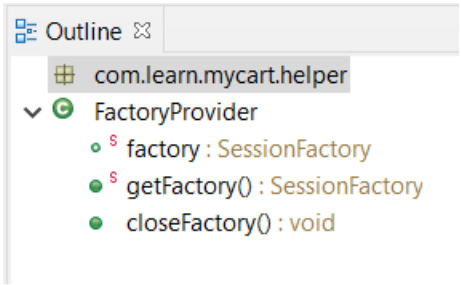


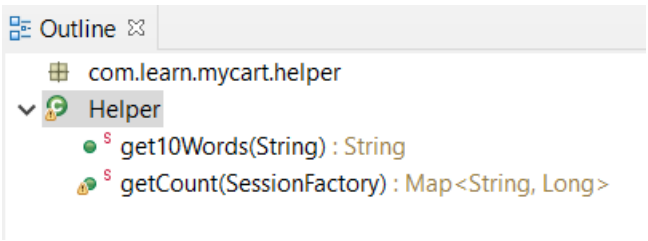




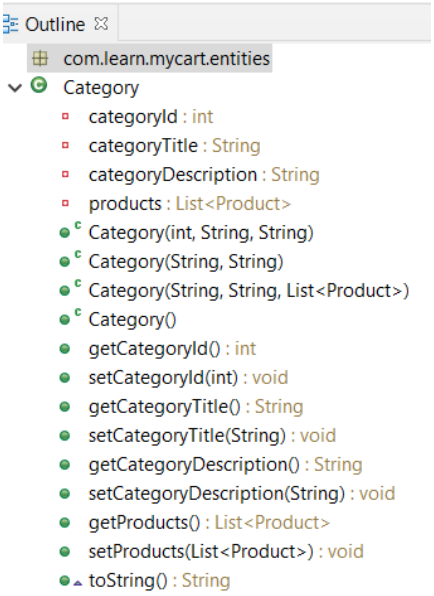


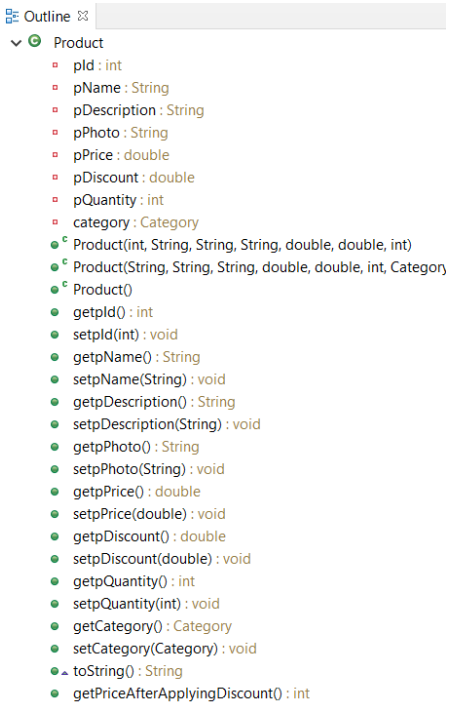
com.learn.mycart.helper

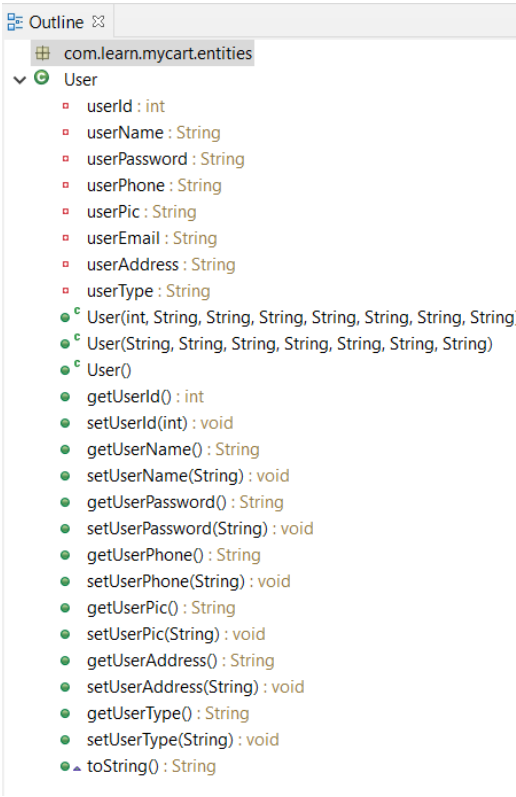




com.learn.mycart.entities







5.3 Hibernate queries :

1. Create database mycart;

   (creating database for the project)

