UrbanKart Ecommerce

By

Suraj Goswami (Roll No.: 10800221012)
Arindol Ghosh (Roll No.: 10800221100)
Binoy Gorai (Roll No.: 10800221117)
Debashis Kar (Roll No.: 10800221120)

Project Report Synopsis

Submitted to Asansol Engineering College in partial fulfillment of the requirements for the degree of

Bachelor of Technology
(Information Technology)
of
MAKAUT

Under the guidance
of

Mr.Santu Mondal
(Assistant Professor)

Department of Information Technology
Asansol Engineering College
Asansol

CERTIFICATE FROM PROJECT GUIDE

This is to certify that the project entitled "URBANKART ECOMMERCE" submitted by Suraj Goswami (Roll No.: 10800221012), Arindol Ghosh (Roll No.: 10800221100), Binoy Gorai (Roll No.: 10800221117), Debashis Kar (Roll No.: 10800221120) for the award of B. Tech. (Information Technology) degree of MAKAUT is absolutely based upon his own work under the supervision of Mr.Santu Mondal, Department of Information Technology, Asansol Engineering College, Asansol, India and that neither his project report nor any part of the report has been submitted for any degree/diploma or any other academic award anywhere before.

Name and Designation of Project Guide:

Mr. Santu Mondal (Assistant Professor)
Information Technology
Asansol Engineering College

.....

(Suraj Goswami)

.....

(Arindol Ghosh)

.....

(Binoy Gorai)

.....

(Debashis Kar)

Minor Project Report:-

Motto of the Project:

The primary objective of our e-commerce project is to broaden the reach of businesses by connecting them with a wider customer base at the most opportune moments, thereby driving sales and enhancing profitability. E-commerce encompasses a range of activities, including the buying and selling of goods, as well as the transfer of funds and data over the internet. Our project's aim is to extend the benefits of online shopping to customers of traditional retail stores, allowing them to purchase items from their preferred shops via the internet, regardless of their location. This system facilitates the convenience of online shopping combined with the familiarity of local stores, offering services such as home delivery. It is adaptable for implementation in any local shop or even large multinational retail chains.

Our project targets individuals who prefer the convenience of online shopping and have access to the internet. We aim to attract customers who value home delivery services and appreciate the ability to compare brands and product quality online. E-commerce offers a swift and straightforward shopping experience, enabling users to purchase and distribute items efficiently. However, we recognize the importance of addressing concerns related to online payment security and product authenticity. Many customers are wary of potential fraud or receiving products that do not match their expectations. While the online shopping process is generally efficient, it must also ensure safety and accuracy to truly meet users' needs.

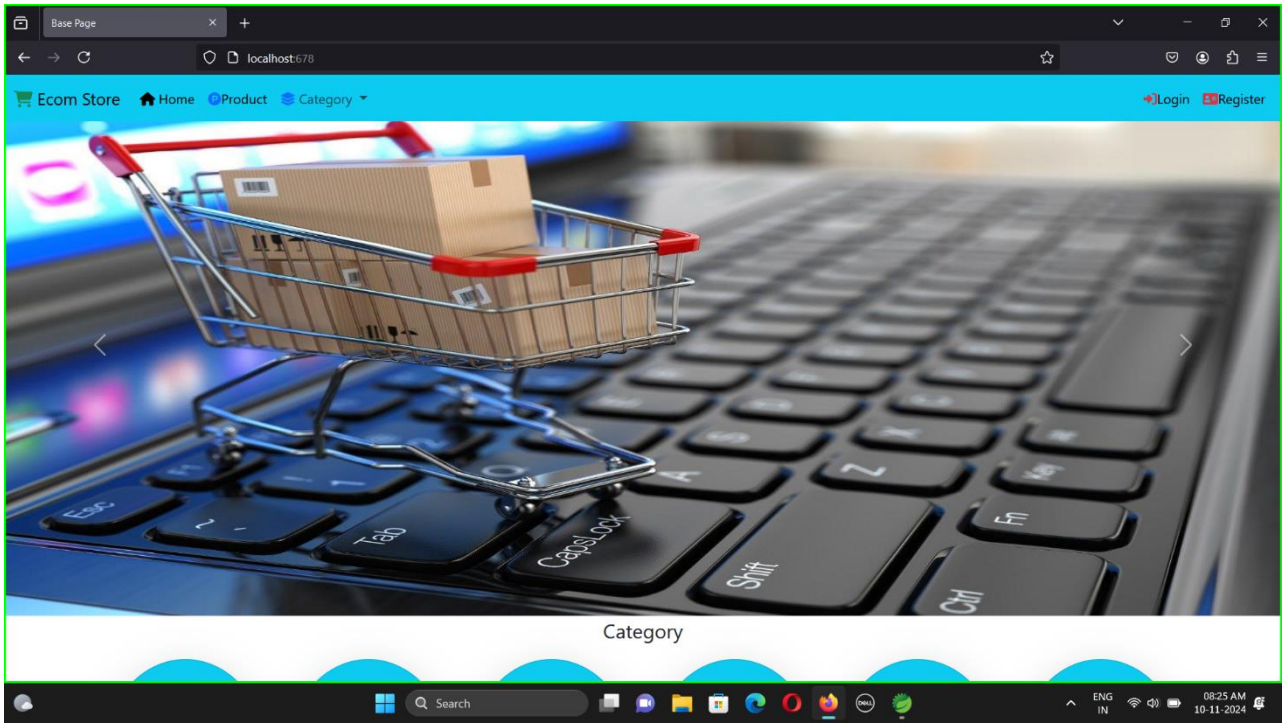
Intended Audience:

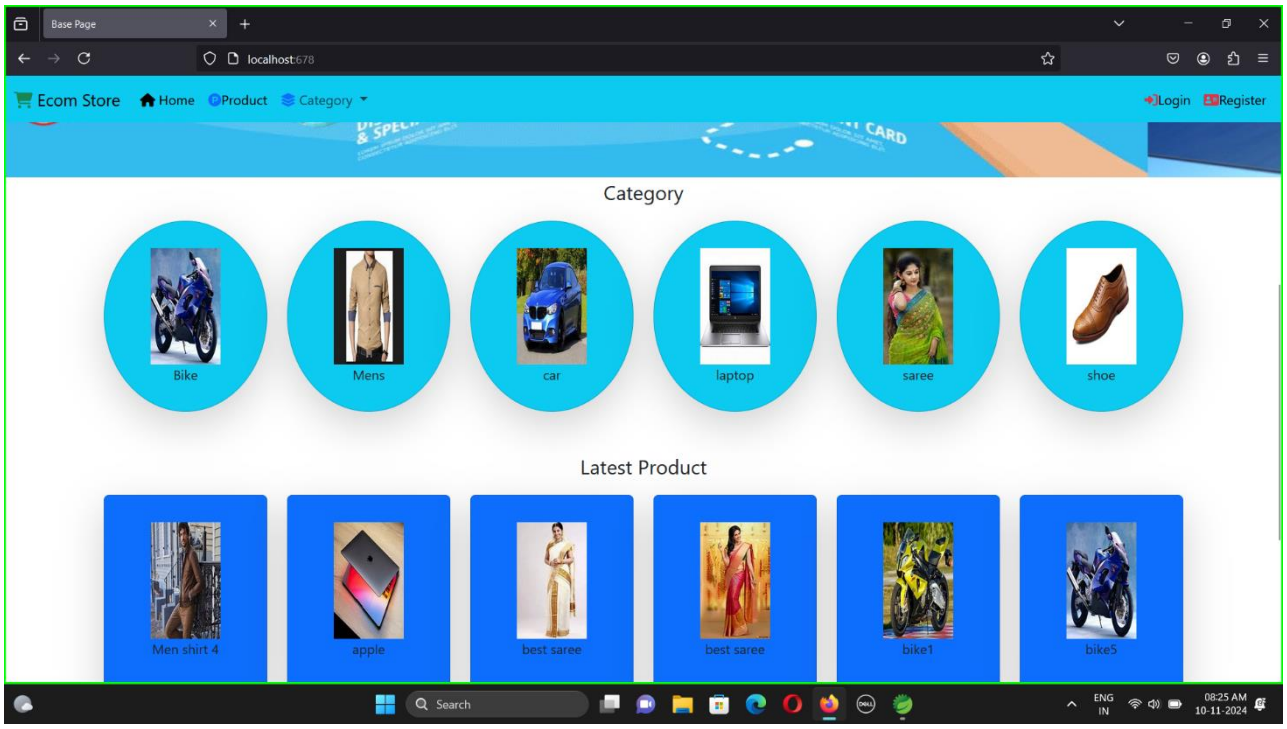
This project is designed for individuals who prefer online shopping and have internet access. Our target audience includes those who seek the convenience of home delivery without additional charges and wish to compare different brands and product qualities before making a purchase. While e-commerce is particularly popular among working adults who may not have time for traditional shopping, it is accessible to anyone with a credit or debit card. Younger consumers also frequently use online shopping services for purchasing items like clothing and games, appreciating the ease and convenience it offers.

Chapter:-1 Introduction

This e-commerce platform offers advanced search, multi-role support (customers/admins), account management, real-time order tracking, and automated email notifications, enhancing user experience and administrative efficiency.

Fundamental features of our side

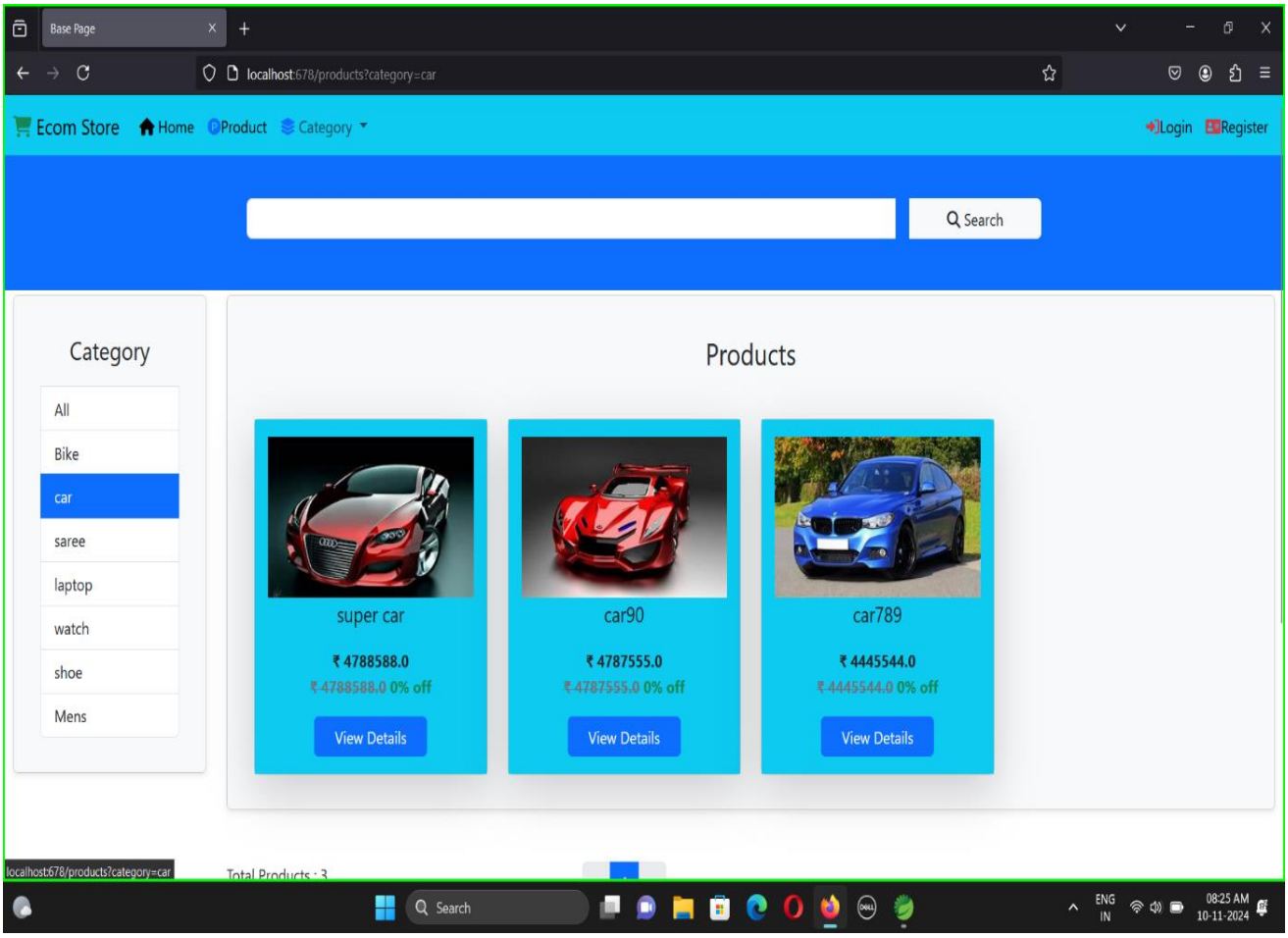




There are some features of our sides as follows:

- **Search Bar:-**

In a Spring Boot application, implementing a search bar involves creating a user-friendly interface that allows users to query a database for specific information. The search bar serves as a critical component in enhancing user experience by enabling quick and efficient data retrieval. To implement a search bar, start by designing the search form in the frontend using HTML and integrating it with a Spring Boot controller. The controller handles incoming search requests and maps them to a method that processes the search query. Typically, the search query is passed as a parameter from the frontend to the backend .In the backend, a service layer interacts with the repository to fetch data from the database based on the search criteria. This can involve simple keyword matching, or more complex queries using JPA (Java Persistence API) or native SQL. The retrieved data is then sent back to the controller, which forwards it to the view layer, displaying the search results to the user.

