REPORT OF ONE MONTH TRAINING

at

STEP GNDEC

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY
(Computer Science and Engineering)

JUNE- JULY, 2025

SUBMITTED BY:

NAME : PRABHDEEP KAUR

UNIVERSITY ROLL NO.(s) :

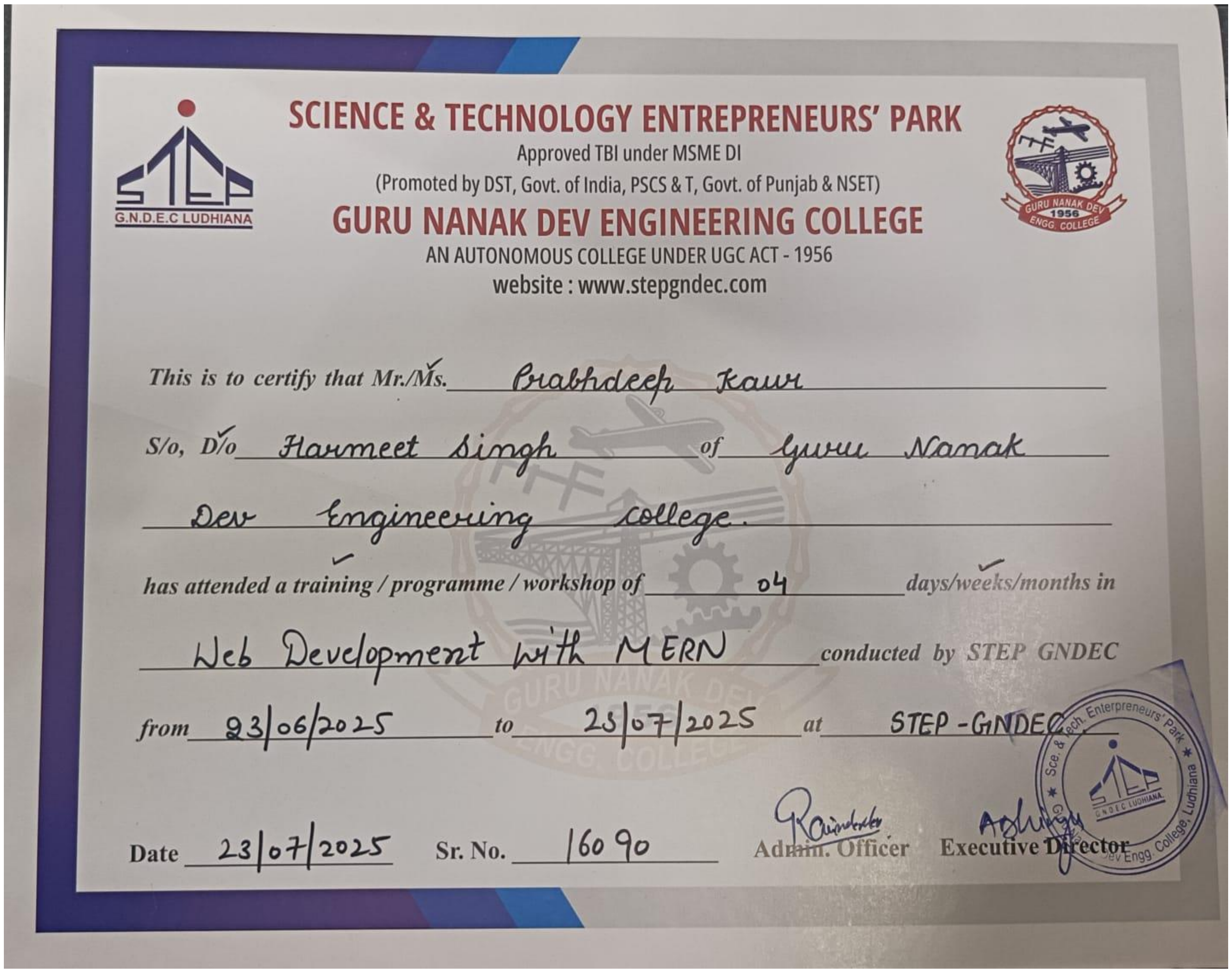
2435240

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

GURU NANAK DEV ENGINEERING COLLEGE LUDHIANA

(An Autonomous College Under UGC ACT)

CERTIFICATE BY INSTITUTE



SCIENCE & TECHNOLOGY ENTREPRENEURS' PARK
Approved TBI under MSME DI
(Promoted by DST, Govt. of India, PSCS & T, Govt. of Punjab & NSET)
GURU NANAK DEV ENGINEERING COLLEGE
AN AUTONOMOUS COLLEGE UNDER UGC ACT - 1956
website : www.stepgndec.com

This is to certify that Mr./[✓ Ms.] Prabhdeep Kaur

S/o, [✓ D/o] Harmeet Singh *of* Guru Nanak Dev Engineering college.

has attended a [✓ training] / programme / workshop of 04 *days/[✓ weeks]/months in*

Web Development with MERN *conducted by STEP GNDEC*

from 23/06/2025 *to* 25/07/2025 *at* STEP-GNDEC.

Date 23/07/2025 Sr. No. 1690 Admin. Officer Executive Director

Sce. & Tech. Entrepreneurs' Park, G.N.D. Engg. College, Ludhiana

GURU NANAK DEV ENGINEERING COLLEGE, LUDHIANA

CANDIDATE'S DECLARATION

I **PRABHDEEP KAUR**, hereby declare that I have undertaken one month training at **STEP GNDEC** during the period from **23 JUNE 2025** to **23 JULY 2025**, in partial fulfillment of the requirements for the award of the degree of **B.Tech (Computer Science Engineering)** at **Guru Nanak Dev Engineering College, Ludhiana**.

The work which is being presented in the training report submitted to the **Department of Electronics and Communication Engineering** at **Guru Nanak Dev Engineering College, Ludhiana**, is an **authentic record of training work** carried out by me.

Signature of the Student

The one month industrial training Viva–Voce Examination of___________________has been held on _____and accepted.

Signature of Internal Examiner Signature of External Examiner

ABSTRACT

During my one-month industrial training at STEP GNDEC, I gained hands-on experience in **Web Development with MERN STACK**, learning both **frontend and backend technologies**. The training focused on building a full-stack online medicine ordering platform, *MediKart*, using the **MERN Stack** — MongoDB, Express.js, React.js, and Node.js.

As part of the training, I learned and applied **HTML, CSS, JavaScript, and Bootstrap** to design responsive and user-friendly web pages. I developed the frontend using **React.js**, creating interactive pages such as Home, Products, Cart, Checkout, and My Orders. On the backend, I worked with **Node.js and Express.js**, learning to handle API requests, secure user authentication with JWT, and integrate a **MongoDB** database for managing users, products, orders, and addresses.

This project provided practical exposure to **state management in React**, **frontend-backend integration**, **database operations**, and **version control using Git and GitHub**. Through this training, I enhanced my understanding of building scalable, dynamic web applications, bridging the gap between theoretical knowledge and real-world web development skill.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to the management and mentors at STEP GNDEC for providing me the opportunity to undergo industrial training in web development. This training gave me practical exposure to full-stack development using the **MERN stack**, including **HTML, CSS, JavaScript, Bootstrap, React.js** and the basics of **MongoDB,Node.js, Express.js**,.

I would also like to express my sincere gratitude to the **Department of Computer Science and Engineering, Guru Nanak Dev Engineering College, Ludhiana**, for incorporating this industrial training as a valuable part of the curriculum. I am especially thankful to **Jaswant Sir** for his constant guidance and motivation, and to **Ms. Kiran (HOD)** for her encouragement and support throughout my academic journey.

This training has given me hands-on experience in **web development, full-stack project implementation, and industry-standard workflows**, and I am truly grateful to everyone who contributed to making this learning experience meaningful and enriching.

ABOUT INSTITUTION

STEP GNDEC is an initiative by **Guru Nanak Dev Engineering College** to foster innovation, entrepreneurship, and practical learning in technology. The organization provides students with exposure to real-world projects, modern development tools, and industry-standard workflows, bridging the gap between academic learning and practical application.

STEP GNDEC focuses on nurturing talent in software development, web technologies, and entrepreneurial ventures. It provides a structured environment for students to work on live projects, gain hands-on experience in web development with MERN STACK, and develop skills that are crucial for the IT industry.
During my training at STEP GNDEC, I got the opportunity to work on the **MediKart project**, a full-stack web application for online medicine ordering. This exposure helped me understand project development cycles, database integration, frontend-backend communication, and best practices in software engineering.

Through such initiatives, **STEP GNDEC** continues to play a vital role in empowering students with technical knowledge, practical experience, and professional skills essential for real-world success.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

In today's digital age, the demand for efficient and user-friendly online platforms is growing across all sectors. Consumers expect seamless access to services and products, along with fast, secure, and reliable online experiences.

During my **web development training with the MERN stack at STEP GNDEC**, I focused on developing **MediKart**, a comprehensive full-stack web application designed to simplify the process of purchasing medicines and healthcare products online. The primary goal of MediKart is to provide users with a **secure, interactive, and intuitive platform** for browsing, ordering, and tracking healthcare products from the convenience of their homes.

The vision behind MediKart encompasses two main objectives: to offer users a **responsive and easy-to-navigate frontend interface** for product selection and order placement, and to streamline **backend operations**, including order management, inventory tracking, and secure payment processing. Every feature, from user authentication to product search, shopping cart integration, and order confirmation workflows, was carefully planned and implemented to enhance operational efficiency and deliver a superior digital experience.

This training provided me with hands-on exposure to the **MERN stack**, including:

- **Frontend development with React.js** for building dynamic and responsive user interfaces.
- **Backend API development using Node.js and Express** for handling requests and managing server logic.
- **Database management with MongoDB** for storing and retrieving application data efficiently.
- **Version control with GitHub**, enabling collaborative development and project versioning.

Through the MediKart project, I gained valuable experience in **full-stack web development**, bridging theoretical knowledge with practical implementation, understanding application architecture, and learning how to manage a real-world web application from **frontend to backend**.