2. Connecting our project to mysql by entering username,password and database name

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
        "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
        "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
    <session-factory>
        <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
        <property name="hibernate.connection.password">root</property>
        <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/mycart</property>
        <property name="hibernate.connection.username">root</property>
        <property name="hibernate.dialect">org.hibernate.dialect.MySQL8Dialect</property>
        <property name="hbm2ddl.auto">update</property>
        <property name="show_sql">true</property>
        <mapping class="com.learn.mycart.entities.User" />
        <mapping class="com.learn.mycart.entities.Category" />
        <mapping class="com.learn.mycart.entities.Product" />
        <mapping class="com.learn.mycart.entities.Recipe"/>
        <mapping class="com.learn.mycart.helper.FactoryProvider" />

    </session-factory>
</hibernate-configuration>
```

3. Creating entities and mapping them

User :

```
@Entity
public class User {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(length=1000,name = "user_id")
    private int userId;
    @Column(length=100,name = "user_name")
    private String userName;
    @Column(length=100,name = "user_password")
    private String userPassword;
    @Column(length=12,name = "user_phone")
    private String userPhone;
    @Column(length=1500,name = "user_pic")
    private String userPic;
    @Column(unique = true)
    private String userEmail;
    private String userAddress;
    @Column(name="user_type")
    private String userType;
```

Category :

```
@Entity
public class Category {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int categoryId;
    private String categoryTitle;
    private String categoryDescription;
    @OneToMany(mappedBy = "category")
    private List<Product> products = new ArrayList<Product>() ;
```

Product :

```
@Entity
public class Product {


    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int pId;
    private String pName;
    @Column(length=3000)
    private String pDescription;
    private String pPhoto;
    private double pPrice;
    private double pDiscount;
    private int pQuantity;
    @ManyToOne
    private Category category;
    public Product(int pId, String pName, String pDescription, String pPhoto, double pPrice, double pDiscount,
            int pQuantity) {
```

Queries to save category :

```
public int saveCategory(Category cat) {
    Session session = this.factory.openSession();
    Transaction tx = (Transaction) session.beginTransaction();
    Serializable catId = session.save(cat);
    try {
        tx.commit();
    }
    catch(Exception e){
        e.printStackTrace();
    }
    session.close();
    return (Integer)catId;

}
```

Queries to return a list of category from database :

```
public <R> List<Category> getCategories() {

    Session s = this.factory.openSession();
    Query<R> query = s.createQuery("from Category");
    List<Category> list = (List<Category>) query.list();
    return list;


}
```

Query to check if user exists in the database :

```
private SessionFactory factory;

public UserDao(SessionFactory factory) {
    this.factory = factory;
}

public User getUserByEmailAndPassword(String email,String password) {
    User user = null;
    try {


        String query = "from User where userEmail=:e and userPassword=:p";
        Session session = this.factory.openSession();
        Query q = session.createQuery(query);
        q.setParameter("e", email);
        q.setParameter("p", password);
        user=(User)q.uniqueResult();


        session.close();
    }catch(Exception e) {
        e.printStackTrace();
    }
return user;
}
```

Queries to save product into database and get all product from database :

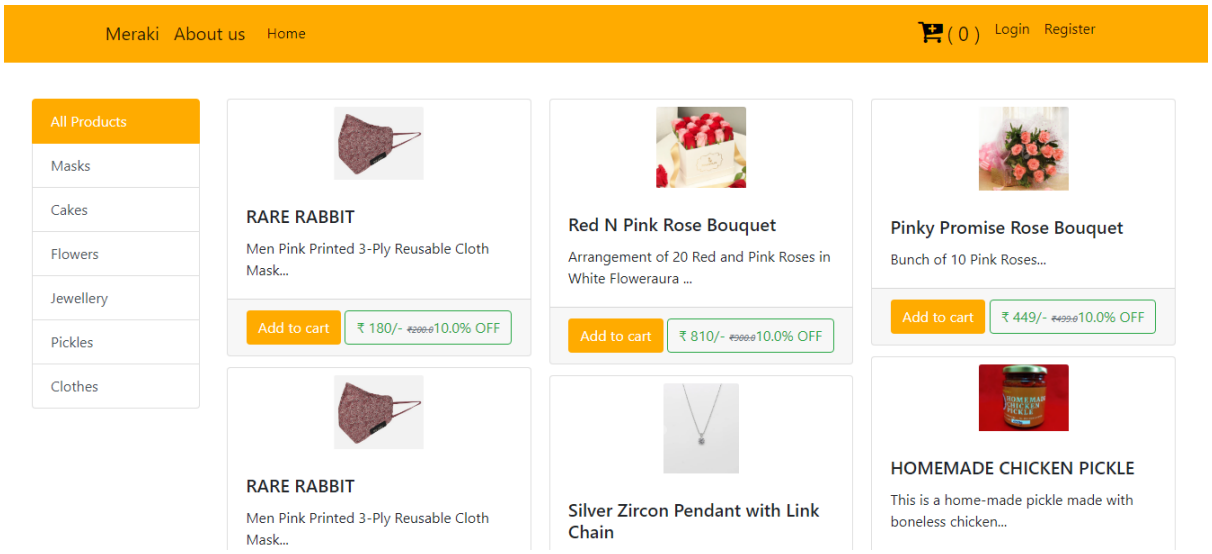
```
public boolean saveProduct(Product product) {
    boolean f=false;
    try {

