A

PROJECT REPORT ON

UrbanKart

ONLINE SHOPPING SYSTEM

SUBMITTED IN PARTIAL FULFILLMENT OF

DIPLOMA IN ADVANCED COMPUTING (PG-DAC)

UNDER THE GUIDANCE OF
Mr. Vinu Josy

PRESENTED BY

Ms. Adiba Khan	**PRN Number 230340120009**
Mr. Amber Selwaria	**PRN Number 230340120023**
Mr. Atul Patankar	**PRN Number 230340120039**
Mr. Abhishek Rajendra Gupta	**PRN Number 230340120072**
Mr. Laxmikant Kulkarni	**PRN Number 230340120103**
Ms. Poonam Patil	**PRN Number 230340120146**

AT

CENTER FOR DEVELOPMENT OF ADVANCED COMPUTING C-DAC,

PUNE

ACKNOWLEDGEMENT

The project “UrbanKart (Online Shopping System)” was a great learning experience for us and we are submitting this work to Advanced Computing Training School (C-DAC ACTS, Pune).

We are very glad to mention the name of Mr. Vinu Josy for his valuable guidance to work on this project.

We are highly grateful to Ms. Risha P. R., Manager of ACTS Training Centre, CDAC, for her guidance and support whenever necessary during the course of our journey to acquire PG-Diploma in Advanced Computing (PG-DAC) through CDAC ACTS, Pune.

Our heartfelt thanks go to Ms. Priyanka Ranade (Course Coordinator, PG_DAC) who gave us all the required support and kind coordination to provide all the necessitiesto complete the project and throughout the course up to the last day of the course.

We would like to express our sincere gratitude towards Mrs. Madhura Anturkar, our faculty for J2SE and J2EE, who was always there for us. Her guidance and support helped us overcome various obstacles and intricacies during the course of our project work. Without her tremendous support, guidance, and efforts, this project would not have been possible.

From:

Mrs. Adiba Khan	PRN Number 230340120009
Mr. Amber Selwaria	PRN Number 230340120023
Mr. Atul Patankar	PRN Number 230340120039
Mr. Abhishek Rajendra Gupta	PRN Number 230340120072
Mr. Laxmikant Kulkarni	PRN Number 230340120103
Ms. Poonam Patil	PRN Number 230340120146

TABLE OF CONTENTS

ABSTRACT

E-commerce websites have become an integral part of our lives, offering convenience and accessibility. They enable us to explore and purchase products from the comfort of our homes, eliminating the need to visit physical stores. This convenience has made online shopping a preferred choice for many, providing a wide range of options and hassle-free shopping experiences.

Our website is like an online store where you can buy all sorts of things easily. We have lots of different products for you to choose from, like clothes and gadgets. You can trust that our website is safe to use, and we'll make sure your purchases get to you quickly. If you ever need help, our friendly customer support team is here for you. Whether you're a regular online shopper or trying it out for the first time, our website makes shopping online simple and enjoyable. Come visit us and discover a better way to shop online!

1. INTRODUCTION

This project focuses on creating an online shopping web application using J2EE, Spring Boot, MySQL, and React. This application offers a convenient way for buyers to shop for products from their homes using the internet, whether on a mobile device or computer. It simplifies the shopping process for customers, similar to how they would shop in a physical store but in a virtual environment.

Online shopping, or e-commerce, involves consumers purchasing products or services online. This web application emulates the experience of shopping at a traditional brick-and-mortar store or a mall, but it's all done online. It's a form of electronic commerce, specifically catering to business-to-consumer (B2C) transactions.

The Online Shopping System is designed to be user-friendly, providing an interactive web interface where users can search for products, see product details, and place orders. Additionally, it offers administrative features to manage products and customers, all presented through an attractive and intuitive user interface.

2. PRODUCT OVERVIEW AND SUMMARY

2.1. PURPOSE

Our project, “UrbanKart”, 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

“UrbanKart” aims to deliver a web-based application that hosts a wide collection of products that users can browse through. Users can place orders for each product. 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. Urbankart 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
- Bootstrap
- React
- Axios

ii. BACK END

- Spring Boot
- Spring Data JPA
- Hibernate
- REST
- JWT Based Security

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.
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 save the product in the cart until they decide to purchase it. When they wish to purchase it, they can place orders for those products .

ii. FOR ADMINS

a. **Login & Logout** – Similar to customers, admins can login & logout to access their account.
b. **Add Category & products** – Admins can add category and products
c. **Delete Customer** –The admins can delete a customer account if they need to for any purpose.
d. **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.
e. **Change Status of Orders** – Manipulate state of orders.

2.4. FEASIBILITY STUDY

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

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

A. TECHNICAL FEASIBILITY

In this type of feasibility study, the system analyst has to 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 doesn't reload the entire page for every HTTP request. It only re-renders the components that need to fetch new data. 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, we assess whether the system can function effectively. We analyze if it's practical for the user department to utilize the application. Operational feasibility means the proposed system is only feasible if users can easily understand and use it.

In our project's design, we prioritized user experience. We created a user-friendly interface with a consistent theme and attractive design to engage users. We used familiar icons and clear instructions, ensuring that users don't need special technical knowledge to use the application. Information is organized logically and consistently, ensuring a seamless and enjoyable user experience throughout 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 “UrbanKart” an economically feasible solution to the organizations that wish to implement it.

3. REQUIREMENTS FULFILLED

3.1. FUNCTIONAL REQUIREMENTS

Following are the functional requirements fulfilled by our project:

- Customers can browse through all available products.
- Customers can place orders for products.
- Admins can manage various product details like inventory, price, adding new products and customers, etc.
- Admins can view all customers.
- JWT Security for Authentication and Authorization

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 and Hibernate 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 “UrbanKart”application:

A. Tables Related to User Details

a. users Table

```
mysql> desc users;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| id            | bigint       | NO   | PRI | NULL    | auto_increment |
| email         | varchar(50)  | NO   | UNI | NULL    |                |
| first_name    | varchar(25)  | YES  |     | NULL    |                |
| last_name     | varchar(25)  | YES  |     | NULL    |                |
| mobile_number | varchar(10)  | NO   |     | NULL    |                |
| password      | varchar(300) | NO   |     | NULL    |                |
| role          | varchar(15)  | YES  |     | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
7 rows in set (0.08 sec)
```

b. delivery_address Table

```
mysql> desc delivery_address;
+--------------+-------------+------+-----+---------+----------------+
| Field        | Type        | Null | Key | Default | Extra          |
+--------------+-------------+------+-----+---------+----------------+
| id           | bigint      | NO   | PRI | NULL    | auto_increment |
| adress_line1 | varchar(50) | NO   |     | NULL    |                |
| adress_line2 | varchar(50) | NO   |     | NULL    |                |
| city         | varchar(50) | NO   |     | NULL    |                |
| state        | varchar(50) | NO   |     | NULL    |                |
| zip_code     | varchar(8)  | NO   |     | NULL    |                |
+--------------+-------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```

B. Tables Related to Orders

a. carts Table

```
mysql> desc carts;
+-------------+--------+------+-----+---------+----------------+
| Field       | Type   | Null | Key | Default | Extra          |
+-------------+--------+------+-----+---------+----------------+
| id          | bigint | NO   | PRI | NULL    | auto_increment |
| created     | date   | YES  |     | NULL    |                |
| total_items | int    | YES  |     | NULL    |                |
| total_price | double | YES  |     | NULL    |                |
| updated     | date   | YES  |     | NULL    |                |
| user_id     | bigint | YES  | MUL | NULL    |                |
+-------------+--------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```

b. cart_items Table

```
mysql> desc cart_items;
+-------------+--------+------+-----+---------+----------------+
| Field       | Type   | Null | Key | Default | Extra          |
+-------------+--------+------+-----+---------+----------------+
| id          | bigint | NO   | PRI | NULL    | auto_increment |
| quantity    | int    | YES  |     | NULL    |                |
| total_price | double | NO   |     | NULL    |                |
| cart_id     | bigint | YES  | MUL | NULL    |                |
| product_id  | bigint | YES  | MUL | NULL    |                |
+-------------+--------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

c. order_details Table

```
mysql> desc order_details;
+-------------+--------+------+-----+---------+----------------+
| Field       | Type   | Null | Key | Default | Extra          |
+-------------+--------+------+-----+---------+----------------+
| id          | bigint | NO   | PRI | NULL    | auto_increment |
| quantity    | int    | YES  |     | NULL    |                |
| total_price | double | YES  |     | NULL    |                |
| order_id    | bigint | YES  | MUL | NULL    |                |
| product_id  | bigint | YES  | MUL | NULL    |                |
+-------------+--------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

d. orders Table

```
mysql> desc orders;
+----------------+--------------+------+-----+---------+----------------+
| Field          | Type         | Null | Key | Default | Extra          |
+----------------+--------------+------+-----+---------+----------------+
| id             | bigint       | NO   | PRI | NULL    | auto_increment |
| delivery_date  | date         | NO   |     | NULL    |                |
| order_date     | date         | NO   |     | NULL    |                |
| shipping_price | double       | YES  |     | NULL    |                |
| status         | varchar(255) | NO   |     | NULL    |                |
| total_price    | double       | NO   |     | NULL    |                |
| address_id     | bigint       | YES  | MUL | NULL    |                |
| user_id        | bigint       | YES  | MUL | NULL    |                |
+----------------+--------------+------+-----+---------+----------------+
8 rows in set (0.00 sec)
```