1.0 Foundational Concepts

During my **web development with MERN STACK training at STEP GNDEC**, I gained a solid understanding of the essential technologies and tools used in modern web applications. The training focused on building a strong foundation in **frontend development, backend basics, and database management**, which prepared me to work on the full-stack project **MediKart**.

Key concepts and skills covered in this training include:

- **HTML & CSS:** Learned to structure web pages and style them effectively, creating responsive layouts suitable for multiple devices.
- **JavaScript:** Gained the ability to add interactivity, manipulate the DOM, and handle events in web applications.
- **Bootstrap:** Applied pre-built components and responsive grid layouts to make websites mobile-friendly and visually appealing.
- **React.js (Frontend):** Learned to develop dynamic, component-based user interfaces, manage state with hooks, and build reusable UI components.
- **Node.js & Express (Backend):** Learned the basics of server-side programming, creating RESTful APIs, and handling HTTP requests and responses.
- **MongoDB (Database):** Gained an understanding of NoSQL databases, collections, documents, and basic CRUD operations.
- **GitHub (Version Control):** Learned to track project versions, collaborate effectively, and manage code changes.
- **Integration & Deployment Basics:** Gained initial experience connecting the frontend, backend, and database, and understanding deployment fundamentals.

This foundational knowledge enabled me to **progress from building simple static websites to creating a full-stack MERN application**. It also provided a practical understanding of how frontend and backend technologies work together to deliver a seamless web experience.

1.1Evolution of Web Technologies

Web technologies have undergone significant evolution over the years, transforming how websites are designed, developed, and experienced. Understanding this evolution is essential to appreciate modern frameworks like **React.js** and full-stack solutions such as the **MERN stack**.

- **Static Web (Early Web):** Initially, websites were static, created with basic **HTML and CSS**. Content was fixed, interactivity was minimal, and any updates required manual editing of HTML files.
- **Dynamic Web (Server-Side Scripting):** Technologies like **PHP**, **ASP**, and later **Node.js** allowed websites to generate content dynamically based on user input. This enabled functionalities like user authentication, forms, and personalized content.
- **Client-Side Interactivity:** The introduction of **JavaScript** allowed interactivity directly in the browser. Users could interact with forms, menus, and dynamic elements without refreshing the page.
- **Responsive Design & CSS Frameworks:** Frameworks such as **Bootstrap** made it easier to create websites that adapt to various devices and screen sizes, improving accessibility and user experience.
- **Modern Frontend Development:** Libraries like **React.js** introduced **component-based architecture**, **virtual DOM**, and **state management**, allowing developers to create scalable and dynamic user interfaces.
- **Full-Stack Development:** Combining frontend technologies (React.js) with backend frameworks (Node.js and Express) and databases (MongoDB) enables developers to build complete web applications, like **MediKart**, with a seamless end-to-end workflow.

This evolution highlights the shift from static pages to **dynamic, interactive, and scalable web applications**, which forms the foundation of modern web development practices.

1.2Importance of Frontend Development

The **frontend** of a web application is the interface through which users interact with the system. It is a critical aspect of web development because it **directly impacts user satisfaction, engagement,**

and retention. During my **web development training with the MERN stack**, I learned that frontend development plays a key role in the success of any application, including **MediKart**, in the following ways:

- **Enhanced User Experience (UX):**
  - Provides a seamless and intuitive interface for users.
  - Simplifies navigation and ensures users can complete tasks efficiently.
  - For MediKart, this includes easy browsing of medicines, quick addition to the shopping cart, and smooth checkout processes.
- **Visual Appeal and Branding:**
  - Creates a professional and trustworthy appearance.
  - Uses **HTML, CSS, and Bootstrap** to design visually engaging and functional layouts.
  - Ensures MediKart is aesthetically pleasing and user-friendly.
- **Dynamic Interactivity:**
  - Implements real-time features with **React.js**.
  - Examples in MediKart include live search filters, instant shopping cart updates, and notifications for order status.
  - Improves user satisfaction and makes the interface feel responsive.
- **Integration with Backend Services:**
  - Frontend communicates with backend to provide a dynamic user experience.
  - React components interact with **Node.js/Express APIs** to fetch product data from **MongoDB**, process orders, and manage user authentication.
  - Ensures MediKart is fully functional, reliable, and responsive.
- **Performance Optimization and Accessibility:**
  - Enhances speed and usability of the web application.
  - Uses React’s virtual DOM for efficient rendering and Bootstrap for responsive design.
  - Ensures accessibility for all users, including those with disabilities.
- **Business Impact:**
  - Improves user engagement, reduces bounce rates, and increases customer

satisfaction.

- A well-designed frontend in MediKart builds user trust and contributes to higher retention rates.

Conclusion:

Frontend development is more than visual design—it is about creating **user-centric, efficient, and accessible applications**. Through my **MERN stack training**, I gained hands-on experience building responsive interfaces with React.js, integrating them with backend services, and ensuring a smooth user experience in MediKart.

1.3Introduction to MERN Stack

During the web development with MERN STACK training, I learned the fundamentals of the **MERN stack** (MongoDB, Express, React, Node.js) and applied them to the **MediKart** project. The MERN stack enables building full-stack web applications using a single language, **JavaScript**, across frontend, backend, and database.

1.3.1 Overview of MERN (MongoDB, Express, React, Node)

- **MongoDB (Database Basics):**
  - A NoSQL database storing data in flexible, JSON-like documents.
  - Learned to create collections, insert documents, and perform queries.
  - Used in MediKart to manage users, products, orders, and addresses.
- **Express.js (Backend Basics):**
  - A Node.js framework for creating server-side applications and APIs.
  - Learned to define routes, handle requests, and send responses.
  - Used in MediKart to manage API endpoints for products, orders, cart, and user authentication.
- **React.js (Frontend):**
  - A JavaScript library for building dynamic and responsive user interfaces.
  - Learned to create components, manage state with hooks, and handle user interactions.
  - Used in MediKart for product listings, shopping cart, checkout, and order tracking.
- **Node.js (Runtime Basics):**
  - JavaScript runtime for executing code on the server side.

    - Learned to start a server, handle requests, and integrate with Express.
    - Enabled communication between MediKart frontend and MongoDB database.

1.3.2 Role of Each Component in Web Applications

- **MongoDB:** Stores and retrieves application data efficiently.
- **Express.js:** Handles backend logic, API routing, and server requests.
- **React.js:** Builds interactive UI components and manages frontend state.
- **Node.js:** Runs backend JavaScript code and connects the frontend with database.

1.3.3 React.js as the Frontend Framework

- Component-based architecture allows modular UI development.
- React Hooks (useState, useEffect) manage state and lifecycle events.
- MediKart's frontend uses React for:
    - Displaying products and categories dynamically.
    - Shopping cart and order placement functionalities.
    - User authentication, profile management, and responsive UI.

Summary:

The MERN stack provides a complete JavaScript-based solution for full-stack web development. Through this training, I gained practical experience in **frontend, backend, and database integration**, applying these concepts directly to the MediKart project to build a fully functional e-commerce platform.

1.4 Frontend Architecture and Component Planning

In modern web development, **frontend architecture** plays a crucial role in building maintainable, scalable, and efficient applications. During my training and development of **MediKart**, I followed a **component-based architecture** using **React.js**, which allowed the separation of concerns, reusability of code, and easier debugging.

Key Points of Frontend Architecture in MediKart

- **Component-Based Structure:**
    - Each UI element, such as Navbar, ProductCard, Cart, and CheckoutPage, was designed as a **separate React component**.
    - Components were **modular and reusable**, making it easy to update or expand

features in the future.

- **Page Organization:**
    - MediKart's pages were structured logically: Home, Product Category, Product Details, Cart, Checkout, My Orders, Seller Dashboard, and Admin sections.
    - This separation ensured that **navigation and state management** remained organized.
- **State Management:**
    - React hooks such as useState, useEffect, and useContext were used to manage **application state** efficiently.
    - Centralized state management via **AppContext** allowed different components to **share data** like user info, cart items, and product listings.
- **Routing and Navigation:**
    - **React Router** was implemented for seamless page transitions and route protection.
    - Public and protected routes ensured that **users and sellers** accessed only relevant pages.
- **UI Planning and Layout:**
    - Pages were designed with **Bootstrap** for responsiveness, ensuring compatibility with mobile, tablet, and desktop screens.
    - Components like Hero Sections, BestSeller listings, Category Cards, and Footer were carefully arranged to **enhance user experience**.
- **Integration with Backend APIs:**
    - Frontend components were designed to **consume APIs** developed in Express.js and Node.js.
    - Features like product listing, cart management, order placement, and user authentication relied on **dynamic API responses**.
- **Reusable Utilities:**
    - Utility functions for **form validation, API calls, and notifications** were abstracted to keep components clean and maintainable.

By following this architecture, MediKart achieved **a clean separation between presentation, business logic, and data management**, allowing for faster development, easier maintenance, and smoother user experience.

1.5 State Management with React Hooks

During my **web development training with the MERN stack at STEP GNDEC**, I learned how to manage application state efficiently in **MediKart** using **React Hooks**. State management is crucial in a full-stack web application to ensure dynamic updates, smooth user interactions, and seamless data flow across components.

Key aspects of state management I learned and applied in MediKart:

- **useState:**
    - Learned to manage **local state** within React components.
    - Applied it to **track form inputs**, **cart items**, and **quantity changes** dynamically.
    - Example: Updating the shopping cart in **Cart.jsx** as users add or remove products.
- **useEffect:**
    - Learned to handle **side effects** such as API calls and data fetching.
    - Applied it to fetch **product details, order history, and categories** from the backend dynamically.
    - Example: Updating the **MyOrders.jsx** page after placing an order to reflect real-time data.
- **useContext (AppContext):**
    - Learned to implement **global state management** without prop drilling.
    - Shared data like **logged-in user information, cart items, and authentication status** across multiple pages.
    - Enabled components like **Navbar**, **Cart**, and **CheckoutPage** to access and update global data efficiently.
- **Custom Hooks:**
    - Learned to create reusable hooks for API calls and other repeated logic.
    - Improved **code reusability**, **modularity**, and **maintainability** in MediKart.