        Session session = this.factory.openSession();
        Transaction tx = session.beginTransaction();
        session.save(product);
        tx.commit();
        session.close();
        return f;


    } catch(Exception e) {

        e.printStackTrace();

    }
    return f;
}

//get all products
public List<Product> getAllProducts(){

    Session s = this.factory.openSession();
    Query query = s.createQuery("from Product");
    List<Product> list = query.list();
    return list;

}
```

Some more queries

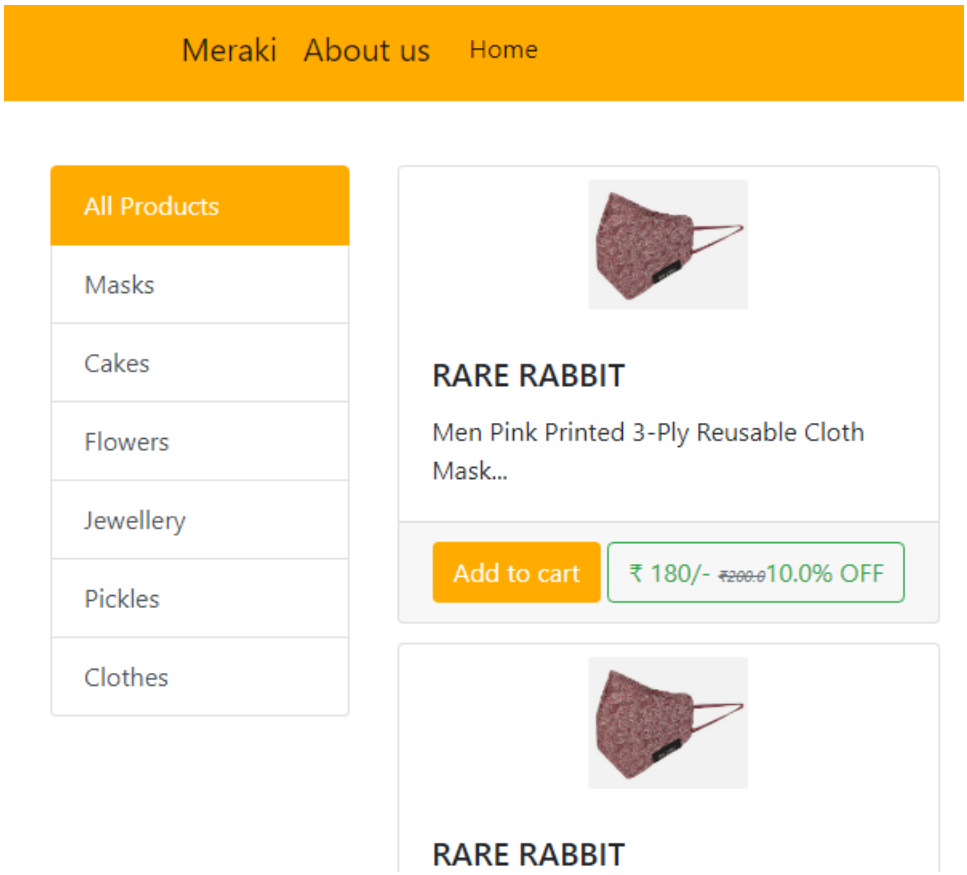
```
public static Map<String, Long> getCount(SessionFactory factory)
{
    org.hibernate.Session session =  factory.openSession();
    String q1="Select count(*) from User";
    String q2="Select count(*) from Product";

    Query query1 = session.createQuery(q1);
    Query query2 = session.createQuery(q2);
    Long userCount = (Long) query1.list().get(0);
    Long productCount = (Long) query2.list().get(0);

    Map<String, Long> map= new HashMap<String, Long>();
    map.put("userCount", userCount);
    map.put("productCount", productCount);

    session.close();
    return map;