C. Tables Related to Product

a. categories Table

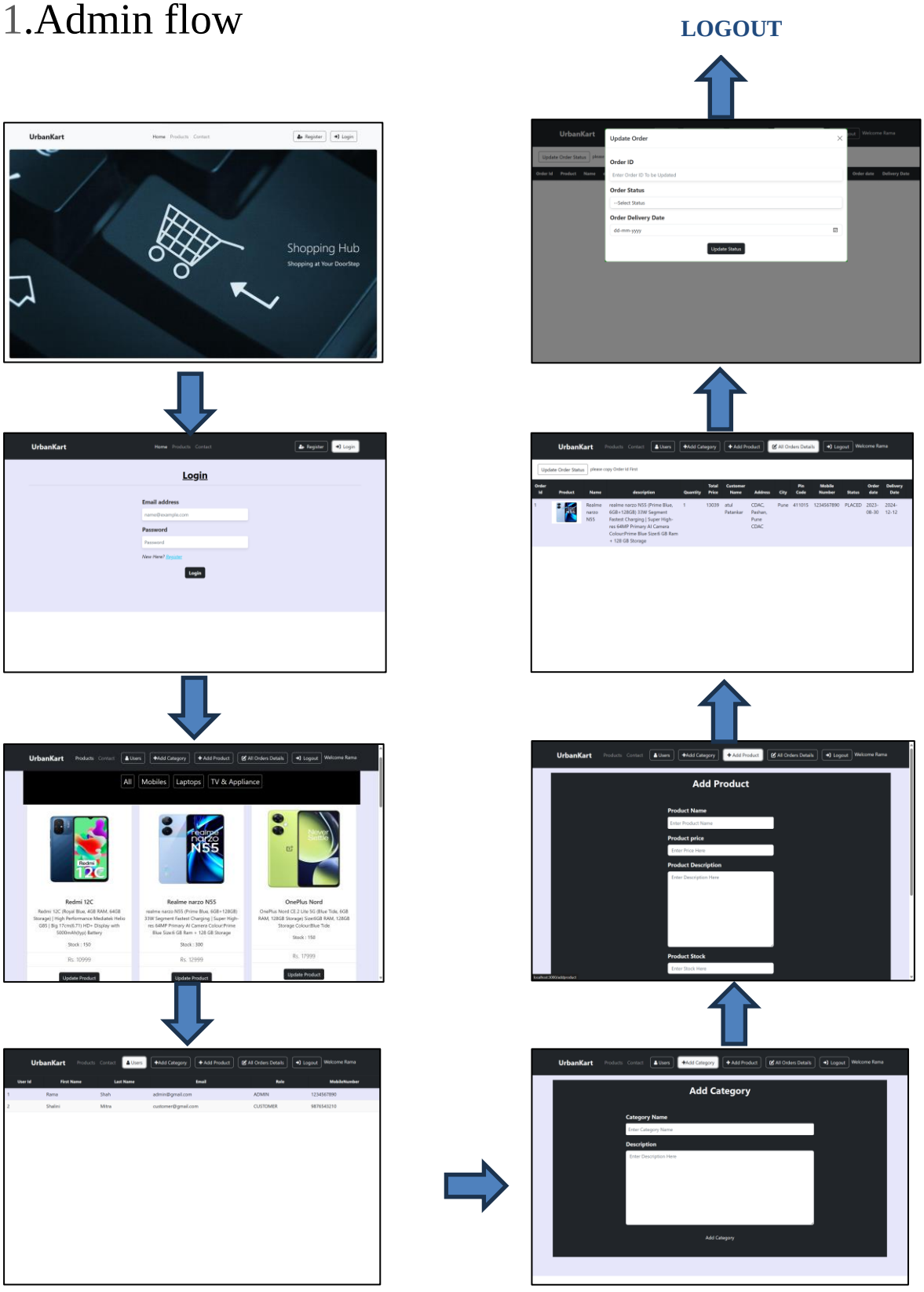
```
mysql> desc categories;
+---------------+---------------+------+-----+---------+----------------+
| Field         | Type          | Null | Key | Default | Extra          |
+---------------+---------------+------+-----+---------+----------------+
| id            | bigint        | NO   | PRI | NULL    | auto_increment |
| category_name | varchar(50)   | NO   |     | NULL    |                |
| description   | varchar(1000) | NO   |     | NULL    |                |
+---------------+---------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)
```

b. products Table

```
mysql> desc products;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| id          | bigint       | NO   | PRI | NULL    | auto_increment |
| description | varchar(255) | YES  |     | NULL    |                |
| exp_date    | date         | YES  |     | NULL    |                |
| image_path  | varchar(255) | YES  |     | NULL    |                |
| name        | varchar(20)  | NO   |     | NULL    |                |
| price       | double       | NO   |     | NULL    |                |
| stock       | int          | YES  |     | NULL    |                |
| category_id | bigint       | YES  | MUL | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
8 rows in set (0.00 sec)
```