1.6 User Interface & User Experience Design

A seamless **user interface (UI)** and smooth **user experience (UX)** are critical for any e-commerce application like **MediKart**, where users need to browse products, add items to the cart, and place orders efficiently.

UI Design Principles Applied in MediKart

- **Consistency:** Uniform color schemes, fonts, and buttons across all pages to ensure a professional look.
- **Simplicity:** Minimalistic design for easy navigation and faster access to key features like product search and checkout.
- **Responsiveness:** UI adapts to various screen sizes using **Bootstrap**, making MediKart accessible on desktops, tablets, and mobile devices.
- **Accessibility:** Ensured readable fonts, visible buttons, and clear navigation for all users.

UX Design Strategies Implemented

- **Intuitive Navigation:** Clear menu structure with categories, bestsellers, and search functionality for quick product discovery.
- **Interactive Components:** Product cards, shopping cart, and forms provide instant feedback to user actions.
- **Checkout Flow:** Step-by-step checkout process for adding addresses, selecting payment options, and confirming orders.
- **User Authentication:** Secure login and registration process to protect user data and enhance trust.
- **Feedback Mechanisms:** Success messages for order placement and alerts for invalid inputs in forms improve interaction clarity.

Tools and Technologies Used for UI/UX

- **React.js:** Built reusable components for product cards, navigation, forms, and dashboards.
- **Bootstrap:** Designed responsive layouts and ensured mobile-friendly views.
- **CSS & HTML:** Customized styling for branding, color themes, and layout adjustments.

1.7 Deployment Overview & Basic DevOps

As part of the **web development training with the MERN stack**, I gained practical experience in deploying and managing a full-stack web application through my project **MediKart**. This included integrating frontend, backend, and database functionalities, and using version control for collaborative development.

Key Learnings:

- **Version Control with GitHub:**
    - Learned to manage project code efficiently.

    - Created branches, committed changes, and merged updates for the MediKart project.
- **Frontend Deployment:**
    - Built the **React.js frontend** for production using npm build.
    - Integrated it with backend APIs to ensure all user actions (like browsing products, adding to cart, and placing orders) worked seamlessly.
- **Backend Deployment:**
    - Configured the **Node.js + Express server** to handle client requests.
    - Used environment variables to secure sensitive data like MongoDB connection URLs.
- **Database Integration:**
    - Connected the **MongoDB database** to manage users, products, addresses, and orders.
    - Ensured reliable data storage and retrieval for smooth operations on MediKart.
- **Testing and Debugging:**
    - Learned to monitor API responses and debug errors.
    - Ensured features like product listing, cart functionality, and order placement worked correctly.
- **Basic DevOps Understanding:**
    - Learned the workflow from development to deployment.
    - Practiced deploying updates, managing server configurations, and handling production-ready builds.

Outcome:

This part of the training provided hands-on exposure to **deploying a full-stack MERN application**, bridging the gap between development and live deployment, and gave practical experience in managing a real-world project like MediKart

1.8 Software, Tools, and Technologies Learned

During the training, I gained practical knowledge of **web development tools and technologies**, which were later applied to the **MediKart** project.

1.8.1 HTML & CSS – Structure and Styling

- Learned the basics of HTML to create webpage structures.
- Studied CSS for styling elements, layout design, and responsive techniques.

- Though not used in MediKart, this knowledge helped understand JSX and component styling in React.

1.8.2 JavaScript – Interactivity and Logic

- Learned JavaScript fundamentals for client-side interactivity and logic.
- Gained understanding of functions, events, DOM manipulation, and ES6 features.
- Applied concepts in MediKart via React.js for dynamic behavior.

1.8.3 Bootstrap – Responsive Design

- Learned how to use Bootstrap classes and components for quick responsive layouts.
- Did not directly use Bootstrap in MediKart, but helped in understanding modern CSS frameworks like Tailwind CSS.

1.8.4 Tailwind CSS – Styling and Responsive Design

- Used Tailwind CSS for building responsive, modern, and reusable UI components in MediKart.
- Styled pages including homepage, product listings, cart, checkout, and dashboards efficiently with utility classes.

1.8.5 React Components & JSX – Dynamic UI Development

- Built reusable components for product cards, navigation bars, modals, dashboards, and order management.
- Managed state using **React Hooks** (useState, useEffect) for live cart updates, order tracking, and user sessions.

1.8.6 GitHub – Version Control and Collaboration

- Managed project codebase, tracked changes, and maintained versions efficiently.
- Learned branching, committing, merging, and resolving conflicts for collaborative development.

1.8.7 VS Code, Node.js, Express.js, MongoDB, and Browser Developer Tools

- **VS Code**: IDE for coding, debugging, and project management.
- **Node.js**: Server-side runtime to handle requests and responses.
- **Express.js**: Built RESTful APIs for products, users, orders, and authentication.
- **MongoDB**: Stored and retrieved data for users, products, orders, and addresses.
- **Browser Developer Tools**: Debugged frontend functionality and network requests.

Outcome:

This combination of learned technologies and tools provided a solid foundation in **full-stack web development**, allowing me to successfully implement **MediKart**, a MERN stack application with modern frontend styling using **Tailwind CSS**.

1.9 Training Benefits and Learning Outcomes

The web development training with the MERN stack significantly contributed to building both **theoretical understanding** and **practical skills**, enabling me to successfully develop the **MediKart** project. The key benefits and learning outcomes are as follows:

- **Application of Theory to Practice**
  The training enabled me to convert theoretical knowledge of web development into practical implementation. Concepts of frontend-backend integration, API development, and database management were reinforced through hands-on coding.
- **Development Understanding**
  I gained a comprehensive understanding of the MERN stack workflow. This includes how React interacts with Node.js/Express APIs and MongoDB basics, forming a complete full-stack application like MediKart.
- **Problem-Solving and Logical Thinking**
  Developing MediKart required solving real-world challenges such as managing product data, handling orders, and implementing secure user authentication. This improved my problem-solving and logical thinking skills.
- **Project Structuring and Component Planning**
  The training emphasized planning frontend components, backend routes, and database schemas, which helped in developing MediKart in an organized and modular way.
- **Hands-On Database and Backend Experience**
  Working with Node.js, Express, and MongoDB provided experience in managing server-side logic, implementing APIs, and handling data storage efficiently.
- **Version Control and Collaboration**
  Learning GitHub taught me proper version control, enabling project tracking, code commits, and collaboration for better workflow management.

- **Confidence in Full-Stack Application Development**
  Completing MediKart provided the confidence to independently design, develop, and deploy full-stack web applications, bridging the gap between academic knowledge and professional-level implementation.

1.10 Report Structure Overview

The report is structured to provide a clear and comprehensive understanding of the **web development training with the MERN stack** and the development of the **MediKart project**. Each chapter focuses on a specific aspect of the training and project work:

- **Chapter 1 – Introduction:**
  Covers the background of the training, theoretical concepts, MERN stack fundamentals, software and tools learned, and key learning outcomes.
- **Chapter 2 – Training Work Undertaken:**
  Details the sequential learning process, methodology followed, and hands-on project development activities for MediKart. Includes weekly progress, component development, and backend/database integration.
- **Chapter 3 – Results and Discussion:**
  Presents the outcomes of the MediKart project, including frontend functionalities, UI interactions, backend operations, and database performance. Discusses challenges faced and solutions implemented during development.
- **Chapter 4 – Conclusion and Future Scope:**
  Summarizes the overall learning experience, project achievements, and potential enhancements or future developments for MediKart.
- **References and Appendix:**
  Includes the sources consulted during the training and additional information such as code snippets, diagrams, or configuration details related to MediKart.

This structure ensures a logical flow from **theoretical knowledge** to **practical application**, highlighting the skills and understanding gained through the **training**.

1.11 Summary

The web development training with the **MERN stack** provided a solid foundation in both **frontend and backend development**, along with database management and version control. Through this training:

- I gained theoretical knowledge and practical experience in **React.js, Node.js, Express.js, and MongoDB**.
- Learned to plan and develop **frontend components** and integrate them with backend APIs.
- Acquired skills in **state management, user interface design, and responsive layouts** using Tailwind CSS in MediKart.
- Built a complete **full-stack application** by combining frontend, backend, and database workflows.
- Learned to use **GitHub** for version control and collaborative project management.

Overall, the training bridged the gap between theory and practice, equipping me with the skills to develop a professional web application like **MediKart**, while understanding the complete **full-stack development lifecycle**.

CHAPTER 2

TRAINING WORK UNDERTAKEN

The training program at STEP GNDEC was a structured and professional experience that provided exposure to multiple aspects of web development using the **MERN stack**—from planning and frontend design to backend basics and database integration. The program revolved around the development of **MediKart**, an online medicine ordering platform, which allowed me to apply the concepts learned during training in a real-world project scenario.

Throughout the training, I worked on developing the project in a step-by-step manner, focusing on both frontend implementation with **React** and backend basics using **Node.js, Express.js, and MongoDB**.

The training was divided into **four weeks**, each covering specific skills and deliverables.

Key Learning Areas and Activities:

- **Version Control with GitHub:**
  - Learned to create repositories, manage branches, and commit changes efficiently.
  - Practiced collaborative workflows to track code changes and maintain project history.
- **Frontend Basics:**
  - Gained knowledge in HTML, CSS, JavaScript, and Bootstrap to understand webpage structure, styling, and interactivity.
  - These fundamentals helped in understanding React component development for MediKart.
- **Introduction to MERN Stack:**
  - Explored React.js to build dynamic and reusable frontend components like product listing pages, shopping cart, and checkout pages.
  - Learned basics of Node.js, Express.js, and MongoDB to handle server-side operations and store data for users, products, and orders.
  - Understood how frontend communicates with backend via APIs.
- **Development Environment Setup:**
  - Installed VS Code, Node.js, npm, and configured the development environment for MediKart.

    - Set up the React frontend project and created the basic folder structure for the application.

