CENTRE FOR DEVELOPMENT OF ADVANCED COMPUTING, KOLKATA
Ministry of Electronics and Information Technology (MeitY), India

PROJECT REPORT

ON

Development of an E-commerce web application "Total Mart"

Post Graduate Diploma
In
Advanced Computing

SUBMITTED BY

NAME	PRN
SUMEET MUKUND ALHAT	220970620043
VISHAL PRAKASH SHINDE	220970620023
SUMEDH DEELIP ZAVARE	220970620020
VAIBHAV SHANKAR KALE	220970620049
CHINMAY PRAMOD PATIL	220970620050
SOURAV SANTOSH SHINDE	220970620031
SWAPNIL BHASHKAR BOKADE	220970620048
SIMRAN SALIM SHAIKH	220970620053

Centre for Development of Advanced Computing, Kolkata
(Plot: E2/1, Block:GP,Sector: V,Salt Lake Electronics Complex, Kolkata-700091)

CERTIFICATE

This is to certify that the project report entitled **–Development of an E-Commerce Web Application "Total Mart"** submitted by **Sumeet Mukund Alhat, Vishal Prakash Shinde, Sumedh Deelip Zavare, Vaibhav Shankar Kale, Chinmay Pramod Patil, Sourav Shantosh Shinde, Swapnil Bhashkar Bokade, Simran Salim Shaikh** is a bonafide work carried out by them under the supervision of **Mr. Sri Debdulal Basak and Mr. Asok Bandyopadhyay.** It is approved for the partial fulfillment of the requirement Centre for Development of Advanced Computing (C- DAC) Kolkata, for the competition of the report. This project report has not been earlier submitted to any other Institute or University for the award of any degree or diploma.

Mr. Debdulal Basak
Joint Director
C-D AC, Kolkata

Mr. Asok Bandyopadhyay
Group Head, ICT&S-2
Associate Director

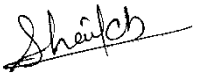
DECLARATION

We hereby declare that the project work entitled **Development of an E-Commerce Web Application "Total Mart"** for Development of Advance Computing(C-DAC), Kolkata, is a record of an original work done by us under the guidance of **Mr. Debdulal Basak,** Joint Director C-DAC, Kolkata**,** and under the supervision of **Mr. Asok Bandyopadhyay**, **Group Head,** Associate Director, ICTS-02, C-DAC, Kolkata. This project work has been performed for the award of ***Post Graduate Diploma in Advanced Computing (PG-DAC SEP 2022)*** course of C- DAC, Kolkata only and this or any similar project will not be used for any other Degree or Diploma's associateship / fellowship.

Signature:
SUMEET ALHAT
PRN- 220970620043

Signature:
VISHAL SHINDE
PRN-220970620023

Signature:
SUMEDH ZAVARE
PRN-220970620020



Signature:
SIMRAN SHAIKH
PRN- 220970620053

Signature:
VAIBHAV KALE
PRN-220970620049

Signature:
CHINMAY PATIL
PRN-220970620050

Signature:
SWAPNIL BOKADE
PRN-220970620048

Signature:
SOURAV SHINDE
PRN-220970620031

ACKNOWLEDGEMENT

We express our utmost sincere regards and deep gratitude to our project guide **Mr. Sri Debdulal Basak, Joint Director, C-DAC, Kolkata** for his valuable guidance towards us to always navigate through the difficulty of project execution. We are very much thankful to **Mr. Asok Bandyopadhyay, Group Head**, Associate Director, ICT&S-2, C-DAC, Kolkata for his valuable support and guidance. Finally, we would like to extend our thanks to all those who have contributed, directly or indirectly to making this project successful.

NAME	PRN
SUMEET MUKUND ALHAT	220970620043
VISHAL PRAKASH SHINDE	220970620023
SUMEDH DEELIP ZAVARE	220970620020
VAIBHAV SHANKAR KALE	220970620049
CHINMAY PRAMOD PATIL	220970620050
SOURAV SANTOSH SHINDE	220970620031
SWAPNIL BHASHKAR BOKADE	220970620048
SIMRAN SALIM SHAIKH	220970620053

TABLE OF CONTENTS

ABSTRACT

E-commerce websites play a big role in our lives, especially in the wake of Covid, as it is not always feasible to visit brick and mortar stores to buy the necessities due to riskof infection. Such websites allow us to browse and shop for products at the comfort ofour homes.

In our project, we aim to provide a similar implementation of such an e-commerce website for the specific case of an online shopping application wherein a user can browse and purchase from assorted catalogue of products.

1. INTRODUCTION

The aim of this project is on the online shopping web application “Total Mart” it is developed using ReactJS, Spring Boot (Rest API, Spring Data JPA), Maven project management, MySQL Database, Embedded Tomcat Server. The web application is very useful where the buyer can directly buy the products from home via internet on mobile or system. The application reduces lot of work load for customer. Online shopping is the process consumers go through to purchase products or Services over the Internet. Anonline shop, e-shopping, e-store, internet shop, web shop, web store, online store, or virtual store evokes the physical analogy of buying products or services at a brick-and-mortar retailer or in a mall. Online shopping is a type of electronic commerce used for business-to-consumer (B2C) transactions. The “Total Mart” is designed to provide an easy to use, interactive web-based interface where users can search for products, view details of the product and order the products. And provide vendor and administration functionality to manage products and customers through an attractive user interface.

2. PRODUCT OVERVIEW AND SUMMARY

2.1 PURPOSE

Our project, “Total Mart,” is a web-based application which aims to provide users with an easy to navigate and visually appealing medium to browse through a category of products and shop for the products they desire.

2.2 SCOPE

“Total Mart” aims to deliver a web-based application that hosts a wide collection of products that users can browse through. Users can place orders for eachproduct. They can view their order history as well. If they no longer wish to be associated with the site, they can deregister themselves. Admins can manage various product details like stock, price, adding new customers, etc. Admins can even delete users if the need arises.

This project does not support the actual logistics and delivery of products and actual payment logic. We are assuming that the organization that implements it will be using third-party payment API which can easily be integrated in our application if needed. Online Shopping System is only an interface for both customers (for browsing and shopping) and admins (for managing inventory and customers).

2.3 OVERVIEW

A. TECHNOLOGIES USED

i. FRONT END

- HTML
- CSS
- JavaScript
- ReactJS

ii. BACK END

- Spring Boot
- Spring Data JPA
- REST API

iii. DATABASE MANAGEMENT SYSTEM

- MySQL

B. FEATURES PROVIDED

i. FOR CUSTOMERS

a. Browse – Customers can browse the homepage to explore the entire collection of products available, as well as view details for individual product.
b. Register, Login & Logout – New customers can register on the site. Existing customers can then login to access their account information and logout when the account is not in use.
c. Add to Cart & Place Orders – If customers find products that they like, they can purchase it. When they wish to purchase it, they can place orders for those products.
d. View Order History – Every customer can view their order history in order to get an idea about their past spendings and how much they saved on each order.

ii. FOR ADMINS

a. Login & Logout – Similar to customers, admins can login & logout to access their account.
b. Manage Inventory – If the admins find that the available stock of some products has depleted, they can replenish it by adding more to the stock.
c. Add Categories – If admin wants to add categories which sorts the products in unique way, then he can specifically describe and the category.
d. Add products - Admin can add products and specify its description and he can add images of the product. He can give number of quantities he has in the inventory.

iii. FOR DELIVERY PERSON

a. Login & Logout – Similar to customers, admins can login & logout to access their account.
b. Change delivery status - Delivery person can the delivery status to: pending, on the way, delivered accordingly. He can also specify the date and time on which the product has been delivered

2.4 FEASIBILITY STUDY

Feasibility is the determination of whether a project is worth undertaking or not. Before recommending the new system, it is important to investigate if it is feasible todevelop it.

Before developing and implementing a system, we must make sure that the system is feasible in the following ways:

A. TECHNICAL FEASIBILITY

In this type of feasibility study, the system analyst must check whether it is possible or not to develop the requested system with the available manpower, software,hardware, etc.
This project makes use of cross-platform software and solutions like Java, and hence can run on any operating system. React, used in front-end, is swift and light weight framework when it comes to delivering the requested page as it does not reload the entire page for every HTTP request. It only re-renders the components that need to fetch newdata. Also, as React is modular in nature, it is easy to develop new components and scale up existing components in order to add new features to the system. The combination of Spring Boot, Spring Data JPA and Hibernate for backend make for a fast, easy to set-up and reliable system to interact with the database, as they are secure and transactional in nature. Since the sensitive data of customers and admins need to be stored in a robust and secure database, MySQL database management system was chosen as it is an industry standard.

B. OPERATIONAL FEASIBILITY

In this type of feasibility study, the operation of the system is considered. An analysisis performed on whether it is feasible for the user department to use the application. Thus, the proposed system is said to be operationally feasible only if clients can understand the system clearly and correctly, and can use it with ease.

In the design of this project, we always kept user experience in mind. We made an effort to have a good user interface with consistent theme and alluring design to keep the users interested and engaged. In our project, the use of universally known icons and instructions that are easy to understand makes sure that the user will not need any special technical know-how to use the application. We made sure that the information available throughout the application is arranged in a logically coherent and consistent manner, guaranteeing that the users will have a smooth and effortless experience and even enjoy using the application.

C. ECONOMIC FEASIBILITY

In this type of feasibility study, the benefits of the system to the organization are considered by taking into consideration the cost-benefit analysis. All the software and technologies used in our project free, open-source, and widely available, with each of the technologies having an extensive community support. This makes “Online Shopping System” an economically feasible solution to the organizations that wish toimplement it.

3. REQUIREMENTS FULLFILLED

3.1 FUNCTIONAL REQUIREMENTS

Following are the functional requirements fulfilled by our project:

- Customers can browse through all available products while going through product categories
- Customers can place orders for products and view their order history.
- Admins can manage various product details like inventory, price, adding new products.
- Admins can assign or place orders to the delivery person.