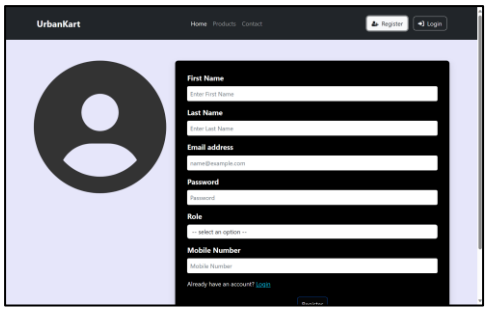
4.2. PAGE FLOW DIAGRAM

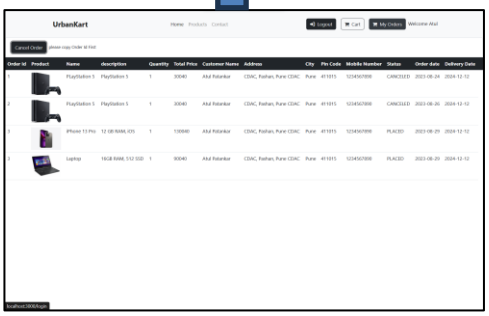
1.Admin flow

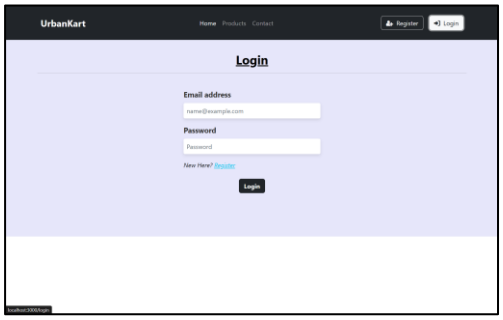


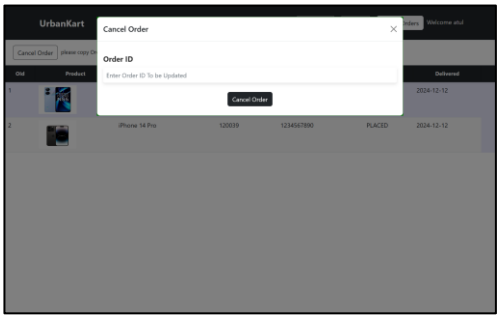
2.Customer Flow

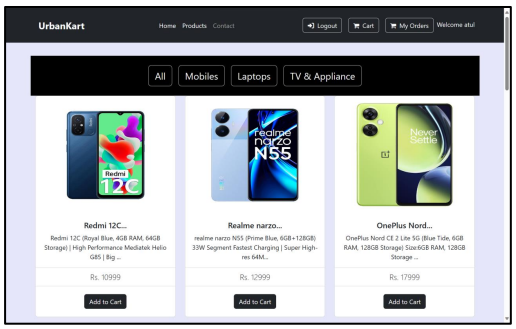
LOGOUT

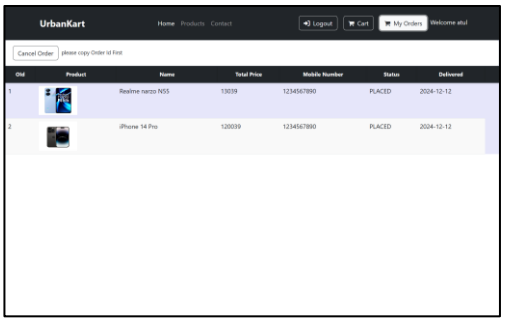


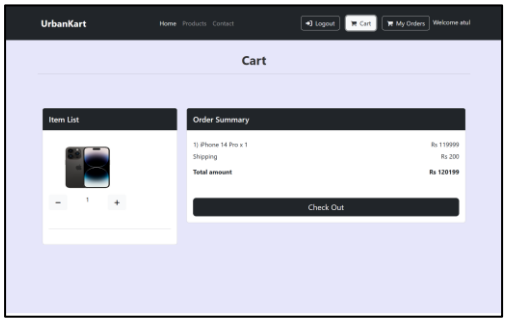


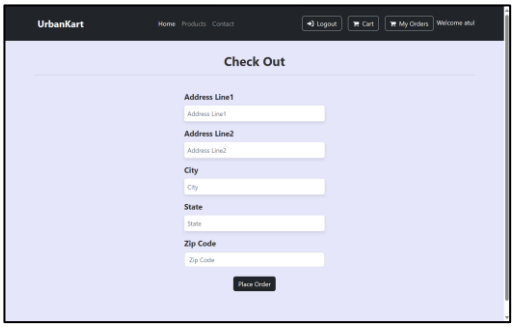




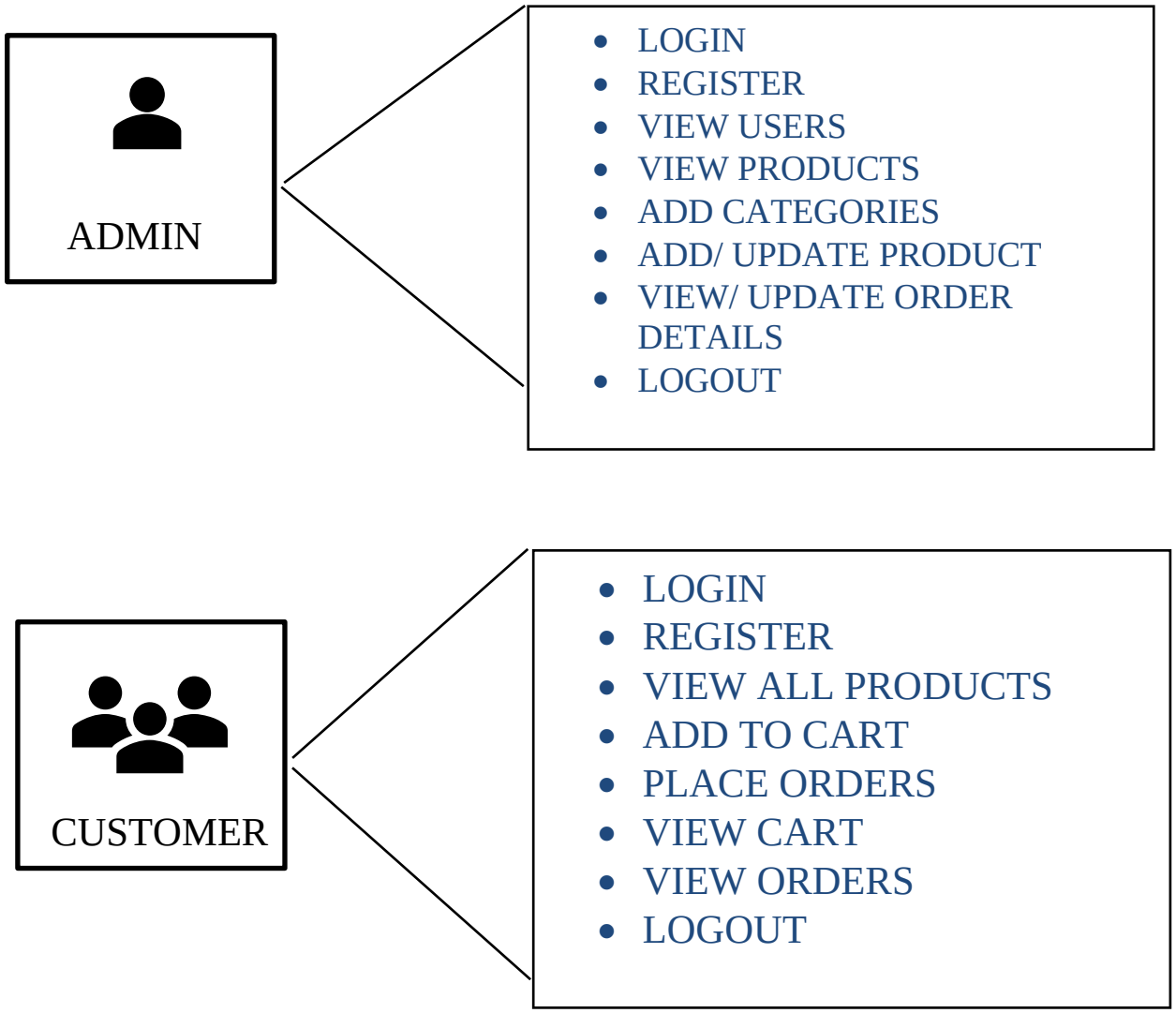




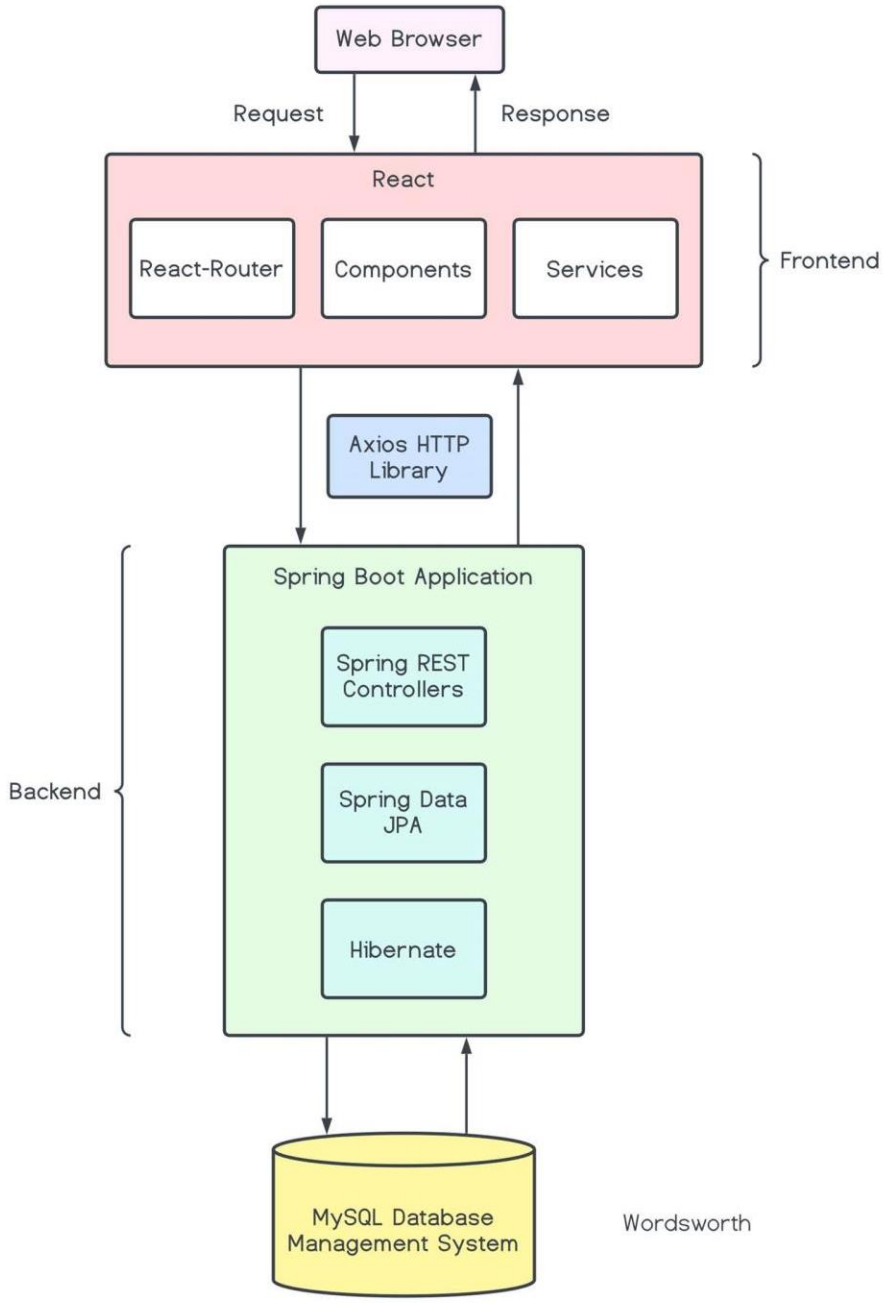




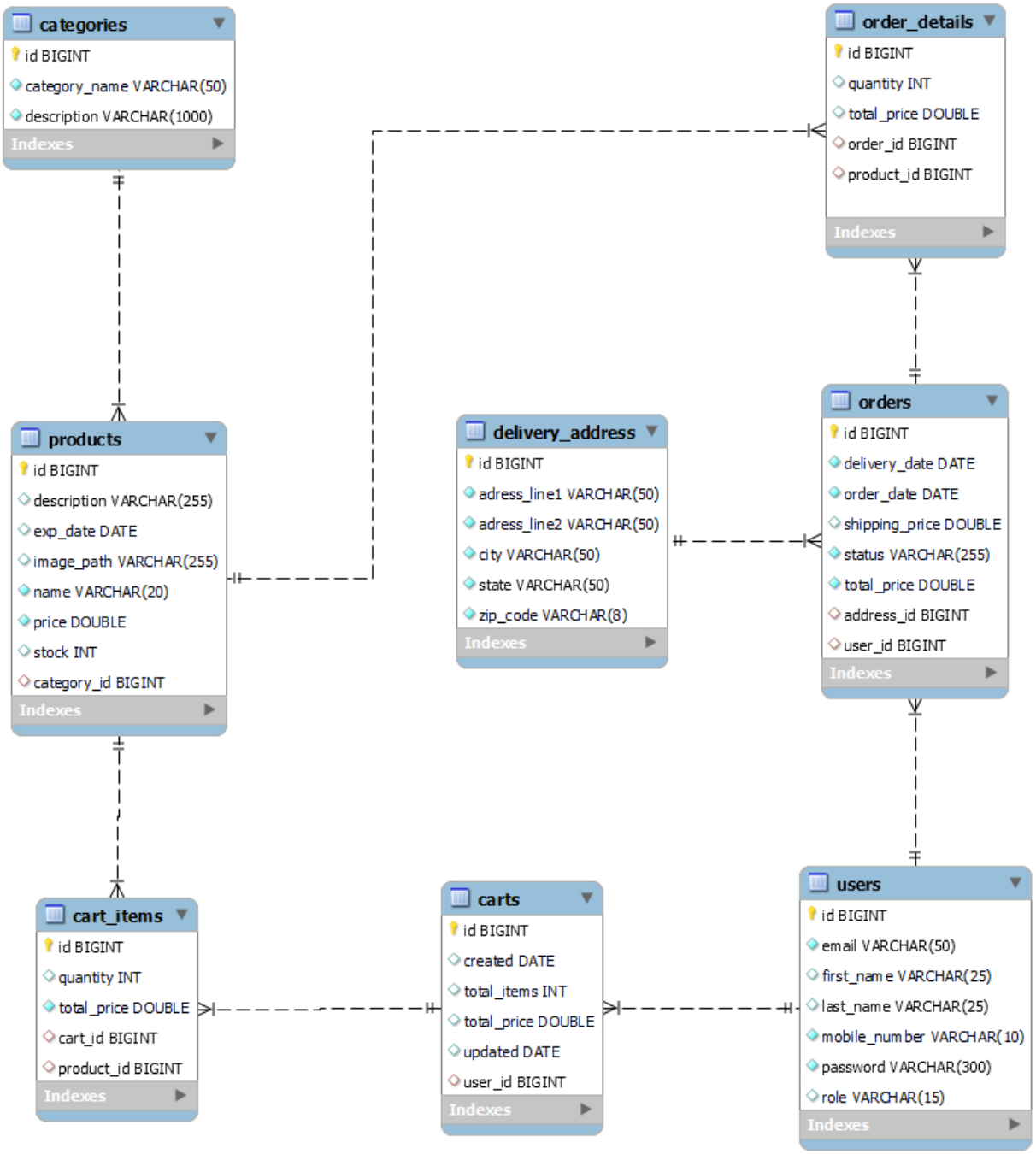
4.3. USE CASE DIAGRAM



4.4. PROJECT ARCHITECTURE



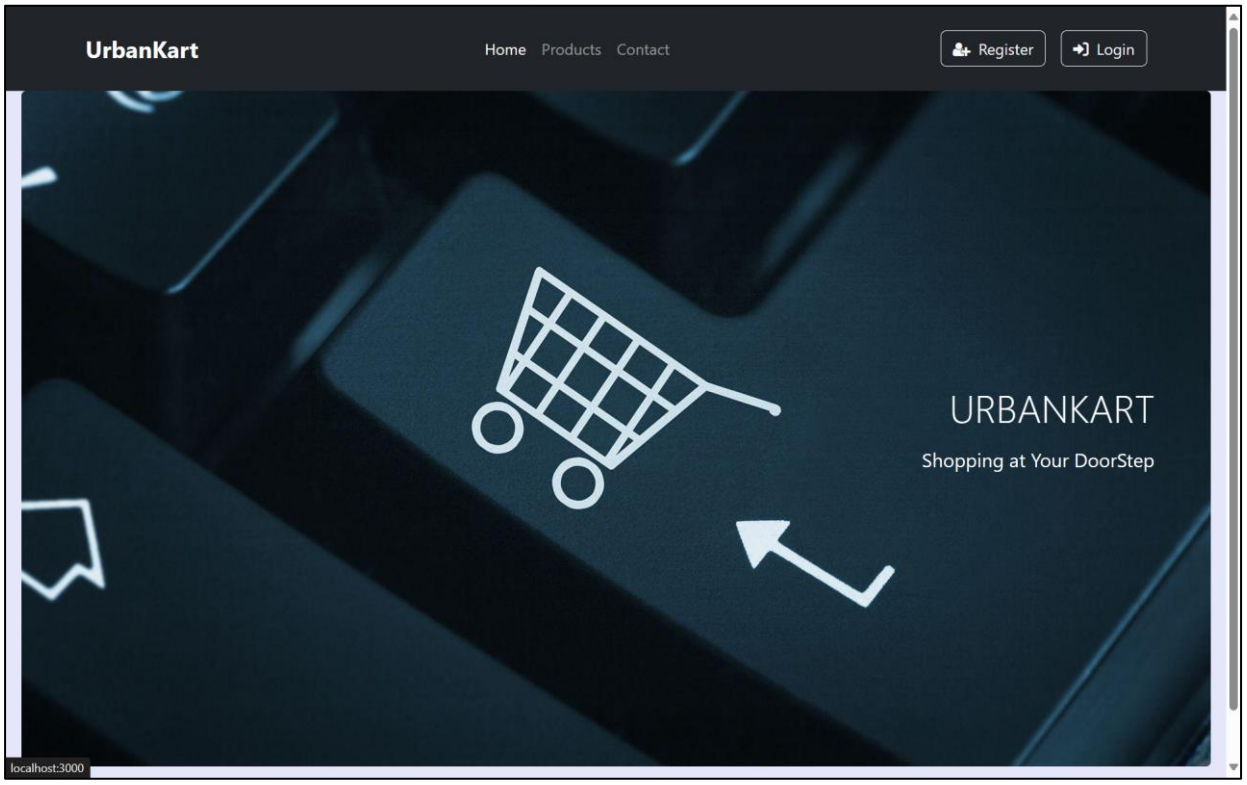
4.5. ER DIAGRAM



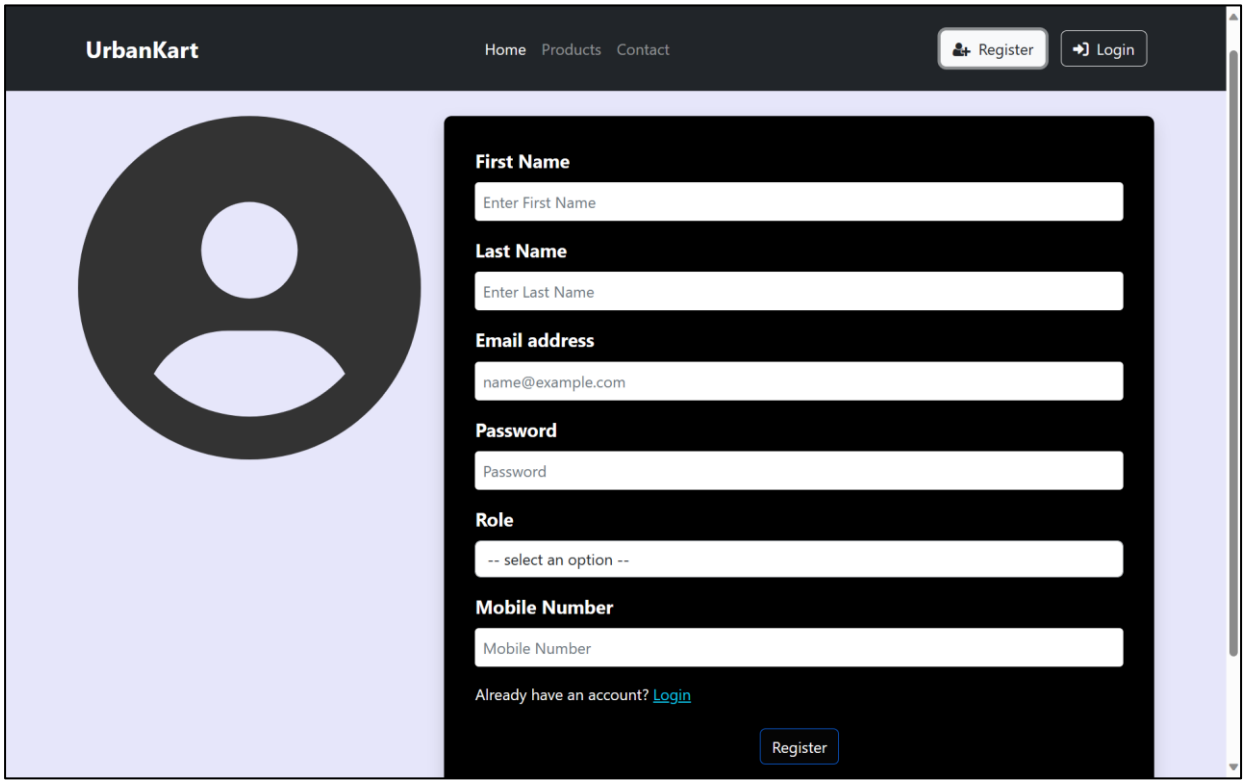
5. PROJECT SCREENSHOTS