2.1 Week 1 – Introduction to Project and Basic Frontend Setup

The first week focused on understanding the objectives of MediKart, getting familiar with web development fundamentals, and setting up the development environment.

Key Activities and Learning Areas:

- **Project Understanding:**
    - Studied the MediKart project scope, an online medicine ordering platform.
    - Learned the workflow of an e-commerce website including user registration, product listing, cart management, and order placement.
    - Analyzed requirements for frontend and backend to plan the initial development steps.
- **Version Control & GitHub:**
    - Created repositories and organized the initial folder structure.
    - Learned to manage branches, commit changes efficiently, and resolve merge conflicts.
    - Practiced collaborative workflows for maintaining project history.
- **Frontend Fundamentals:**
    - Studied HTML, CSS, JavaScript, and Bootstrap for webpage structure, styling, and interactivity.
    - Practiced creating layouts, forms, buttons, and navigation bars.
- **Development Environment Setup:**
    - Installed VS Code, Node.js, and npm for frontend and backend development.
    - Installed essential npm packages like React, Axios, and React Router DOM.
    - Initialized the React frontend project and set up the basic folder structure.
- **Initial Planning:**
    - Sketched wireframes and planned components like Navbar, Product Card, Cart, and Checkout pages.
    - Outlined basic API endpoints for user authentication, product fetching, and order placement.

Outcome:

- Gained strong understanding of version control and project management with GitHub.
- Learned frontend basics and how to apply them in React.
- Fully set up development environment ready for component development.
- Clear roadmap for implementing MediKart's frontend and backend features.

2.2 Week 2 – React Fundamentals and Basic Backend Integration

The second week focused on enhancing frontend development skills using **React.js** and gaining foundational knowledge of backend technologies like **Node.js, Express.js, and MongoDB**.

Key Learning Areas and Activities:

- **React.js Fundamentals:**
    - Developed reusable components like Navbar, ProductCard, Hero Section, Footer, and Category components.
    - Learned JSX syntax for dynamic content rendering.
    - Implemented state management with React Hooks (useState, useEffect) for shopping cart, product listings, and order updates.
    - Practiced props passing and conditional rendering to manage data flow between parent and child components.
- **Routing and Navigation:**
    - Implemented React Router for pages like Home, Product Category, Product Details, Cart, Checkout, Add Address, and My Orders.
    - Configured dynamic routes to display product details based on product IDs.
    - Ensured smooth navigation and state persistence across pages.
- **Backend Basics:**
    - Explored Node.js as a server-side runtime.
    - Learned Express.js to create APIs for products, user authentication, carts, and orders.
    - Studied MongoDB for storing users, products, addresses, orders, and cart items.
- **Frontend-Backend Integration:**
    - Connected React frontend with backend APIs to fetch and display data dynamically.
    - Linked forms like Add Address and Checkout to backend endpoints.
    - Implemented basic user authentication using JWT tokens.

    - Managed API responses and error handling using Axios.

- **Development Tools and Practices:**
    - Continued using GitHub for version control.
    - Used VS Code for React and Node.js development.

Outcome:

- Gained strong understanding of React component development, state management, and basic backend integration.
- Able to develop interactive and dynamic features for MediKart.

2.3 Week 3 – Advanced Frontend Development and Backend Connectivity

The third week focused on building dynamic frontend components and connecting them with backend APIs to enhance the MediKart application.

Key Learning Areas and Activities:

- **Advanced React Components:**
    - Developed Product Card, Cart, Checkout, Navbar, and User Dashboard components.
    - Implemented dynamic rendering for products, categories, and orders fetched from backend APIs.
    - Managed state across components using React Hooks and Context API.
- **Form Implementation and Validation:**
    - Built Add Address and Checkout forms with input validations.
    - Linked forms to backend endpoints using Axios.
    - Ensured secure handling of user data during login, signup, and order placement.
- **Interactive Features and UX Enhancements:**
    - Enabled live cart updates and dynamic order confirmations.
    - Improved checkout flow for better usability.
    - Tested component interactions and ensured state persistence across pages.
- **Development Tools and Practices:**
    - Used GitHub for version control and project management.
    - Debugged components and API responses using VS Code and browser tools.

Outcome:

- Successfully created interactive and dynamic frontend components.
- Connected React components to backend APIs and implemented functional features like cart management and order placement.

2.4 Week 4 – Backend Finalization and Database Integration

The fourth week focused on completing the backend, integrating MongoDB, and making MediKart fully functional.

Key Learning Areas and Activities:

- **Backend Completion:**
  - Finalized Node.js and Express.js backend setup for MediKart.
  - Implemented API routes for user authentication, product listing, cart management, and order placement.
  - Ensured proper communication between frontend and backend.
- **Database Integration (MongoDB):**
  - Connected the backend to MongoDB for storing users, products, carts, and orders.
  - Verified database connections and ensured data consistency for all features.
- **Frontend-Backend Connectivity:**
  - Linked React components to backend APIs.
  - Tested login, signup, cart updates, order placement, and order history features.
  - Managed API responses and errors for seamless user experience.
- **Development Tools and Practices:**
  - Continued using GitHub for version control.
  - Used VS Code and browser developer tools for testing, debugging, and optimizing functionality.
  - Organized project structure for maintainability.

Outcome:

- MediKart became a fully functional MERN stack application.
- Users could register, browse products, add items to cart, place orders, and view order history.
- The application successfully integrated frontend, backend, and database components.

CHAPTER 3

RESULTS AND DISCUSSION

This chapter presents the overall results, implementation outcomes, and discussions based on the development of the *MediKart* project. The project, created during my industrial training at STEP GNDEC, aimed to design and develop a full-stack online medicine ordering platform using the **MERN Stack** — MongoDB, Express.js, React.js, and Node.js.

The system was built with the goal of providing users with an easy and secure way to order medicines online while enabling sellers to manage their product listings efficiently. The project involved multiple development phases — frontend design, backend API creation, database integration, and complete testing.

This chapter elaborates on the **functional results, UI design, backend performance, and learning outcomes** derived during development.

3.1 Functional Outcomes of MediKart Application

The MediKart application successfully met the objectives defined at the beginning of the training. It offers both **user** and **seller** functionalities, ensuring a complete e-commerce experience tailored for the medical and healthcare domain.

Main Functional Modules:

1. User Module:

- **User Authentication and Authorization:**
  Implemented secure signup and login functionality using JWT (JSON Web Tokens). Each user session is authenticated using tokens to ensure security and data privacy.
- **Product Browsing and Searching:**
  Users can explore the list of available medicines fetched dynamically from the MongoDB database. Products are categorized for better accessibility.
- **Cart and Checkout System:**
  The cart allows users to add, update, or remove items dynamically. Data persistence ensures that cart items remain intact during navigation.

At checkout, users can add delivery addresses and confirm their orders.

- **Order Placement and Tracking:**
  Once an order is placed, it gets stored in the database and can be viewed in the user's **My Orders** section.

2. Seller Module:

- **Seller Login and Authentication:**
  Sellers log in securely through their own portal using JWT-based authentication.
- **Product Management:**
  Sellers can add new medicines, edit existing details, and delete products when out of stock. All product data is stored in MongoDB through Express.js APIs.
- **Order Management:**
  Sellers can view all customer orders related to their products, helping them manage deliveries and inventory efficiently.

3. Application-Wide Features:

- **Dynamic API Integration:**
  The frontend communicates with the backend using Axios for API calls. Each component fetches and displays real-time data from the database.
- **Data Validation:**
  All user inputs (in forms like registration, address, and product addition) are validated on both frontend and backend.
- **Error Handling:**
  Proper error messages and toasts are shown for invalid inputs, authentication errors, or network failures.

Result:

By the completion of the project, the MediKart application functioned as a full-stack e-commerce platform where users could browse, purchase, and track their medicines while sellers could efficiently manage their products and orders.

3.2 UI Components and Interaction Flow

The **frontend of MediKart** was built using **React.js** for dynamic and modular rendering, **Bootstrap** for responsive design, and **React Router DOM** for smooth, single-page application navigation. The goal was to create an **intuitive and engaging interface** for both users and sellers, ensuring that every action is easy to perform and data is seamlessly updated through backend APIs.

The following subsections describe each major UI component in **logical order of user interaction**, along with recommended screenshot placements.

---

1. Navigation and Layout

The **layout components** define the overall structure of the application and are consistent across all pages.

- **Navbar:**
  - Provides quick access to **Home, Products, Cart, My Orders**, and **Login/Signup** pages.
  - Dynamically updates based on login status:
    - Shows **user options** if a user is logged in.
    - Shows **seller options** if a seller is logged in.
  - Ensures a consistent navigation experience throughout the site.

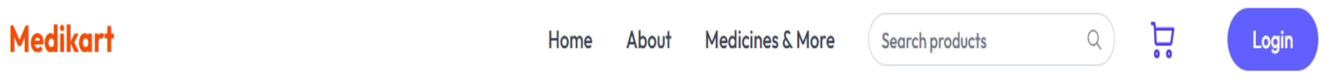
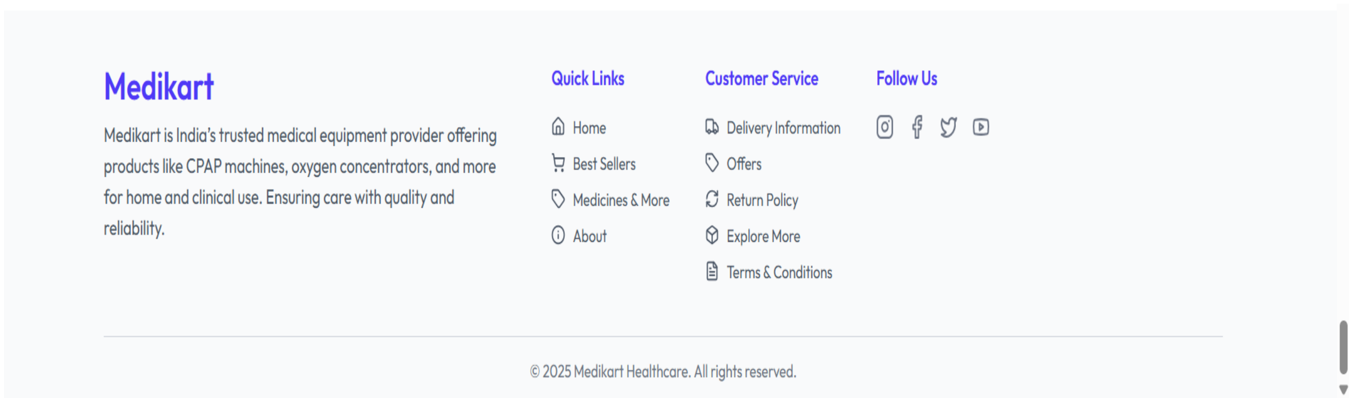


***Figure 1.1** –Navbar*

- **Footer:**
  - Displays **contact information, newsletter subscription**, and **policy links**.
  - Provides additional navigation for users who scroll to the bottom of the page.