3.2. NON-FUNCTIONAL REQUIREMENTS

Following are the non-functional requirements fulfilled by our project:

- Since the application uses lightweight and established software components that are also cross-platform, it is remarkably performant and has good support for every operating system.
- The use of React for front end and Spring Boot, Spring Data JPA for back end delivers quick response times to admins and customers alike.
- Card-style UI and well-known icons and symbols used throughout the application provides a consistent theme and user-friendly interface that anyone can grasp easily, even without a technical background.

4. PROJECT DESIGN

4.1 DATA MODEL

The following tables depict the database design used for “Total Mart” application:

a. user table

```
mysql> desc user;
+------------+--------------+------+-----+---------+----------------+
| Field      | Type         | Null | Key | Default | Extra          |
+------------+--------------+------+-----+---------+----------------+
| id         | int          | NO   | PRI | NULL    | auto_increment |
| email_id   | varchar(255) | YES  |     | NULL    |                |
| first_name | varchar(255) | YES  |     | NULL    |                |
| last_name  | varchar(255) | YES  |     | NULL    |                |
| password   | varchar(255) | YES  |     | NULL    |                |
| phone_no   | varchar(255) | YES  |     | NULL    |                |
| role       | varchar(255) | YES  |     | NULL    |                |
| address_id | int          | YES  | MUL | NULL    |                |
+------------+--------------+------+-----+---------+----------------+
8 rows in set (0.02 sec)
```

b. product table

```
mysql> desc product;
+-------------+---------------+------+-----+---------+----------------+
| Field       | Type          | Null | Key | Default | Extra          |
+-------------+---------------+------+-----+---------+----------------+
| id          | int           | NO   | PRI | NULL    | auto_increment |
| description | varchar(255)  | YES  |     | NULL    |                |
| image_name  | varchar(255)  | YES  |     | NULL    |                |
| price       | decimal(19,2) | YES  |     | NULL    |                |
| quantity    | int           | NO   |     | NULL    |                |
| title       | varchar(255)  | YES  |     | NULL    |                |
| category_id | int           | YES  | MUL | NULL    |                |
+-------------+---------------+------+-----+---------+----------------+
7 rows in set (0.02 sec)
```

c. orders table

```
mysql> desc orders;
+-------------------+-------------+------+-----+---------+----------------+
| Field             | Type        | Null | Key | Default | Extra          |
+-------------------+-------------+------+-----+---------+----------------+
| id                | int         | NO   | PRI | NULL    | auto_increment |
| delivery_assigned | varchar(255) | YES  |     | NULL    |                |
| delivery_date     | varchar(255) | YES  |     | NULL    |                |
| delivery_person_id | int        | NO   |     | NULL    |                |
| delivery_status   | varchar(255) | YES  |     | NULL    |                |
| delivery_time     | varchar(255) | YES  |     | NULL    |                |
| order_date        | varchar(255) | YES  |     | NULL    |                |
| order_id          | varchar(255) | YES  |     | NULL    |                |
| quantity          | int         | NO   |     | NULL    |                |
| product_id        | int         | YES  | MUL | NULL    |                |
| user_id           | int         | YES  | MUL | NULL    |                |
+-------------------+-------------+------+-----+---------+----------------+
11 rows in set (0.00 sec)
```

d. category table

```
mysql> desc category;
+-------------+-------------+------+-----+---------+----------------+
| Field       | Type        | Null | Key | Default | Extra          |
+-------------+-------------+------+-----+---------+----------------+
| id          | int         | NO   | PRI | NULL    | auto_increment |
| description | varchar(255) | YES  |     | NULL    |                |
| title       | varchar(255) | YES  |     | NULL    |                |
+-------------+-------------+------+-----+---------+----------------+
3 rows in set (0.01 sec)
```

e. cart table

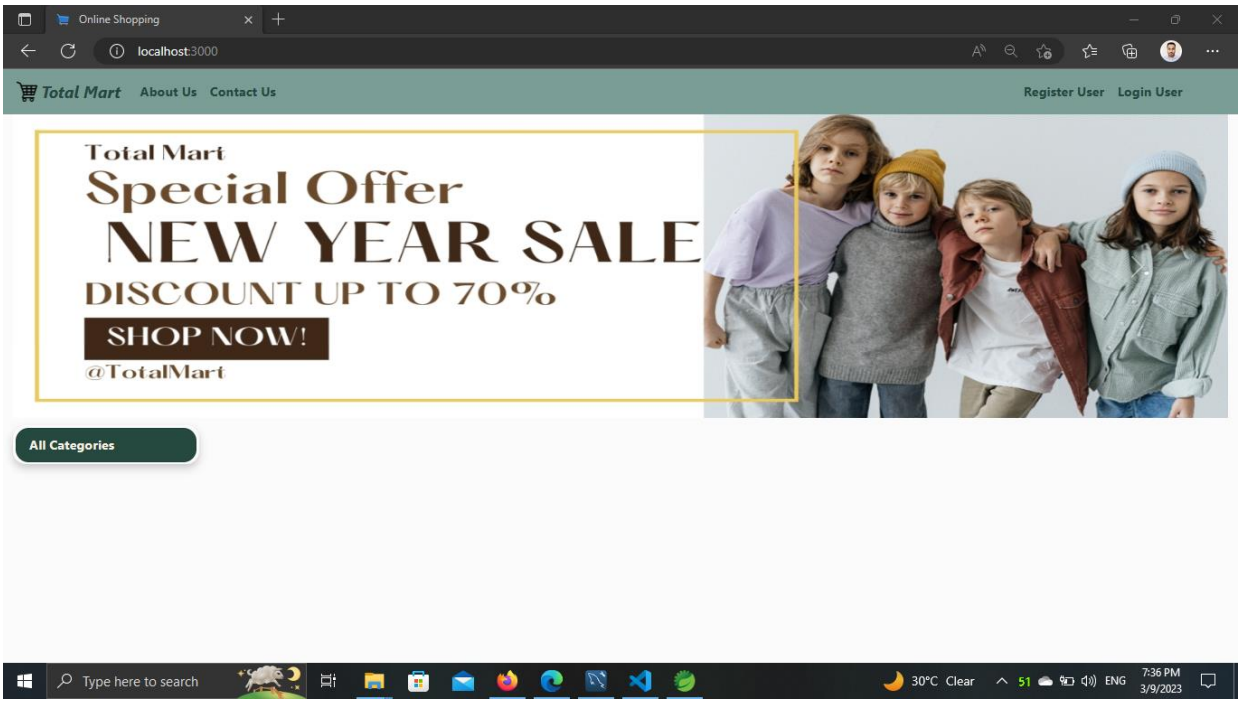
```
mysql> desc cart;
+------------+------+------+-----+---------+----------------+
| Field      | Type | Null | Key | Default | Extra          |
+------------+------+------+-----+---------+----------------+
| id         | int  | NO   | PRI | NULL    | auto_increment |
| quantity   | int  | NO   |     | NULL    |                |
| product_id | int  | YES  | MUL | NULL    |                |
| user_id    | int  | YES  | MUL | NULL    |                |
+------------+------+------+-----+---------+----------------+
4 rows in set (0.02 sec)
```

f. address table

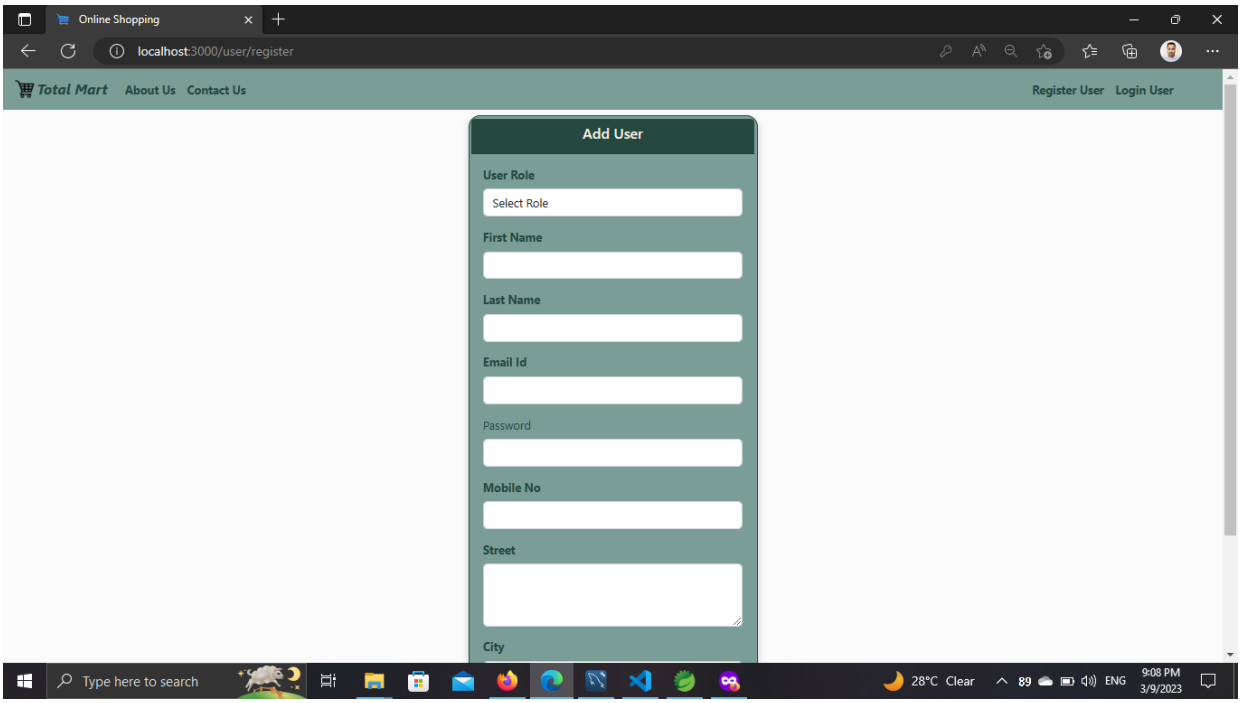
```
mysql> desc address;
+---------+--------------+------+-----+---------+----------------+
| Field   | Type         | Null | Key | Default | Extra          |
+---------+--------------+------+-----+---------+----------------+
| id      | int          | NO   | PRI | NULL    | auto_increment |
| city    | varchar(255) | YES  |     | NULL    |                |
| pincode | int          | NO   |     | NULL    |                |
| street  | varchar(255) | YES  |     | NULL    |                |
+---------+--------------+------+-----+---------+----------------+
4 rows in set (0.01 sec)
```

