A PROJECT REPORT ON

ONLINE MEDICAL STORE

SUBMITTED TO

CHANDIGARH UNIVERSITY

By

Sweta Dey
24MCI10247

In partial fulfillment for the award of the degree of

MASTER OF COMPUTER APPLICATION

IN
ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

UNIVERSITY INSTITUTE OF COMPUTING,

CHANDIGARH UNIVERSITY

ACKNOWLEDGEMENT

In performing our project, I would like to show my gratitude to all the faculty members of University Institute of Computing, Chandigarh University for giving us a good guideline for the project throughout numerous consultations. I would also like to expand my deepest gratitude to all those who have directly and indirectly guided me in completion of this project.

In addition, a thank you to my Prof. Dr. Banita Mandal, my mentor who introduced me to the Art of Computer Programming, and her passion for the "underlying structures" had lasting effect. I also thank the University of Chandigarh for consent to include copyright pictures as a part of our paper.

Many people, especially my classmates have made valuable comment suggestions on this project which gave me an inspiration to improve my application.

I thank all the people for their help directly and indirectly to complete my assignment.

CONTENT

1. **Preface**
2. **Problem Statement**
3. **Introduction**
   - Objective
   - Scope of the Project
   - Survey of Technology
   - Requirements
4. **Literature Review**
   - Functionality Overview
   - Technologies Used
5. **System Design and Architecture**
6. **Snapshots**
7. **Benefits**
8. **Challenges and Limitations**
9. **Conclusion**
   - Future Scope
10. **References**

PREFACE

Online Medical Store plays a vital role in modern healthcare systems by ensuring the efficient handling of medicine inventories, prescriptions, customer orders, and overall operations within a pharmacy. With the growing demand for digitization in healthcare, traditional manual systems often fall short in managing large volumes of data, leading to errors, inefficiencies, and delayed service.

This project introduces a full-stack Online Medical Store developed using a React frontend, Node.js backend, and MongoDB database. It allows users to browse medical products, view details, manage a shopping cart, and place orders. On the admin side, the system supports product management and contact message handling. Integration with Stripe enables secure online payments, while RESTful APIs ensure smooth communication between client and server components.

While the current system provides a robust foundation for pharmacy operations, future enhancements can include user authentication, real-time stock updates, prescription uploads, and analytics dashboards. This initiative aims to streamline pharmacy workflows and enhance the user experience, making digital pharmacy access more efficient, accurate, and scalable.

PROBLEM STATEMENT

Given the need for efficient and scalable pharmacy operations, the goal is to develop a web-based Online Medical Store that allows users to browse medical products, view detailed information, manage a shopping cart, and securely place orders. On the backend, the challenge is to design a RESTful API for handling product data, customer interactions, and payments, while maintaining data integrity and real-time responsiveness.

In practice, online medical involves coordinating product inventories, customer service, and order processing — all of which require accurate, reliable systems. While this full-stack application offers a strong foundation through technologies like React, Node.js, MongoDB, and Stripe, real-world deployments may require enhancements such as user authentication, role-based access control, prescription uploads, and real-time inventory tracking to fully meet industry demands.

3. INTRODUCTION

Online Medical is a vital part of the healthcare system, ensuring that medications are dispensed accurately, inventories are well-maintained, and patient-related data is handled securely. As the demand for efficient and responsive healthcare services grows, traditional manual systems often fall short due to inefficiencies and human error. Core operations like product management, order tracking, payment processing, and customer communication increasingly require digital solutions that are streamlined, reliable, and easy to use.

This project introduces a full-stack Online Medical store built using modern web technologies to address these challenges. The frontend, developed with React, provides an intuitive interface for users to explore products, access detailed medical information, and manage their shopping cart. The backend, powered by Node.js and Express, delivers a robust API layer for handling product data, user interactions, and secure payments through Stripe integration. MongoDB serves as the database, efficiently storing information related to products, contact submissions, and transactions.

To enhance the user experience, the system supports features like category-based browsing, real-time cart updates, and a built-in contact form for feedback or queries. RESTful APIs and well-structured database queries ensure smooth interaction between frontend and backend components. While the current version establishes a strong foundation, the system can be further enhanced by incorporating user authentication, prescription upload features, order tracking, and role-based access for administrators. Overall, this project showcases how full-stack development can be effectively used to digitize pharmacy workflows and improve access to essential medical services.

3.1 Objective

- **Develop a pharmacy management web application**
  The system will allow users to browse medical products, view details, and manage a shopping cart.
  It will streamline pharmacy operations and enhance the customer experience.

- **Implement a full-stack architecture using MERN**
  The project uses MongoDB, Express, React, and Node.js for robust and scalable development.
  It ensures efficient data handling and seamless communication between frontend and backend.

- **Integrate secure payment functionality**
  Stripe is used to handle payment processing securely within the application.
  This allows users to make purchases directly from the platform with confidence.

- **Create a responsive and user-friendly interface**
  The React frontend offers intuitive navigation, category-based browsing, and dynamic product views.
  It provides a smooth user experience across devices.