5.1. CUSTOMER

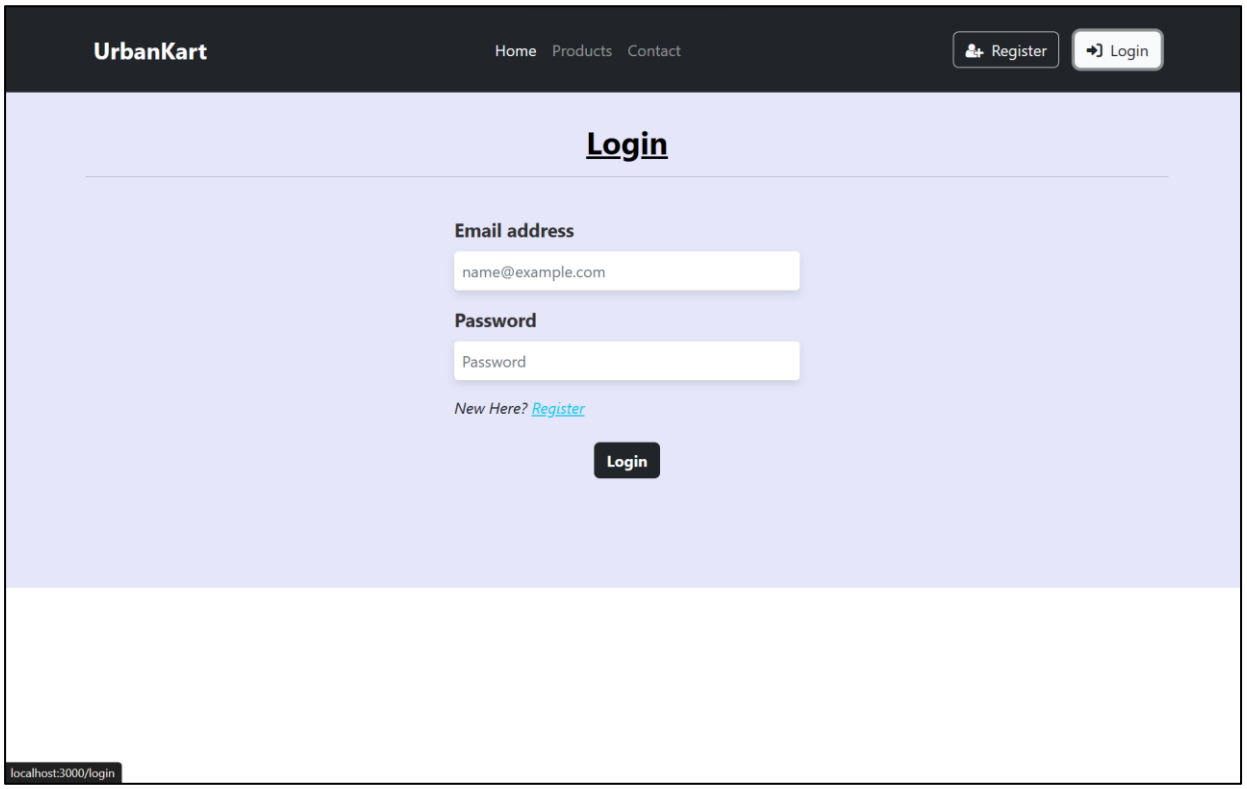
HOME PAGE



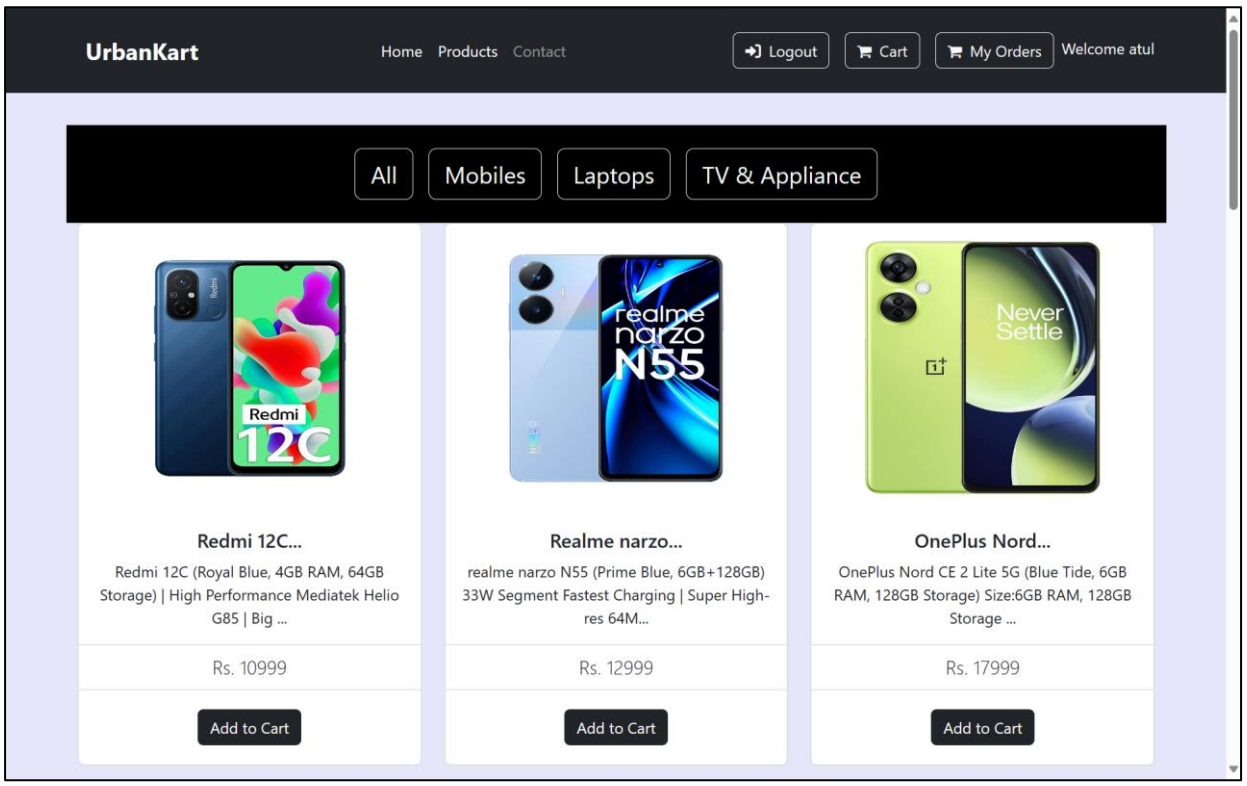
REGISTRATION PAGE



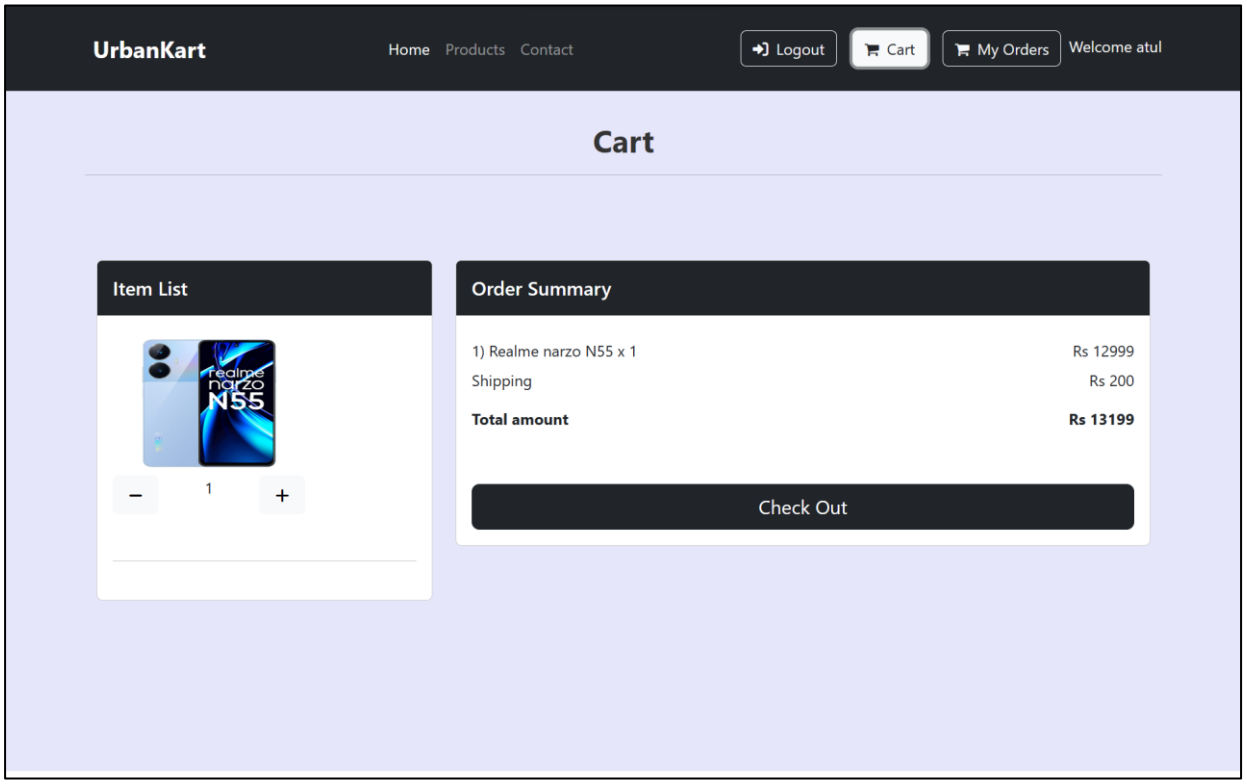
LOGIN PAGE



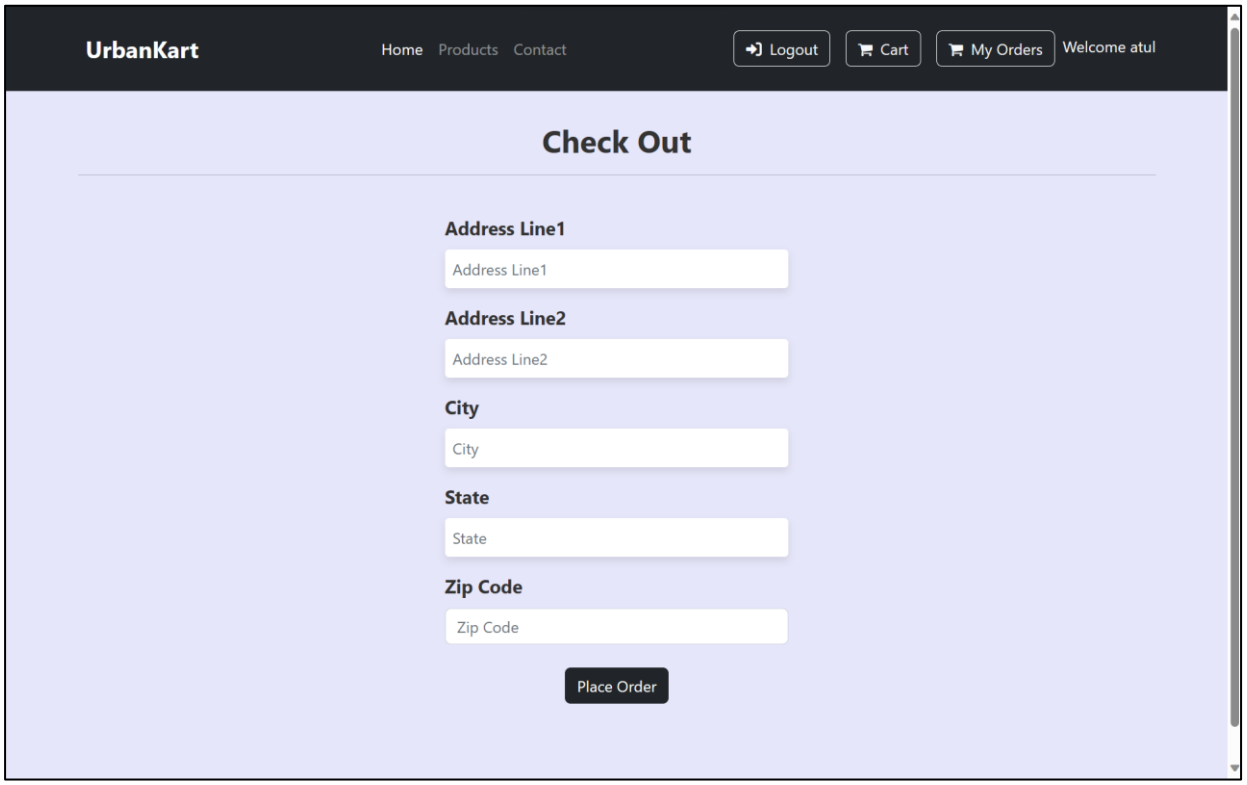
VIEW PRODUCTS FROM CATEGORIES



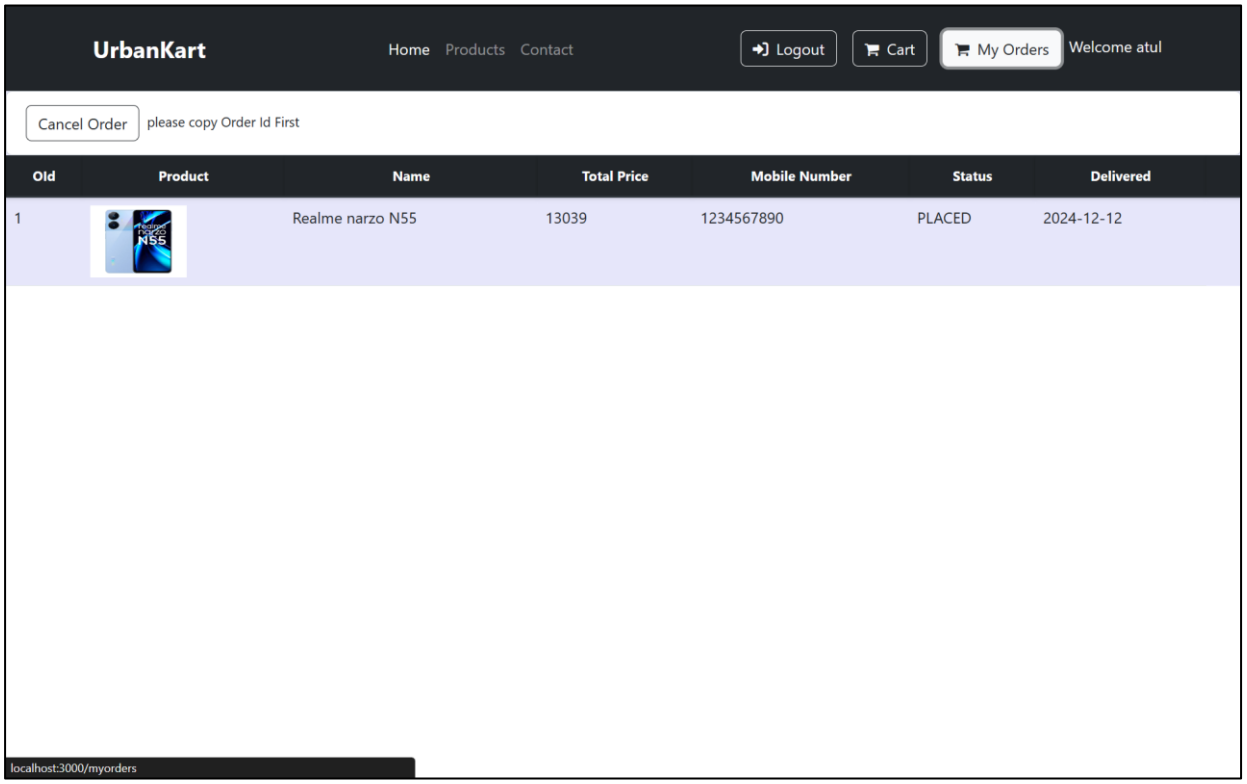
MY CART VIEW



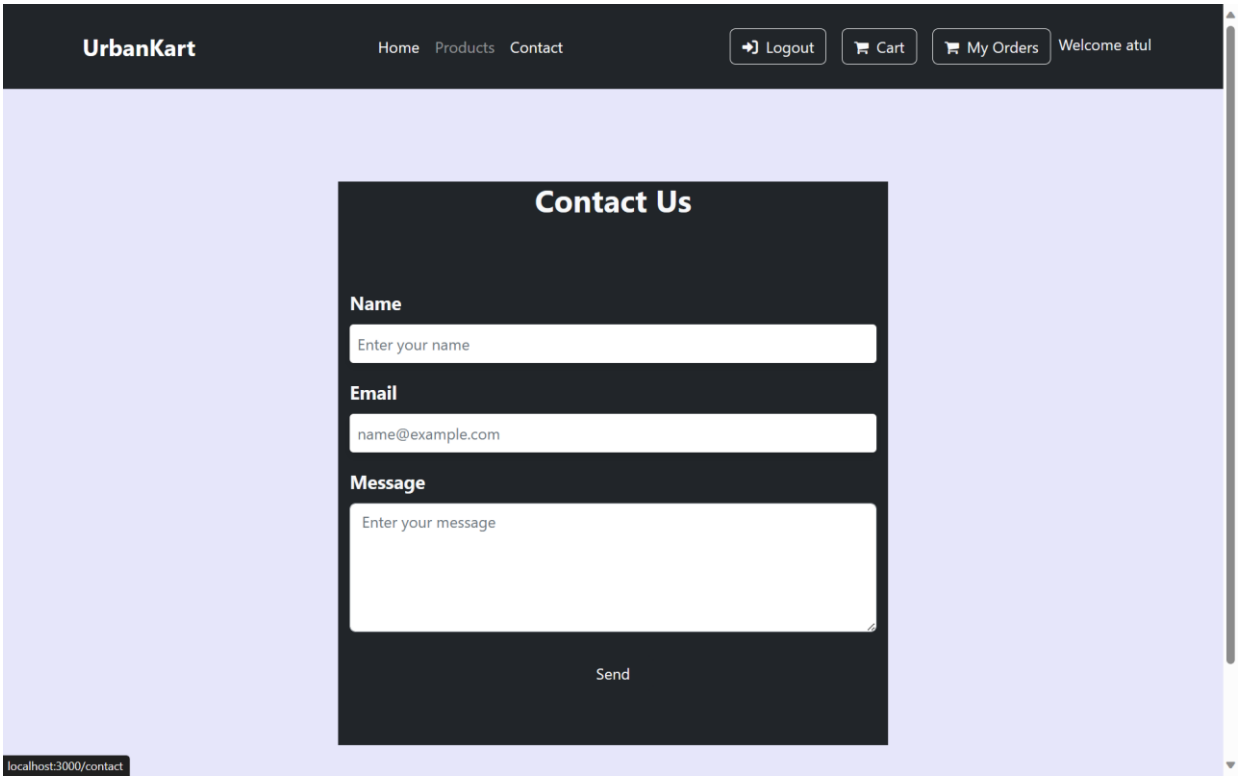
ADDRESS CHECKOUT PAGE



ORDER PLACED

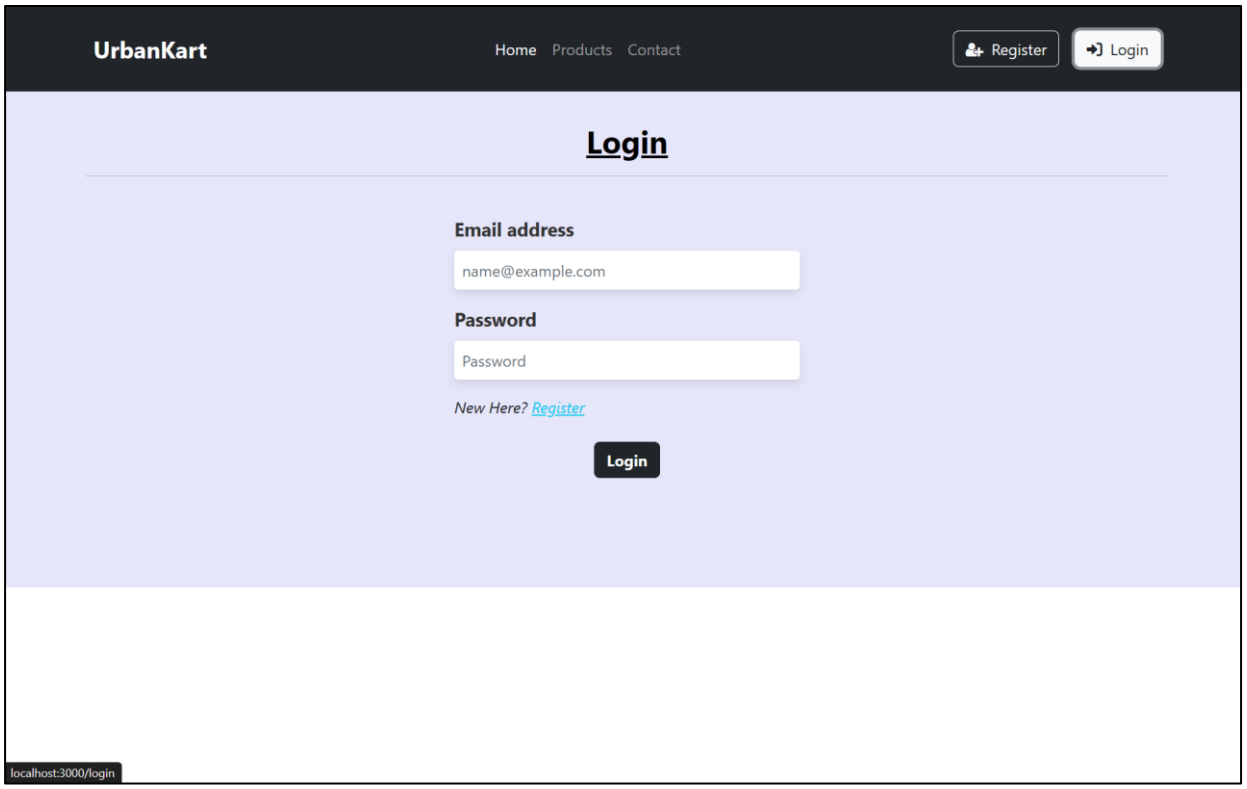


CONTACT US

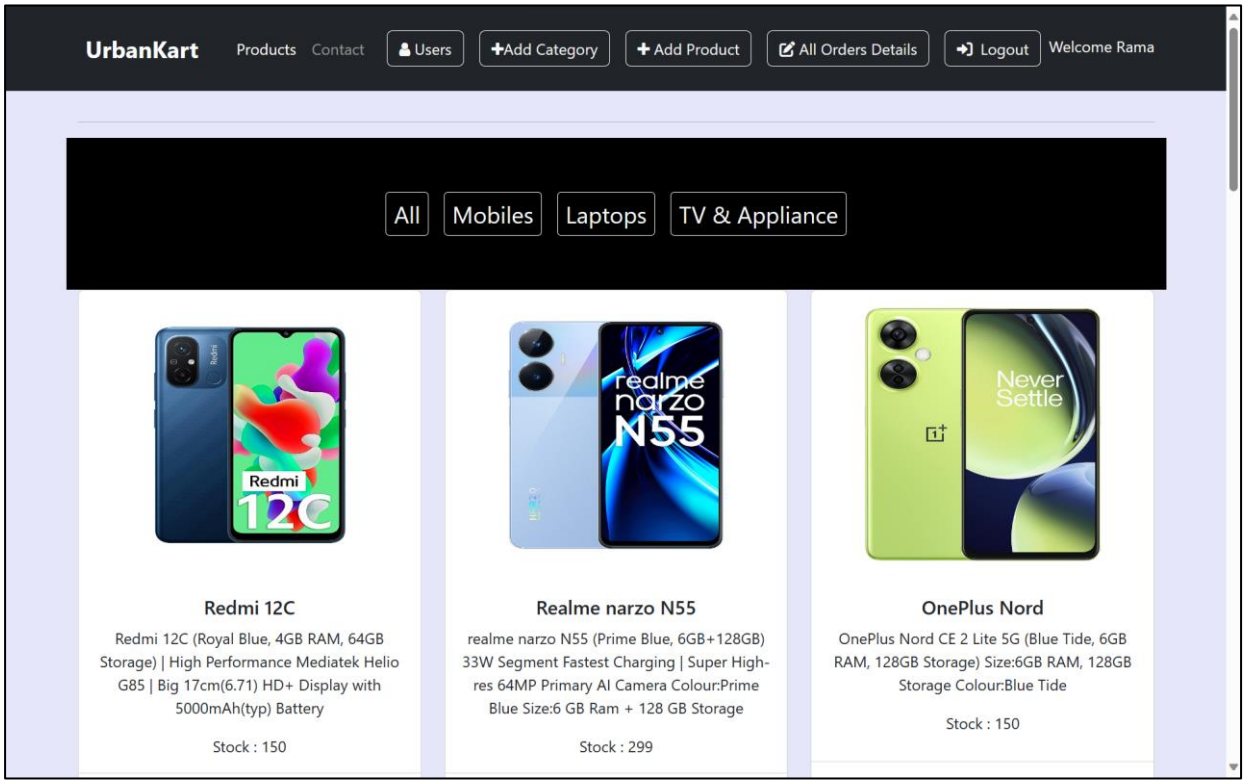


5.1. ADMIN

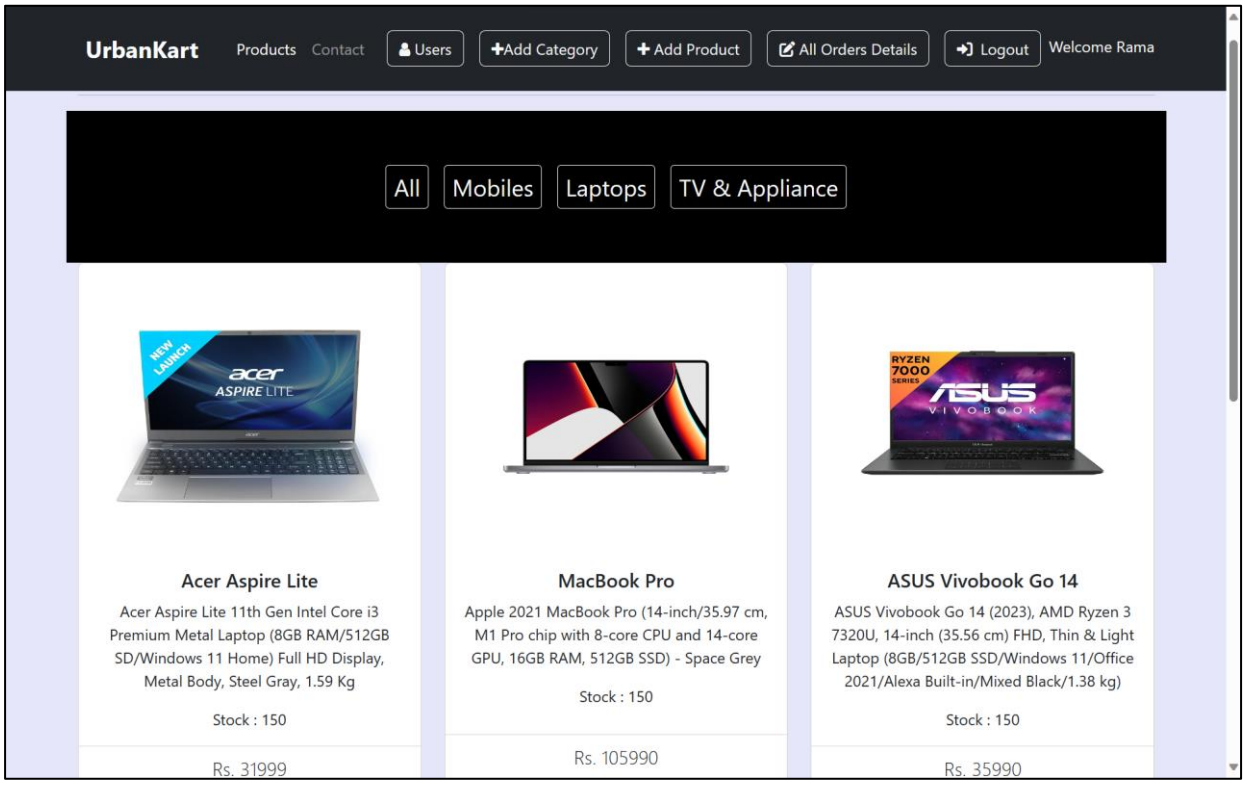
LOGIN



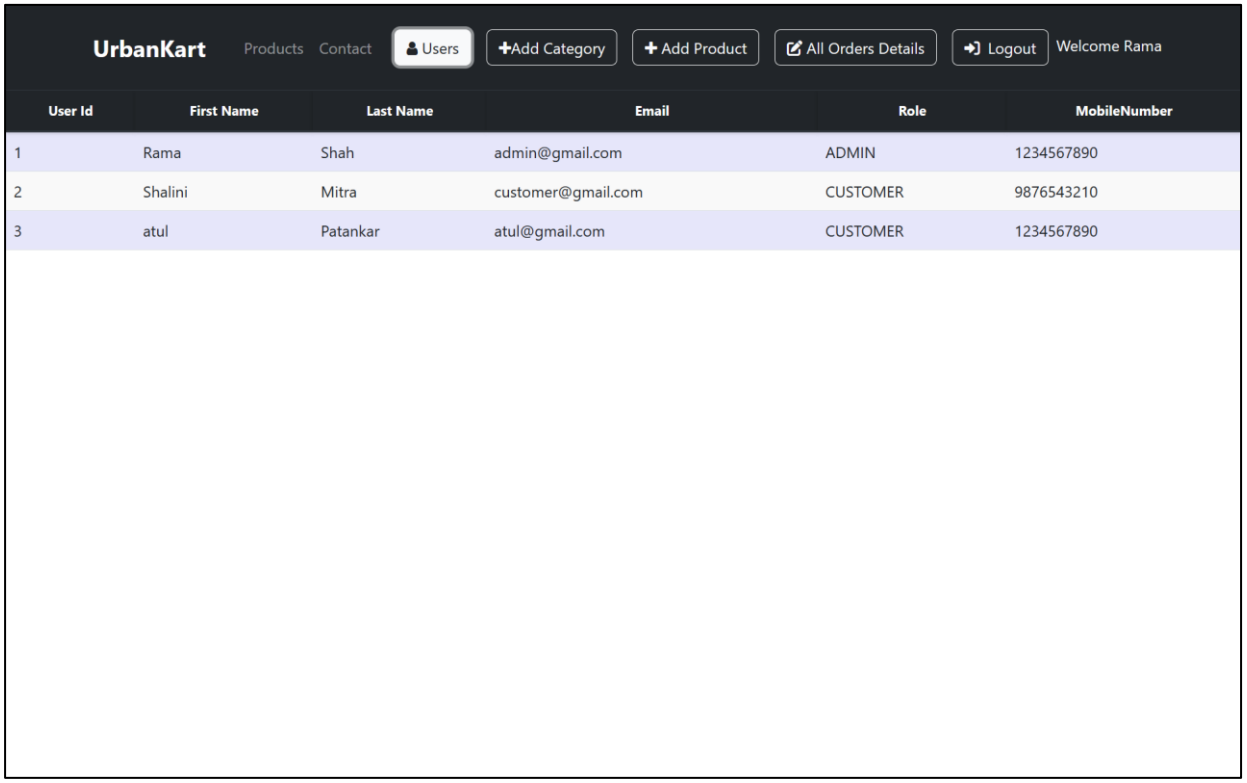
ADMIN HOME PAGE



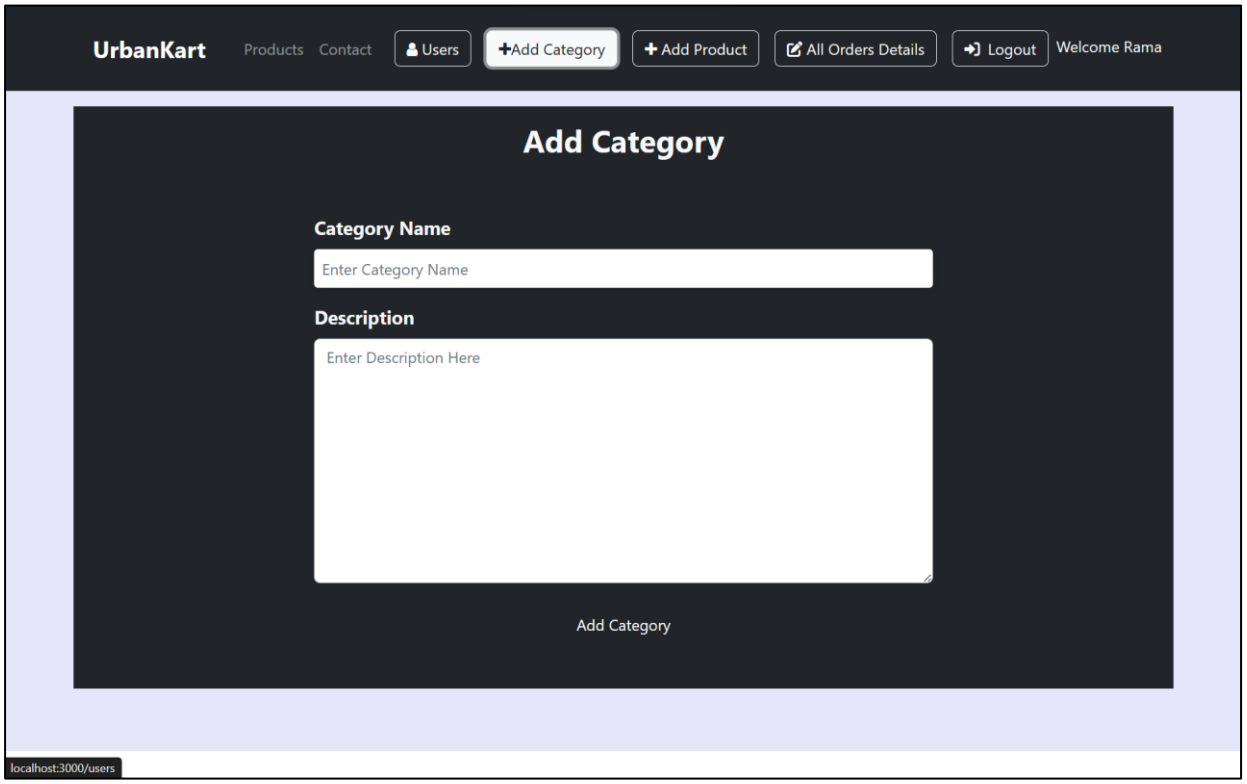
VIEW PRODUCT BY CATEGORIES



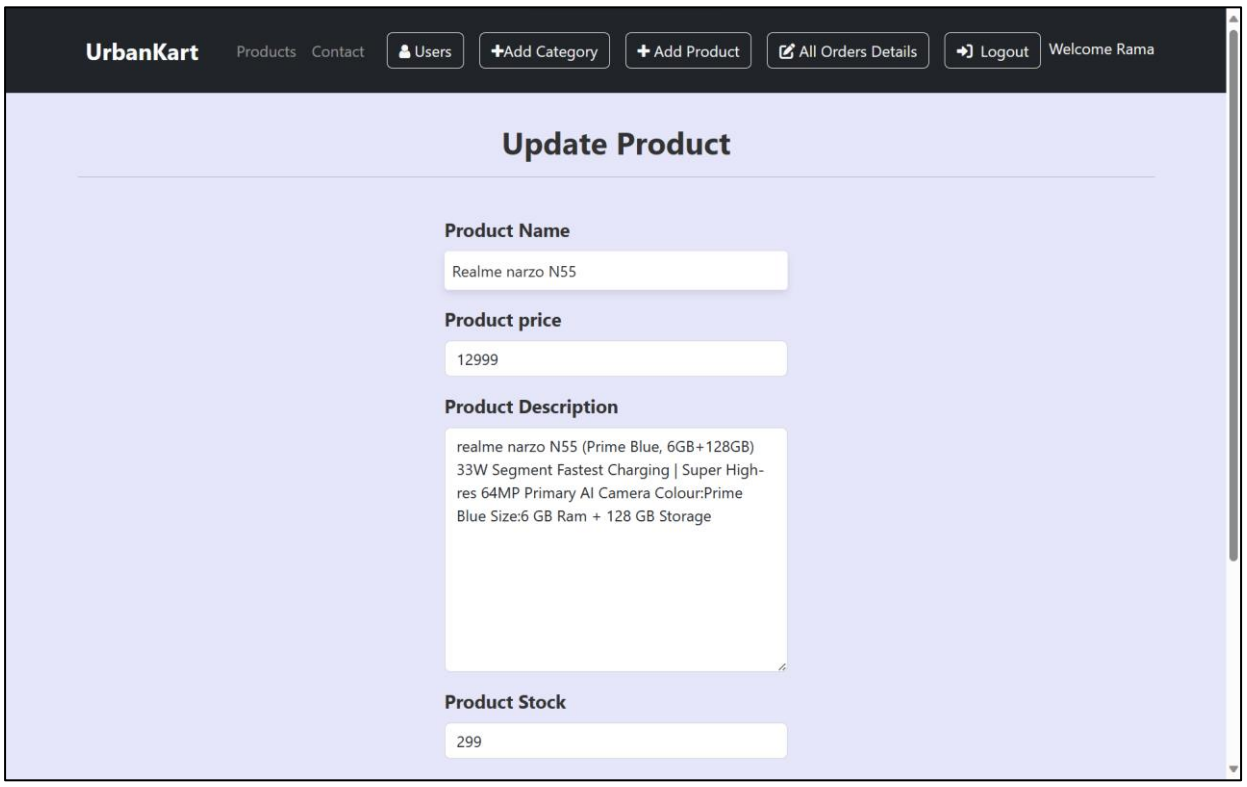