4.2 PAGE FLOW DIAGRAM

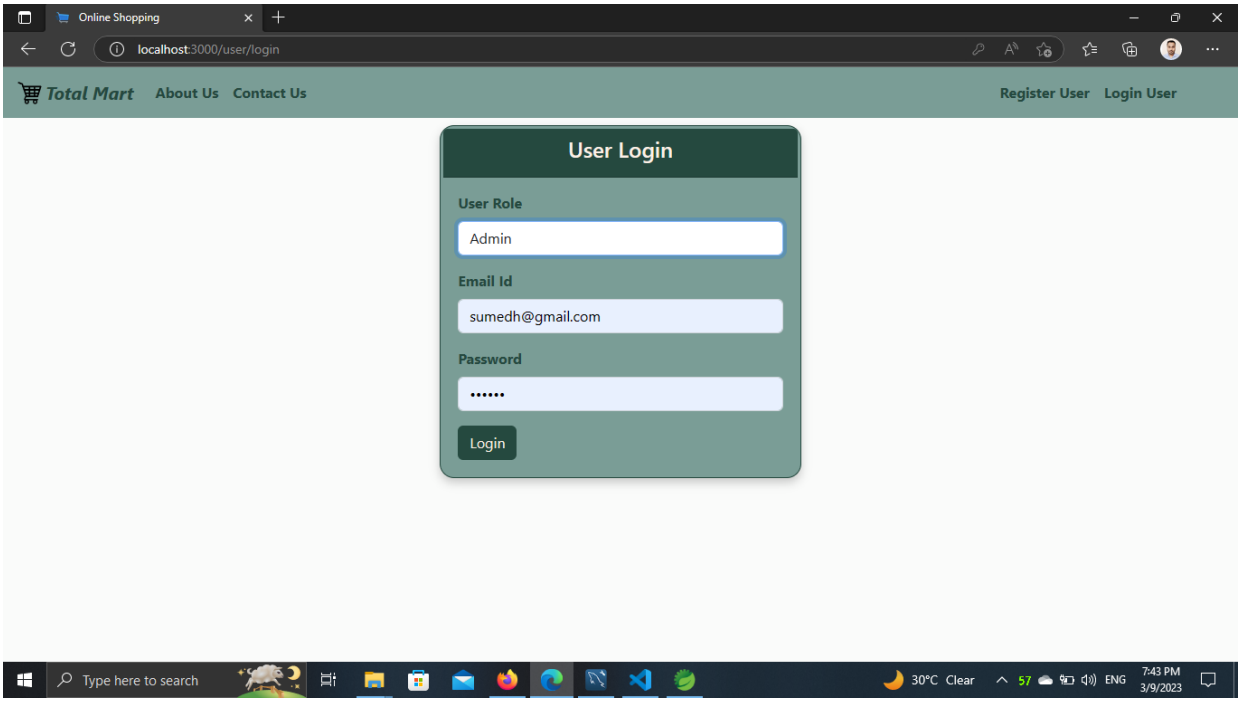
1. Home page



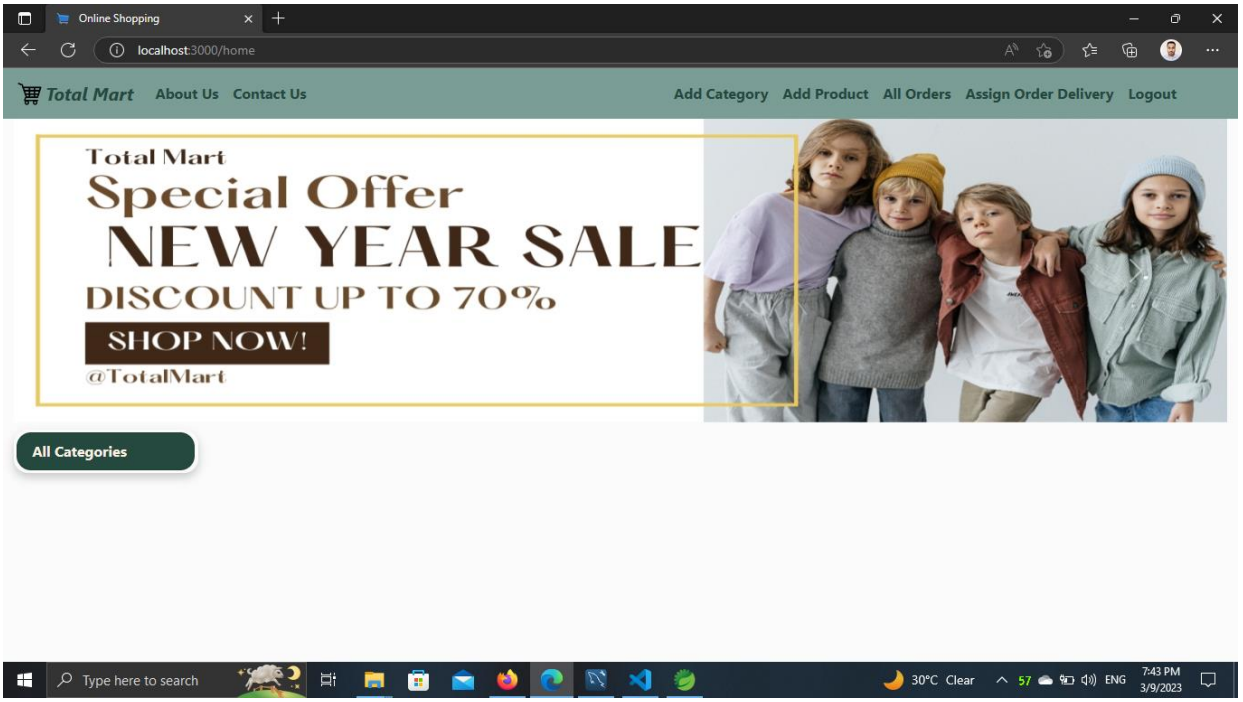
2. Add user: (admin / customer / delivery person)