Medikart

Medikart is India's trusted medical equipment provider offering products like CPAP machines, oxygen concentrators, and more for home and clinical use. Ensuring care with quality and reliability.

Quick Links
- Home
- Best Sellers
- Medicines & More
- About

Customer Service
- Delivery Information
- Offers
- Return Policy
- Explore More
- Terms & Conditions

Follow Us

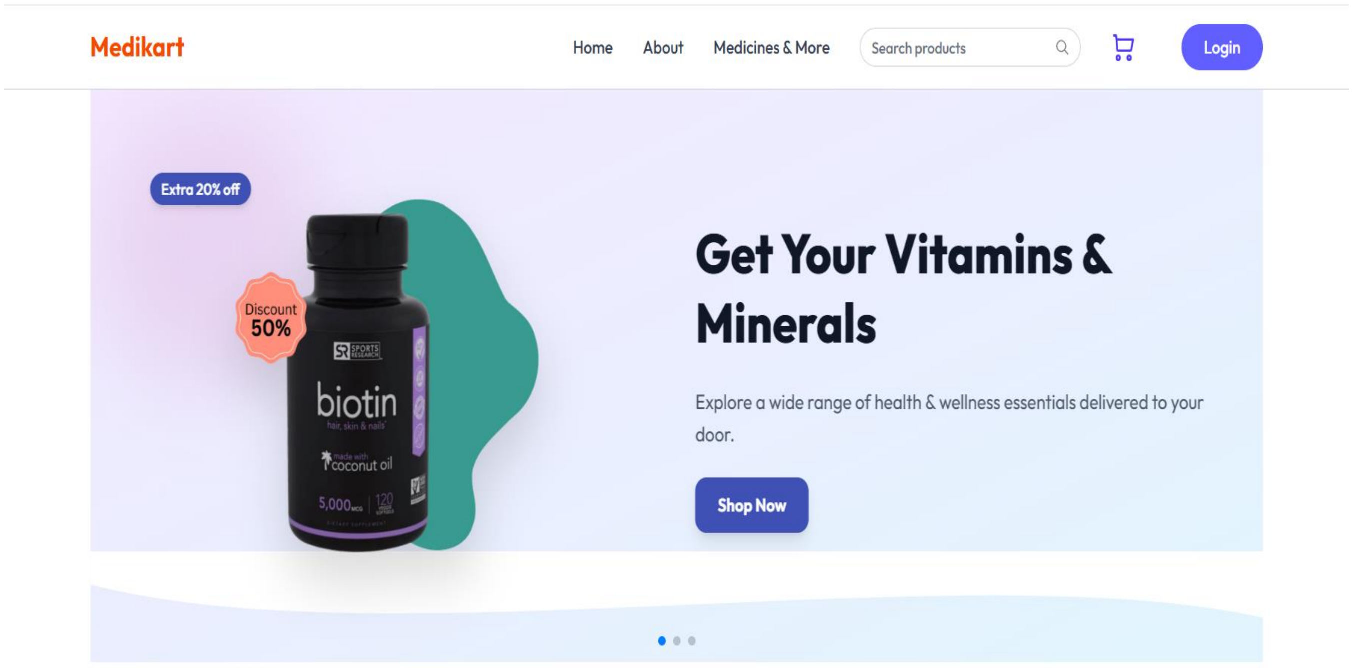
© 2025 Medikart Healthcare. All rights reserved.

Figure1.2*- Footer*

2. Home Page

The **Home Page** serves as the **first impression** of the platform and introduces users to the services offered.

- **Hero Section:**
    - Highlights the platform with promotional banners and call-to-action buttons like *"Shop Now"*.
    - Sets the visual theme for the website.



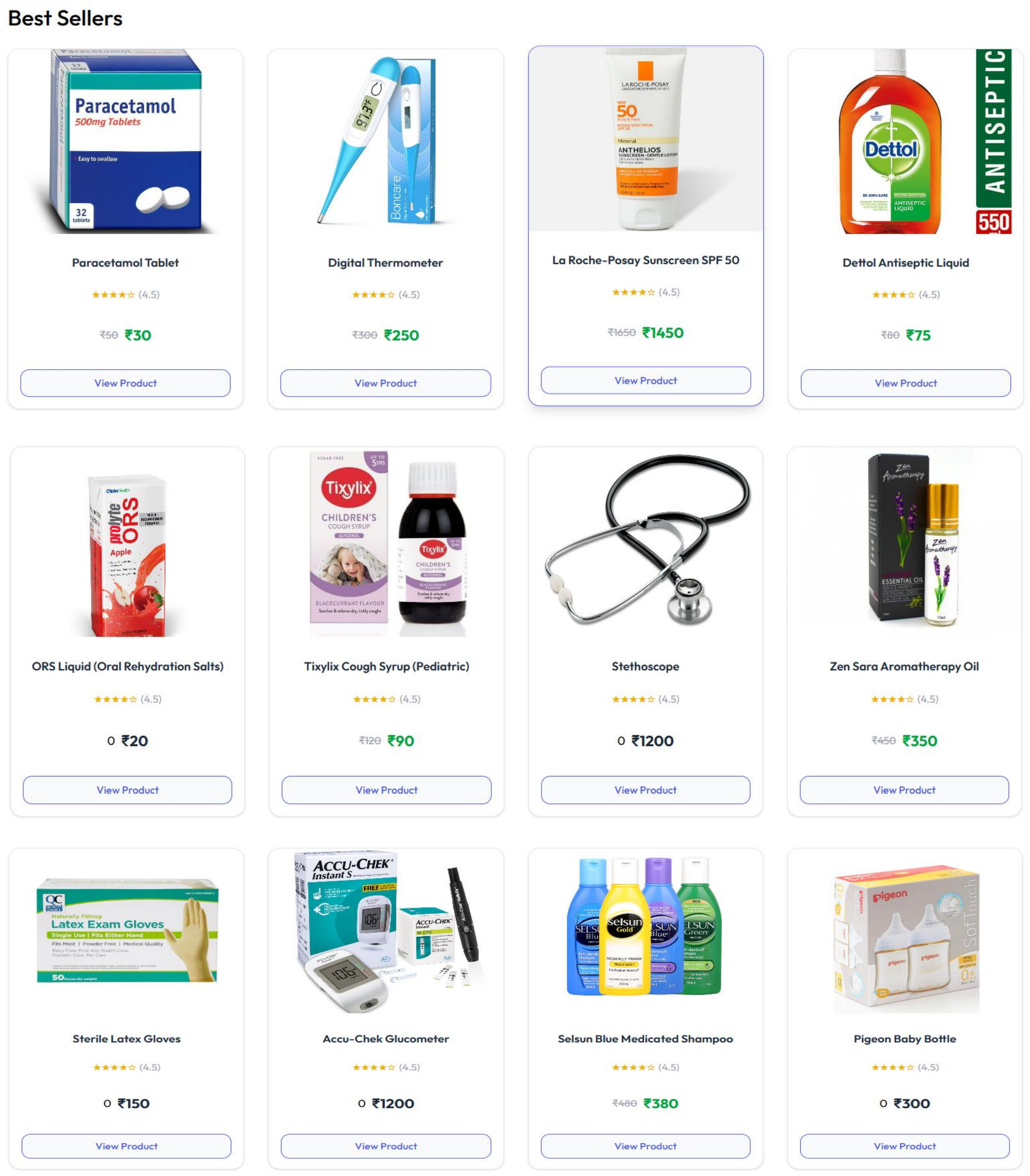
Get Your Vitamins & Minerals

Explore a wide range of health & wellness essentials delivered to your door.

Figure 1.3 *–Hero Section*

Best Seller Section:

- Showcases popular medicines with **real-time data fetched from MongoDB**.
- Encourages users to quickly access frequently purchased items.



Best Sellers

Paracetamol Tablet
★★★★☆ (4.5)
~~₹50~~ ₹30

Digital Thermometer
★★★★☆ (4.5)
~~₹300~~ ₹250

La Roche-Posay Sunscreen SPF 50
★★★★☆ (4.5)
~~₹1650~~ ₹1450

Dettol Antiseptic Liquid
★★★★☆ (4.5)
~~₹80~~ ₹75

ORS Liquid (Oral Rehydration Salts)
★★★★☆ (4.5)
o ₹20

Tixylix Cough Syrup (Pediatric)
★★★★☆ (4.5)
~~₹120~~ ₹90

Stethoscope
★★★★☆ (4.5)
o ₹1200

Zen Sara Aromatherapy Oil
★★★★☆ (4.5)
~~₹450~~ ₹350

Sterile Latex Gloves
★★★★☆ (4.5)
o ₹150

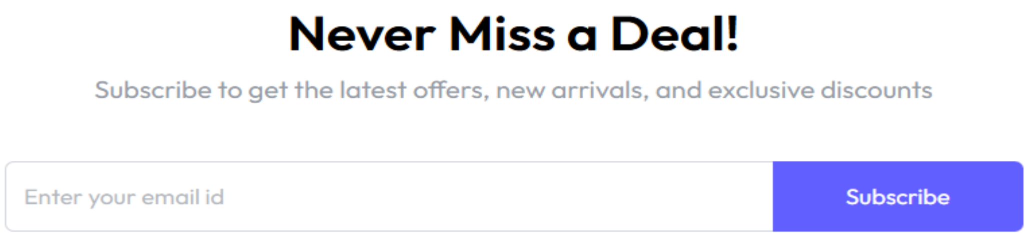
Accu-Chek Glucometer
★★★★☆ (4.5)
o ₹1200

Selsun Blue Medicated Shampoo
★★★★☆ (4.5)
~~₹480~~ ₹380

Pigeon Baby Bottle
★★★★☆ (4.5)
o ₹300

Figure 1.4- *Best Seller Section*

- **Newsletter Subscription Section:**
  - This section of the website is designed to encourage users to subscribe to the platform's newsletter.