- **Enable backend product and contact management**
  The backend allows for adding and managing products, and handling user contact messages.
  Admin-level interactions are supported for data control and system updates.

- **Ensure reliable API communication and data storage**
  RESTful APIs facilitate structured data access and operations between client and server. MongoDB stores product, order, and contact information efficiently.

- **Prepare for future scalability and enhancements**
  The system is designed to support features like user authentication and prescription uploads.
  Future improvements may include order tracking, admin dashboards, and role-based access control.

3.2 Scope of the Project

- **Comprehensive Pharmacy Management System**
  The project aims to develop a full-stack pharmacy management system that facilitates browsing medical products, managing a shopping cart, and placing orders. It simplifies pharmacy workflows and improves accessibility for users.

- **Backend Integration with MongoDB and Express**
  The backend is built using Node.js and Express, with MongoDB as the primary database for storing product, contact, and order data. RESTful APIs handle all server-side operations efficiently.

- **User-Friendly React Frontend**
  A dynamic React-based frontend provides an intuitive interface for users to explore product categories, view detailed information, and interact with the system in real time.

- **Secure Online Payment Functionality**
  The system integrates Stripe to handle secure online payments, allowing users to purchase medicines with confidence and reliability.

- **Product and Contact Data Handling**
  Admins can add and manage product data, and the system stores user-submitted contact messages for customer support. Data is managed through structured Mongoose models.

- **Deployment-Ready Architecture**
  The application is configured to serve the frontend through the backend in a production environment. It ensures smooth deployment on local or cloud servers.

- **Future Enhancements and Scalability**
  The project is built with scalability in mind and can be extended with features like user authentication, prescription uploads, role-based access control, and inventory tracking to meet real-world pharmacy needs.

3.3 Survey of Technology

Operating System: Windows Operating System
Software used: VS Code, Mongo DB, Postman
Language used: Python, React JS

3.4 Requirements

Hardware requirements for running this project are as follows:
OS: - Windows XP and above
RAM: Ideally 2GB or above
Graphics: Integrated / Min 2GB (discrete)
Hard disk: Min 128 GB

Software requirements for running this application are as follows:
RAM: 1 GB or above.
Memory Space: 500 MB or above.
Languages and Platform used:
Platform: Output window
Language: Python, React JS

4. LITERATURE REVIEW

4.1 Functionality Overview

The Pharmacy Management System is a full-featured web-based application designed to streamline the process of browsing, managing, and purchasing pharmaceutical products. It enhances user experience through a clean, responsive interface and a well-structured backend that works cohesively to ensure efficient pharmacy operations. Below is a comprehensive overview of the core functionalities offered by the application:

A) Product Browsing and Categorization

Users can explore a wide range of medical products categorized under sections such as Dermatology, Depression, Dental Care, Fracture Care, Women's Health, and more. The system displays each product with relevant details including name, image, price, dosage, indication, and side effects. This structured browsing experience allows users to find the right medicine quickly and easily.

B) Product Details Page

Each product can be viewed on a dedicated page that shows detailed specifications. Users can view the product's price, availability, usage instructions, and potential side effects. They can also select the desired quantity and add it to their cart directly from the product detail page.

C) Add to Cart and Cart Management

Users can add products to their shopping cart and update the quantity or remove items as needed. The cart is persistent and stored in the browser using local storage, allowing users to continue shopping without losing their selections. This functionality ensures a smooth e-commerce experience from product selection to checkout.

D) Contact Form

The application includes a contact page where users can submit queries or feedback by entering their full name, email, message, and city. Submitted contact forms are stored in the backend for administrative review, enabling effective customer support and engagement.

E) Payment Integration

Stripe is integrated into the application to enable secure online payment processing. Users can proceed to payment after reviewing their cart, and their card details are handled safely via Stripe's API. Upon successful payment, a confirmation is logged, ensuring transaction traceability and user assurance.

F) Admin Product Management (via API)

Although a full admin dashboard is not yet implemented visually, the backend supports admin operations like creating new products through API endpoints. This sets the groundwork for future administrative interfaces, such as dashboards for product inventory, order tracking, and customer interactions.

G) RESTful API Architecture

The application backend uses RESTful API principles to manage routes for products, contacts, and payments. This modular approach improves scalability and allows for easy integration of new features such as user login, order history, and prescriptions in the future.

H) Responsive Frontend Design

Built with React and Bootstrap, the frontend is fully responsive and optimized for various devices including desktops, tablets, and smartphones. The user interface is intuitive, lightweight, and delivers a consistent experience across different screen sizes.

Together, these features create a robust and interactive Pharmacy Management System that enhances both user experience and operational efficiency. This system serves as a solid foundation for building a complete digital pharmacy platform in the healthcare domain.

4.2 Technologies Used

The Pharmacy Management System is developed using the MERN stack, which includes four modern and efficient technologies: **MongoDB**, **Express.js**, **React.js**, and **Node.js**. This technology stack allows for seamless full-stack JavaScript development, enabling fast, scalable, and maintainable code. Additionally, the project utilizes supporting libraries and tools to enhance functionality, security, and user experience across the platform.