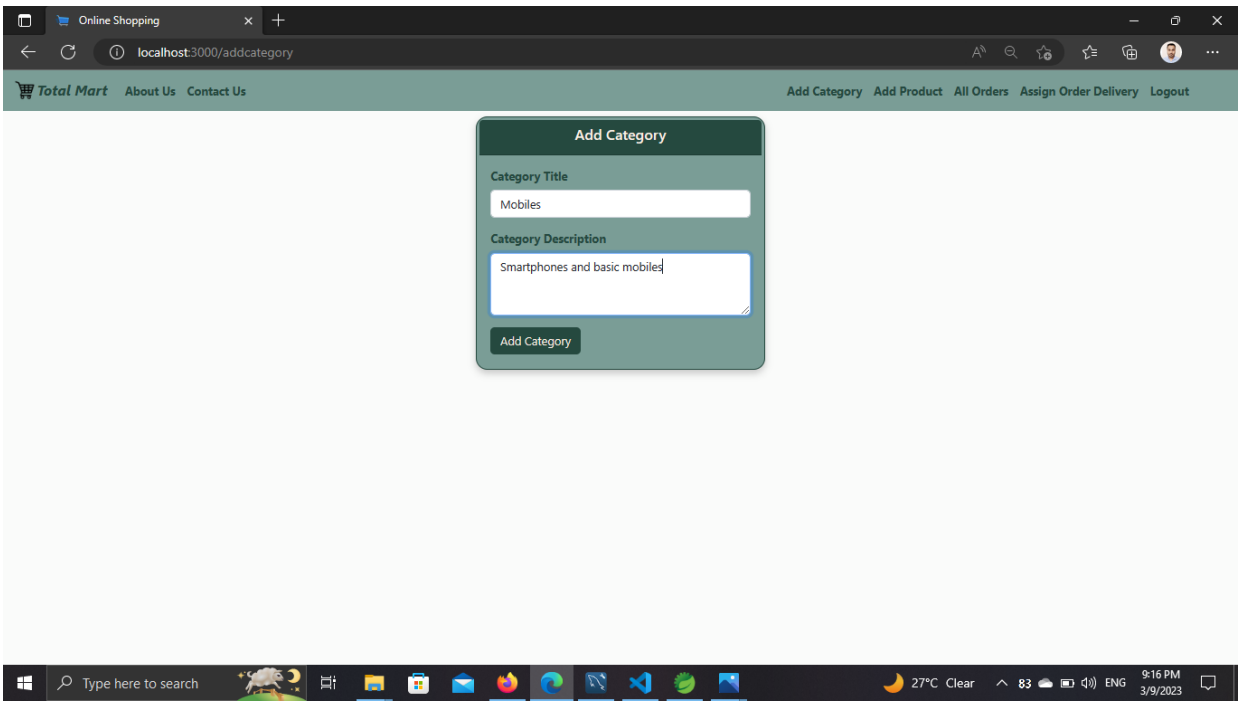
3. User login page



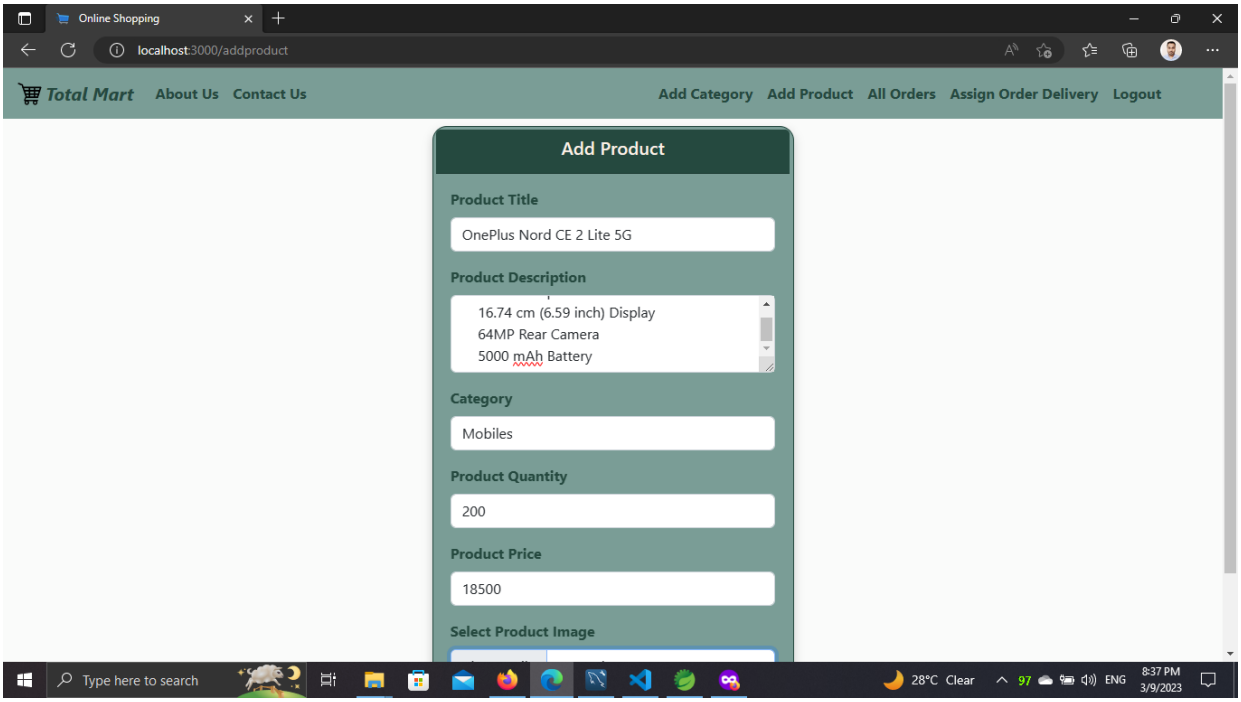
4. Admin login homepage



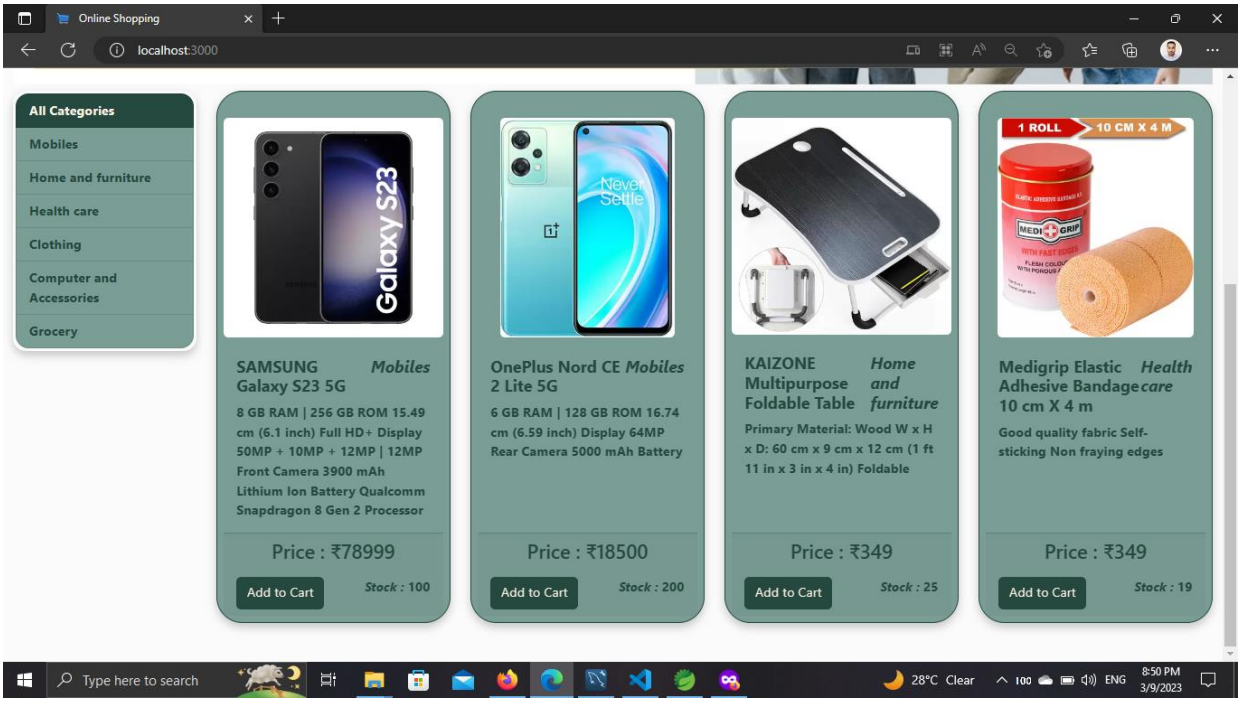
5. Create category page



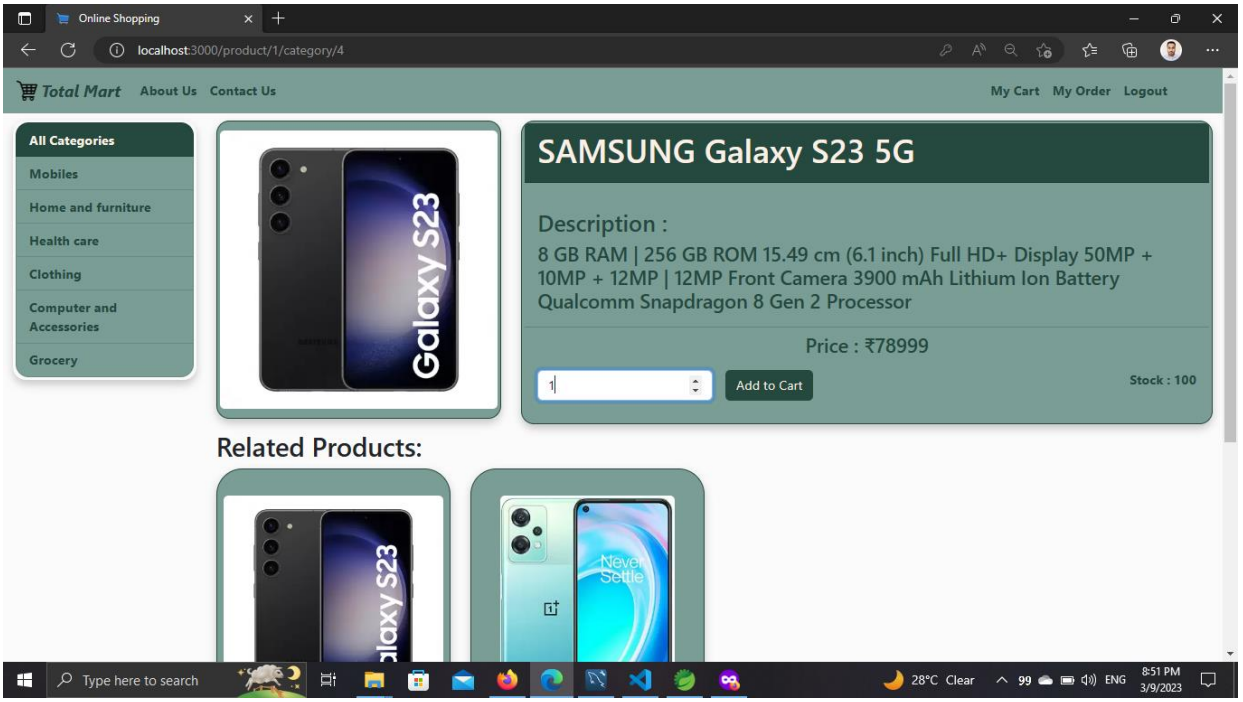
6. Fill product information and add image