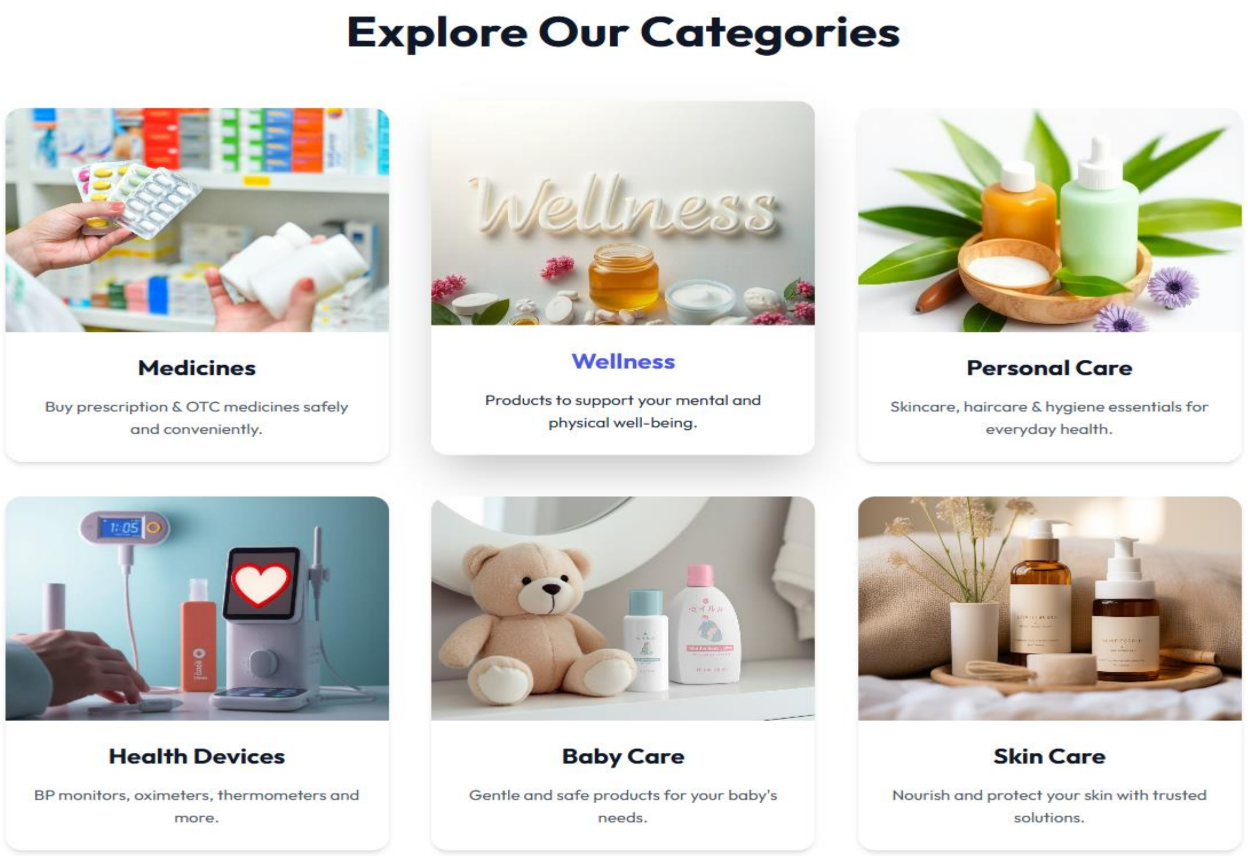


Never Miss a Deal!

Subscribe to get the latest offers, new arrivals, and exclusive discounts

Figure 1.5 *- Newsletter Subscription Section*

- **Category Section:**
  - Organizes products into categories such as Tablets, Syrups, Supplements, and Personal Care.
  - Allows users to filter products easily based on type.



Explore Our Categories

Medicines

Buy prescription & OTC medicines safely and conveniently.

Wellness

Products to support your mental and physical well-being.

Personal Care

Skincare, haircare & hygiene essentials for everyday health.

Health Devices

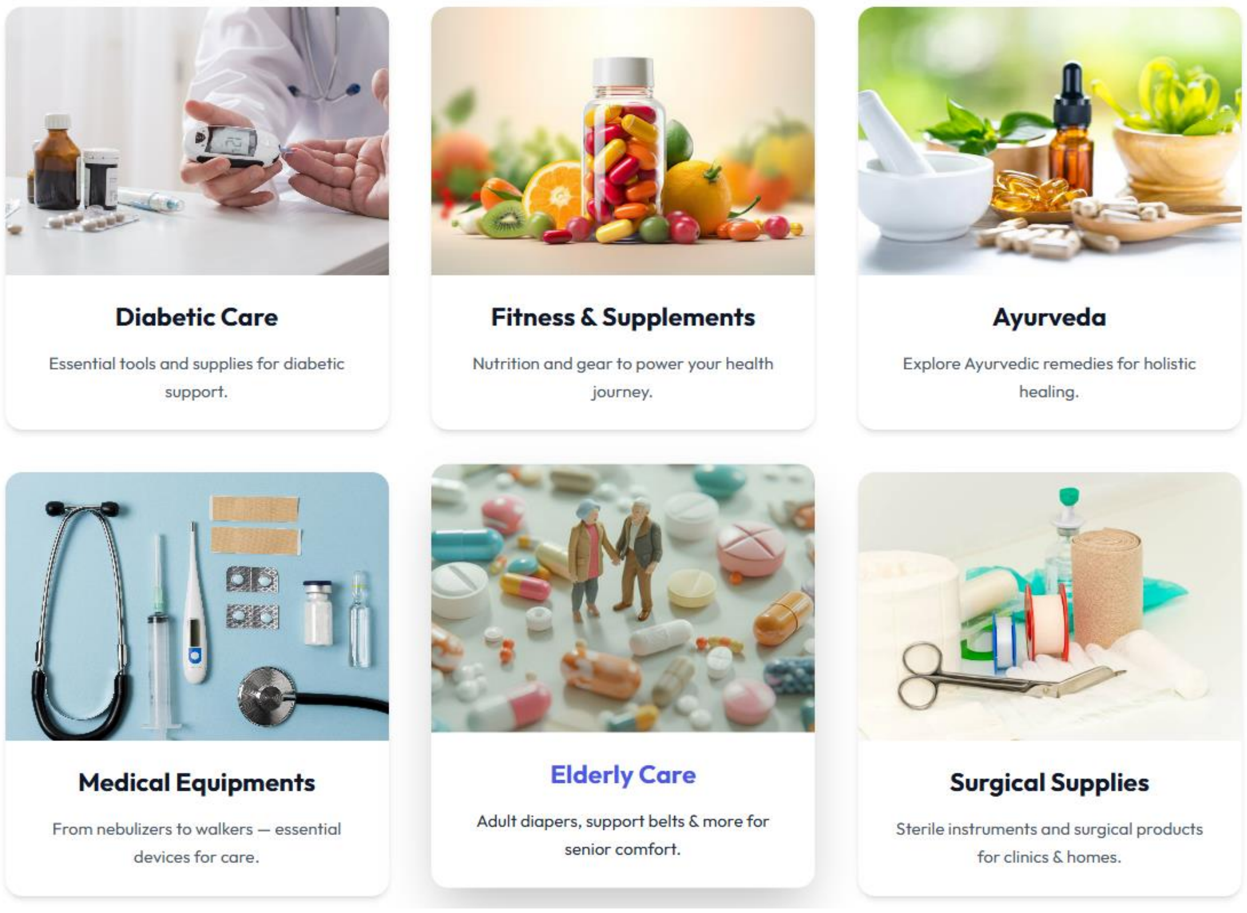
BP monitors, oximeters, thermometers and more.

Baby Care

Gentle and safe products for your baby's needs.

Skin Care

Nourish and protect your skin with trusted solutions.


Diabetic Care

Essential tools and supplies for diabetic support.

Fitness & Supplements

Nutrition and gear to power your health journey.

Ayurveda

Explore Ayurvedic remedies for holistic healing.

Medical Equipments

From nebulizers to walkers — essential devices for care.