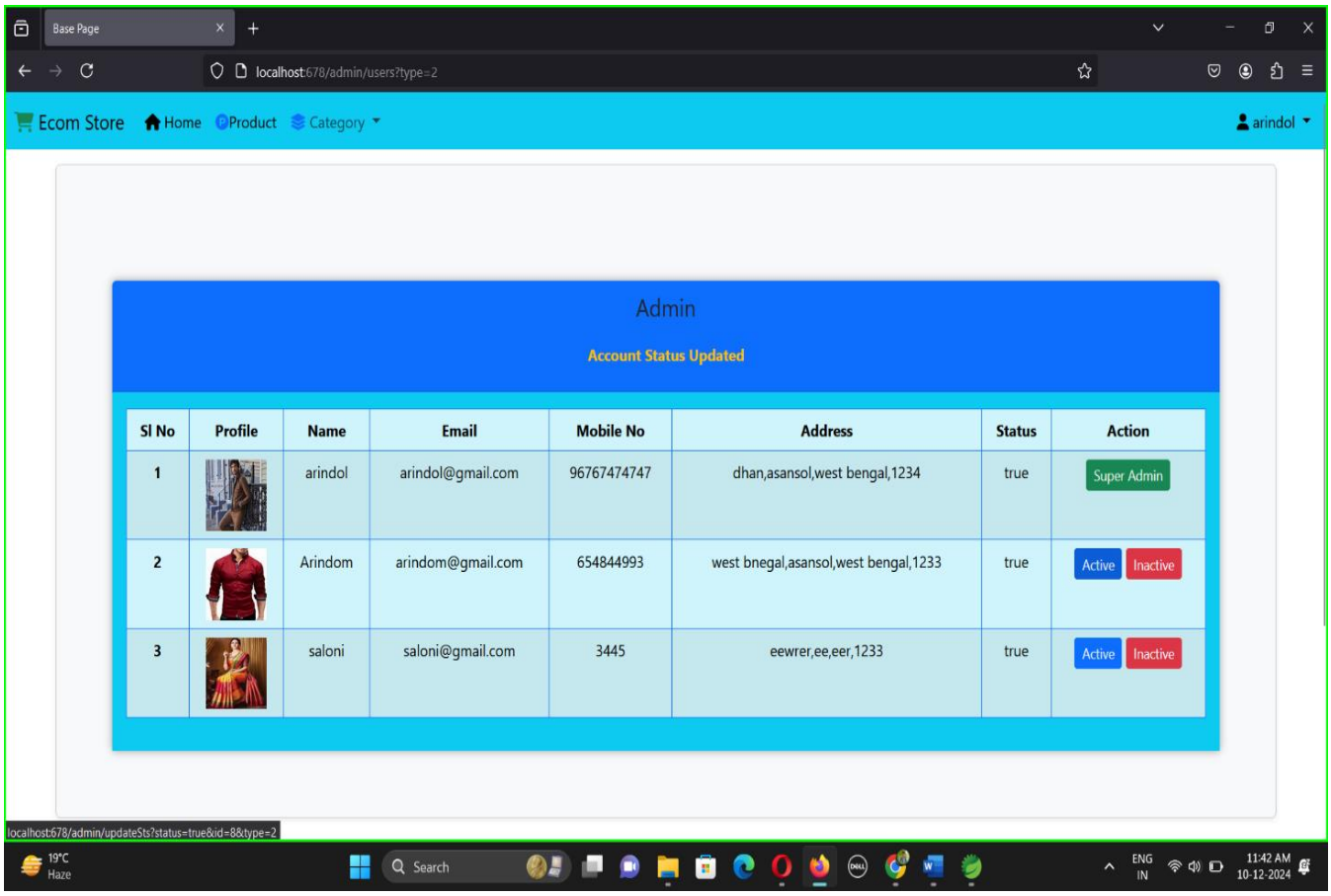


➢ Multiple Admin

Spring Security, a powerful authentication and authorization framework, is used to enforce role-based access. Configure Spring Security to handle authentication (verifying user identity) and authorization (granting or denying access based on roles). You can define security rules in a configuration class by extending WebSecurityConfigurerAdapter and overriding the configure method to specify access controls. For instance, you can restrict access to certain endpoints so that only users with the ADMIN role can access administrative functionalities, while regular users can access general features.

In addition to role-based access control, it is crucial to implement user management features such as registration, login, and profile management. This typically involves creating user registration and login forms, handling user authentication, and managing user sessions.

Overall, Spring Boot simplifies role management through integration with Spring Security, providing a comprehensive approach to secure and manage multiple user roles efficiently within an application.