VIEW ALL USERS



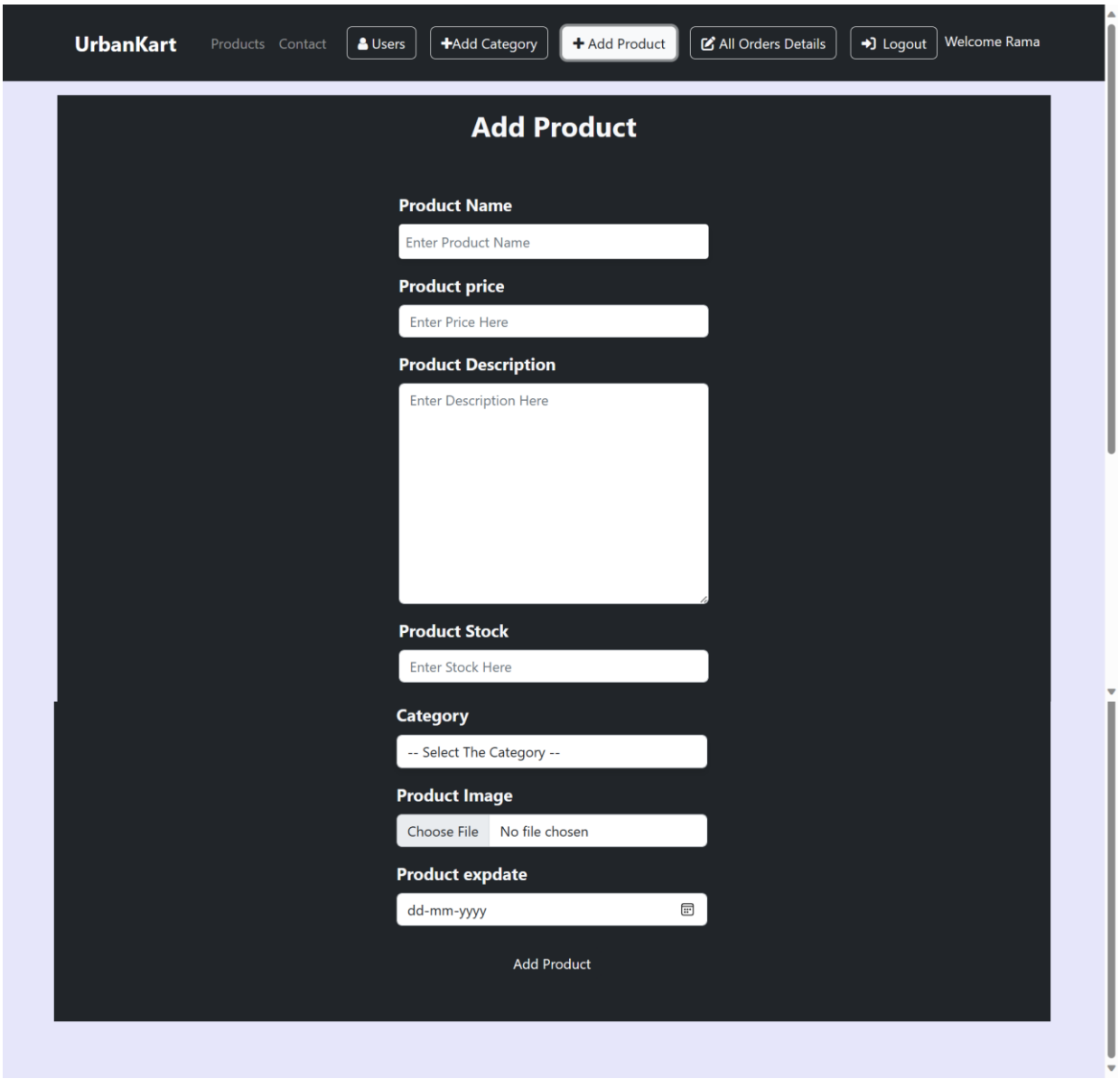
ADD CATEGORY



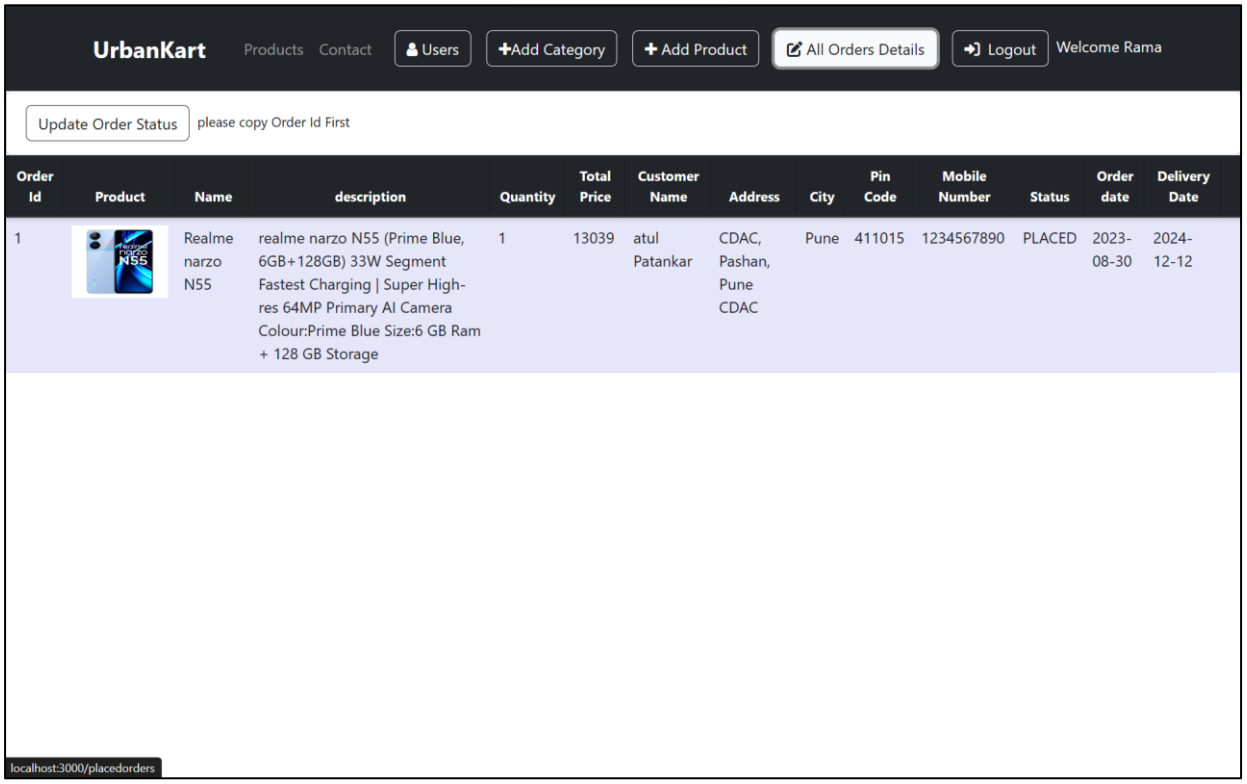
UPDATE PRODUCT



ADD PRODUCT


UrbanKart
Products
Contact
Users
+Add Category
+ Add Product
All Orders Details
Logout
Welcome Rama
Add Product
Product Name
Enter Product Name
Product price
Enter Price Here
Product Description
Enter Description Here
Product Stock
Enter Stock Here
Category
-- Select The Category --
Product Image
Choose File
No file chosen
Product expdate
dd-mm-yyyy
Add Product

ALL ORDER DETAILS



6. 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.

In the course of 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

S. No.	Description	Outcome	Result