}
```

5.4 Screenshots

Home Page



Shop By Category Page :



About us page :

Meraki About us Home (0) Login Register

About us

It is not just about ideas, it's also about making ideas happen

We have witnessed the fall of a number of enterprises, in the wake of The global pandemic, covid-19. People, in multitudes, have lost the Security and comfort of their jobs, and have been left to fend for themselves. But in these miserable times, we have seen a number of Entrepreneurs rise up. Homemade masks, sweets and many other Similar small business have taken birth. This project aims to provide These new entrepreneurs with a platform to widen their reach..

Mission Statement

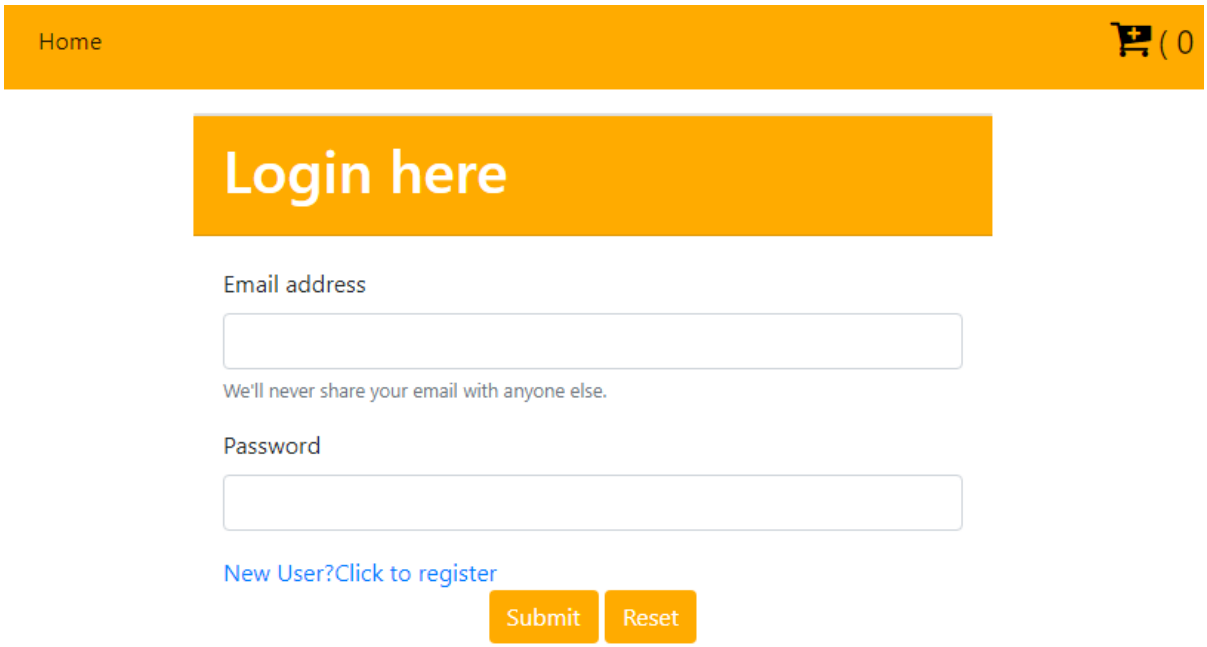
Most of these enterprises are operating from peoples homes. A Majority of the entrepreneurs are women. In almost all cases, the Production, management, book-keeping etc is being done by only one Person. This project will take the burden of management, off the Shoulders of the entrepreneurs. Their only job would be maintaining / improving production quality.