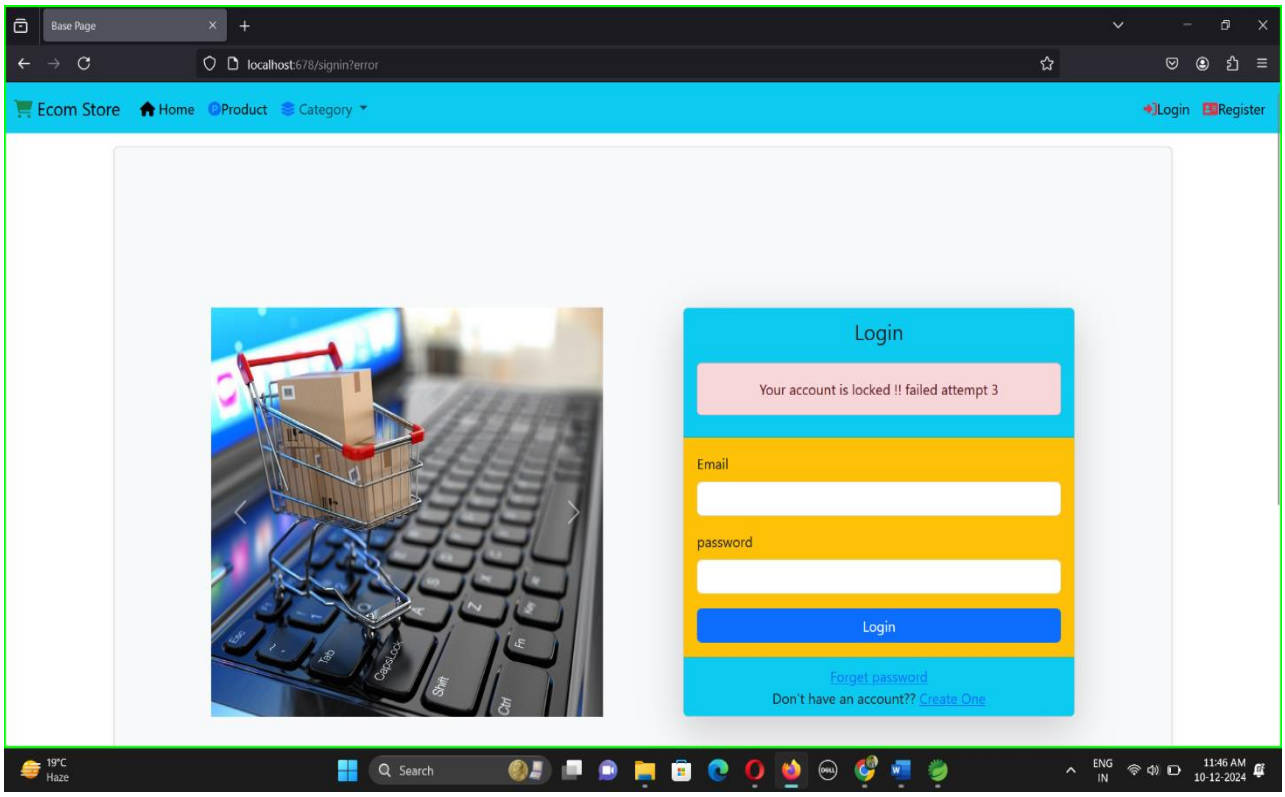


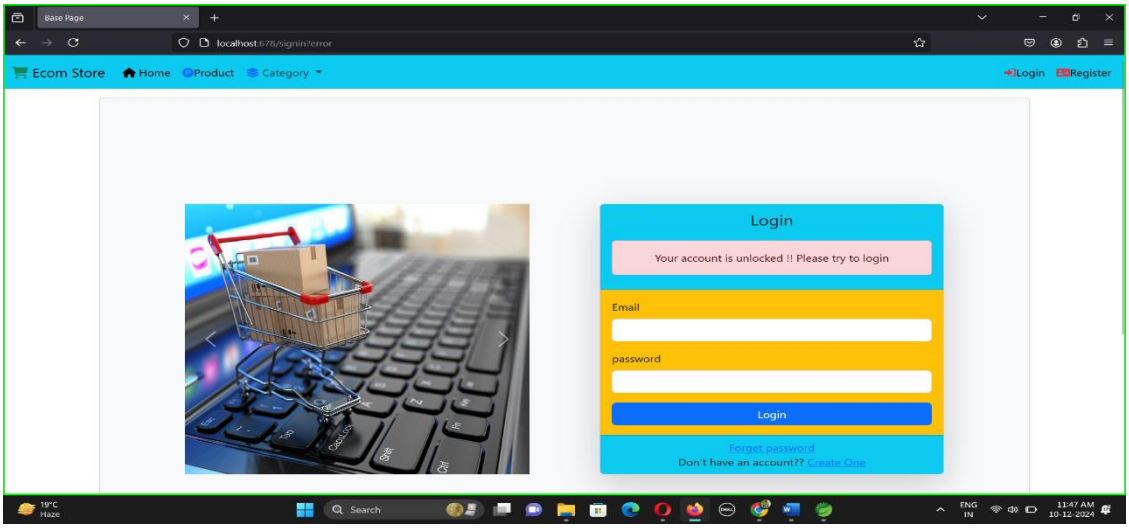
➢ User Account Block & Unblock

In a Spring Boot application, managing user account status, such as blocking and unblocking accounts, is crucial for maintaining security and user management. This functionality ensures that user access can be controlled based on certain criteria, such as suspicious activities or policy violations.

To implement account blocking and unblocking, start by adding an account status field to your user entity. This field, such as isBlocked, typically uses a boolean value to indicate whether an account is currently blocked. You can also use an enumeration for more detailed statuses if needed

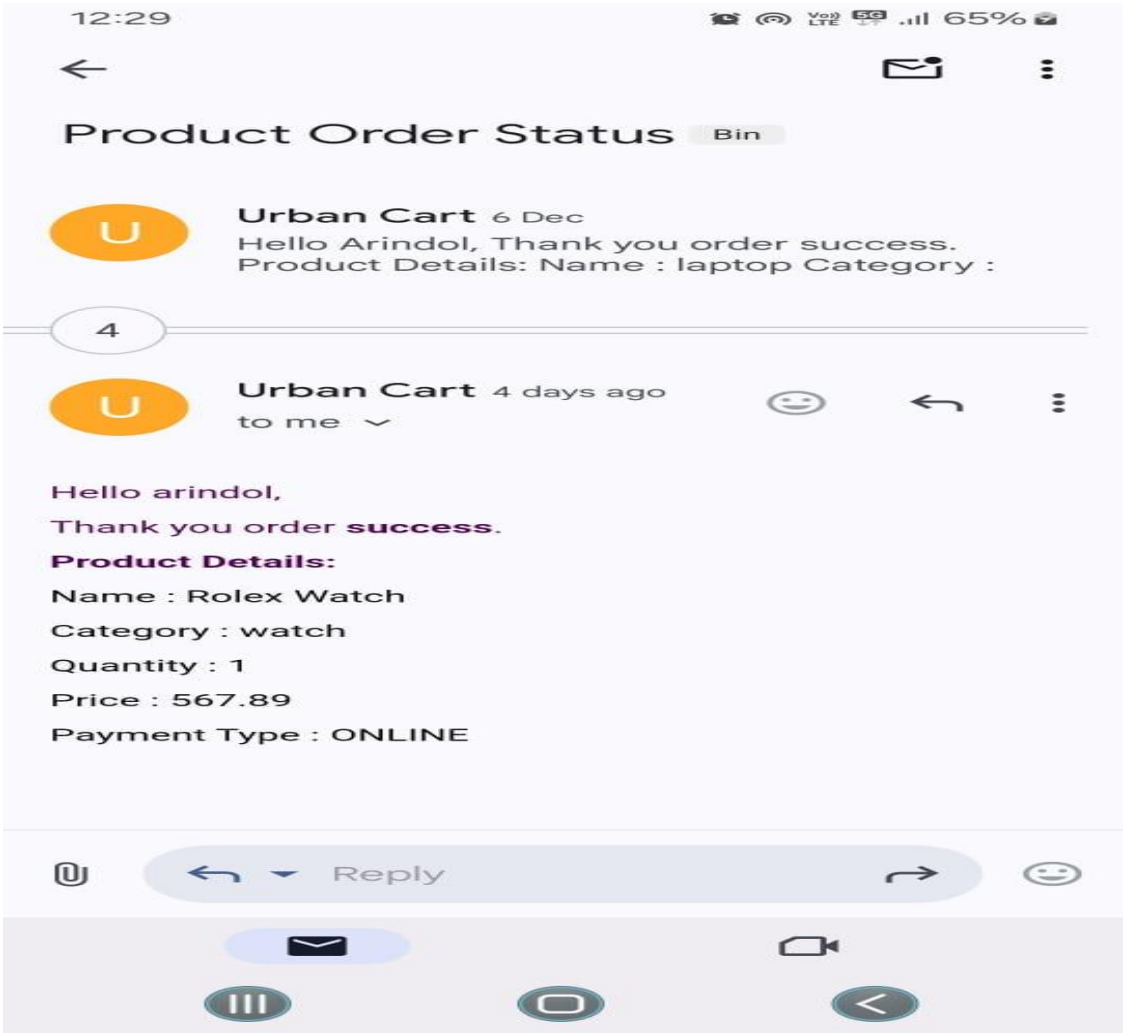
The backend logic for blocking and unblocking accounts is handled through service methods. These methods interact with the repository layer to update the isBlocked status of a user. For instance, a blockUser(Long userId) method would set the isBlocked field to true, while an unblockUser(Long userId) method would set it to false.