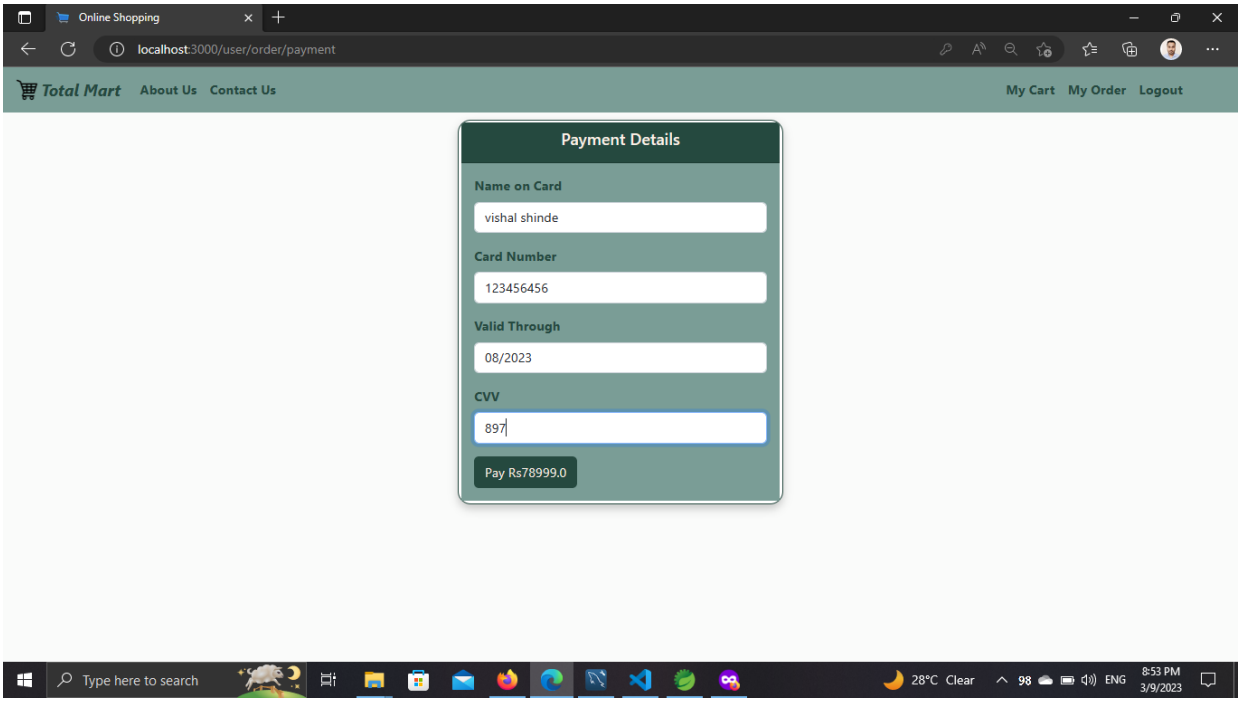
7. All products display



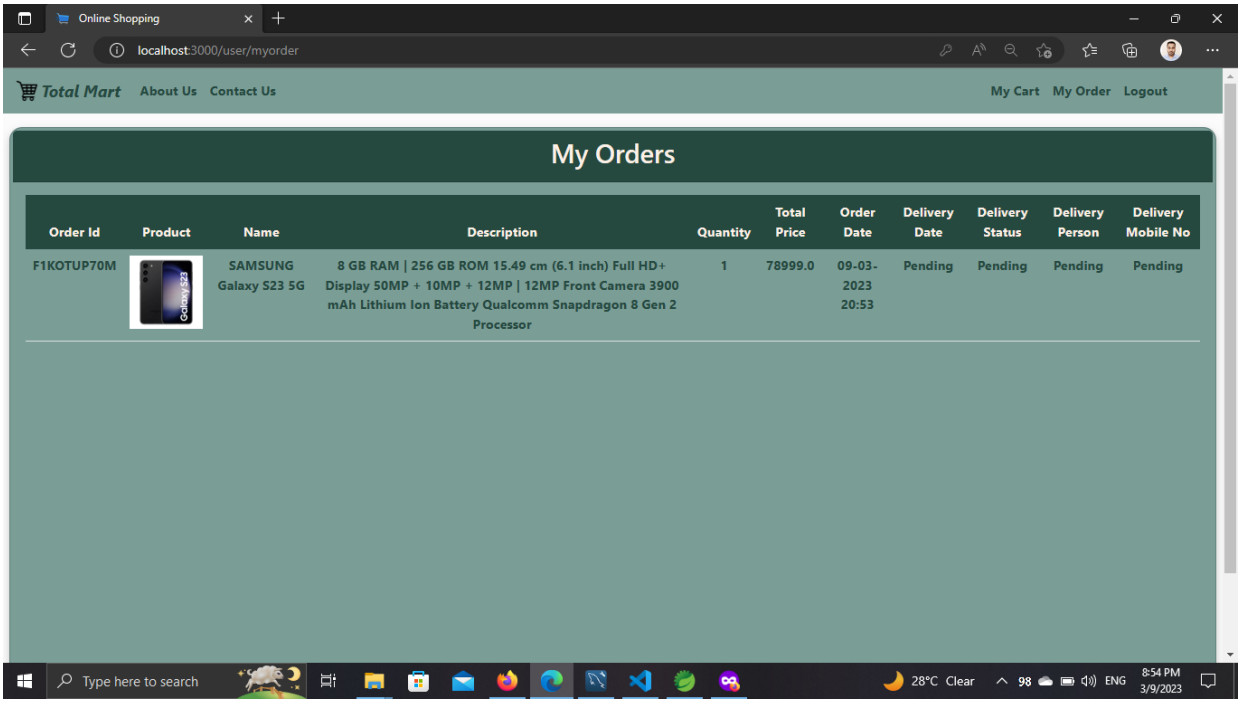
8. Add to cart product



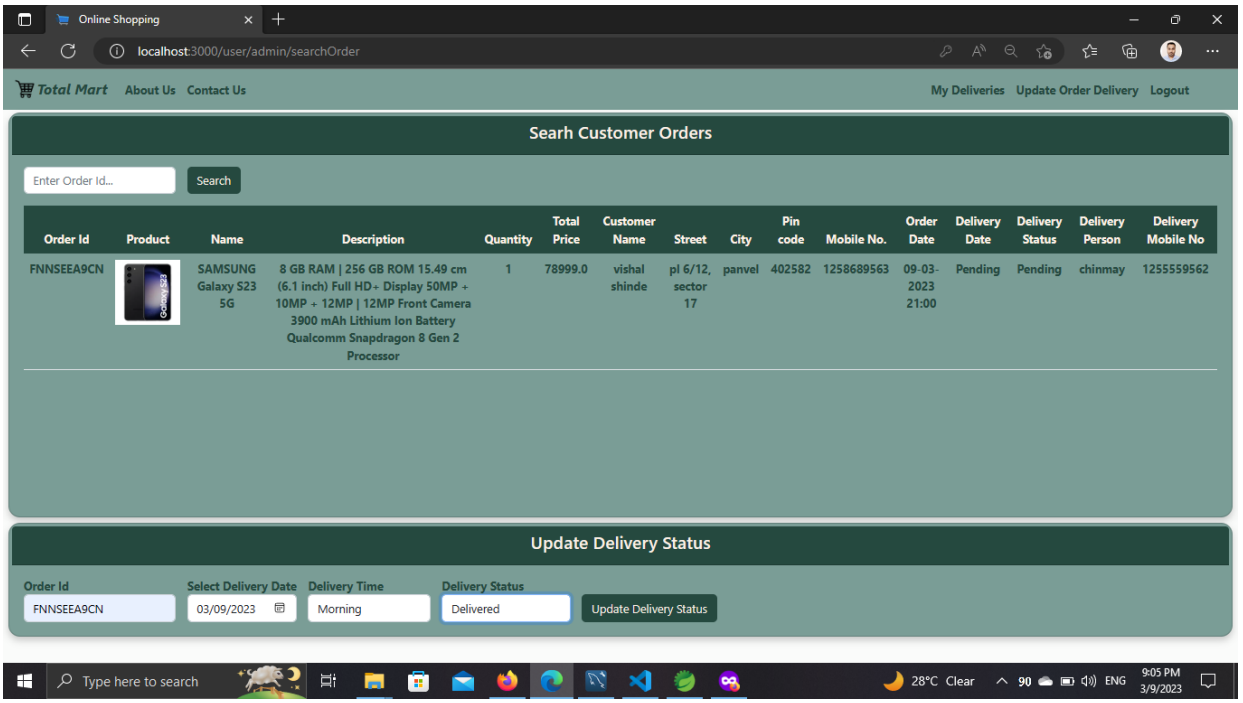
9. Payment page



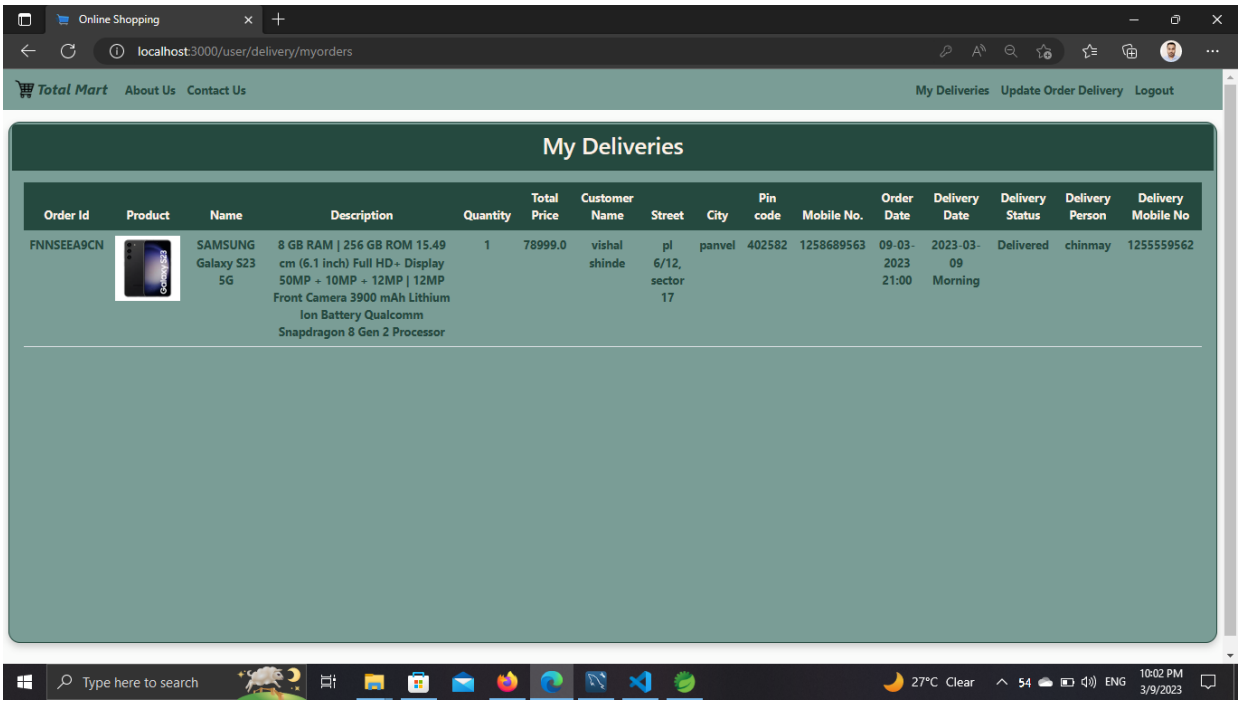
10. My orders page



