INSTITUTE FOR ADVANCED COMPUTING AND SOFTWARE DEVELOPMENT, AKURDI, PUNE

"E-Medicine Hub"

PG-DAC August 2024

Submitted By:

Group No: 29

Roll No.	**Name of Student**
248010	Amruta Thore
248042	Komal Sahu

Mr. Harshal Waghchaure
Project Guide

Mr. Rohit Puranik
Centre Coordinator

ABSTRACT

E-Medicine Hub is a comprehensive platform designed to integrate online e-medicine management with e-commerce functionalities. It enables pharmacies and healthcare providers to manage efficiently manage their medicine stock while offering customers a seamless and secure online purchasing experience. The platform leverages **React and Redux** for dynamic and responsive frontend, while **Spring Boot** powers the backend with a robust API. Spring Security ensures secure authentication and authorization, while payments are securely processed through **Razorpay**.

E-Medicine is a scalable and efficient application aimed at revolutionizing the pharmaceutical industry by simplifying inventory management and online medicine sales.It addresses challenges faced by pharmacies and medical suppliers in tracking stock, processing, orders, and ensuring compliance with healthcare regulations.By integrating **React for the frontend** and **Spring Boot for the backend**, the platform provides a seamless user experience and optimized operational efficiency.

ACKNOWLEDGEMENT

I take this occasion to thank God, almighty for blessing us with his grace and taking our endeavor to a successful culmination. I extend my heartfelt thanks to our esteemed guide, **Mr. Harshal Waghchaure** for providing me with the right guidance and advice at the crucial juncture and showing me the right way. I sincerely thank our respected Centre Co-Ordinator, Mr. Rohit Puranik, for allowing us to use the available facilities. I would also like to thank the other faculty members at this occasion. Last but not least, I would like to thank my friends and family for the support and encouragement they have given me during our work.

Komal Sahu (240841220082)
Amruta Thore (240841220018)

Table of Contents

1. INTRODUCTION

E-Medicine Hub: A Comprehensive Inventory Management and E-Commerce Solution. It is a comprehensive platform designed to integrate inventory management with e-commerce functionalities. It allows businesses to efficiently track and manage their inventory while providing customers with a seamless online shopping experience of medicine. The platform uses technologies like React and Redux for a responsive front-end, and Spring Boot for a robust backend API. Security is enhanced through Spring Security for authentication and authorization, and payments are securely processed via Razor-pay.

E-Medicine Hub is tailored to meet the needs of retail and wholesale businesses, ensuring real-time inventory updates, automated stock adjustments, and a user-friendly interface for customers. This integration of inventory management with e-commerce helps businesses reduce manual errors, improve operational efficiency, and provide a superior customer experience.

In the **E-Medicine Hub** system, both **User** and **Admin** roles play crucial roles in the management and operation of the platform. Each role comes with specific responsibilities, permissions, and access levels to ensure that the system operates smoothly and securely.

1. Admin Role

The **Admin role** has full control over the platform's functionalities. Admins are typically **business owners, managers, or IT personnel** who require comprehensive access to manage inventory, users, and overall system settings.

❖ **Responsibilities:**

- **User Management:**
  - Create, manage, and delete user accounts.
  - Assign roles and permissions to users.
  - Monitor user activities and ensure compliance with company policies.

- **Inventory Management:**
  - Add, update, or delete medicine information.
  - Monitor stock levels and ensure real-time updates.
  - Set reorder levels to automate stock replenishment.

- **Order Management:**
  - View, process, and manage customer orders.
  - Oversee order fulfillment, including shipping and delivery processes.
  - Handle returns and exchanges.

- **Payment Management:**
  - Oversee and manage payments processed through Razorpay.
  - Resolve payment disputes and refund requests.

- **System Configuration:**
  - Configure system settings, including tax rates, discount rules, and shipping options.
  - Manage and update platform content, such as banners, promotions, and announcements.

- **Reporting & Analytics:**
  - Generate and analyze reports on sales, inventory levels, user activities, and financial transactions.
  - Use data to make informed business decisions.

- **Security Management:**

- Implement and enforce security policies using Spring Security.
- Monitor security logs and respond to potential security breaches.

Access Level:

- Full access to all system features and data.
- Ability to override settings and actions performed by users.
- Access to sensitive financial and personal information of users and customers.

2. User Role

Users in the E-Medicine Hub typically refer to customers or employees with limited access. Their access level is more restricted compared to Admins, focusing on tasks related to inventory tracking, order placement, or basic system interactions.

Responsibilities:

- **Inventory Access (for employees):**
  - View product details, stock levels, and inventory status.
  - Update inventory data (with limited permissions).

- **Order Placement (for customers):**
  - Browse medicines, add items to the cart, and place orders.
  - Track order status and view order history.

- **Profile Management:**
  - Update personal information such as name, address, and contact details.
  - Manage account settings, including password changes.

- **Payments:**
  - Make payments for orders via the integrated Razorpay gateway.
  - View payment history and download receipts.

Access Level:

- Limited access based on assigned permissions.

  Can view and interact with specific features (e.g., order placement, product browsing) but cannot alter system settings.

  No access to sensitive system configurations, financial data, or user management functionalities.

Role-Based Access Control (RBAC)

The system implements Role-Based Access Control (RBAC) using **Spring Security**. This ensures that users only have access to the functionalities they need based on their roles. For instance:

- Admins have unrestricted access to all resources and can manage both the system and its users.
- Users have limited access, often restricted to viewing and interacting with med, placing orders, and managing their profiles.

Security Considerations

Both roles are secured through robust authentication mechanisms provided by Spring Security. This includes:

- **Authentication:** Verifying the identity of users before granting access to the system.
- **Authorization:** Ensuring that users can only access the resources permitted by their role.
- **Session Management:** Secure handling of user sessions to prevent unauthorized access.

The Role-based structure of E-Medicine Hub ensures that tasks are properly delegated, maintaining system security and efficiency. Admins handle high-level management and decision-making, while users interact with the platform to fulfill their specific needs without compromising security..

1.1 Purpose

The purpose of the E-Medicine Hub is to provide a comprehensive and efficient platform for managing and selling medicine in a retail or wholesale environment. The system is designed to streamline various business operations related to inventory management, order processing, and customer interactions, ultimately enhancing the overall productivity and profitability of the business.

1.2 Scope

The **scope of the E-Medicine Hub Project** encompasses the design, development, deployment, and maintenance of a comprehensive inventory management and sales platform that integrates modern technologies to meet the needs of retail and wholesale businesses. This scope is defined by the functionalities that the system will provide, the technologies used, and the stakeholders involved.

1.3 Objective of Inventory Sell-Mart

The objectives of the E-Medicine Hub project outline the primary goals that the system aims to achieve, providing clear guidance on its design, implementation, and operational focus. These objectives ensure that the system effectively supports the needs of its users while driving business efficiency and growth.

1. Efficient Medicine Management
2. Streamlined Sales Processes
3. Enhanced User Experience
4. Robust Security and Access Control
5. Scalable and Flexible Architecture
6. Comprehensive Reporting and Analytics
7. Cost Optimization
8. Enhanced Customer Experience
9. Compliance and Audit Readiness

10. Support for Growth and Expansion

2. SOFTWARE REQUIREMENT SPECIFICATION

The functional requirements for E-Medicine Hub outline the specific features and capabilities that the system must provide to meet the needs of its users. These requirements are essential for guiding the development process and ensuring that the final product aligns with the business objectives.

2.1 Functional Requirements for E-Medicine Hub

1. **User Management**
   - **User Registration:**
     The system shall allow new users to create an account by providing personal details, such as name, email, and password.
   - **User Authentication:**
     The system shall authenticate users during login using their registered email and password.
   - **Role-Based Access Control:**
     The system shall support role-based access, where different users (Admin, Employee, Customer) have different permissions.
   - **Profile Management:**
     Users shall be able to view and update their profiles, including personal details and passwords.
2. **Medicine Management**
   - **Product Management:**
     The system shall allow admins to add, update, and delete products from the medicine.
     The system shall store product details such as name, description, price, category, and images.
   - **Stock Level Monitoring:**
     The system shall monitor stock levels in real time and display current medicine stock.

- **Medicine Adjustment:**

  The system shall allow manual adjustments to medicine levels to account for discrepancies or damage.

3. Sales Management

- **Order Placement:**

  Customers shall be able to place orders for products from the E-Medicine Hub.

- **Order Processing:**

  The system shall process orders, updating medicines levels accordingly.

- **Payment Processing:**

  The system shall integrate with Razorpay to handle online payments securely.

  The system shall support various payment methods, including credit/debit cards, UPI, and wallets.

4. Customer Management

- **Customer Profiles:**

  The system shall maintain profiles for each customer, including order history and personal details.

- **Order History:**

  Customers shall be able to view their order history.

- **Loyalty Programs:**

  The system shall support the creation and management of loyalty programs for customers.

5. Security

- **Data Encryption:**

  The system shall encrypt sensitive data, such as payment information and personal details.

- **Authentication and Authorization:**

  The system shall enforce strong authentication mechanisms and ensure that users have access only to the functionalities allowed by their roles.

2.2 Non-Functional Requirements for E-Medicine Hub

1. Performance

- Response Time:

  The system shall respond to user actions within 2 seconds under normal operating conditions.

- Scalability:

  The system shall handle an increasing number of users and transactions without performance degradation. It should support at least 10,000 concurrent users.

- Throughput:

  The system shall process at least 100 transactions per second during peak usage times.

2. Reliability

- Availability:

  The system shall have an uptime of 99.9% over a 12-month period, ensuring high availability for users.

- Fault Tolerance:

  The system shall continue to operate in the event of hardware or software failures, with minimal disruption to users.

- Error Handling:

  The system shall gracefully handle errors and provide meaningful feedback to users when issues occur.

3. Usability

- User Interface:

  The system shall have a user-friendly interface that is easy to navigate, with clear instructions and minimal learning curve.

4. Maintainability

- Modularity:

  The system shall be designed in a modular fashion, allowing for easy updates and enhancements to individual components without affecting the entire system.

- Code Quality:

  The system shall follow coding best practices, with well-documented, clean, and

maintainable code.

- Testing:

  The system shall undergo thorough testing, including unit tests, integration tests, and user acceptance tests, to ensure quality and stability.