Frontend Technologies

A) React.js

React.js is a component-based JavaScript library used for building dynamic user interfaces. In this project, it manages multiple screens such as product listings, product details, cart, contact form, and admin access. React's reusable components and reactive state management make the application fast, modular, and easy to maintain.

B) Bootstrap 5

Bootstrap 5 is used for responsive design and styling. It provides pre-built components like grids, buttons, navbars, and modals that help ensure the interface is consistent and user-friendly across all screen sizes. The use of Bootstrap enhances the visual appeal and accessibility of the application.

C) React Router

React Router is responsible for client-side routing in the application. It allows users to navigate between different screens like /, /product/:id, /contact, and /cart without full page reloads, ensuring a seamless and interactive experience.

Backend Technologies

A) Node.js

Node.js allows JavaScript to be used on the server side, enabling full-stack development with one programming language. It is asynchronous and event-driven, making it ideal for handling user requests, payment processing, and database interactions efficiently.

B) Express.js

Express.js is a lightweight and flexible web framework built on Node.js. It handles routing, middleware, and API endpoints for products, contacts, and payments. Express simplifies the process of building a RESTful API that connects the frontend to the database.

C) MongoDB

MongoDB is a NoSQL, document-oriented database used to store product details, contact submissions, and cart information. Its flexible schema design and native JSON-like format make it ideal for storing and retrieving structured product and user data dynamically.

D) Mongoose

Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js. It provides schemas and models to define and manage collections like Products and Contacts. Mongoose helps maintain data integrity and simplifies database operations.

Other Tools and Libraries

A) Stripe

Stripe is used to handle secure online payment processing. It enables the application to collect user payment information safely, process transactions, and return real-time confirmation without storing sensitive data on the server.

B) Dotenv

Dotenv is used for managing environment variables securely, such as API keys and database URIs. It allows sensitive configuration values to be stored in a .env file and accessed safely throughout the application.

C) Nodemon

Nodemon is a development utility that automatically restarts the server whenever code changes are detected, greatly improving the development workflow.

D) Git & GitHub

Git is used for version control, while GitHub serves as the remote repository for collaboration, source code hosting, and project backup.

5. SYSTEM DESIGN AND ARCHIECTURE

5.1 System Features

The Pharmacy Management System is designed to offer a comprehensive, user-friendly solution for browsing, managing, and purchasing pharmaceutical products online. The system integrates essential e-commerce features with healthcare product categorization, delivering a responsive and interactive experience. It blends simplicity and utility to meet the needs of both consumers and administrators in a digital pharmacy environment.

A) Product Browsing and Detailed Information

The application allows users to explore a wide range of medicines categorized under sections such as Dermatology, Depression, Dental Care, Fracture Support, and Women's Health. Each product displays:

- Name and image
- Indication and dosage details
- Side effects
- Price and availability

This helps users make informed decisions before adding items to their cart.

B) Add to Cart and Checkout

Users can add desired products to a shopping cart for review before checkout. The cart supports:

- Adding multiple products
- Quantity selection and updates
- Removing items as needed
- Local storage persistence

The cart reflects real-time changes and ensures a smooth checkout process.

C) Contact Form for Customer Support

The system features a contact page where users can:

- Submit inquiries using their full name, email, city, and message
- Receive backend acknowledgment upon submission
- Allow the admin to view saved contact messages via the database

This creates a communication channel between customers and pharmacy admins.

D) RESTful API Architecture

The backend includes robust API endpoints for:

- Retrieving all products or single product details
- Creating new products (admin use)
- Saving contact form data
- Handling payment processing

This modular and scalable structure supports future integration and feature expansion.

E) Admin Product Handling (via Backend)

Although a visual admin panel is not yet implemented, the backend currently supports:

- Creating new product entries via POST requests
- Managing data using Mongoose schemas
- Saving and retrieving product and contact data in MongoDB

This backend support lays the foundation for future admin dashboards.

F) Responsive Interface and Navigation

Built with React.js and styled using Bootstrap 5, the frontend:

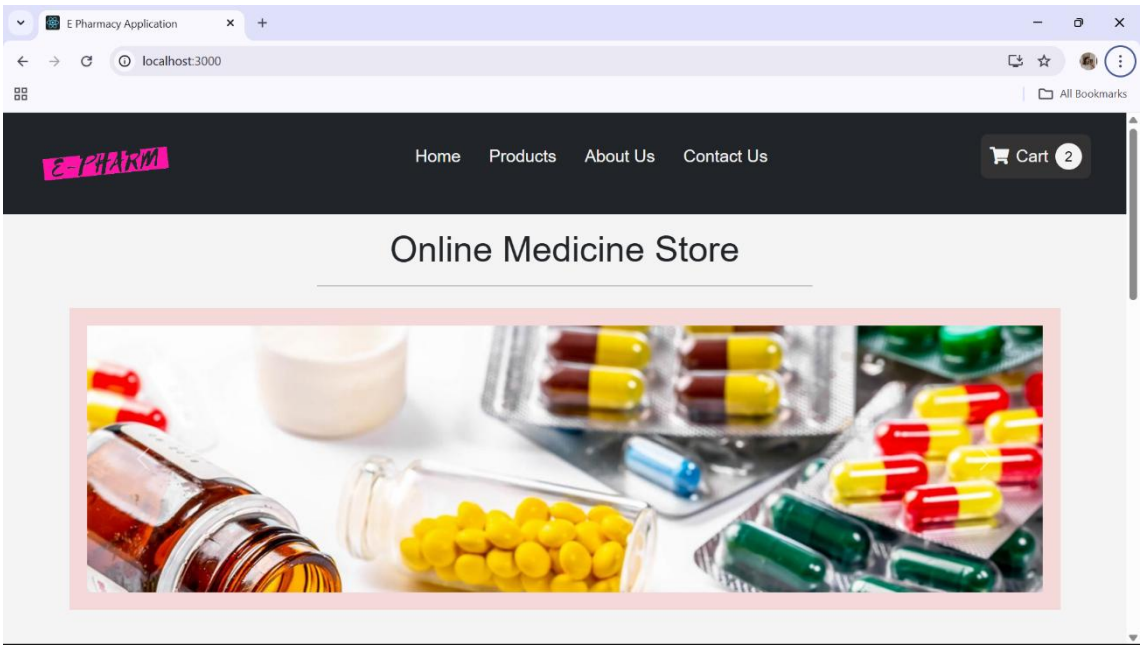
- Adapts smoothly to desktops, tablets, and mobile devices
- Provides intuitive navigation through categories and routes
- Ensures a visually appealing and accessible interface across all platforms

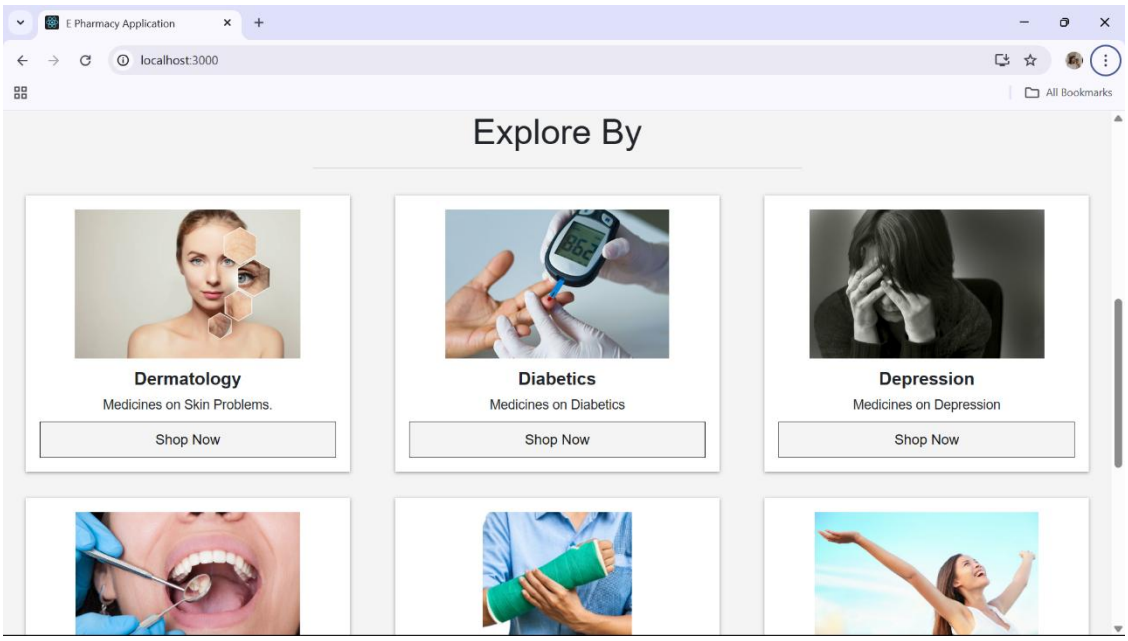
G) Real-Time User Interaction and Data Handling