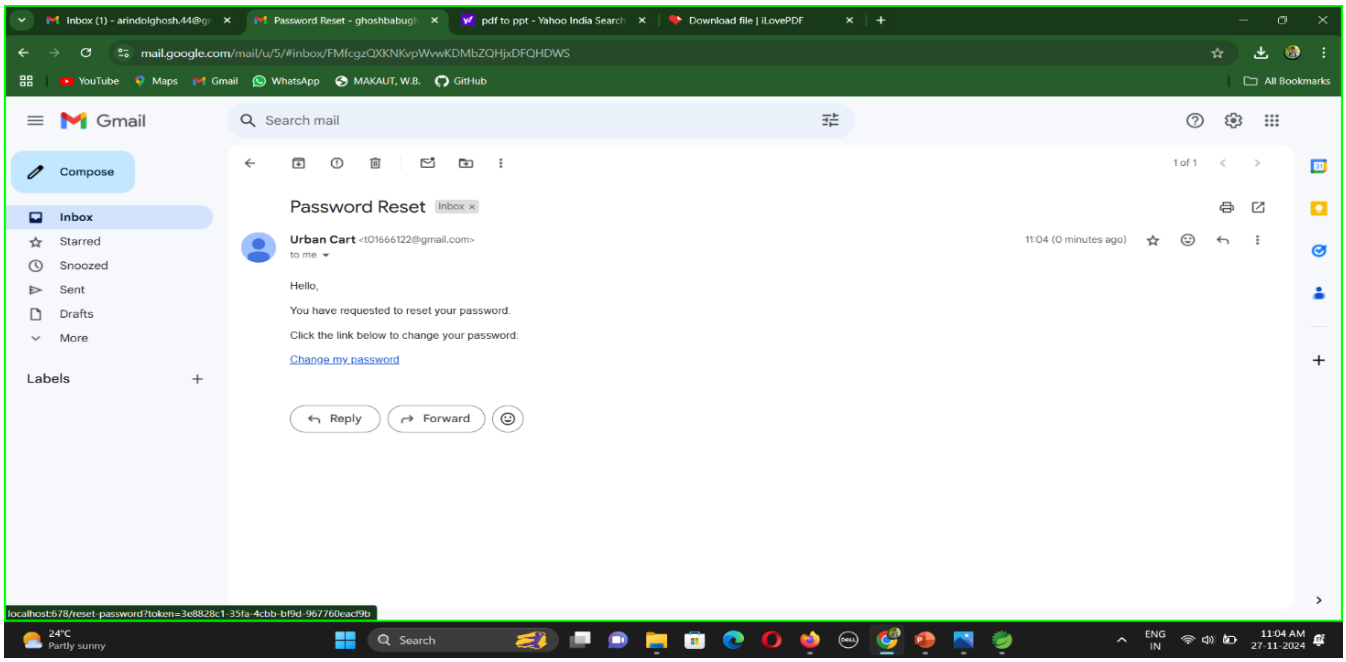
- **Order Status Email Notification**

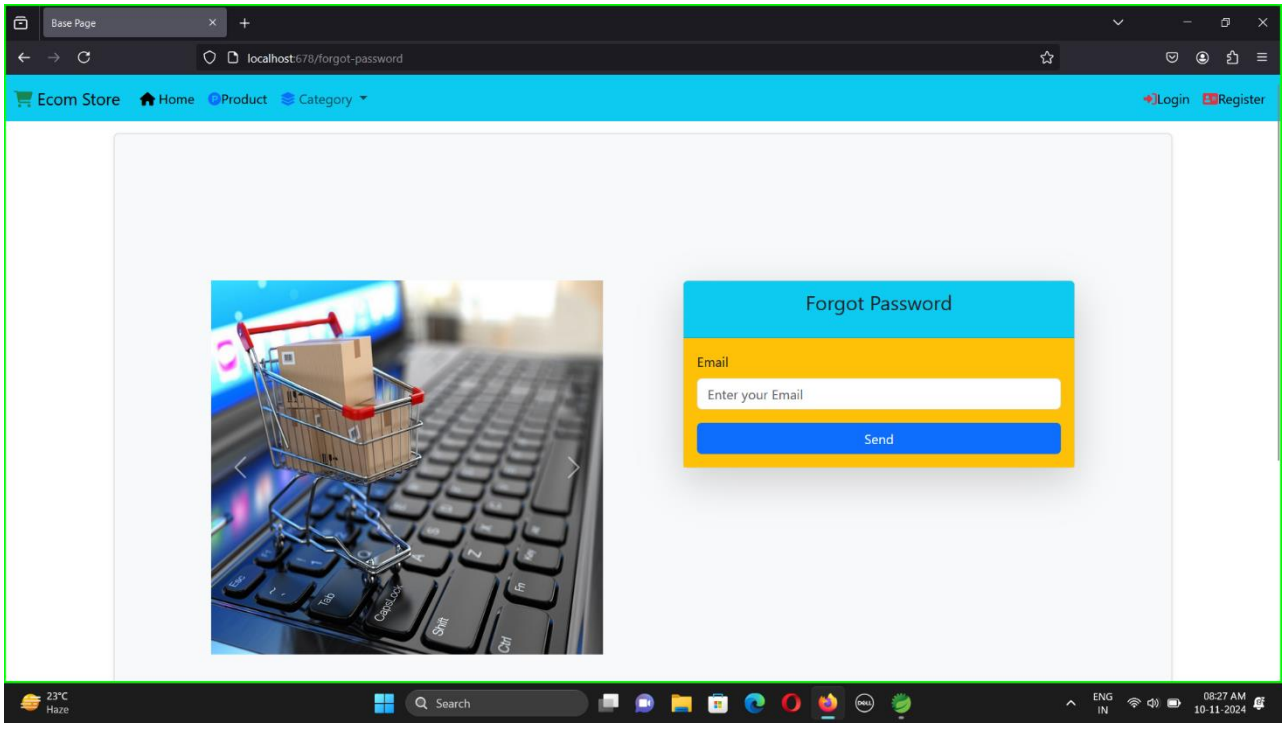
  In Spring Boot, manage order status with service methods updating status fields, and send real-time email notifications using JavaMailSender(smtp) for efficient order tracking and customer communication.