- **Other Requirements:**

Hardware and Network Interfaces:

Back-end Server Configuration:

- Intel(R) core(TM) i3-1005G1 CPU Processor
- 12 GB RAM

Front-end Client Configuration:

- Intel(R) core(TM) i3-1005G1 CPU Processor
- 128 MB SDRAM
- 10 GB Hard Disk Drive
- 104 Keys Keyboard
- PS2 Mouse with pad

Software Interfaces:

Software configuration for back-end Services:

- Java EE (version -jdk17)
- Spring Boot (version 3.4.1), JPA, Razor Pay, Spring Authentication
- MySQL 8.0.37
- STS 4.26.0

Software configuration for front-end Services:

- ReactJS 20, Redux 5.0.1
- HTML (version 5), CSS (version 3), JS (ES15)
- Bootstrap 5.3.3
- VS Code 1.97.0

3. DIAGRAMS

3.1 Entity Relationship Diagram:

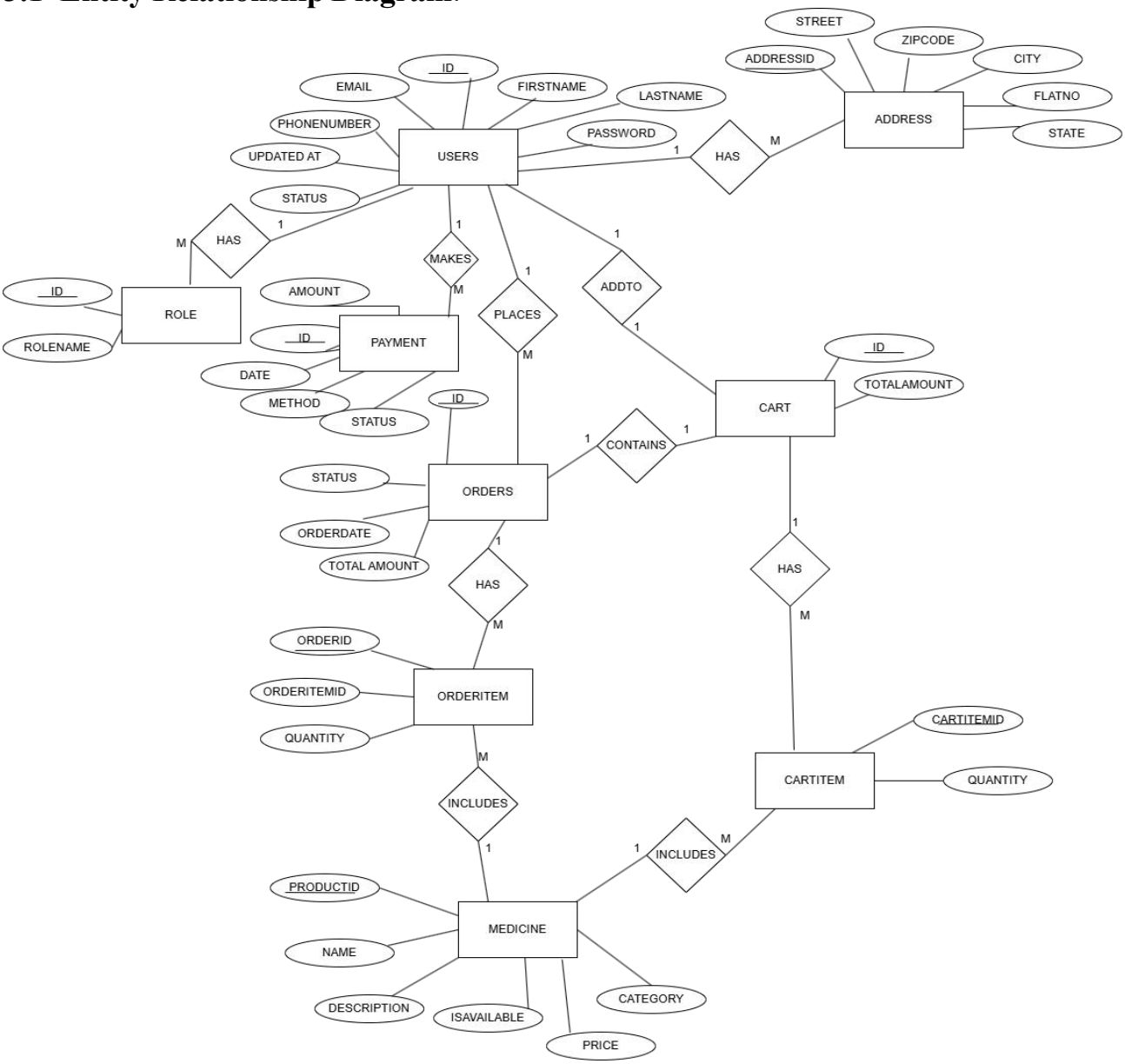


Fig. ER Diagram for E-Mdicine Hub

3.2 Use Case Diagram:

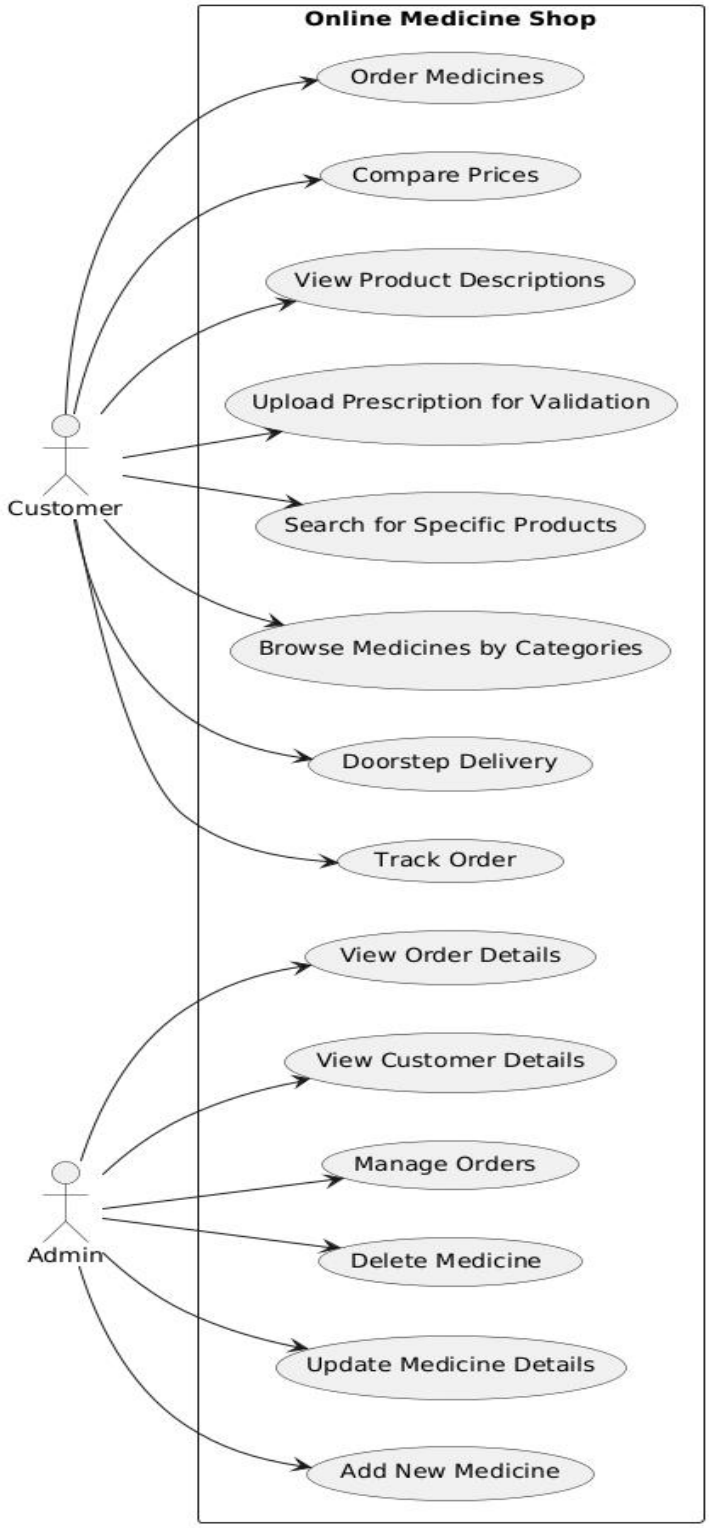
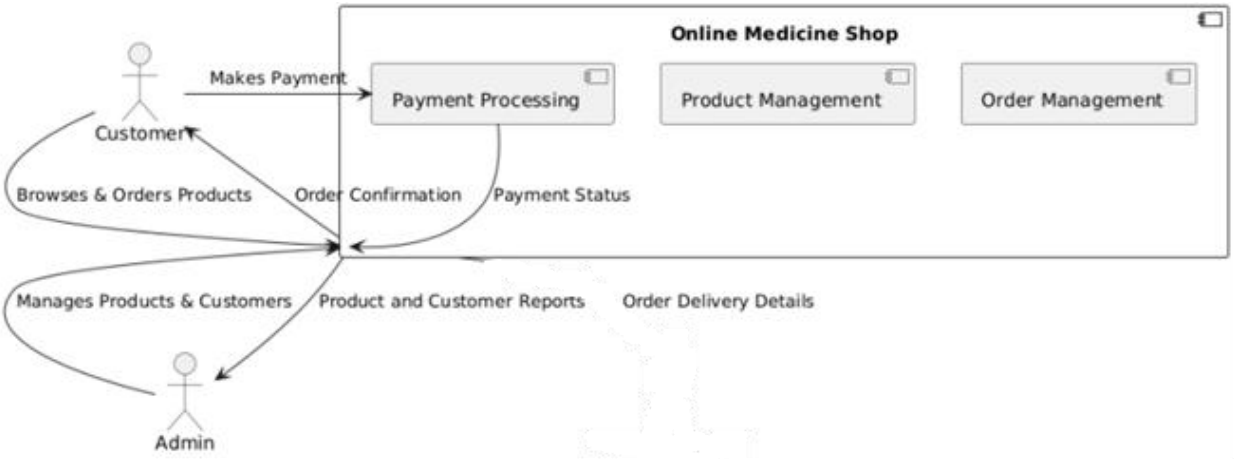