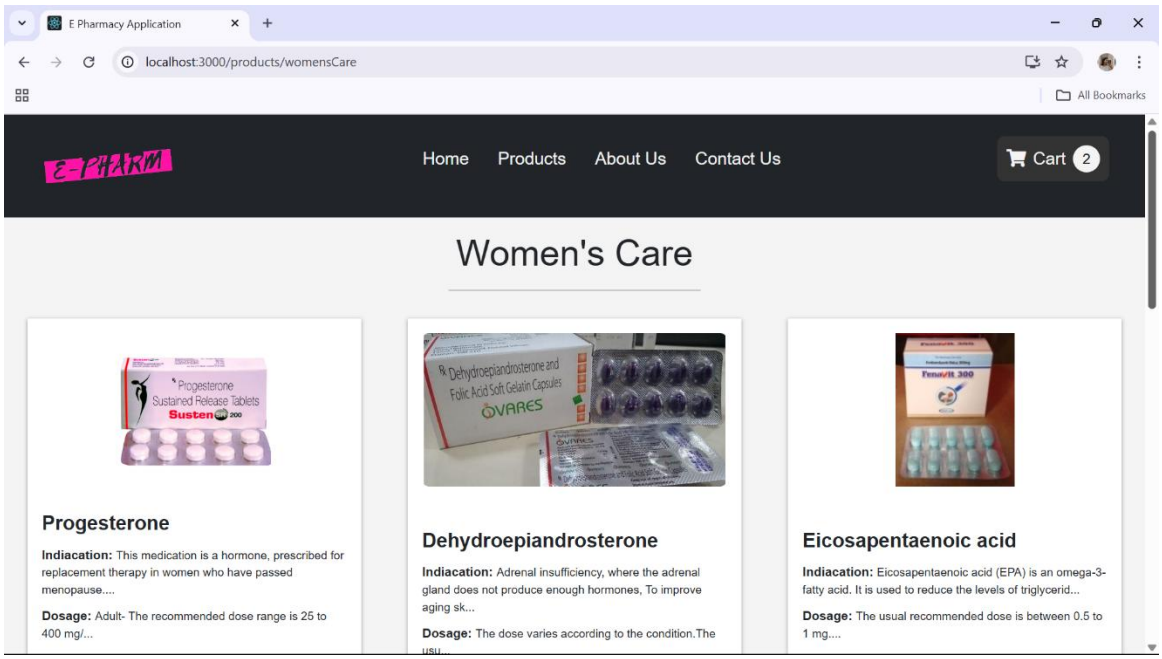
All user interactions (such as adding items to the cart, viewing product details, or submitting contact forms) are handled in real time via seamless frontend-backend communication. This ensures:

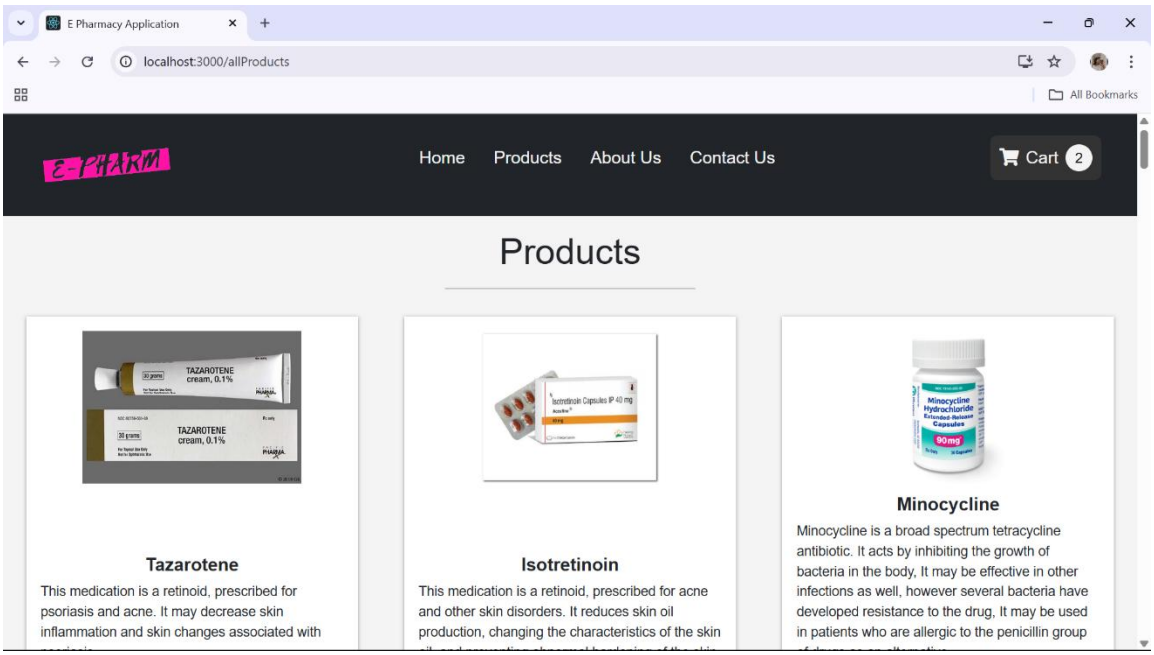
- Quick updates to the UI
- Consistent user experience
- Accurate database syncing for all operations