Our Team

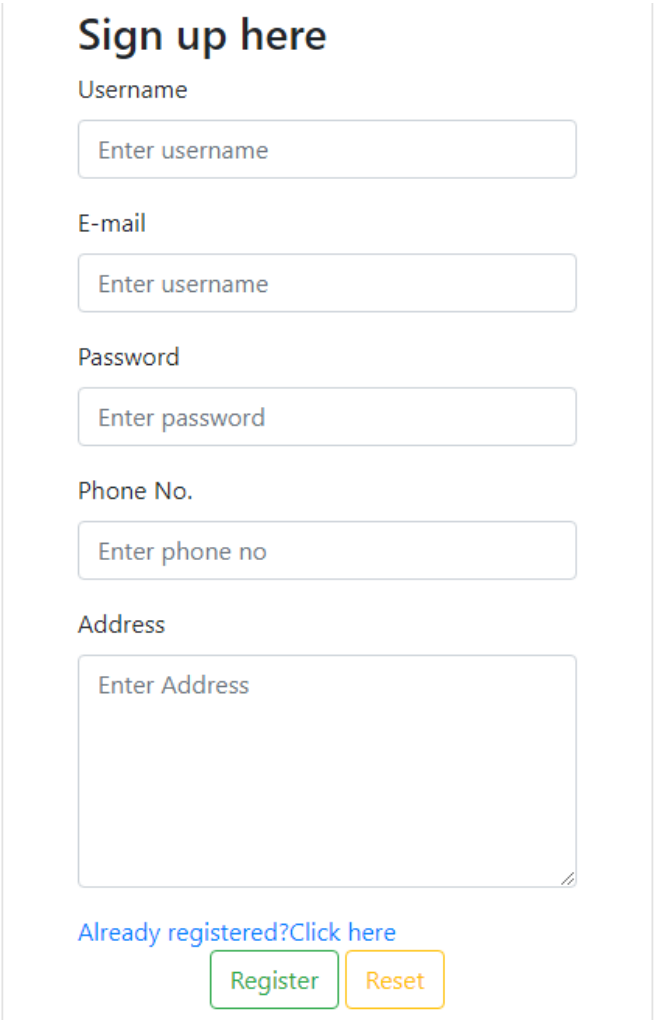
J	S	A
Jesica Dsouza 8933 SE COMPS B	Sakshi Shetty 8972 SE COMPS B	Ameaza Rodrigues 8967 SE COMPS B

Login Page :

User can login by entering email address and valid password :

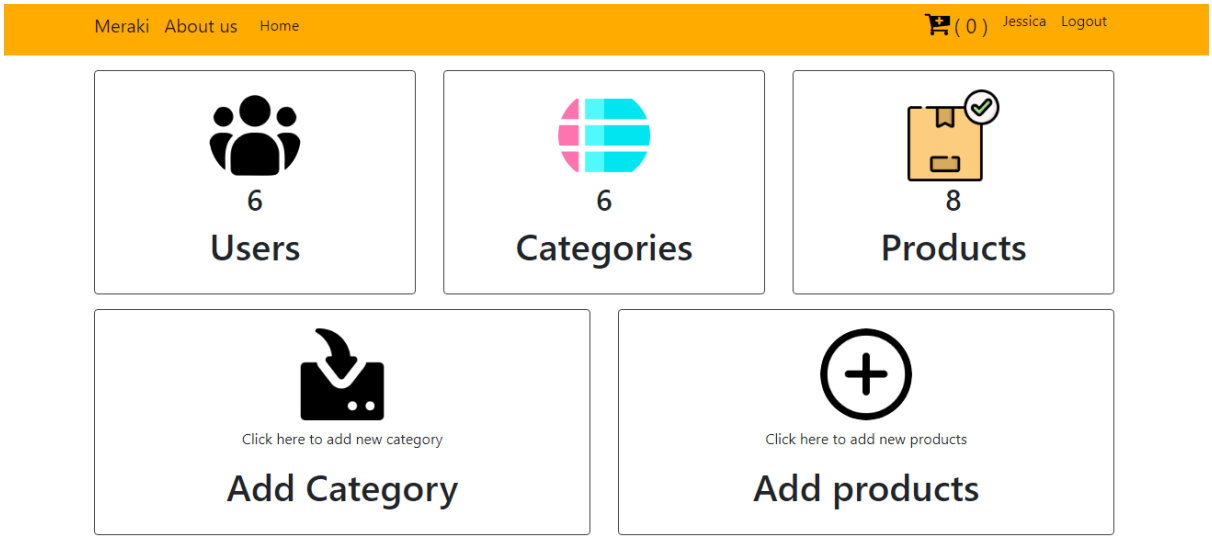


Registration Page :

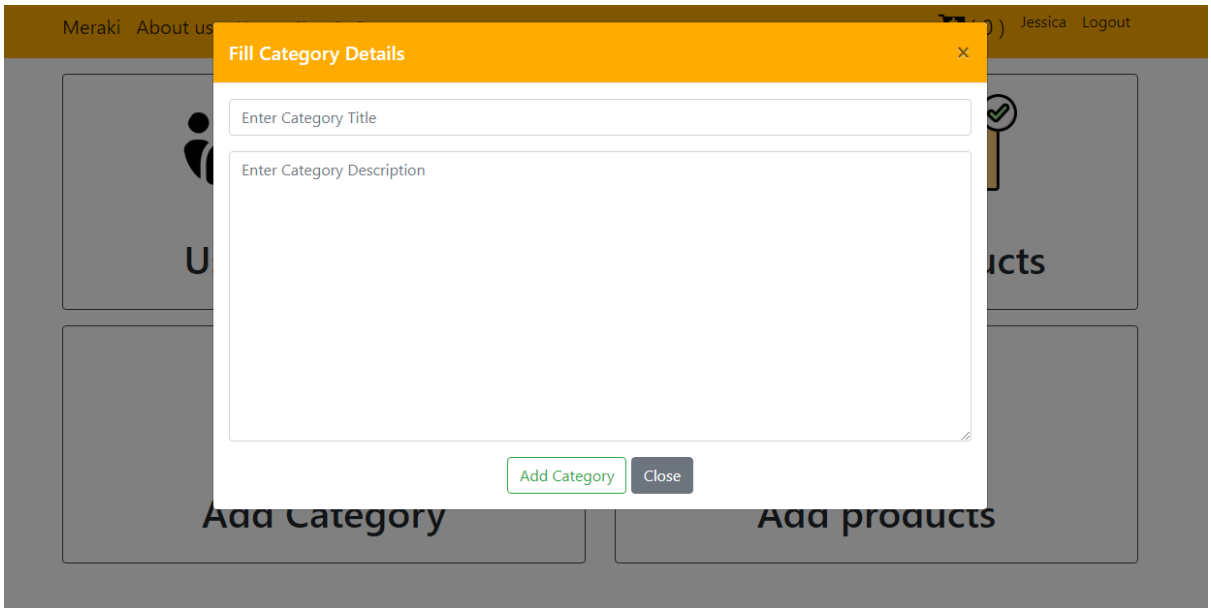


ADMIN PAGE :

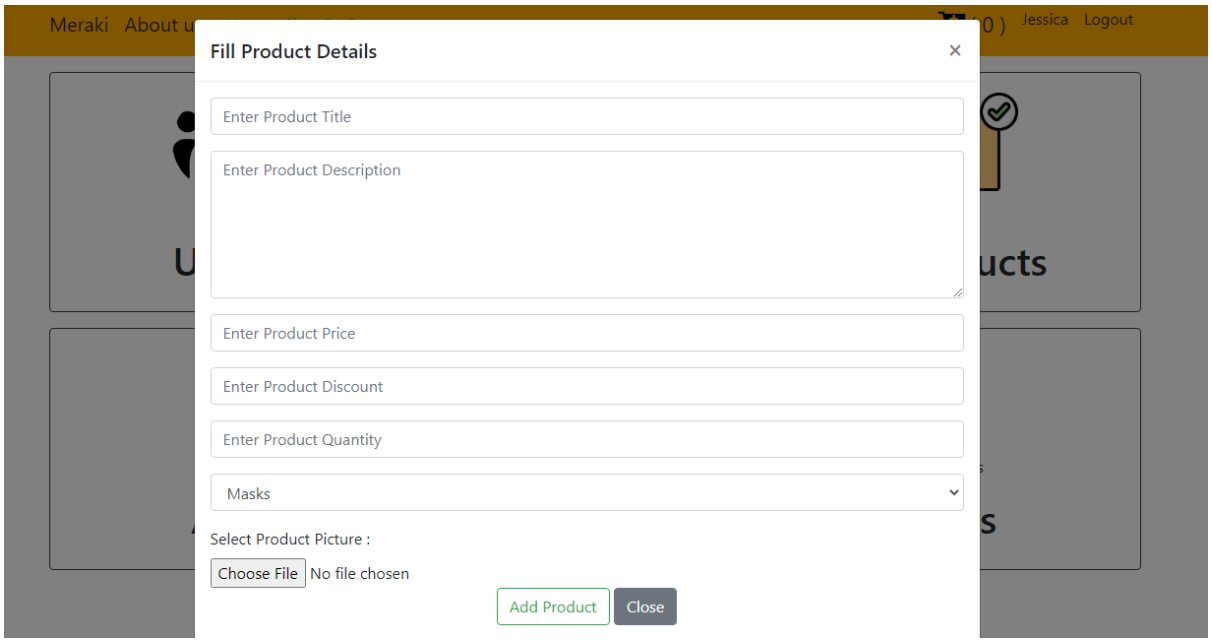
(Here the admin can view the actual count of the users, categories and products. Also, add new categories & products)



Add Category form :