➢ Forgot Password Email Notification

A "Forgot Password" email notification in Spring Boot involves configuring an email service using JavaMailSender. Generate a unique, time-limited token linked to the user account. Use this token in a secure password reset URL. Send the email with the link using a template. Ensure HTTPS and token validation for security.





Chapter:-2 Project Details

A system requirement essay defines essential functional and non-functional specifications, ensuring successful design, development, and shared stakeholder understanding.

2.1 System Requirement:-

- **Software**
- Web Server: Apache Tomcat
- Database: MYSQL
- Frontend: HTML,CSS,JS,THYMELEAF,REACTJS.
- Backend: LOMBOK,SPRING SECURITY,SERVLET JSP,SPRING DATA JPA,SPRING VALIDATION,MYSQL DRIVER,SPRING WEB, SPRING BOOT DEVTOOLS
- **Hardware**
- Processor: i3 or above
- RAM:4GB or above
- Storage:8GB or above

2.2Definition and Theories:

Spring boot:

Spring Boot is an open-source Java framework offered by the Spring ecosystem, specifically designed to simplify the development of web applications and microservices. By providing an opinionated approach, it reduces the complexity of configuration, allowing developers to focus on application logic rather than setup. Key features like auto-configuration, embedded servers (such as Tomcat or Jetty), and an extensive set of pre-built dependencies streamline the development process, making it faster and more efficient.

Spring Boot eliminates much of the boilerplate code traditionally required in Java applications, providing a production-ready setup right out of the box. It supports RESTful APIs, database integration, and security features with minimal effort. Its scalability, combined with robust community support and compatibility with cloud-native architectures, makes Spring Boot a top choice for building modern, high-performance applications.

Reactjs:

React.js is a widely-used JavaScript library designed for building dynamic, interactive, and reusable user interface components. Created and maintained by Facebook, it allows developers to construct complex user interfaces with ease by breaking them into smaller, manageable pieces. React leverages a virtual DOM to optimize rendering and improve performance, making it highly suitable for building efficient single-page applications. Its declarative syntax simplifies UI updates, ensuring a seamless user experience. With a strong ecosystem and community support, React has become a go-to choice for modern front-end web application development.

Spring Data JPA:

Spring Data JPA, based on the Java Persistence API (JPA), simplifies database interactions using Object-Relational Mapping (ORM). ORM maps Java objects directly to database tables, streamlining data persistence. The @Query annotation in Spring Data JPA allows developers to define custom queries, offering flexibility and precision when executing database operations beyond standard methods.

- **Spring Validation:**

  Spring's Validator interface is used for validating objects, reporting validation failures to an Errors object. Validators implement custom logic to check data integrity and capture issues in the Errors object for processing. This approach is flexible and allows detailed error handling. For instance, a simple Java data object can be validated by implementing and configuring this interface effectively.

- **Apache Tomcat:**

  Tomcat is a reliable web server and servlet container known for efficiency in deploying Java applications. Spring Boot, on the other hand, is a powerful framework for creating standalone applications, streamlining setup, and development. Together, they offer a robust solution for building and running scalable, production-ready Java applications with minimal configuration and enhanced performance.

- **Spring Security**

  Spring Security is a robust, customizable framework for authentication and access control in Java applications. It is the de-facto standard for securing Spring-based projects, offering features for both authentication and authorization. With flexible configurations, it enables developers to protect applications against security threats while integrating seamlessly into the Spring ecosystem.

- **MYSQL:**

  MySQL is an open-source relational database management system widely used for managing and storing data. Its name combines "My," after co-founder Michael Widenius's daughter, and "SQL," which stands for Structured Query Language. MySQL is known for its speed, reliability, and ease of use, making it a popular choice for web applications and data-driven projects.