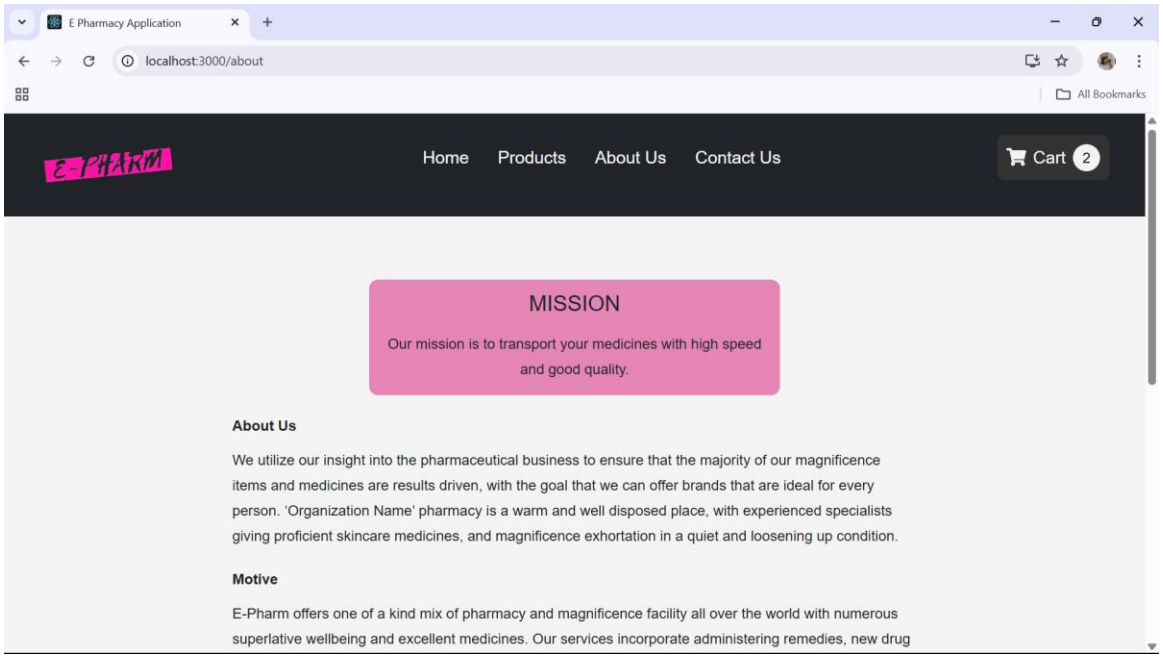
6. SNAPSHOTS

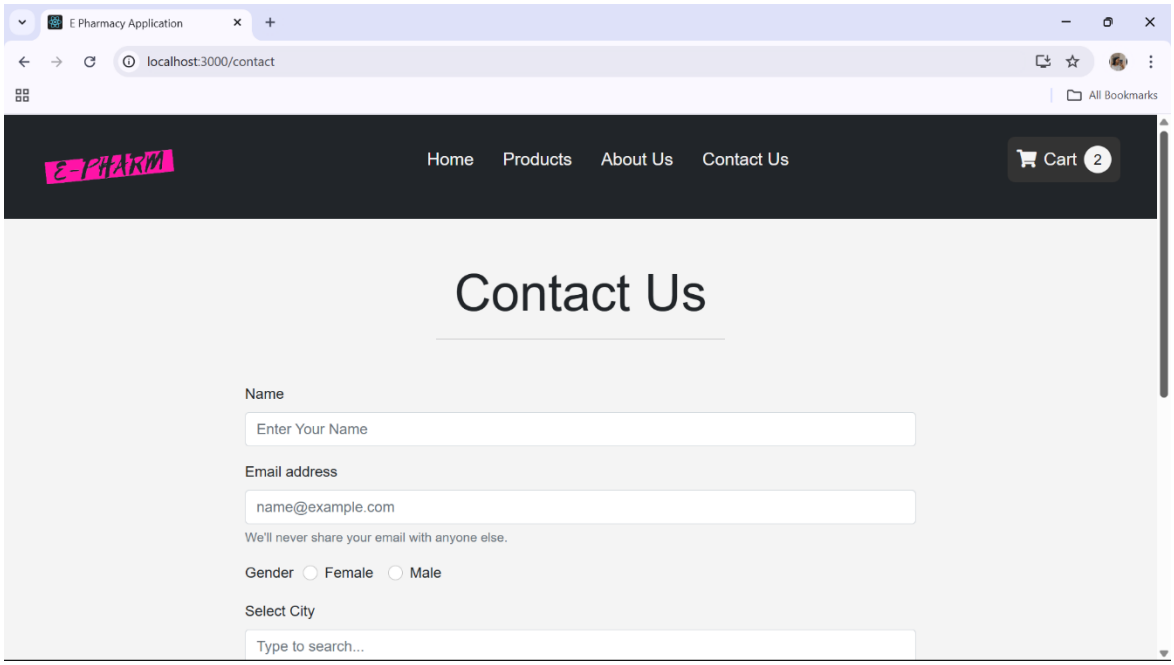


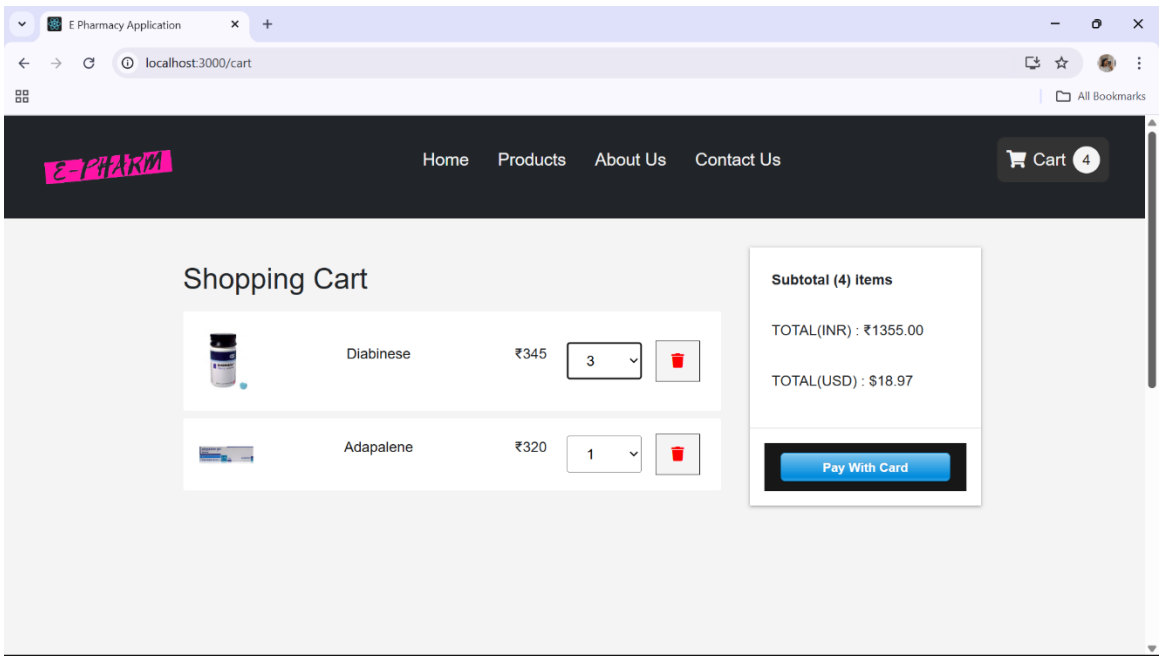

E Pharmacy Application
localhost:3000
All Bookmarks
Explore By
Dermatology
Medicines on Skin Problems.
Shop Now
Diabetics
Medicines on Diabetics
Shop Now
Depression
Medicines on Depression
Shop Now


E Pharmacy Application
localhost:3000/products/womensCare
All Bookmarks
E-PHARM
Home
Products
About Us
Contact Us
Cart 2
Women's Care
Progesterone
Indiacation: This medication is a hormone, prescribed for replacement therapy in women who have passed menopause....
Dosage: Adult- The recommended dose range is 25 to 400 mg/...
Dehydroepiandrosterone
Indiacation: Adrenal insufficiency, where the adrenal gland does not produce enough hormones, To improve aging sk...
Dosage: The dose varies according to the condition.The usu...
Eicosapentaenoic acid
Indiacation: Eicosapentaenoic acid (EPA) is an omega-3-fatty acid. It is used to reduce the levels of triglycerid...
Dosage: The usual recommended dose is between 0.5 to 1 mg....


E Pharmacy Application
localhost:3000/allProducts
All Bookmarks
E-PHARM
Home
Products
About Us
Contact Us
Cart 2
Products
Tazarotene
This medication is a retinoid, prescribed for psoriasis and acne. It may decrease skin inflammation and skin changes associated with
Isotretinoin
This medication is a retinoid, prescribed for acne and other skin disorders. It reduces skin oil production, changing the characteristics of the skin
Minocycline
Minocycline is a broad spectrum tetracycline antibiotic. It acts by inhibiting the growth of bacteria in the body, It may be effective in other infections as well, however several bacteria have developed resistance to the drug, It may be used in patients who are allergic to the penicillin group