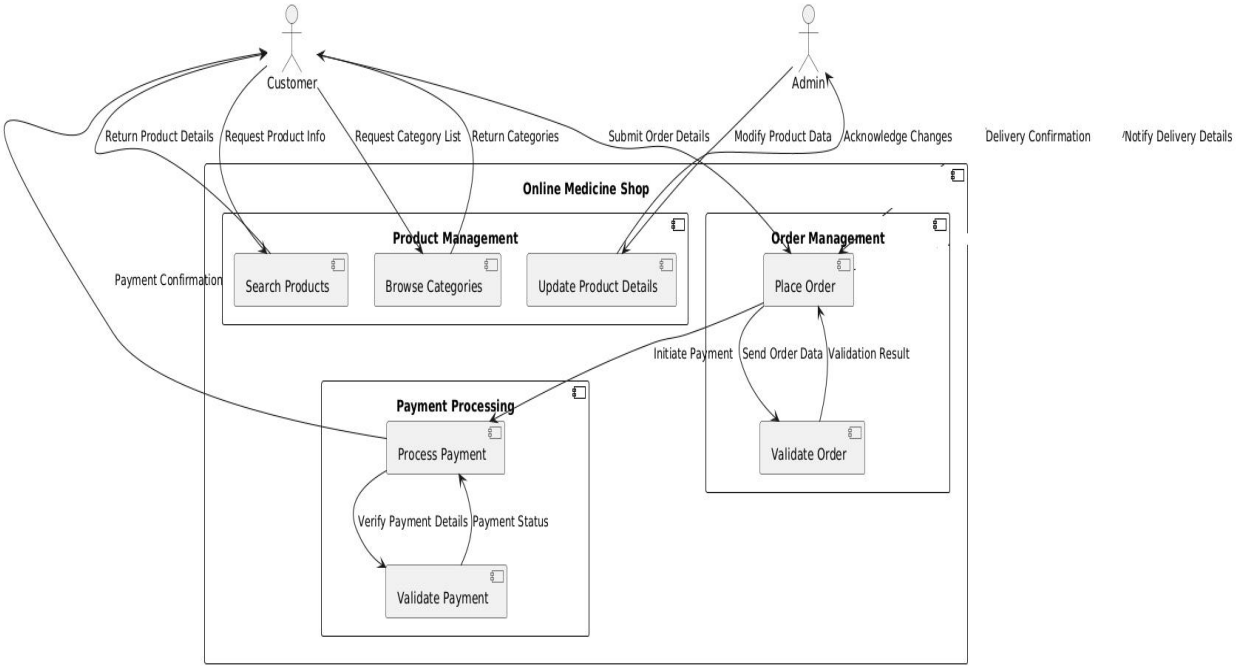
Fig. Use Case Diagram for E-Medicine Hub

3.3 Data Flow Diagram:

DFD Level 0:

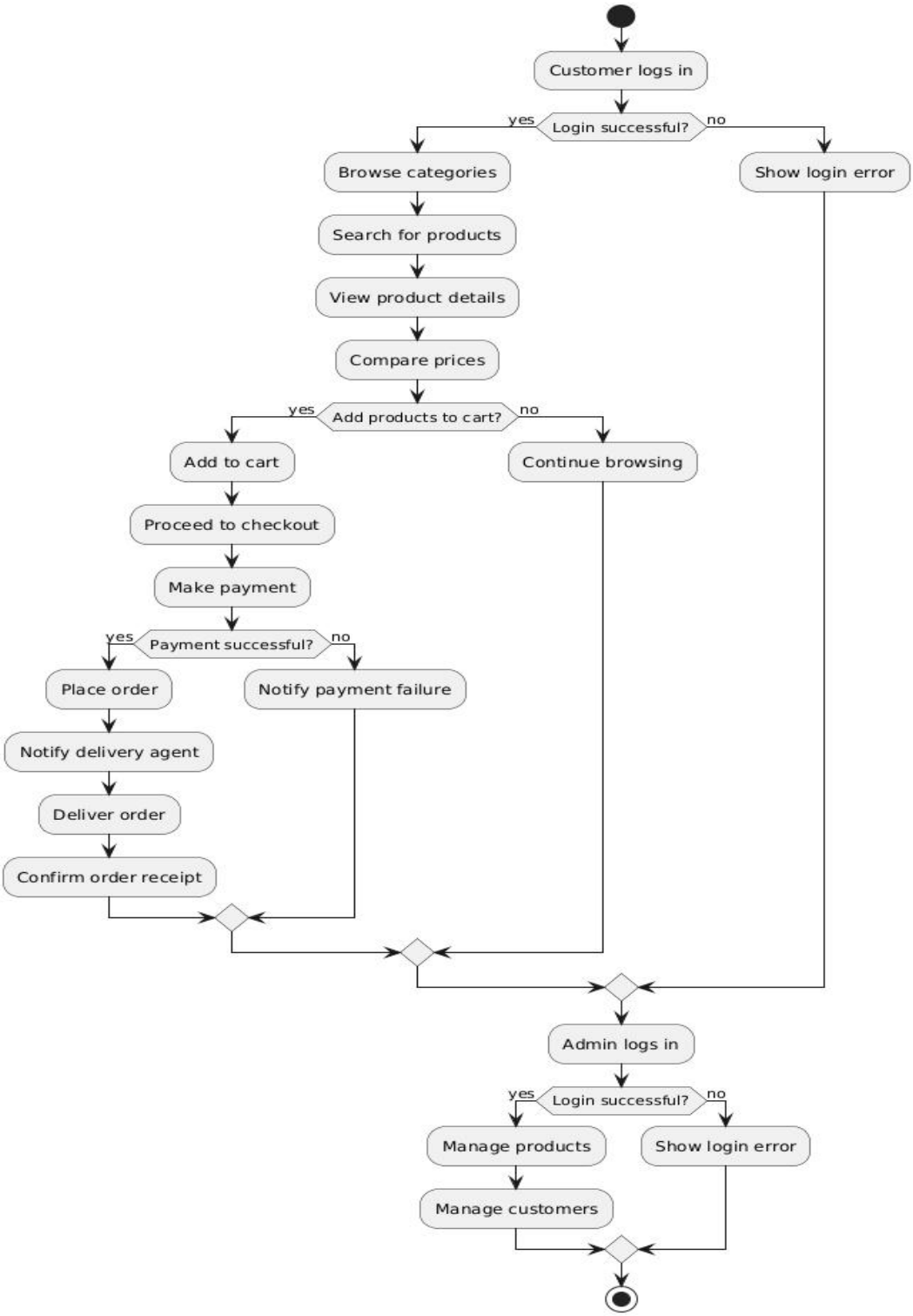


DFD level 1 :