1.	Register	New customer details saved in the database.	Passed
2.	Login	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

S. No.	Description	Outcome	Result
1.	Login	Fetched authenticated user details saved in database.	Passed
2.	Add New Product	The details of a new product were updated in the database.	Passed
3.	Manage Product Inventory	The inventory of the products was updated in the database.	Passed
4.	Logout	The session was cleared.	Passed

7. CONCLUSION

Our team developed the "UrbanKart" application to simplify online buying and selling. We utilized cutting-edge, versatile technologies that work well across different platforms. Additionally, we consciously chose open-source software to minimize production costs.

We also paid careful attention to the user experience, ensuring that our website is easy to navigate and provides a smooth experience.

In summary, "UrbanKart" is an excellent choice for any product trading business looking to establish an online presence. We are confident that its numerous features and visually appealing design will significantly enhance the business's prospects.

8. FUTURE SCOPE

Throughout our course, we've strived to make our project user-friendly and packed it with as many features as possible within our project timeline. Nevertheless, there are additional features that could enhance our application:

1. Highlighting top-selling products to promote customer favorites.
2. Offering personalized discounts based on a customer's purchase history and the quantity of products they buy.
3. Allowing customers to interact with feedback by upvoting, downvoting, or reporting it.
4. Expanding payment options beyond just credit cards.
5. Sending email invoices to customers after confirming their purchases.
6. Implementing a 'reset password' feature for users who forget their passwords.
7. Enhancing security with CAPTCHA on the login page.
8. Sending email notifications to users when items in their cart, previously out of stock, become available.

These additions would further enrich the user experience and functionality of our application.

9. 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