E Pharmacy Application
localhost:3000/about
All Bookmarks
E-PHARM
Home
Products
About Us
Contact Us
Cart 2
MISSION
Our mission is to transport your medicines with high speed and good quality.
About Us
We utilize our insight into the pharmaceutical business to ensure that the majority of our magnificence items and medicines are results driven, with the goal that we can offer brands that are ideal for every person. 'Organization Name' pharmacy is a warm and well disposed place, with experienced specialists giving proficient skincare medicines, and magnificence exhortation in a quiet and loosening up condition.
Motive
E-Pharm offers one of a kind mix of pharmacy and magnificence facility all over the world with numerous superlative wellbeing and excellent medicines. Our services incorporate administering remedies, new drug


E Pharmacy Application
localhost:3000/contact
All Bookmarks
E-PHARM
Home
Products
About Us
Contact Us
Cart 2
Contact Us
Name
Enter Your Name
Email address
name@example.com
We'll never share your email with anyone else.
Gender Female Male
Select City
Type to search...


E Pharmacy Application
localhost:3000/cart
All Bookmarks
E-PHARM
Home
Products
About Us
Contact Us
Cart 4
Shopping Cart
Diabinese
₹345
3
Adapalene
₹320
1
Subtotal (4) items
TOTAL(INR) : ₹1355.00
TOTAL(USD) : $18.97
Pay With Card

7. BENEFITS

- **Streamlined Online Pharmacy Experience**
  The system allows users to browse, view, and purchase medicines easily from the comfort of their homes, reducing the need to physically visit a pharmacy and improving access to healthcare products.