- **THYMELEAF:**

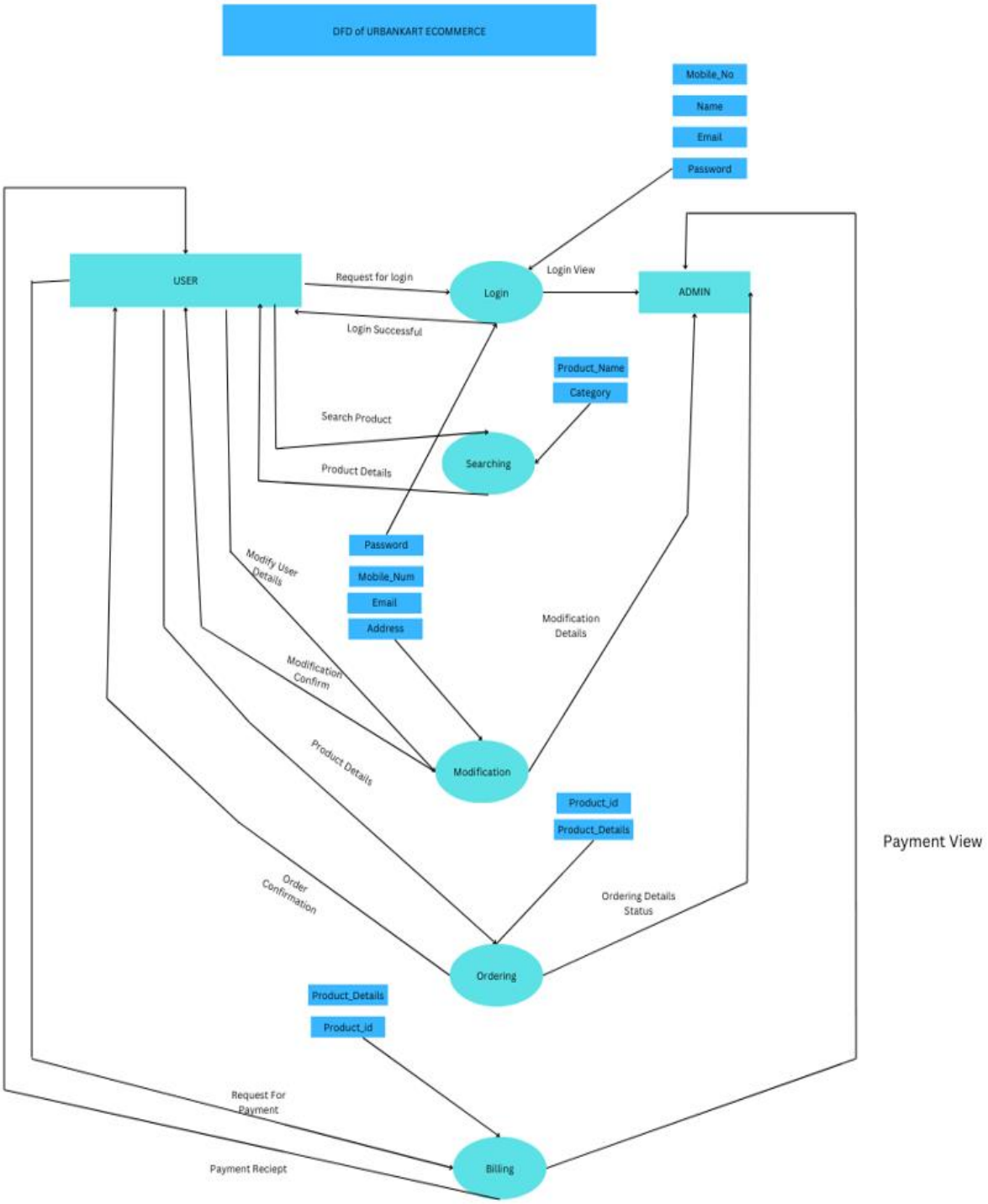
  Thymeleaf is a Java-based templating library used to build dynamic web applications. It supports XHTML/HTML5 rendering, making it ideal for creating modern, server-side web applications. Thymeleaf integrates seamlessly with Spring, offering a natural templating syntax that enhances the development experience. It allows developers to easily create flexible, maintainable, and efficient web interfaces.

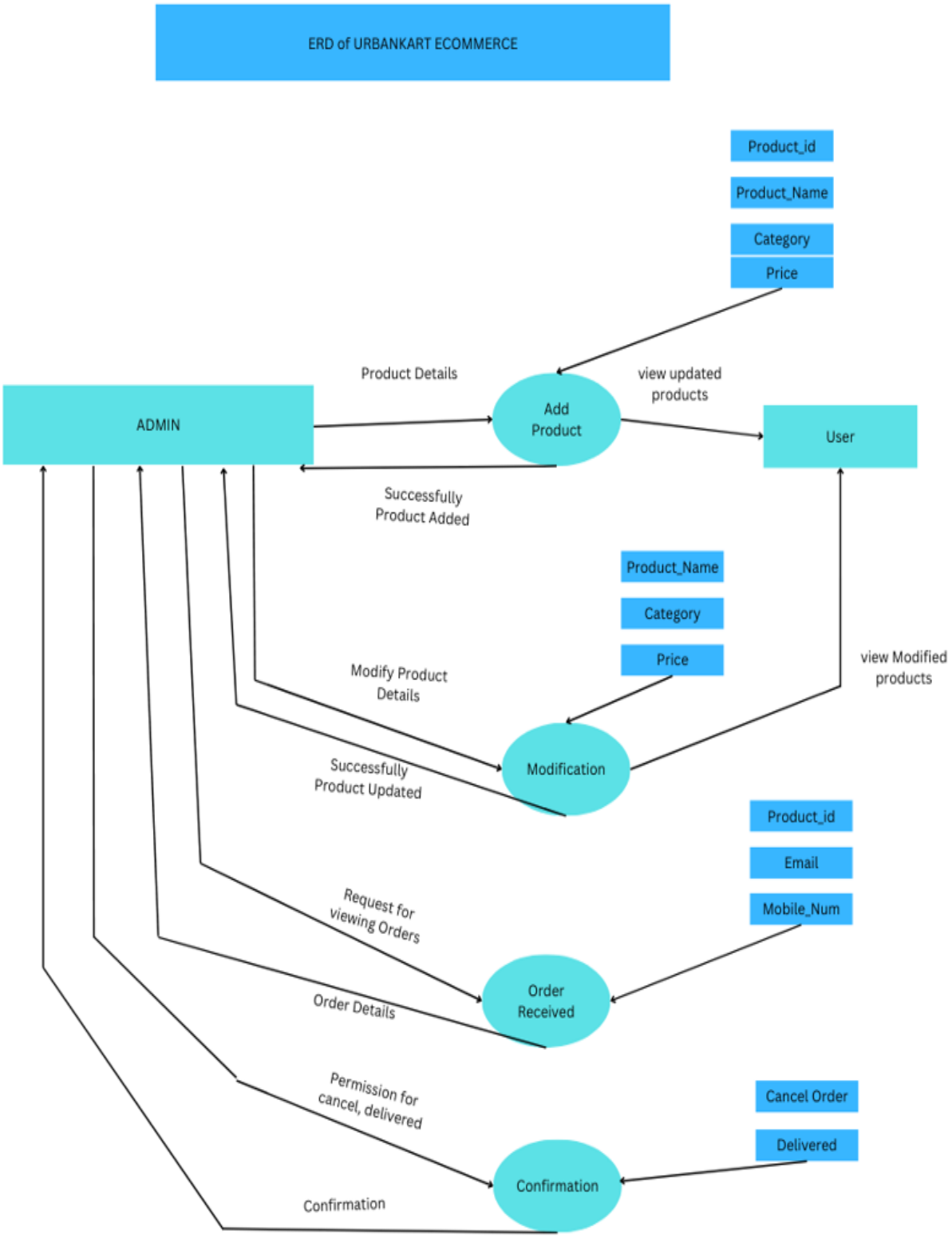
- **Spring Boot Devtools:**

Spring Boot DevTools provides automatic restart functionality, allowing developers to quickly see changes made to their application without needing to manually restart the server. Any changes to class files trigger a restart of the application, speeding up the development process.