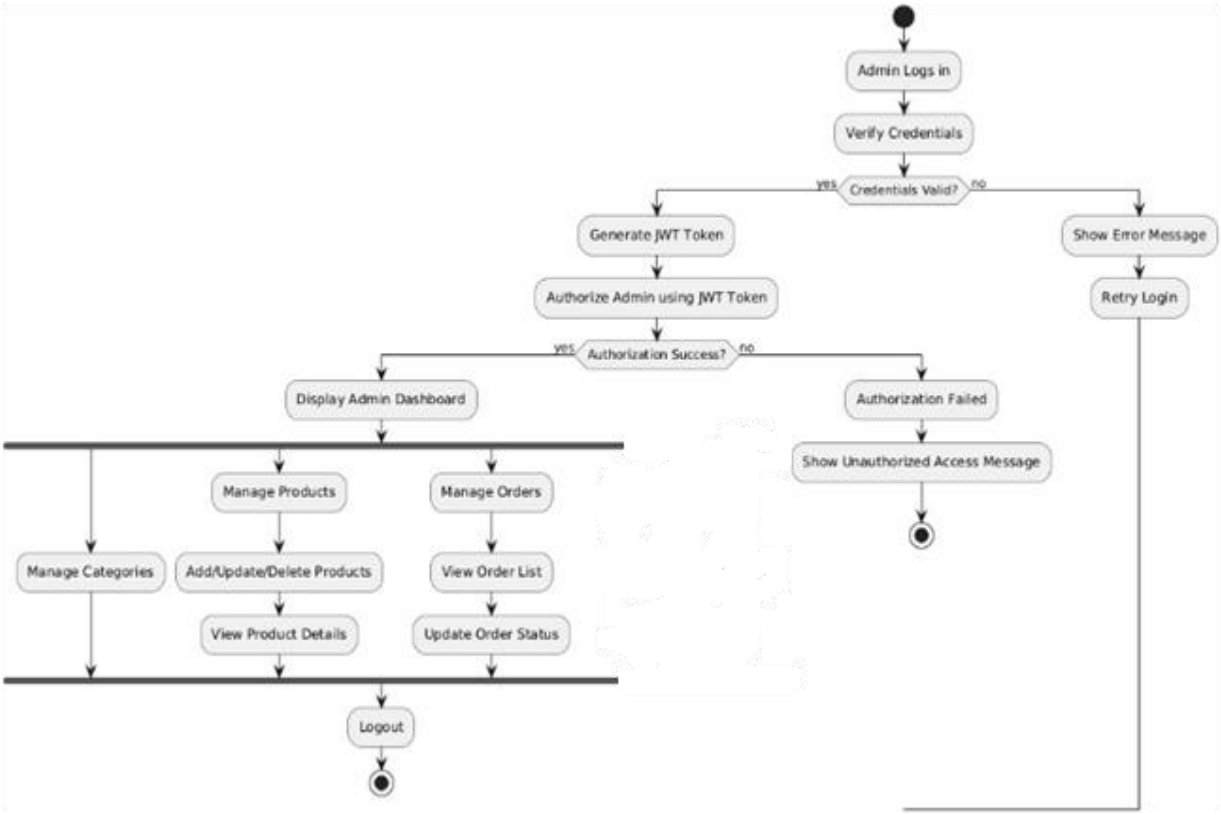


3.4 Activity Diagram :

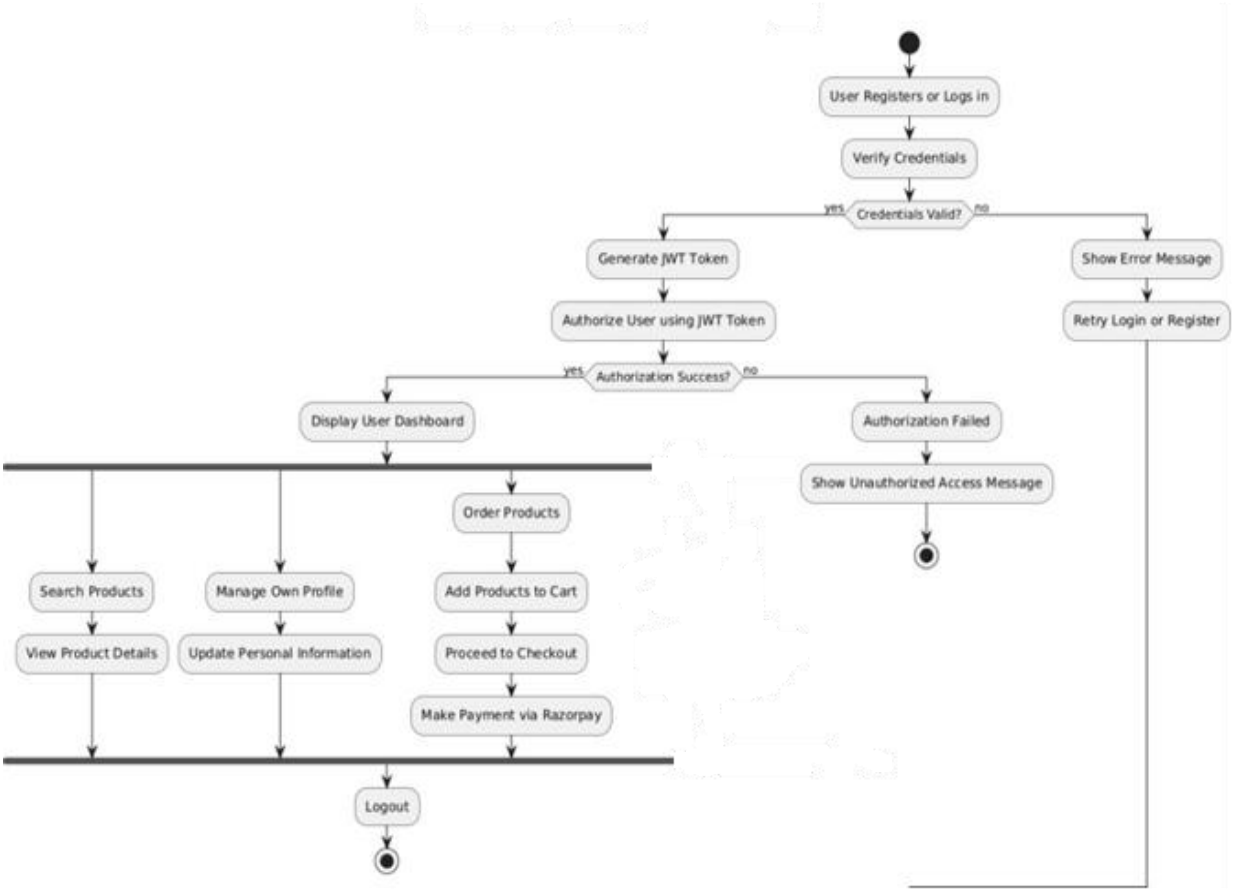
1. Login Activity Diagram



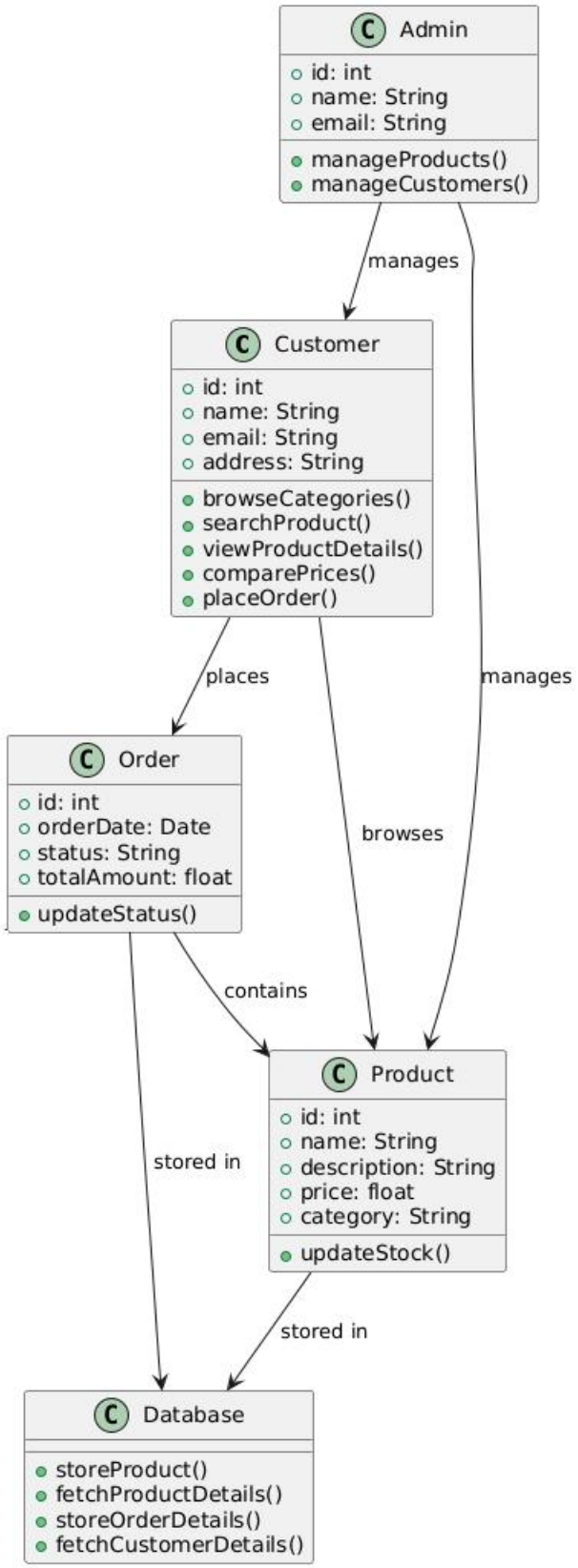
2. Admin Activity Diagram:



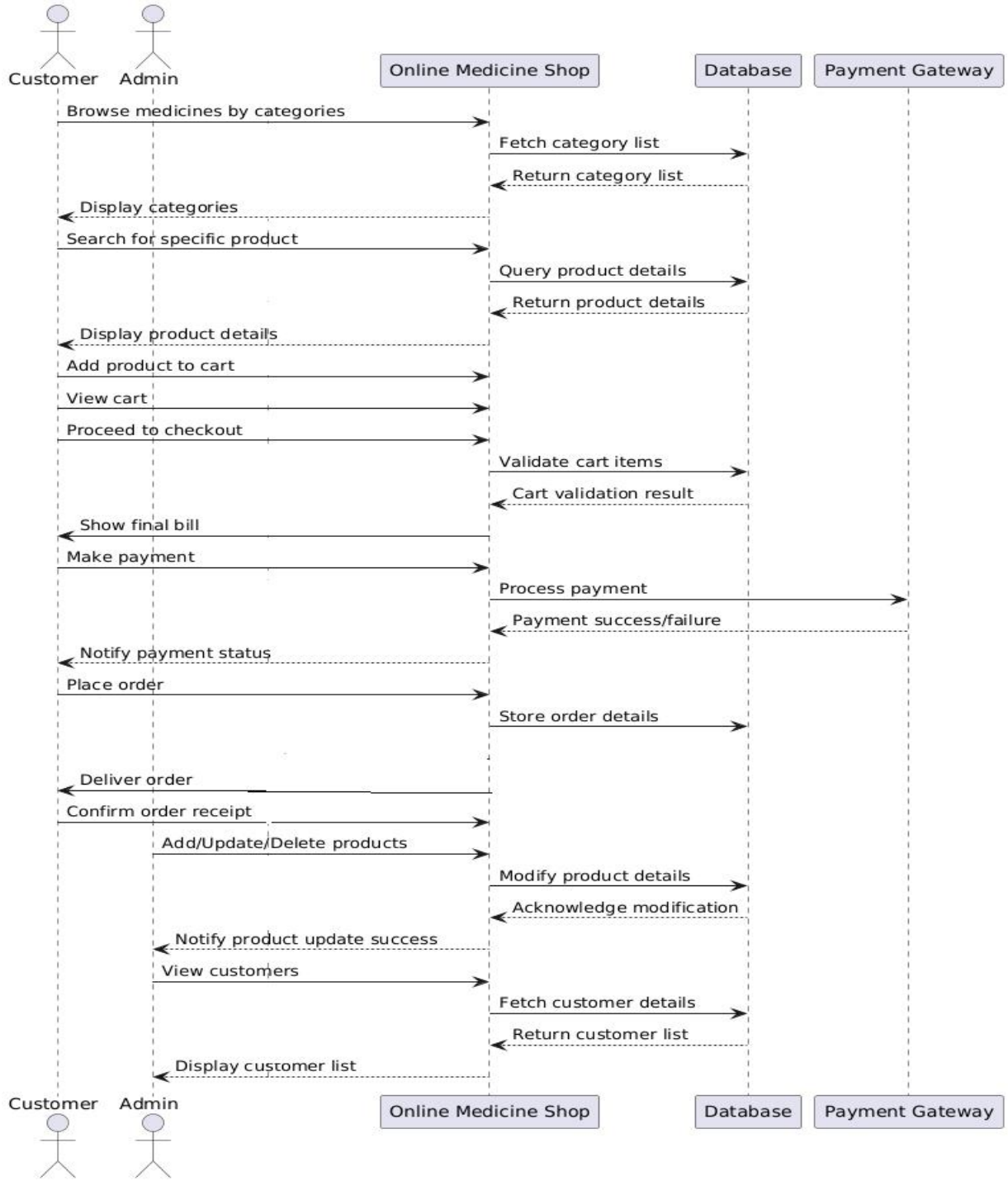
3. User Activity Diagram



3.5 Class Diagram:

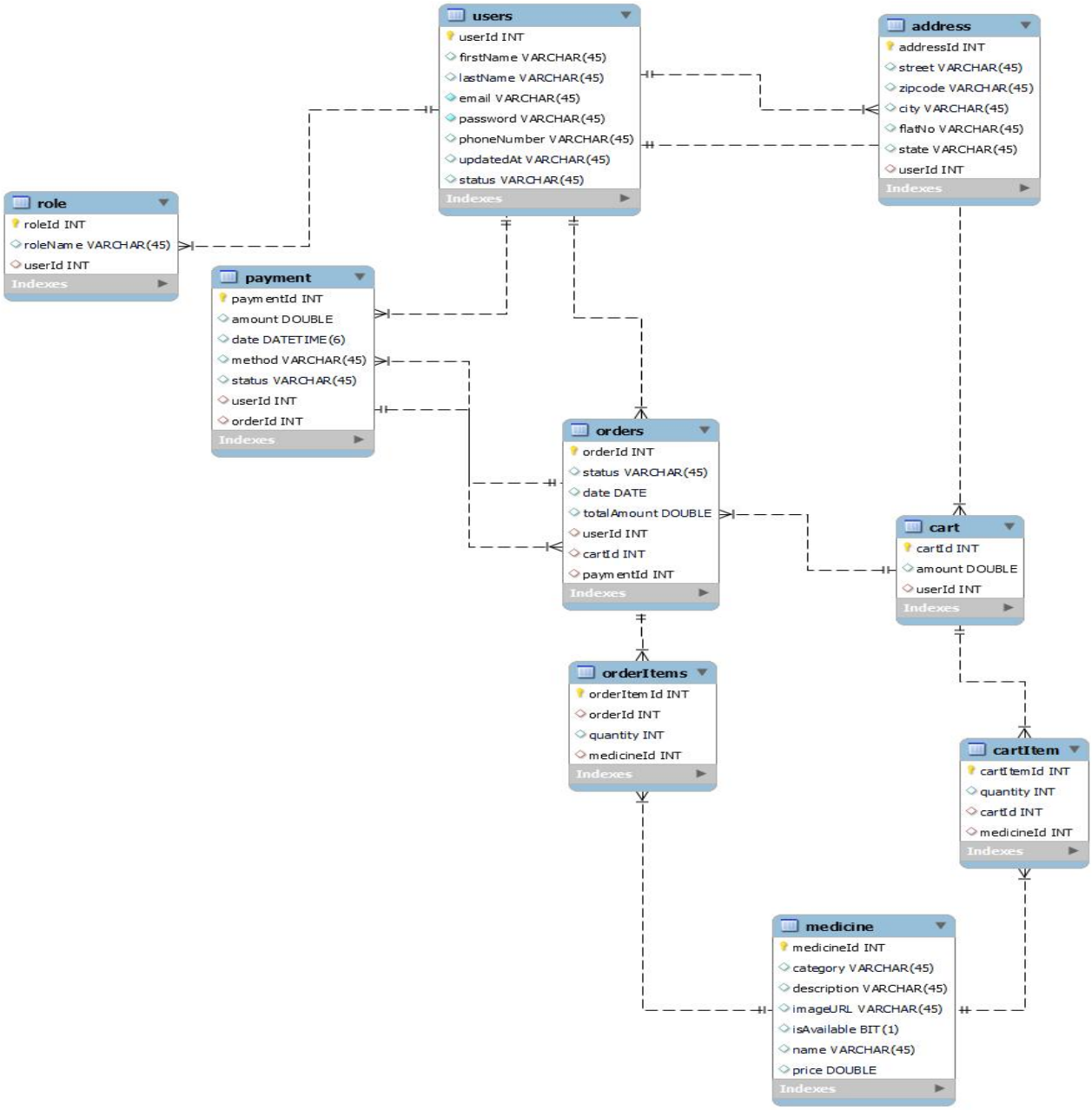


3.6 Sequence Diagram



4. DATABASE DESIGN

4.1 Design:



4.2 Tables:

The following table structures depict the database design.

```
mysql> desc users;
+---------------------+-------------------------------+------+-----+---------+-------+
| Field               | Type                          | Null | Key | Default | Extra |
+---------------------+-------------------------------+------+-----+---------+-------+
| id                  | int                           | NO   | PRI | NULL    |       |
| created_at          | datetime(6)                   | YES  |     | NULL    |       |
| email               | varchar(100)                  | NO   | UNI | NULL    |       |
| first_name          | varchar(255)                  | NO   |     | NULL    |       |
| last_name           | varchar(255)                  | YES  |     | NULL    |       |
| password            | varchar(255)                  | NO   |     | NULL    |       |
| phone_number        | varchar(255)                  | YES  |     | NULL    |       |
| updated_at          | datetime(6)                   | YES  |     | NULL    |       |
| user_account_status | enum('ACTIVE','DEACTIVATE')   | YES  |     | NULL    |       |
+---------------------+-------------------------------+------+-----+---------+-------+
9 rows in set (0.05 sec)
```

Table 1: user

```
mysql> desc orders;
+--------------+-----------------------------------------------------+------+-----+---------+----------------+
| Field        | Type                                                | Null | Key | Default | Extra          |
+--------------+-----------------------------------------------------+------+-----+---------+----------------+
| order_id     | int                                                 | NO   | PRI | NULL    | auto_increment |
| order_date   | datetime(6)                                         | YES  |     | NULL    |                |
| status       | enum('CANCELLED','DELIVERED','PENDING','SHIPPED')   | YES  |     | NULL    |                |
| total_amount | double                                              | NO   |     | NULL    |                |
| cart_id      | int                                                 | YES  | MUL | NULL    |                |
| payment_id   | bigint                                              | YES  | UNI | NULL    |                |
| user_id      | int                                                 | YES  | MUL | NULL    |                |
+--------------+-----------------------------------------------------+------+-----+---------+----------------+
7 rows in set (0.00 sec)
```

Table 2: orders

```
mysql> desc medicine;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| product_id   | int          | NO   | PRI | NULL    | auto_increment |
| category     | varchar(255) | YES  |     | NULL    |                |
| description  | varchar(255) | YES  |     | NULL    |                |
| image_urls   | varchar(500) | YES  |     | NULL    |                |
| is_available | bit(1)       | NO   |     | NULL    |                |
| name         | varchar(255) | YES  |     | NULL    |                |
| price        | double       | NO   |     | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
7 rows in set (0.00 sec)
```

Table 3: Medicine

```
mysql> desc user_roles;
+---------+--------+------+-----+---------+-------+
| Field   | Type   | Null | Key | Default | Extra |
+---------+--------+------+-----+---------+-------+
| user_id | int    | NO   | PRI | NULL    |       |
| role_id | bigint | NO   | PRI | NULL    |       |
+---------+--------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

Table 4: UserRoles

```
mysql> desc payement;
+----------------+-----------------------------------------+------+-----+---------+----------------+
| Field          | Type                                    | Null | Key | Default | Extra          |
+----------------+-----------------------------------------+------+-----+---------+----------------+
| payment_id     | bigint                                  | NO   | PRI | NULL    | auto_increment |
| payment_amount | double                                  | YES  |     | NULL    |                |
| payment_date   | datetime(6)                             | YES  |     | NULL    |                |
| payment_method | enum('CASH','CREDIT_CARD','UPI')        | YES  |     | NULL    |                |
| payment_status | enum('FAILED','PENDING','SUCCESSFUL')   | YES  |     | NULL    |                |
| order_id       | int                                     | YES  | UNI | NULL    |                |
| user_id        | int                                     | NO   | MUL | NULL    |                |
+----------------+-----------------------------------------+------+-----+---------+----------------+
7 rows in set (0.00 sec)
```

Table 5 : payment_information

```
mysql> desc cart;
+--------------+--------+------+-----+---------+----------------+
| Field        | Type   | Null | Key | Default | Extra          |
+--------------+--------+------+-----+---------+----------------+
| cart_id      | int    | NO   | PRI | NULL    | auto_increment |
| total_amount | double | YES  |     | NULL    |                |
| user_id      | int    | YES  | UNI | NULL    |                |
+--------------+--------+------+-----+---------+----------------+
3 rows in set (0.00 sec)
```

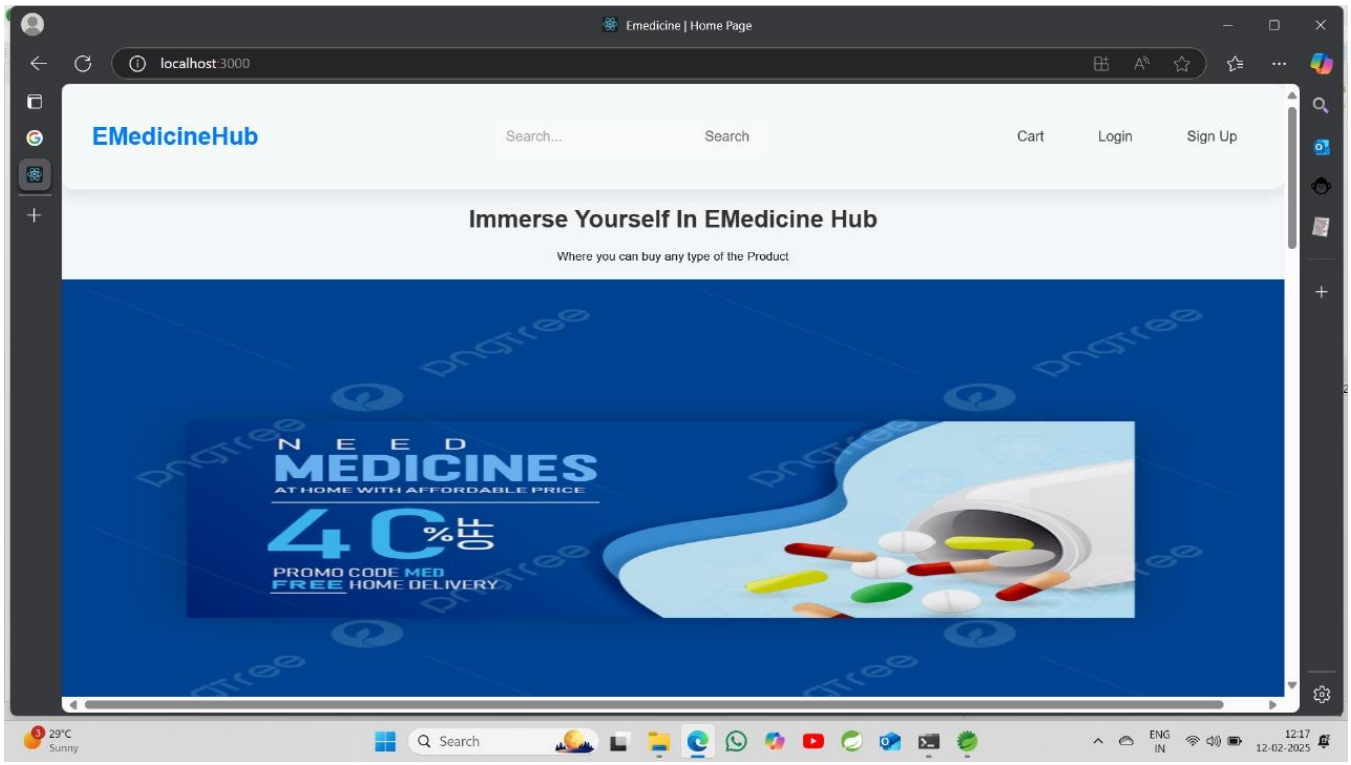
Table 6 : Cart