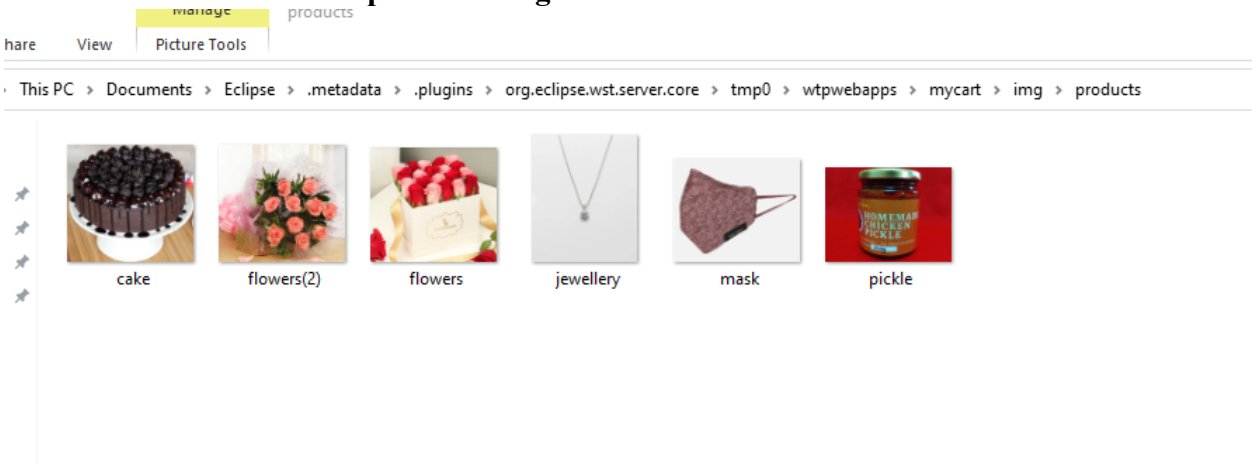


Add Product Form :

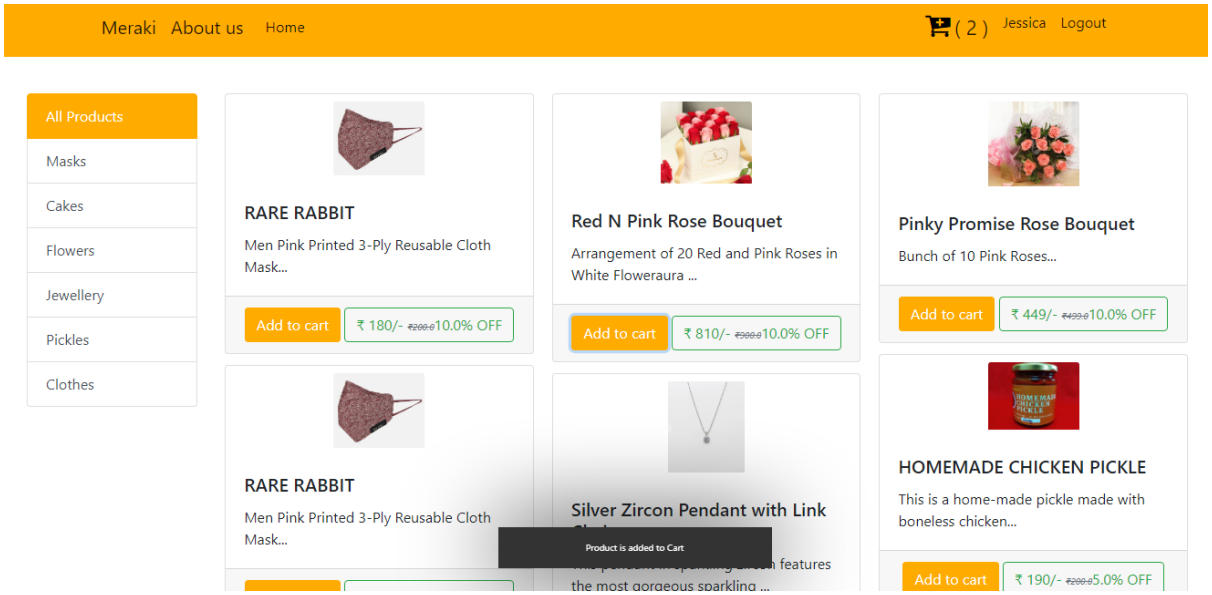


Seller Page:

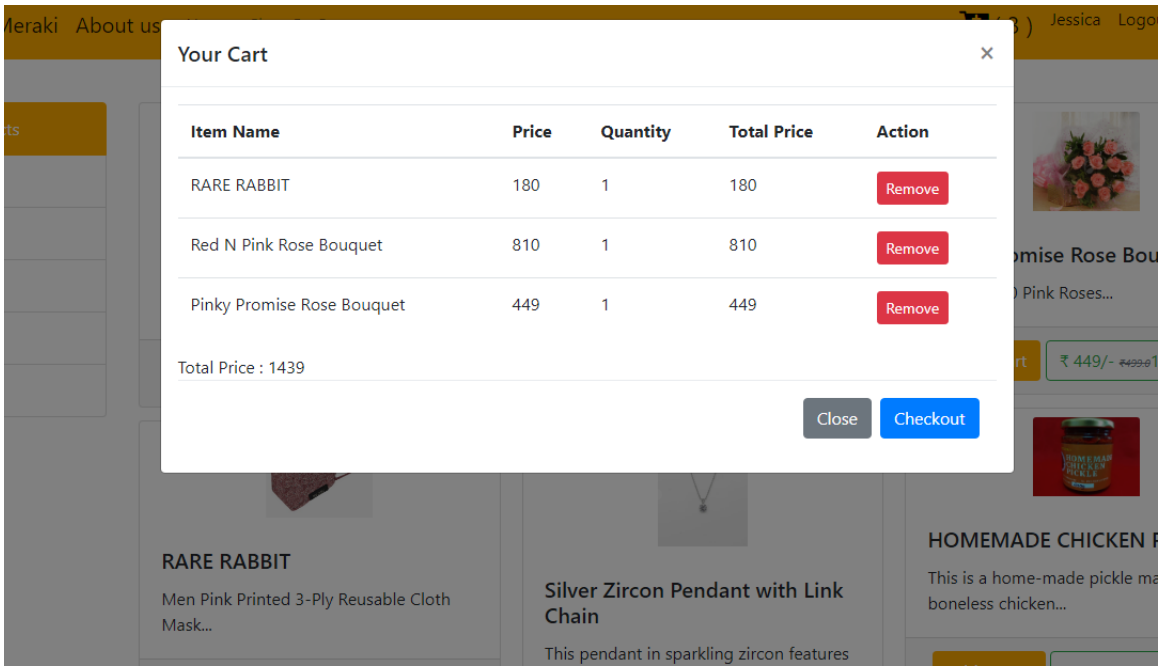
The folder in which all the product images are saved :



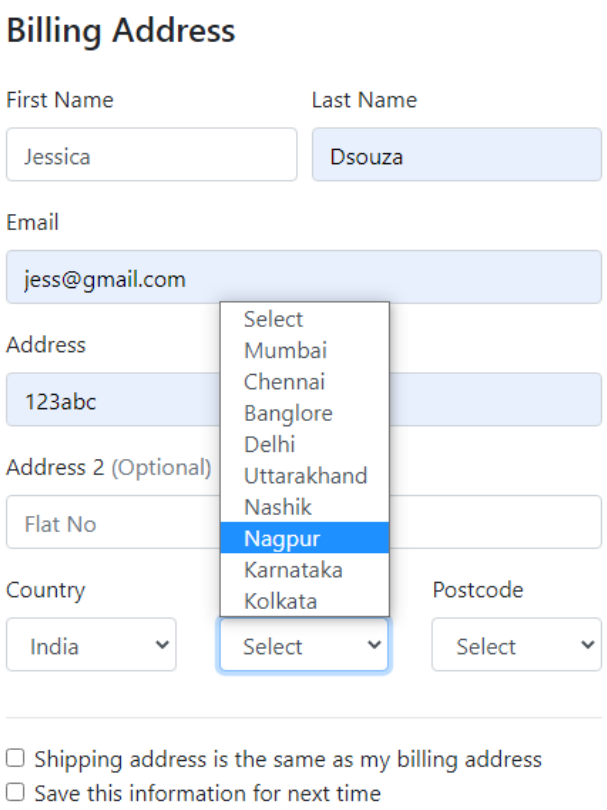
Adding items to cart :

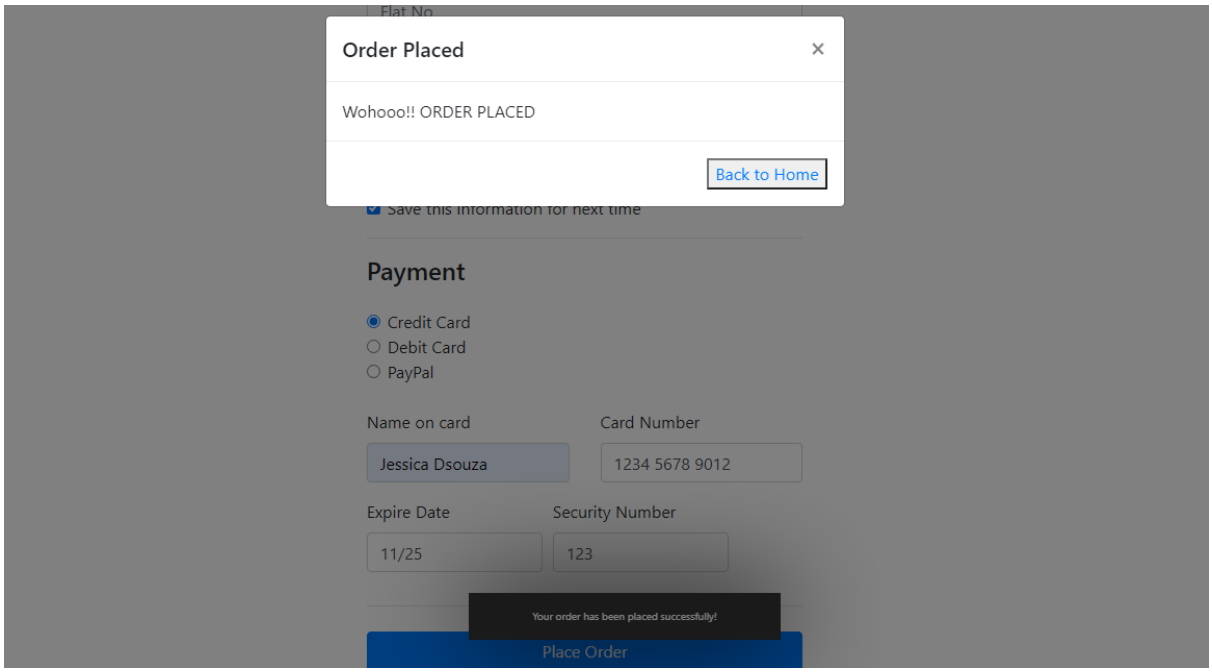


CART :



CheckOut Page :




Flat No
Order Placed
Wohooo!! ORDER PLACED
Back to Home
Save this information for next time
Payment
Credit Card
Debit Card
PayPal
Name on card
Jessica Dsouza
Card Number
1234 5678 9012
Expire Date
11/25
Security Number
123
Your order has been placed successfully!
Place Order

Chapter 6

CONCLUSION

Thus, in this project we have created a web application which is easy to access and user friendly. Such E-commerce websites help to increase the reach of your product. Here, the Sellers can easily add the product to the website which can further also be managed by the admin. With even basic understanding of technology this application can be easily used.

REFERENCES

https://www.javatpoint.com/hibernate-tutorial