Elderly Care

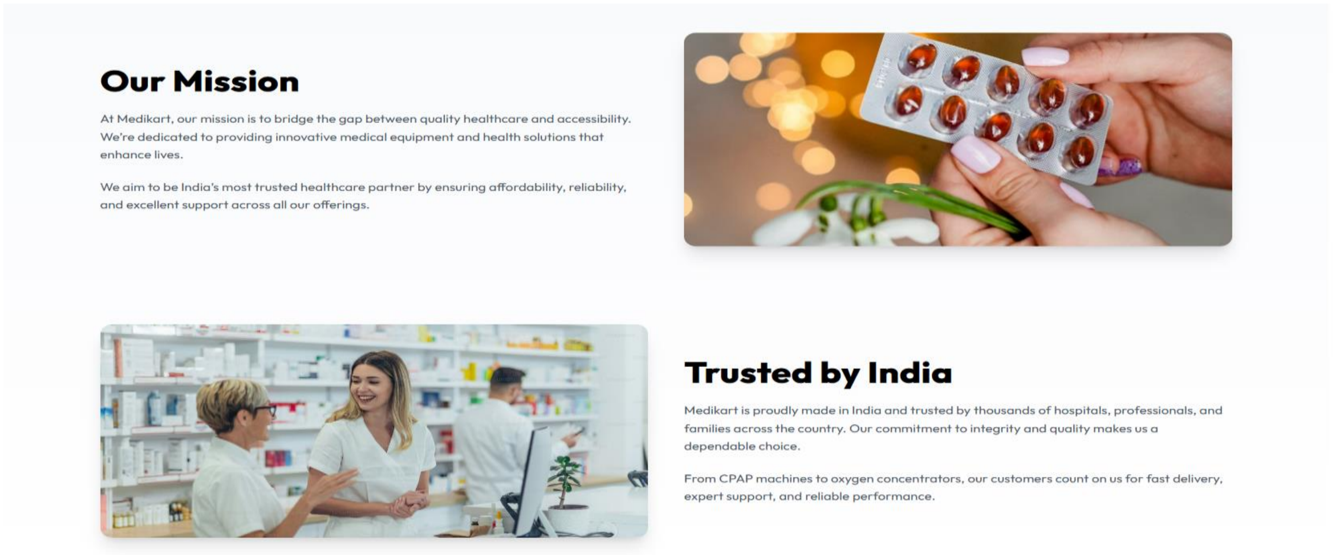
Adult diapers, support belts & more for senior comfort.

Surgical Supplies

Sterile instruments and surgical products for clinics & homes.

Figure 1.6: *category section*

- **Feature & Mission Sections:**
    - Communicates the core **values, safety standards**, and benefits of using MediKart.


Our Mission

At Medikart, our mission is to bridge the gap between quality healthcare and accessibility. We're dedicated to providing innovative medical equipment and health solutions that enhance lives.

We aim to be India's most trusted healthcare partner by ensuring affordability, reliability, and excellent support across all our offerings.

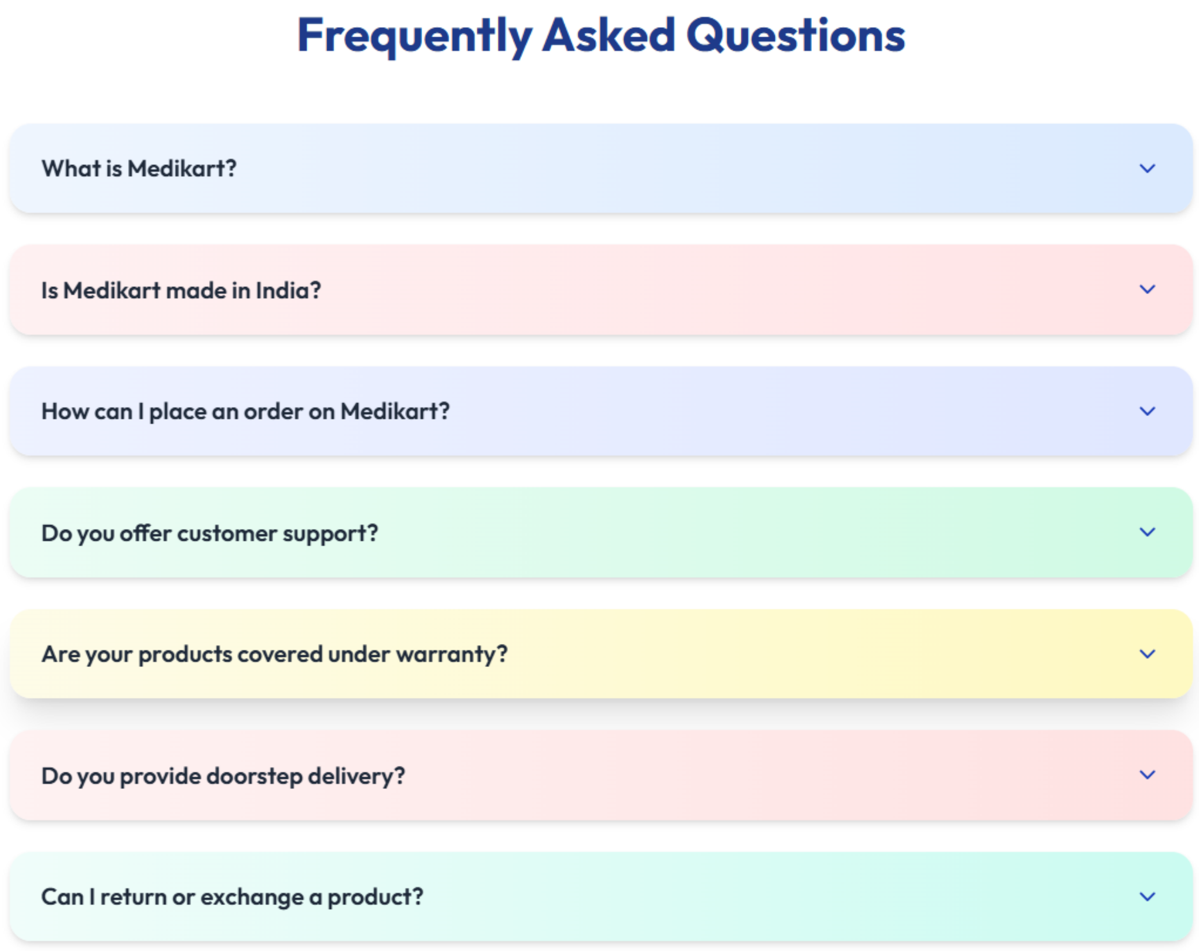
Trusted by India

Medikart is proudly made in India and trusted by thousands of hospitals, professionals, and families across the country. Our commitment to integrity and quality makes us a dependable choice.

From CPAP machines to oxygen concentrators, our customers count on us for fast delivery, expert support, and reliable performance.

Figure 1.7- *Feature & Mission section*

- **FAQ (Frequently Asked Questions) Section:**
  - The FAQ section is intended to address common queries related to the system, provide guidance to users, and improve overall usability and understanding of the application.



Frequently Asked Questions

What is Medikart?

Is Medikart made in India?

How can I place an order on Medikart?

Do you offer customer support?

Are your products covered under warranty?

Do you provide doorstep delivery?

Can I return or exchange a product?

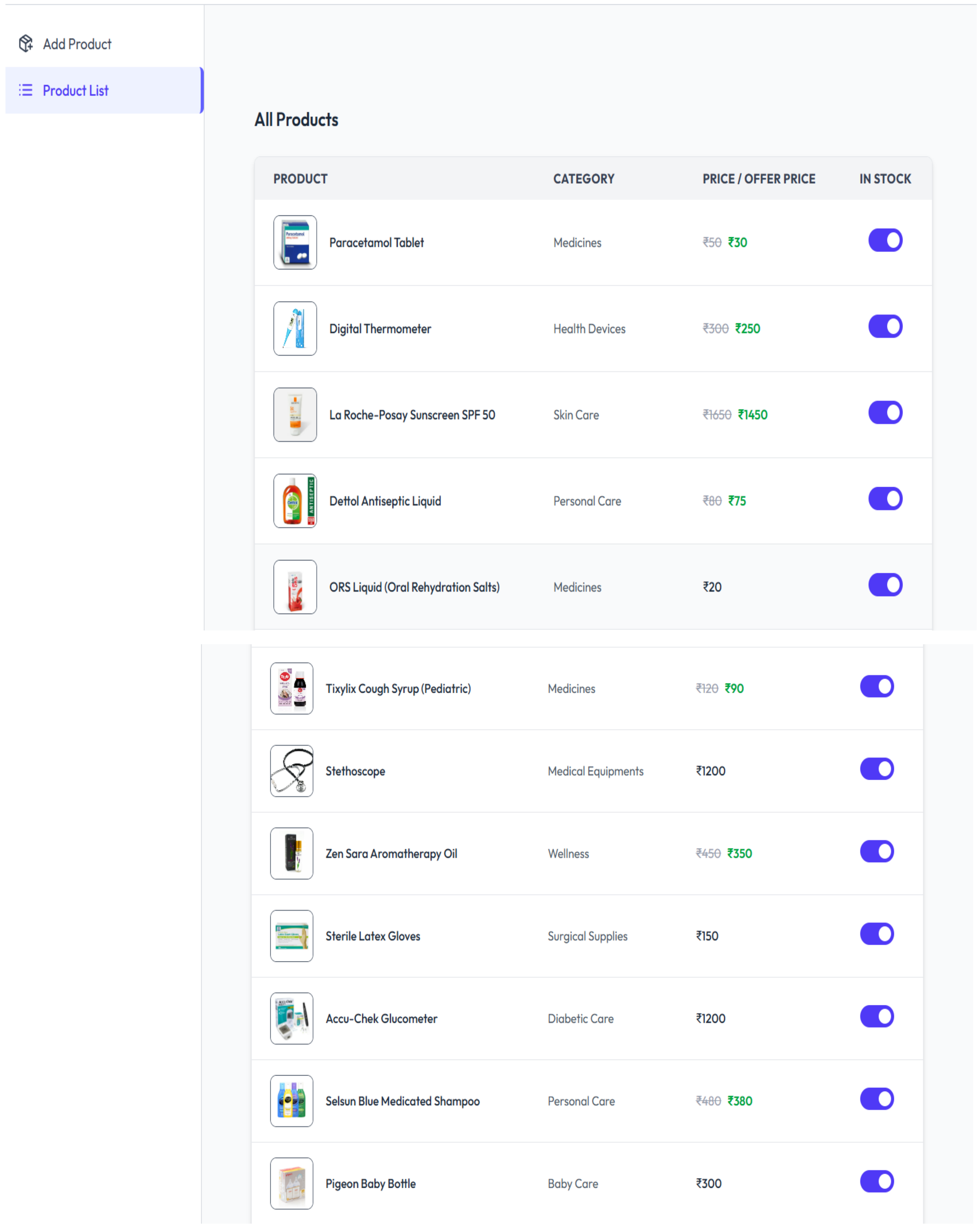
Figure 1.8- *Frequently Asked Questions Section*

---

3. Product Pages

These pages are central to the shopping experience, combining frontend display and backend data interaction.