```
mysql> desc address;
+-----------+--------------+------+-----+---------+----------------+
| Field     | Type         | Null | Key | Default | Extra          |
+-----------+--------------+------+-----+---------+----------------+
| addressid | int          | NO   | PRI | NULL    | auto_increment |
| street    | varchar(255) | NO   |     | NULL    |                |
| zip_code  | varchar(255) | NO   |     | NULL    |                |
| city      | varchar(10)  | NO   |     | NULL    |                |
| flat_no   | varchar(255) | NO   |     | NULL    |                |
| state     | varchar(10)  | NO   |     | NULL    |                |
| userid    | int          | YES  | MUL | NULL    |                |
+-----------+--------------+------+-----+---------+----------------+
7 rows in set (0.00 sec)
```

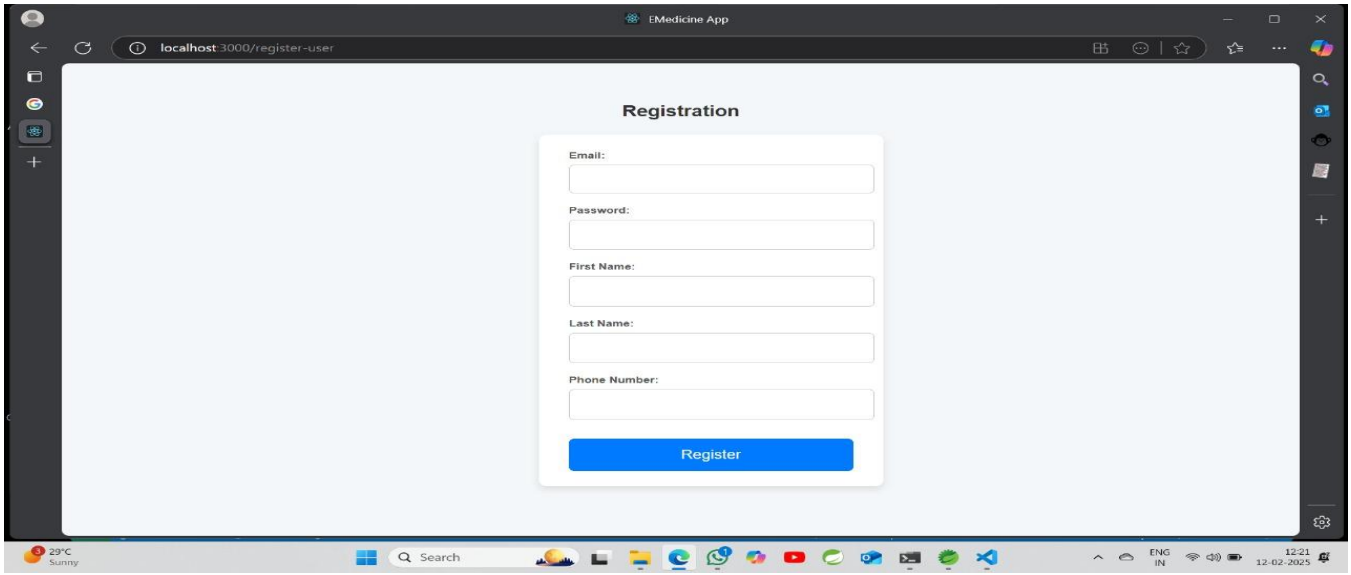
Table 7: Address

5. SNAPSHOTS

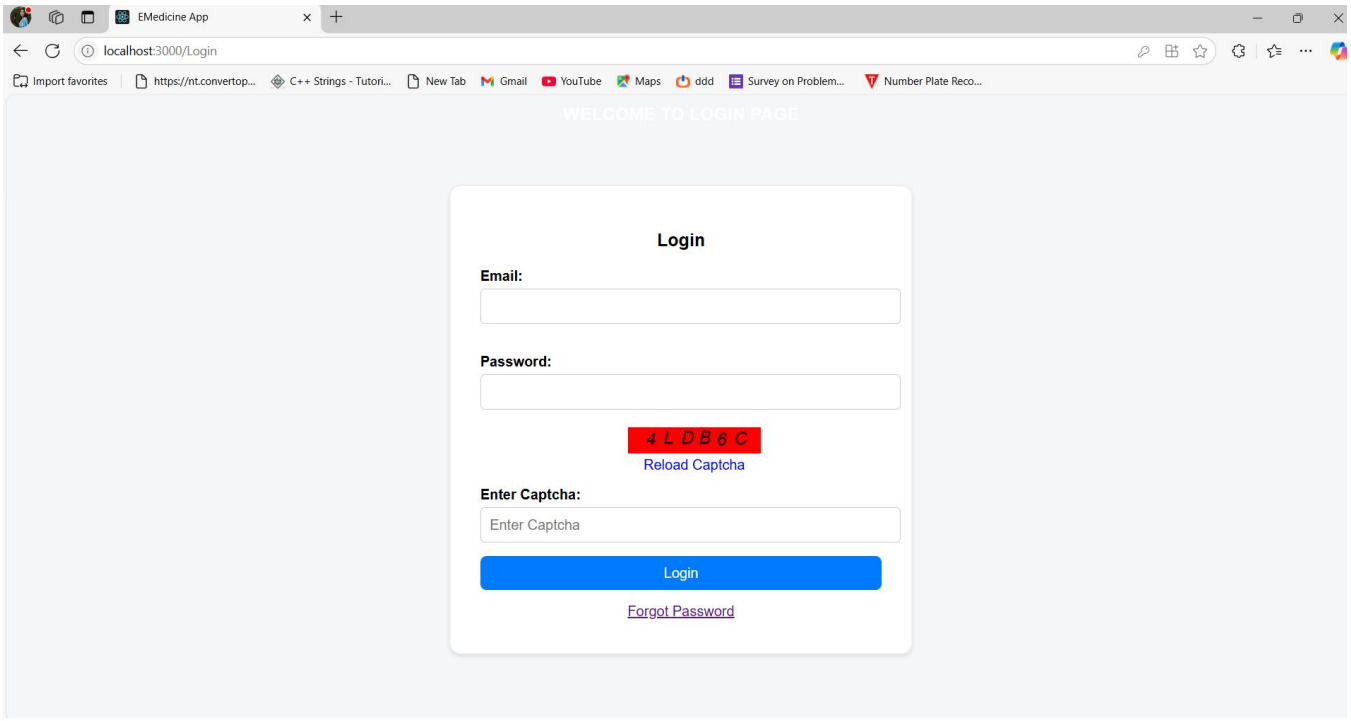
Home page:



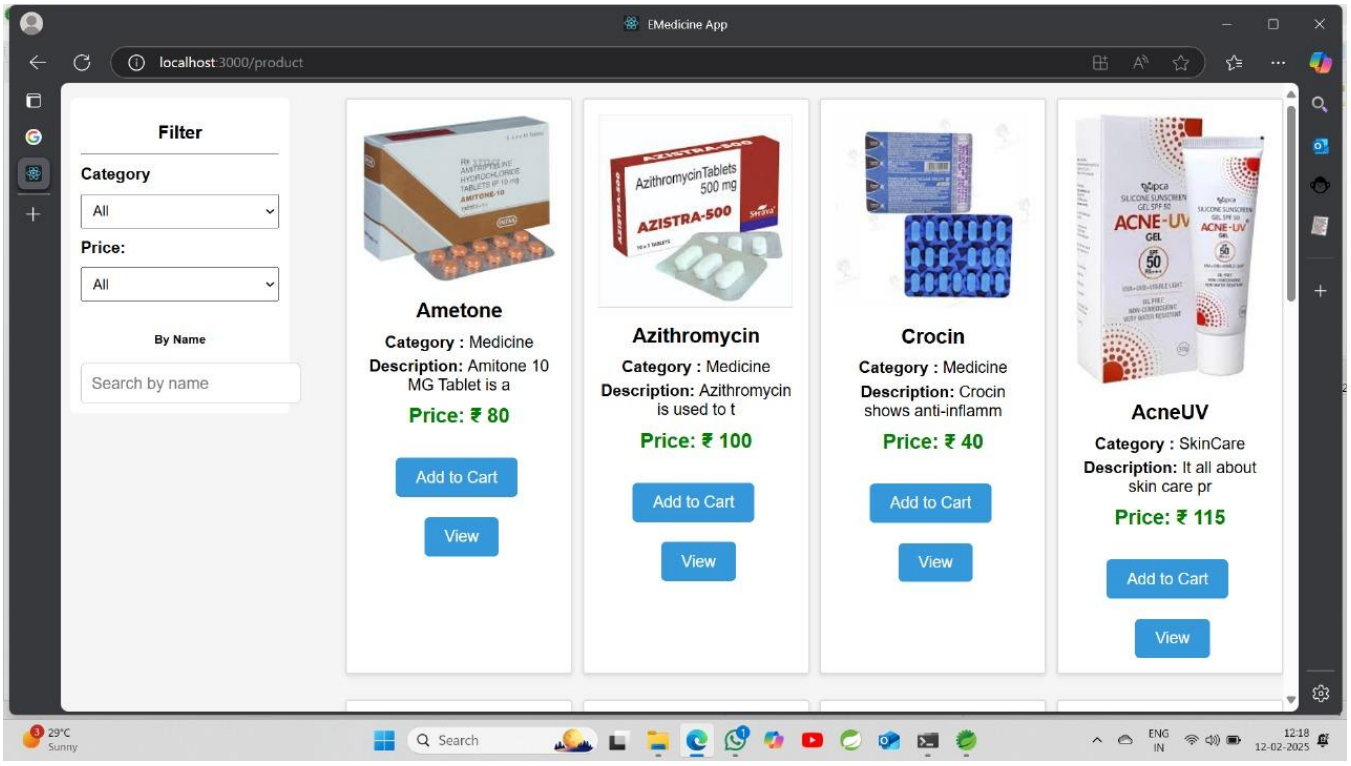
SignUp:



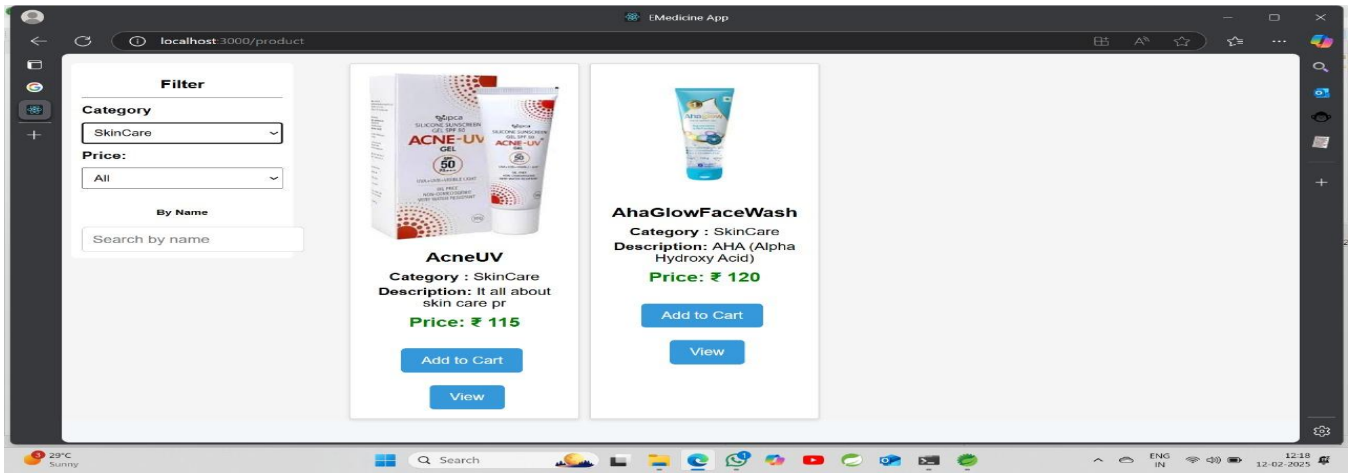
Login:

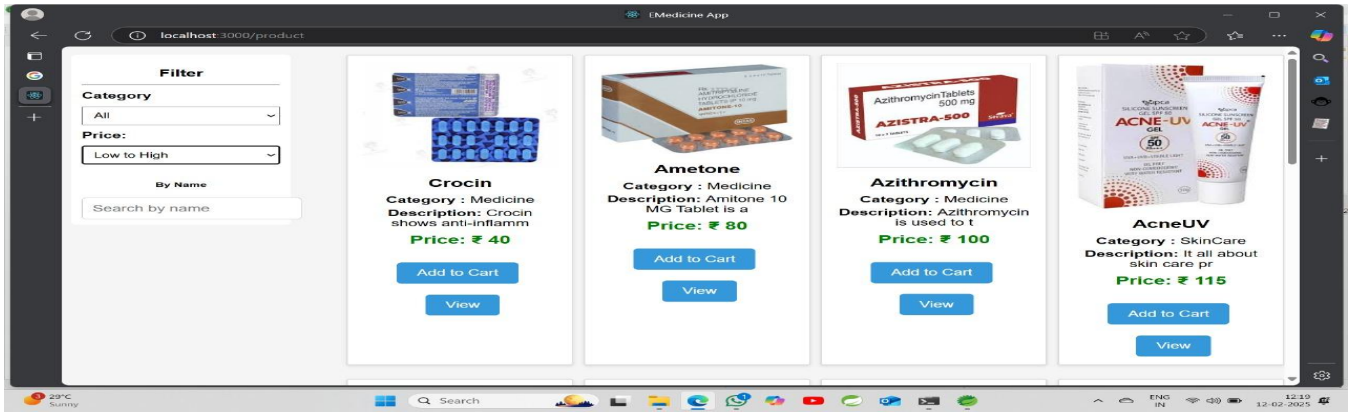


Product Page:

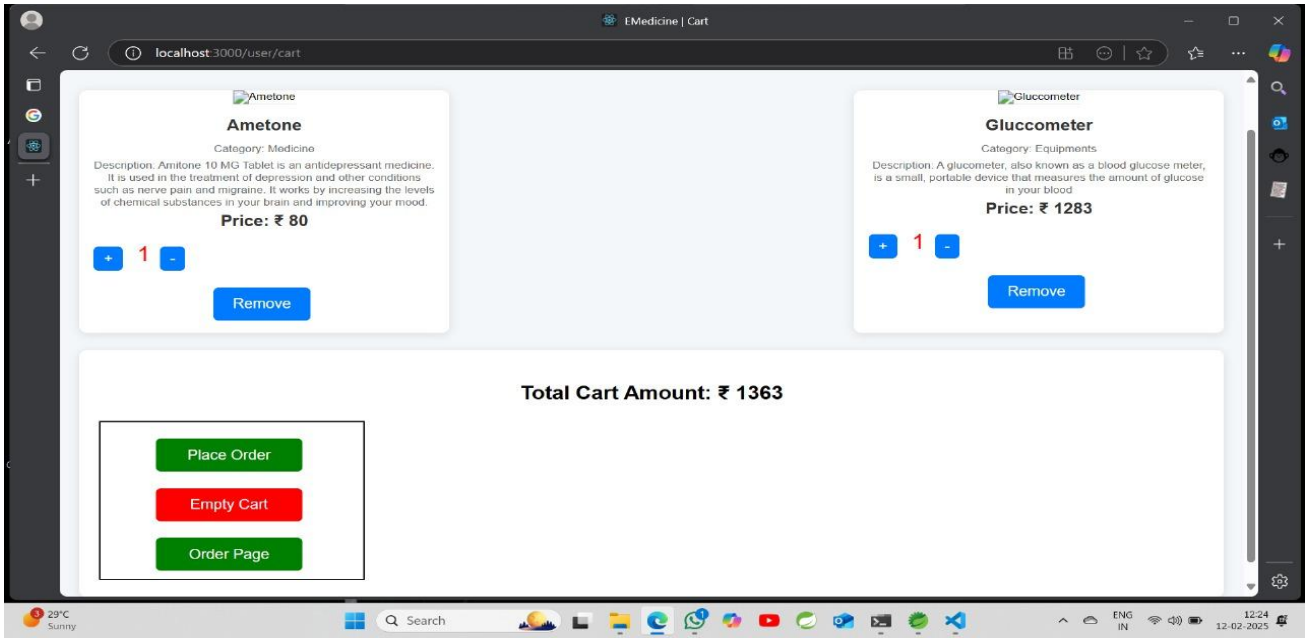


Filtered Product:

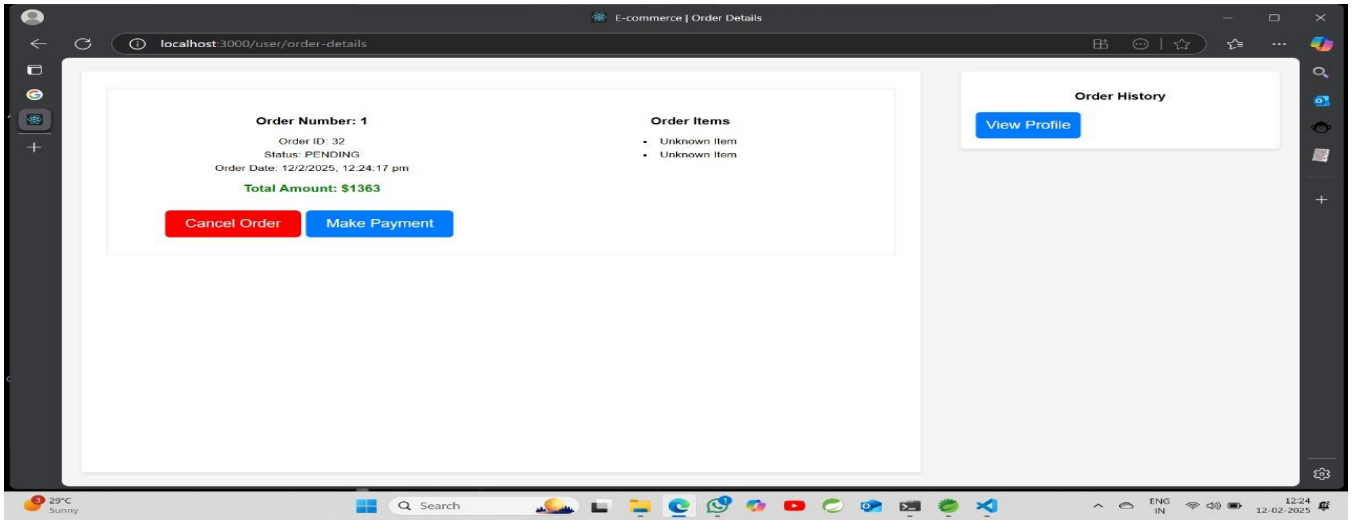




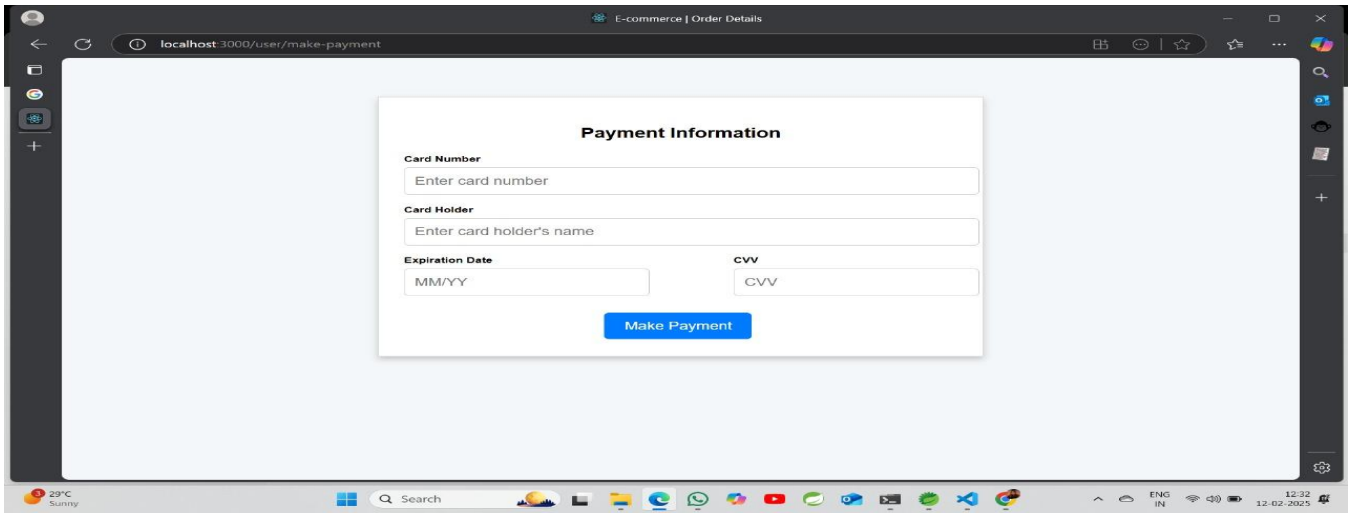
Cart Items:



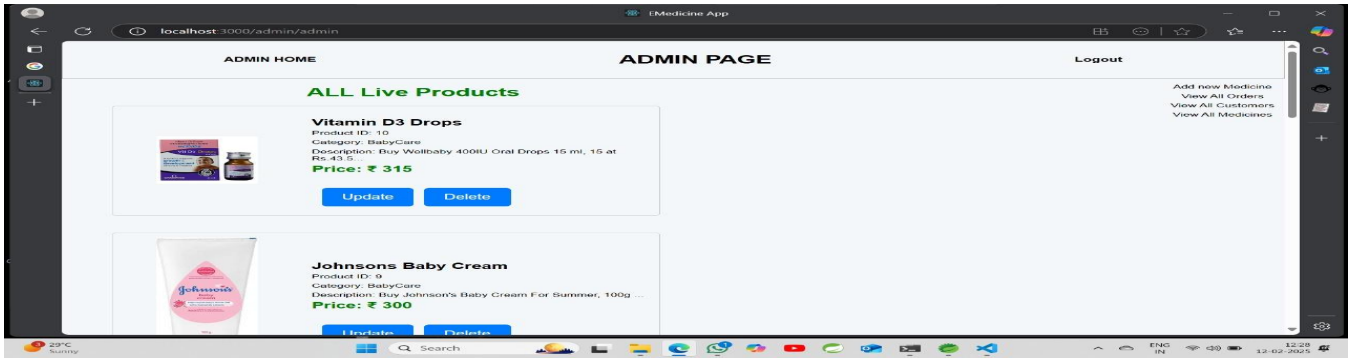
Order Page:



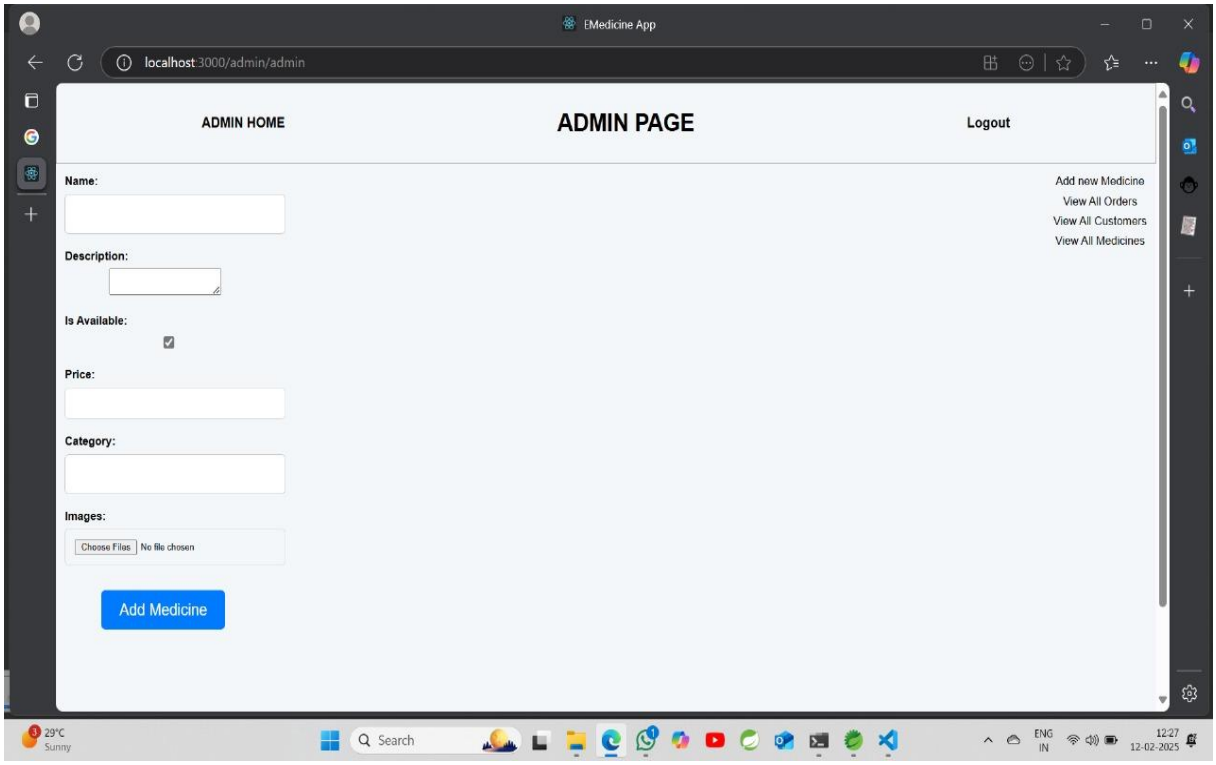
Payment Page:



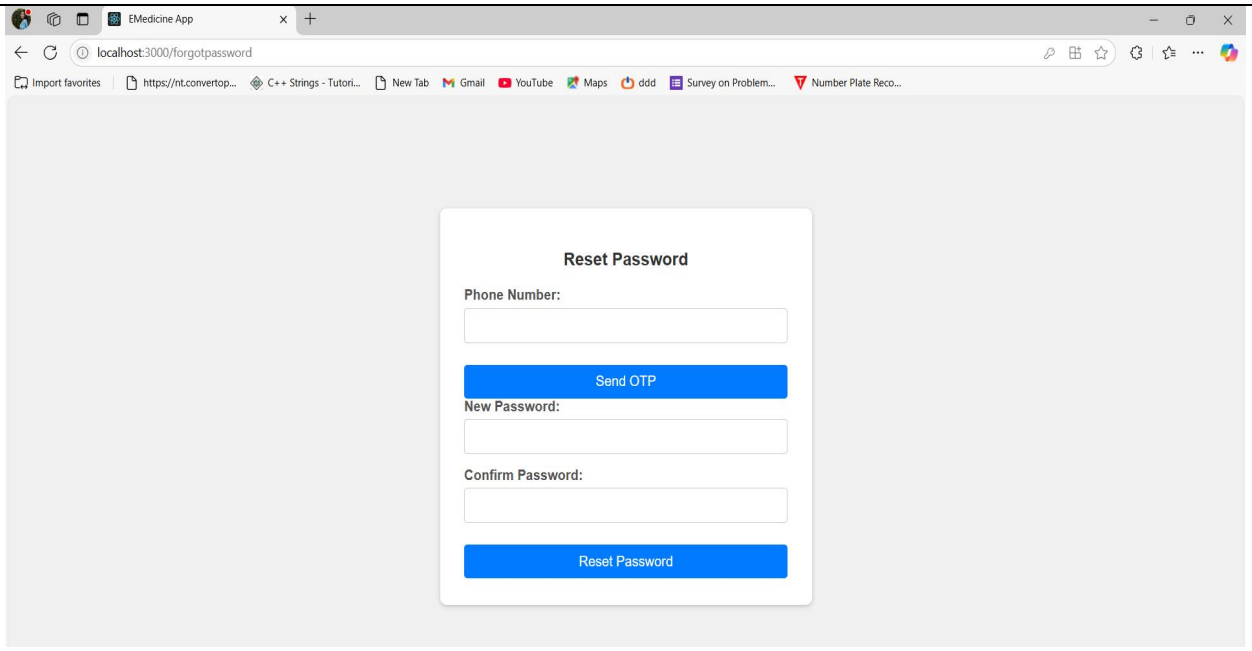
Admin Page:



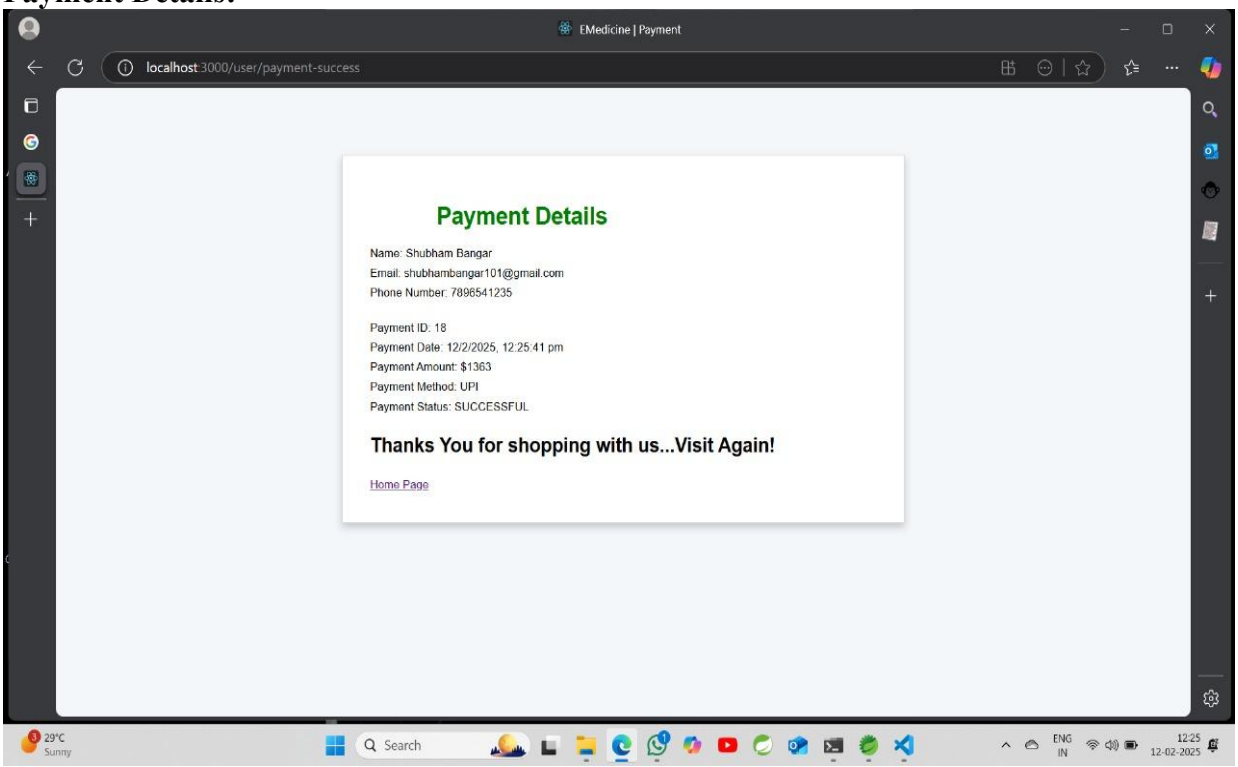
Add New Medicine:



Reset Password:



Payment Details:



6. CONCLUSION

In conclusion, **E-Medicine Hub** addresses key requirements for medicines and order management systems, providing a solid foundation for further development and improvement. Its modular design and adherence to security best practices position it well for continued success and adaptation in the evolving e-commerce landscape.

The **E-Medicine Hub** project exemplifies a sophisticated approach to medicine and order management by leveraging cutting-edge technologies and methodologies. This application is designed to meet the needs of both users and administrators, ensuring a seamless experience for managing medicines, orders, and payments while adhering to stringent security standards.

Its blend of modern technologies, robust security, and user-centric design positions it as a valuable tool for businesses seeking to enhance their e-commerce capabilities. The project's architecture and features provide a strong foundation for ongoing development, making it adaptable to future technological advancements and market trends.

6.1 Future Scope:

1. AI-Powered Health Recommendations