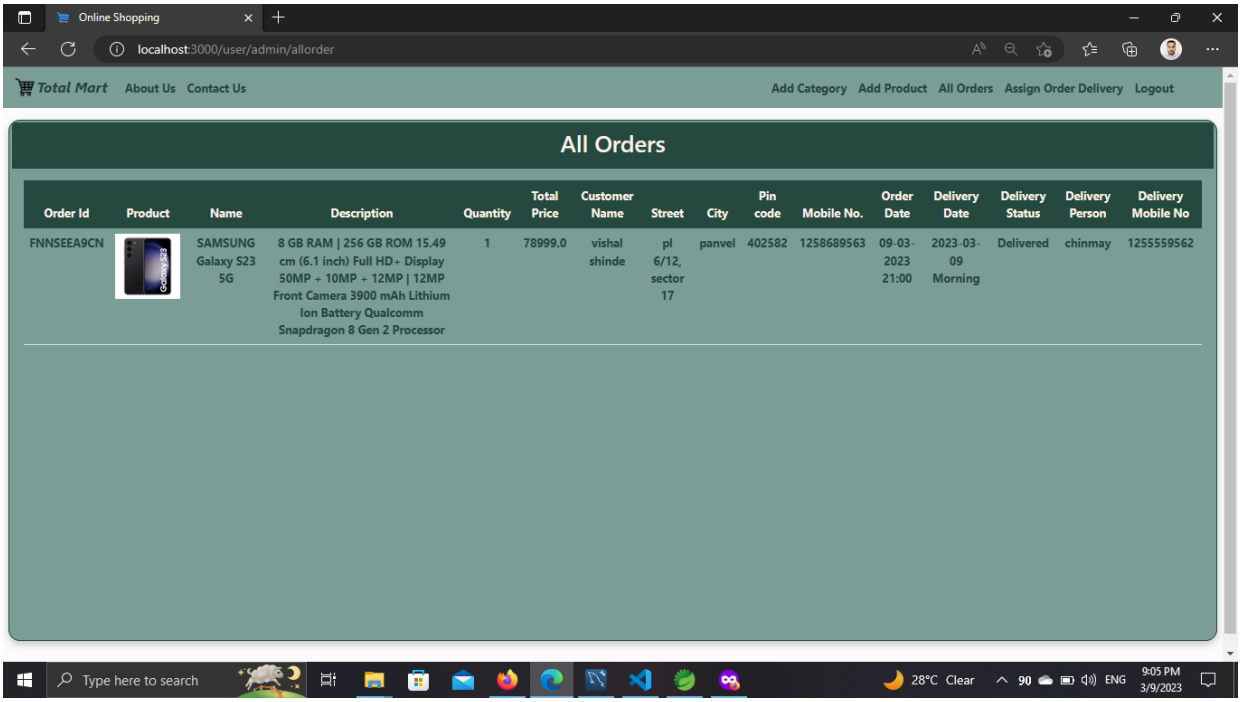
11. Update delivery status (by Delivery person)



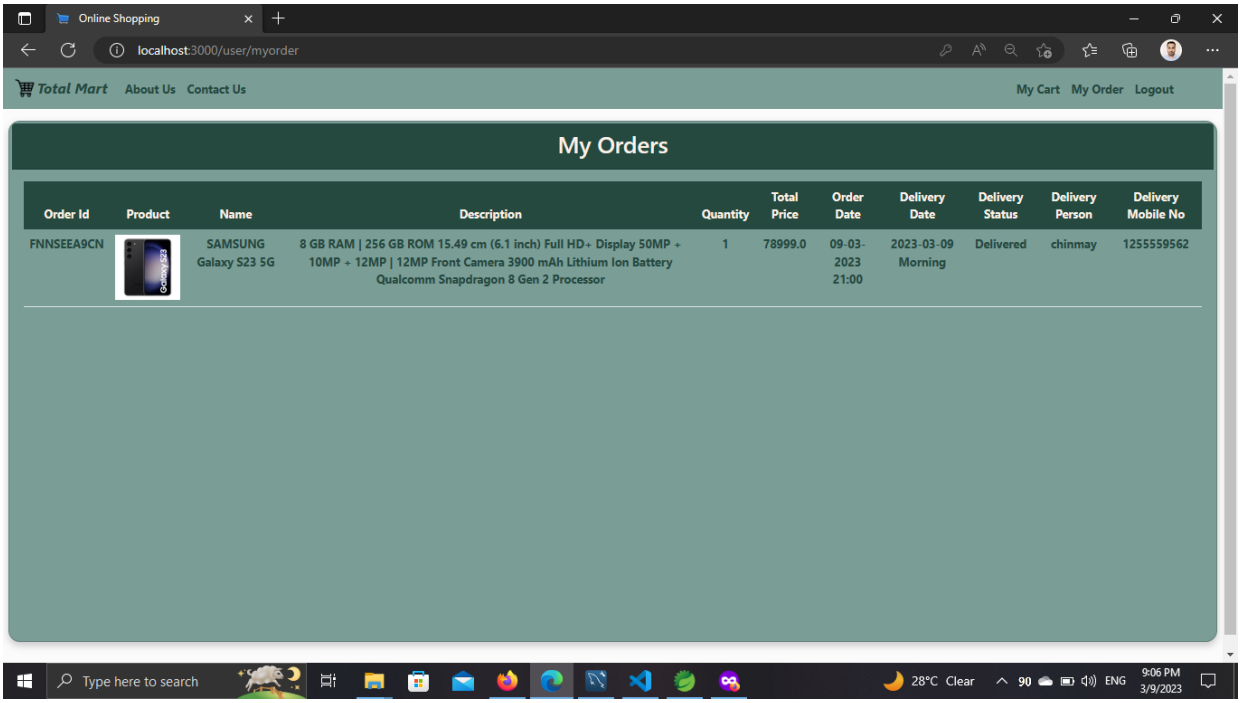
12. Deliveries pending/done status (by Delivery person)



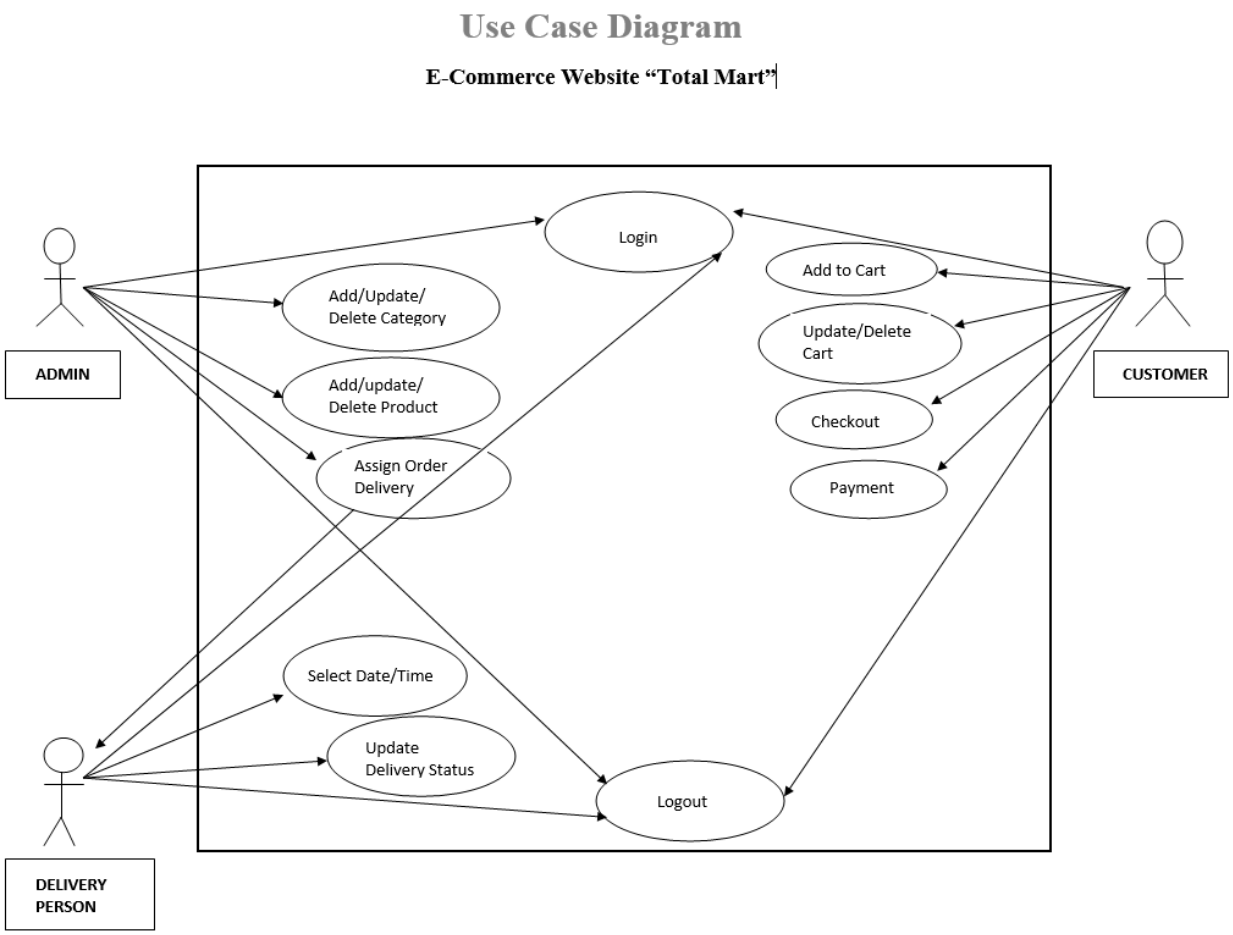
13. Delivery success (Admin view page)