- **Improved Operational Efficiency**
  By managing products, customer inquiries, and payments through a centralized system, pharmacy staff can reduce manual errors and save time on daily tasks, leading to more efficient workflows.

- **Secure and Convenient Payment Processing**
  With Stripe integration, users can safely complete transactions online, ensuring data protection and enhancing trust in the platform.

- **Better Customer Engagement**
  The built-in contact form allows users to submit feedback or inquiries directly through the platform, helping pharmacy staff respond to concerns quickly and maintain strong customer relationships.

- **Real-Time Product Interaction**
  Users can instantly view product availability, select quantities, and update their shopping cart dynamically. This enhances user satisfaction and supports a smooth purchasing journey.

- **Scalability and Flexibility**
  The modular architecture of the system allows for future upgrades such as user login, prescription uploads, order tracking, and role-based admin controls — making the system adaptable to a full-scale pharmacy platform.

8. CHALLENGES AND LIMITATIONS

- **Lack of User Authentication**
  The current system does not support user registration or login, limiting personalization, order tracking, and secure user data management.

- **No Admin Dashboard Interface**
  While product creation and data handling are supported via API, the absence of a visual admin panel restricts real-time inventory updates and efficient product management by pharmacy staff.

- **Static Product Dataset**
  All product data must be manually entered into the system. There is no integration with real-time inventory or external pharmacy databases, which can affect product availability accuracy.

- **Limited Error Handling on Forms**
  The contact form and cart operations have minimal validation, which can lead to issues if users submit incomplete or incorrect information.

- **No Prescription Upload Feature**
  The system currently does not support uploading prescriptions, which limits its usability for medications that legally require doctor approval.

- **Scalability and Performance Constraints**
  While suitable for small to medium-scale use, the system may face performance issues if scaled to handle large inventories, heavy traffic, or multiple user roles without backend optimization.

9. CONCLUSION

The Pharmacy Management System project successfully demonstrates how modern web technologies can streamline online pharmacy operations. By integrating React for the frontend, Node.js and Express for the backend, MongoDB for data storage, and Stripe for secure payments, the system provides a responsive and functional platform for browsing, purchasing, and managing medicines. While the application covers core features like product display, cart management, and contact handling, it currently lacks advanced capabilities such as user authentication, prescription uploads, and administrative dashboards. With future enhancements, this system can evolve into a full-scale digital pharmacy platform capable of serving both users and pharmacy staff more effectively.

10.1 Future Scope

- **User Authentication and Role Management**
  Implementing login functionality with role-based access control will enable personalized user experiences and secure admin operations.

- **Prescription Upload and Verification**
  Allowing users to upload prescriptions for restricted medicines will align the system with real-world pharmacy requirements and legal compliance.

- **Admin Dashboard Development**
  A dedicated interface for admins to add, edit, or remove products and view contact messages will improve inventory control and customer service.

- **Order Tracking and History**
  Adding order tracking, history, and status updates will help users monitor their purchases and improve post-sale engagement.

- **Real-Time Inventory Integration**
  Connecting the platform to a live inventory system will ensure product availability data remains accurate and up-to-date.

- **Cloud Deployment and Scalability**
  Hosting the platform on cloud services like Render, Vercel, or AWS will enhance performance, scalability, and accessibility across wider regions.

10. REFERENCES

- **W3Schools – React JS Tutorial, Node.js Tutorial, Express.js Tutorial**
  https://www.w3schools.com/react/
  → Used for understanding React components, JSX, routing, and state management.
  https://www.w3schools.com/nodejs/
  → Reference for building server-side applications, file handling, and middleware concepts.
  https://www.w3schools.com/express/
  → Used to implement RESTful APIs and middleware in the Node.js backend.

- **MongoDB – Official Documentation**
  https://www.mongodb.com/docs/
  → For creating and managing MongoDB collections, schemas, and CRUD operations.

- **Mongoose – ODM Documentation**
  https://mongoosejs.com/docs/
  → Used for defining schemas and interacting with the MongoDB database in Node.js.

- **Stripe – Official Developer Docs**
  https://stripe.com/docs
  → For integrating secure online payment processing into the system.

- **React Router – Documentation**
  https://reactrouter.com/en/main
  → For implementing client-side routing and navigation in the React frontend.

- **MDN Web Docs – JavaScript**
  https://developer.mozilla.org/en-US/docs/Web/JavaScript
  → General reference for JavaScript syntax, functions, and DOM manipulation.

- **Bootstrap – Official Docs (v5)**
  https://getbootstrap.com/docs/5.0/getting-started/introduction/
  → Used to design responsive and mobile-friendly layouts across all devices.

- **Dotenv – NPM Package Documentation**
  https://www.npmjs.com/package/dotenv
  → Used for managing environment variables securely in Node.js applications.

- **Postman – API Testing Guide**
  https://learning.postman.com/docs/getting-started/introduction/
  → Used during development to test RESTful API endpoints.

- **Git & GitHub – Guides**
  https://docs.github.com/en/get-started
  → Used for version control, collaboration, and code repository management.