- Implement AI chatbots for basic medical advice and symptom checking.
- Use machine learning to analyze patient history and suggest medicines or lifestyle changes.

2. Telemedicine Integration

- Provide virtual doctor consultations within the platform.
- Enable video calling and online prescription generation for seamless healthcare access.

3. Automated Medicine Delivery

- Partner with local pharmacies and logistics services for same-day delivery.
- Implement drones or autonomous delivery bots for faster service in urban areas.

7. REFERENCES

1. https://docs.spring.io/spring-security/reference/
2. https://docs.spring.io/spring-boot/index.html/
3. https://www.amazon.com/Agile-Software-Development-Principles-Patterns/dp/0135974445
4. Garcia-Molina, H., Ullman, J. D., & Widom, J. (2008). *Database Systems: The Complete Book*. Prentice Hall.
5. Codd, E. F. (1970). *A Relational Model of Data for Large Shared Data Banks*. Communications of the ACM, 13(6), 377-387.
6. Fowler, M. (2003). *Patterns of Enterprise Application Architecture*. Addison-Wesley.
7. Evans, E. (2003). *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison-Wesley.
8. Soni, D., Nord, R. L., & Hofmeister, C. (1995). *Software Architecture in Industrial Applications*. Proceedings of the 17th International Conference on Software Engineering, 196-207.