2.3 DFD(DATA FLOW DIAGRAM):-

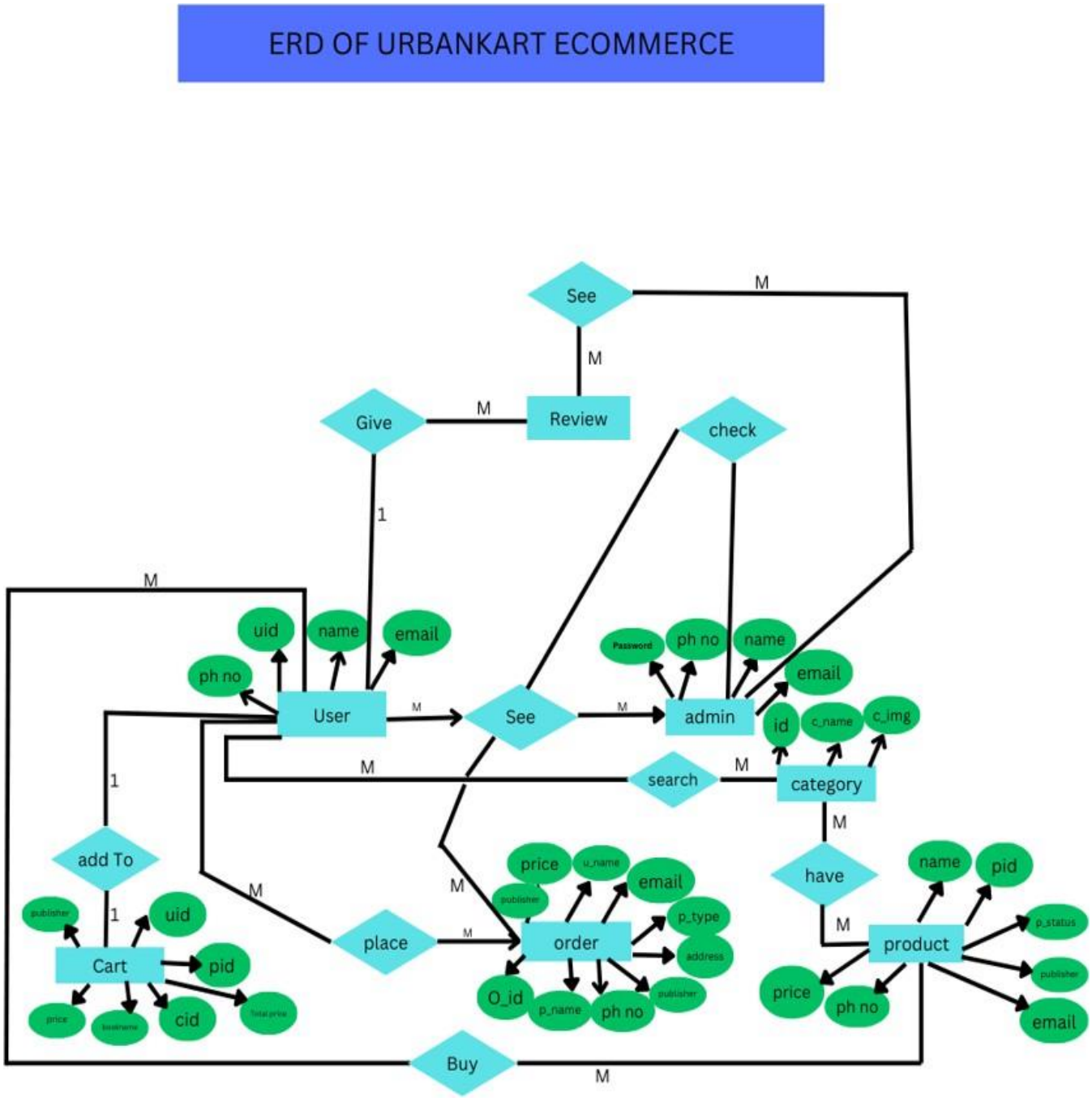
A data flow diagram (DFD) is a visual representation of how data moves through a system or process. It uses symbols like circles, rectangles, and arrows, along with text labels, to show data inputs, outputs, storage points, and the routes between them. DFDs are often used in software engineering to design software architecture and foundations.




ERD of URBANKART ECOMMERCE
Product_id
Product_Name
Category
Price
Product Details
ADMIN
Add Product
view updated products
User
Successfully Product Added
Product_Name
Category
Price
Modify Product Details
view Modified products
Modification
Successfully Product Updated
Product_id
Email
Mobile_Num
Request for viewing Orders
Order Received
Order Details
Permission for cancel, delivered
Cancel Order
Delivered
Confirmation
Confirmation

2.4 ERD(ENTITY RELATIONSHIP DIAGRAM):-

Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.



Chapter:-3 Future Work , Conclusion , Reference

Future Work

A product filter, real-time transaction processing, and feedback-based approach are essential for enhancing user experience and operational efficiency in e-commerce. Product filters allow users to refine search results, improving product discovery. Real-time transactions ensure swift, accurate processing of purchases, boosting customer satisfaction. A feedback-based approach enables continuous improvement by gathering user insights, helping to tailor products and services to customer needs, ultimately driving business growth and customer loyalty.

Conclusion

The e-commerce project was a success, showcasing the effective use of a full-stack technology stack. By combining Spring Boot with React, Servlet JSP, HTML, and CSS, we were able to build a robust, scalable, and user-friendly e-commerce platform. The project demonstrated the importance of integrating modern frontend frameworks with a strong backend, resulting in a well-rounded and efficient application. This project serves as a solid foundation for future enhancements, such as implementing advanced features like AI-based recommendations, real-time notifications, and a more sophisticated order management system.

Reference

1. Spring Boot in Action by Craig Walls (3 January 2016)
ISBN-13 : 978-1617292545

2. Spring Boot: Up and Running by Mark Heckler (28 February 2021)
ISBN-13 : 978-1492076988

3. Spring Microservices in Action by John Carnell (6 July 2017)
ISBN-13 : 978-1617293986

4. Pro Spring Boot 2 (https://start.spring.io/)
5. Spring Boot Cookbook by Alex Antonov (28 September 2015)
ISBN-13 : 9781785289118
6. GFG (https://www.geeksforgeeks.org/what-is-dfddata-flow-diagram/?ref=gcse_ind)
7. JavaTpoint (https://www.javatpoint.com/spring-boot-tutorial)