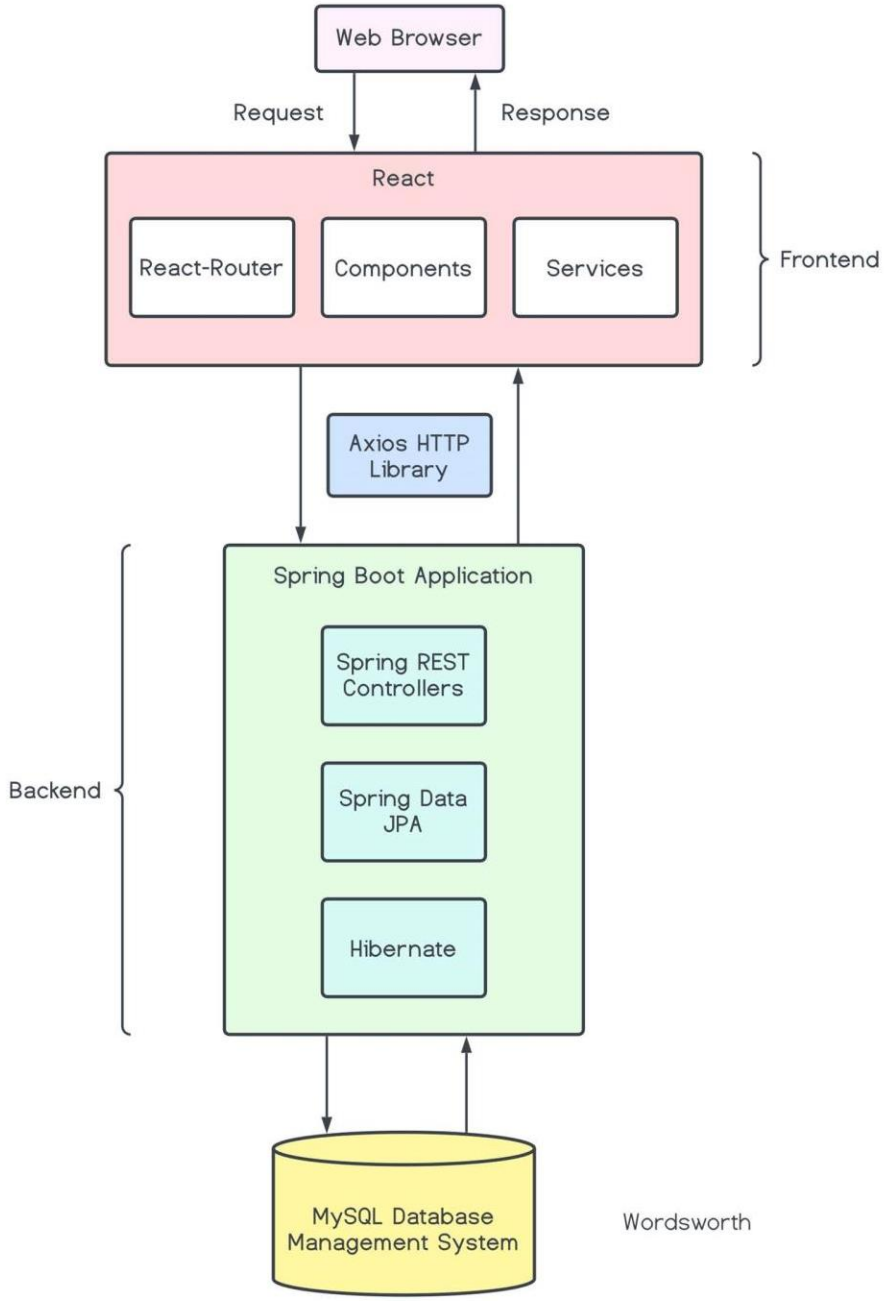
41. Delivery success (Customer view page)



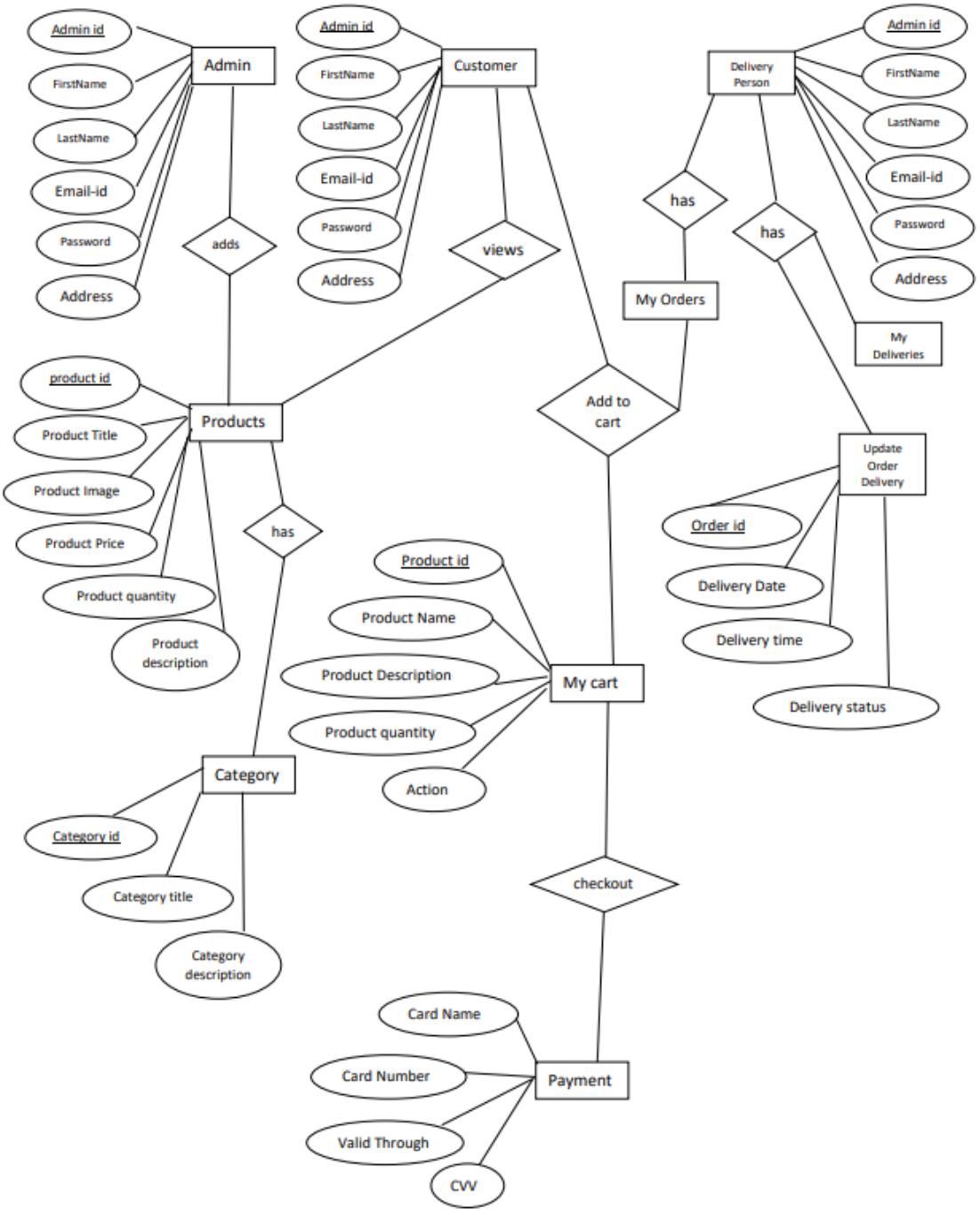
4.3 USE CASE DIAGRAM



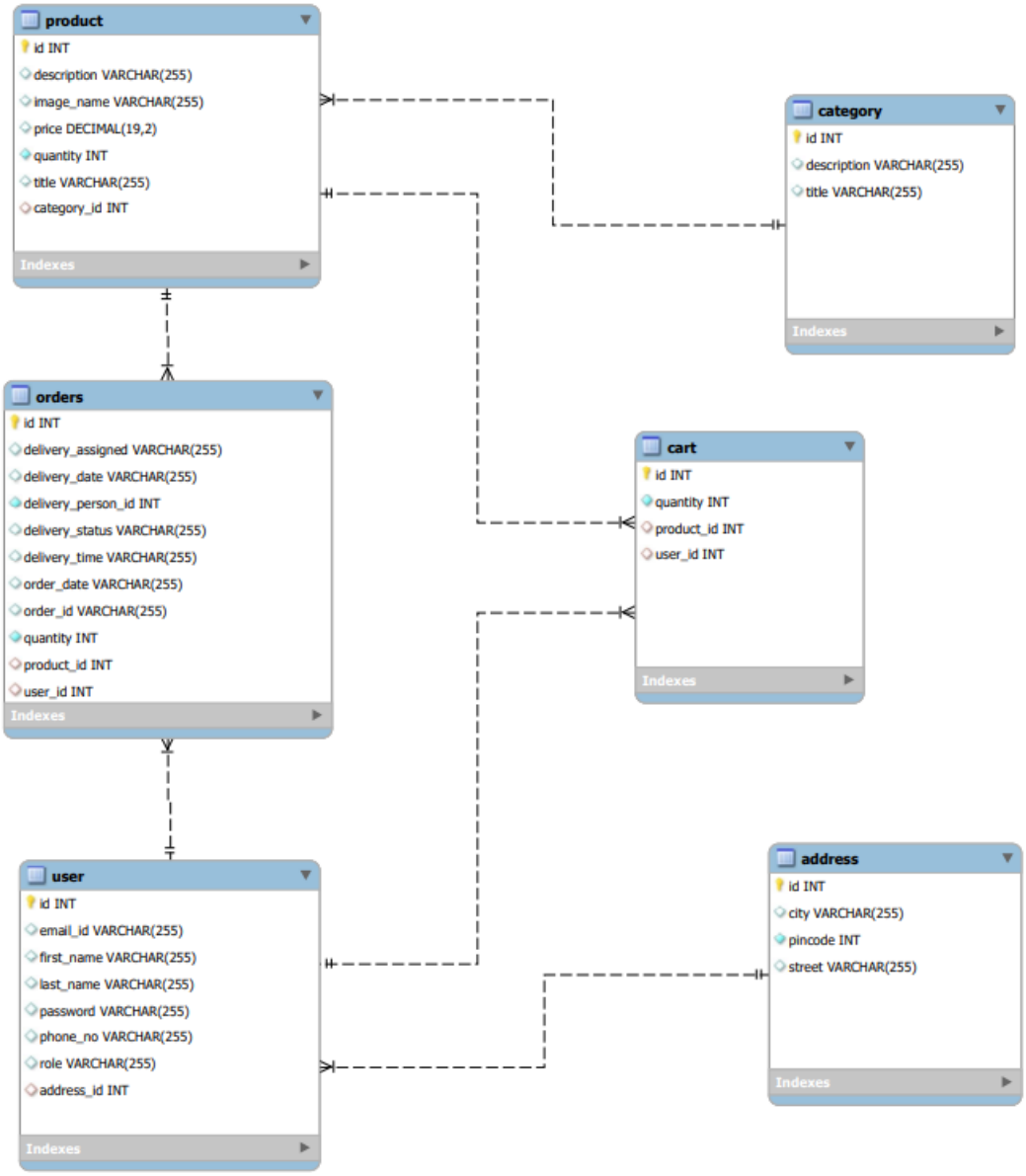
4.4 PROJECT ARCHITECTURE



4.5 ER DIAGRAM



4.6 CLASS DIAGRAM



4.7 Lines of code (Images)

Backend:

Java Application Properties:

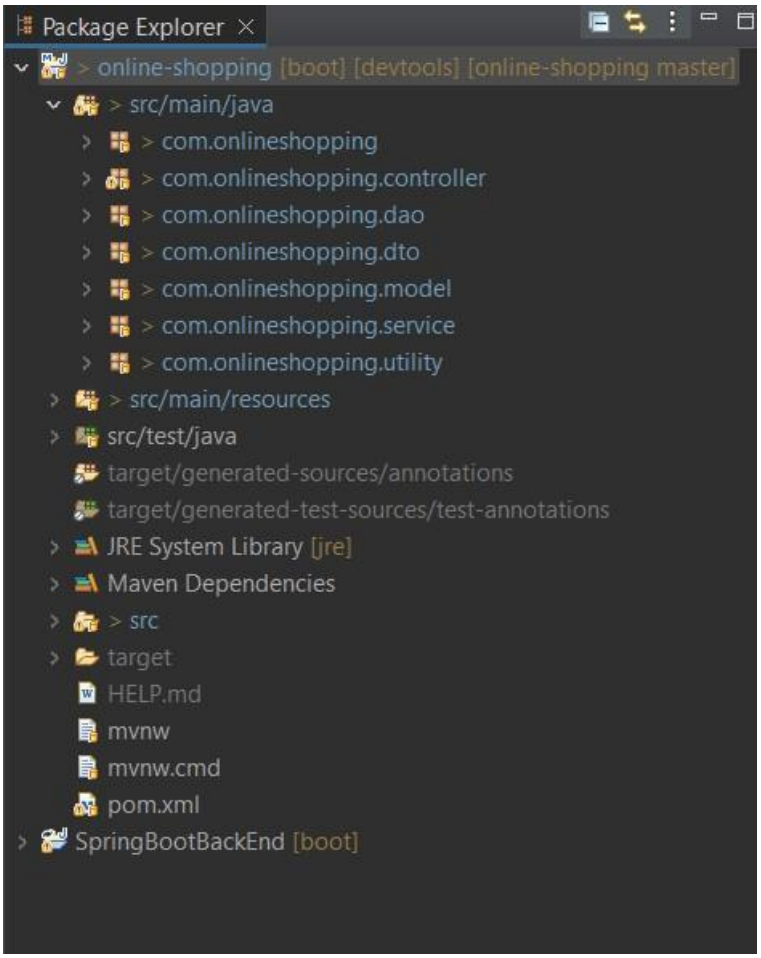
application.properties

```
# MySQL Properties
spring.datasource.url=jdbc:mysql://localhost:3306/TotalMart?createDatabaseIfNotExist=true&useUnicode
spring.datasource.username=root
spring.datasource.password=sumeet1996
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.jpa.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL57Dialect

disk.upload.basepath=D:/CDAC Project Final/GROUP-A/images

spring.jackson.serialization.fail-on-empty-beans=false
```

Folder Structure:



Dependencies configuration:

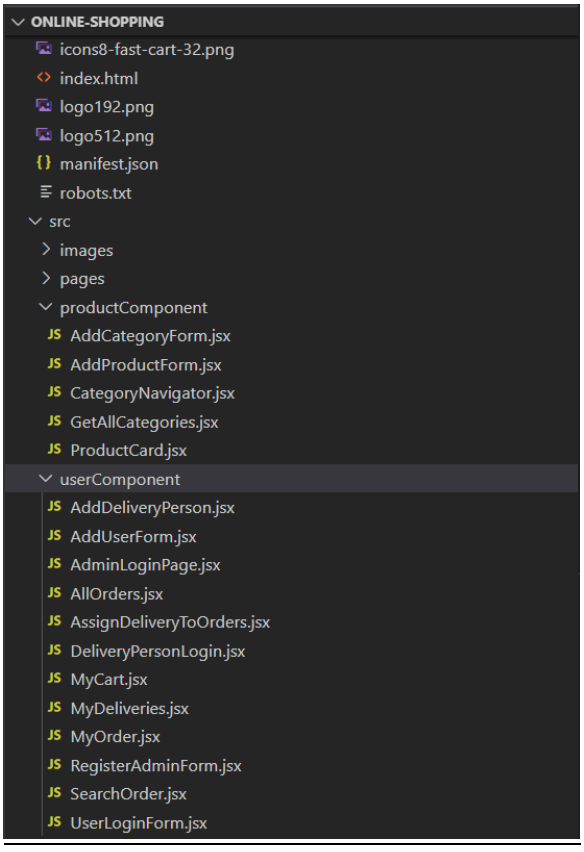
application.properties | online-shopping/pom.xml

```
    </parent>
    <groupId>com.onlineshopping</groupId>
    <artifactId>online-shopping</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>online-shopping</name>
    <description>online shopping project using  spring boot + react</description>
    <properties>
        <java.version>11</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-devtools</artifactId>
            <scope>runtime</scope>
            <optional>true</optional>
        </dependency>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <scope>runtime</scope>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
        <!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-databind -->
```

Frontend:

Folder Structure:



5. TESTING

One of the main purposes of testing is to validate and verify that the system works as intended. No program or system design is perfect. However, if we implement the system without proper testing, then it may cause problems and lead to a bad user experience.

Testing and checking outcomes of each test gives us the best chance to detect and correct errors before the system is implemented in a production environment.

During our project, we made an effort to manually test each component. In all cases, we obtained the desired results as demonstrated below.

A. CUSTOMER FEATURES TEST

#	Description	Outcome	Result
1.	Register as Customer	New customer details were saved in the Database.	Passed
2.	Login as Customer	Fetched authenticated user details saved in database.	Passed
3.	Browse Category	Fetched list of all products from the database.	Passed
4.	Add products to Cart	The products along with necessary details were saved in database in the customer's cart.	Passed
5.	Place Order	The cart items associated with the customer were saved in the form of a placed order in the database.	Passed
6.	Logout	The session was cleared.	Passed

B. ADMIN FEATURES TEST

#	Description	Outcome	Result
1.	Sign in as Admin	Fetched authenticated user details saved in database.	Passed
2.	Add New Category	New category and all its respective details were saved in database	Passed
3.	Add New Product	The details of a new product were updated in the database.	Passed
4.	Assign orders to Delivery person	The delivery person orders were assigned successfully and reflected the same in the database.	Passed
5.	Logout	The session was cleared.	Passed

C. DELIVERY PERSON TEST

#	Description	Outcome	Result
1.	Sign in as Delivery person	Fetched authenticated user details saved in database.	Passed
2.	Check out assigned deliveries	Assigned deliveries were fetched from the database	Passed
3.	Change delivery status	Delivery status: pending, on the way, delivered. Done smoothly	Passed
4.	Change delivery time and date	Time and Date of the delivery entered by the delivery person, reflected in the admin and customer view page.	Passed
5.	Logout	The session was cleared.	Passed

6. CONCLUSION

“Total Mart” application was developed by our project team to simplify the online sale and purchase of products. We tried using the latest technologies that are cross-platform and robust. Each software we used was open-source in nature,which keeps the cost of production at a minimum.

We were also meticulous about the user experience aspect of our application so that navigating our website is an easy and seamless experience.

In conclusion, “Total Mart” as an application would be a good choice for any product trading business that wishes to enter the online market. We are confident that the numerous features and visually appealing look of the application willcertainly give a big boost to the business.

7. FUTURE SCOPE

Using whatever we have learnt over the duration of this course, we tried to make our project as user-friendly and gave it as many features as possible in the limited time allotted for the project work. That said, there are certainly more features that can be added to our application. Some of those are mentioned below:

1. The most purchased products can be highlighted as customer favorite to promote those products further.
2. Discounts can be given on a per-user basis depending on the customer's purchase history as well as how many products they buy at the same time.
3. Customers can upvote/downvote/report feedbacks.
4. Additional payment means can be added other than cards.
5. After a confirmed purchase, an email with the invoice of the orders can be sent to the customer.
6. In case the user forgets the password, a 'reset password' functionality can be added.
7. CAPTCHA can be added to login page.
8. An email notification can be sent to the users for an item in their cart, which may have been out of stock, but is now available.

8. REFERENCES

Following is the list of websites we referred during the course of our project:

1. https://getbootstrap.com/docs/5.1/getting-started/introduction/
2. https://reactjs.org/docs/getting-started.html
3. https://www.baeldung.com/
4. https://www.w3schools.com/
5. https://docs.spring.io/springdata/jpa/docs/current/reference/html/#reference
6. https://javaee.github.io/javaee-spec/javadocs/
7. https://javadoc.io/doc/org.springframework.data/spring-datajpa/latest/index.html