- **Product Listing Page:**
  - Dynamically fetches all products from the backend API using **Axios**.
  - Displays products in a **grid layout** using the reusable ProductCard component.
  - Allows **sorting and searching** for better accessibility.

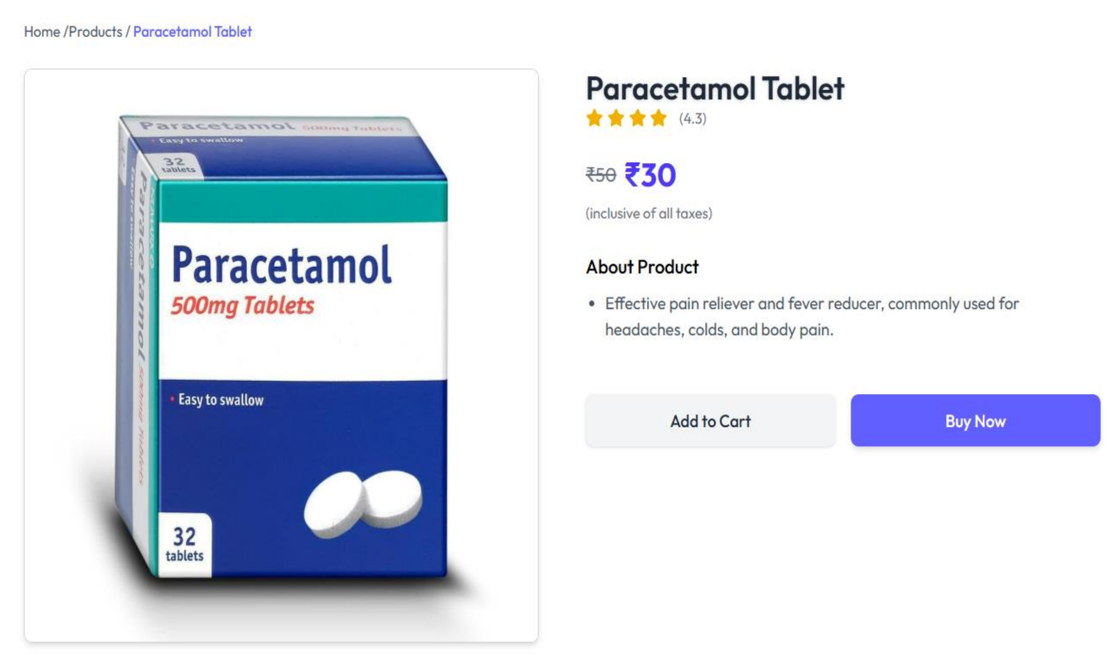


All Products

PRODUCT	CATEGORY	PRICE / OFFER PRICE	IN STOCK
Paracetamol Tablet	Medicines	~~₹50~~ ₹30	✓
Digital Thermometer	Health Devices	~~₹300~~ ₹250	✓
La Roche-Posay Sunscreen SPF 50	Skin Care	~~₹1650~~ ₹1450	✓
Dettol Antiseptic Liquid	Personal Care	~~₹80~~ ₹75	✓
ORS Liquid (Oral Rehydration Salts)	Medicines	₹20	✓
Tixylix Cough Syrup (Pediatric)	Medicines	~~₹120~~ ₹90	✓
Stethoscope	Medical Equipments	₹1200	✓
Zen Sara Aromatherapy Oil	Wellness	~~₹450~~ ₹350	✓
Sterile Latex Gloves	Surgical Supplies	₹150	✓
Accu-Chek Glucometer	Diabetic Care	₹1200	✓
Selsun Blue Medicated Shampoo	Personal Care	~~₹480~~ ₹380	✓
Pigeon Baby Bottle	Baby Care	₹300	✓

***Figure 2.1** – product list*

- **Product Details Page:**
  - Shows detailed information about a selected product:
    - Product images
    - Name and description
    - Price and available quantity
  - Includes an **Add to Cart** button, which updates the cart in real time.



Home /Products / Paracetamol Tablet

Paracetamol Tablet

(4.3)

~~₹50~~ ₹30

(inclusive of all taxes)

About Product

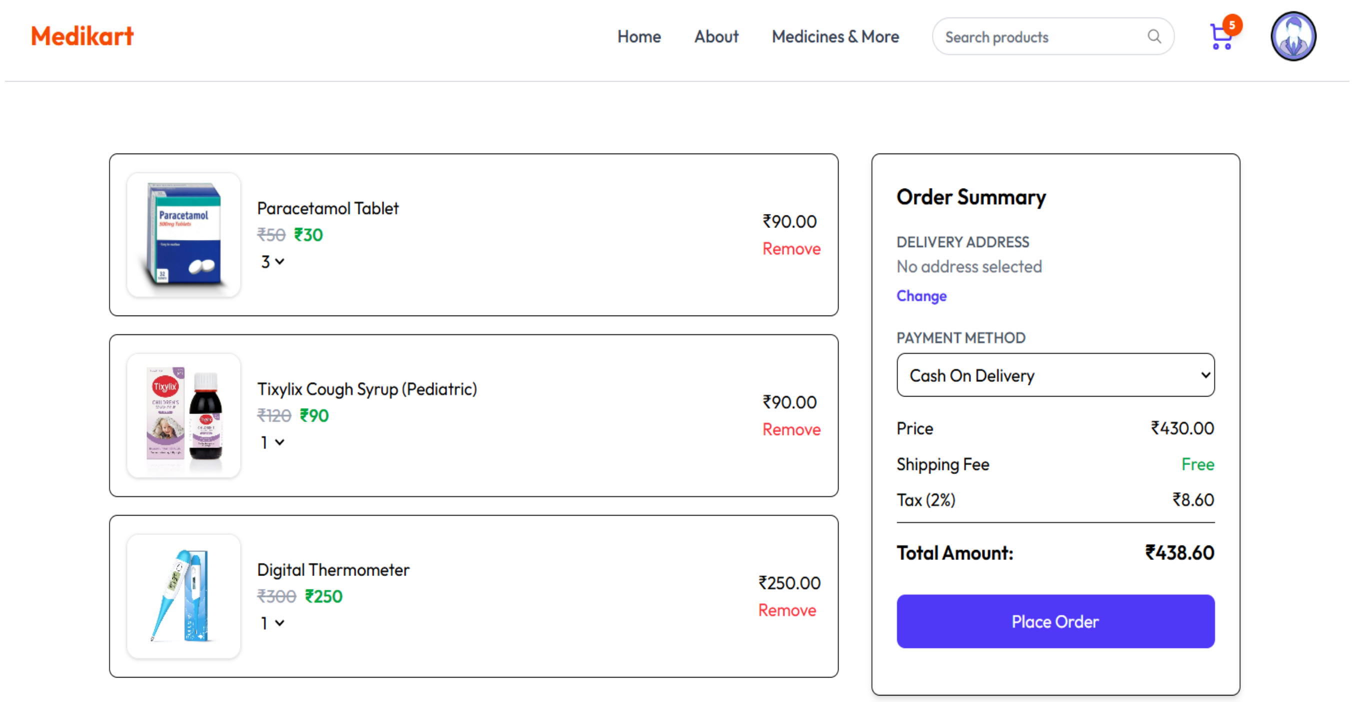
- Effective pain reliever and fever reducer, commonly used for headaches, colds, and body pain.

Figure 2.2- *Product Details page*

4. Cart and Checkout

The **Cart and Checkout system** integrates user selections with backend data storage to maintain order integrity.

- **Cart Page:**
  - Displays selected products with quantity adjustment options and total price updates dynamically.
  - Users can remove items before proceeding to checkout.


Medikart

Home About Medicines & More Search products

Paracetamol Tablet ~~₹50~~ ₹30 3	₹90.00	Remove
Tixylix Cough Syrup (Pediatric) ~~₹120~~ ₹90 1	₹90.00	Remove
Digital Thermometer ~~₹300~~ ₹250 1	₹250.00	Remove

Order Summary

DELIVERY ADDRESS
No address selected
Change

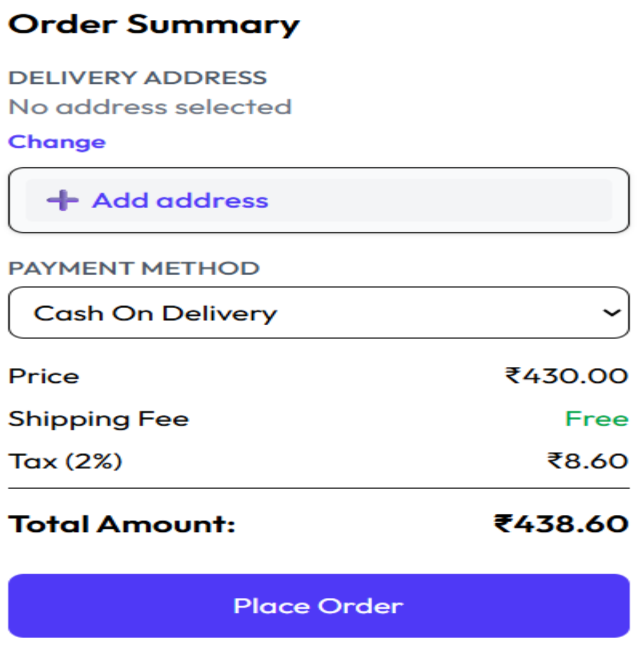
PAYMENT METHOD
Cash On Delivery

Price	₹430.00
Shipping Fee	Free
Tax (2%)	₹8.60
Total Amount:	₹438.60

Place Order

Figure 3.1- cart page

- **Add Address Page:**
    - Allows users to save new addresses directly into the database using backend APIs.
    - Ensures smooth integration between the frontend form and the backend server.


Order Summary

DELIVERY ADDRESS
No address selected
Change

Add address

PAYMENT METHOD
Cash On Delivery

Price	₹430.00
Shipping Fee	Free
Tax (2%)	₹8.60
Total Amount:	₹438.60

Place Order

Figure 4.1- Add address

When clicking on add address button :

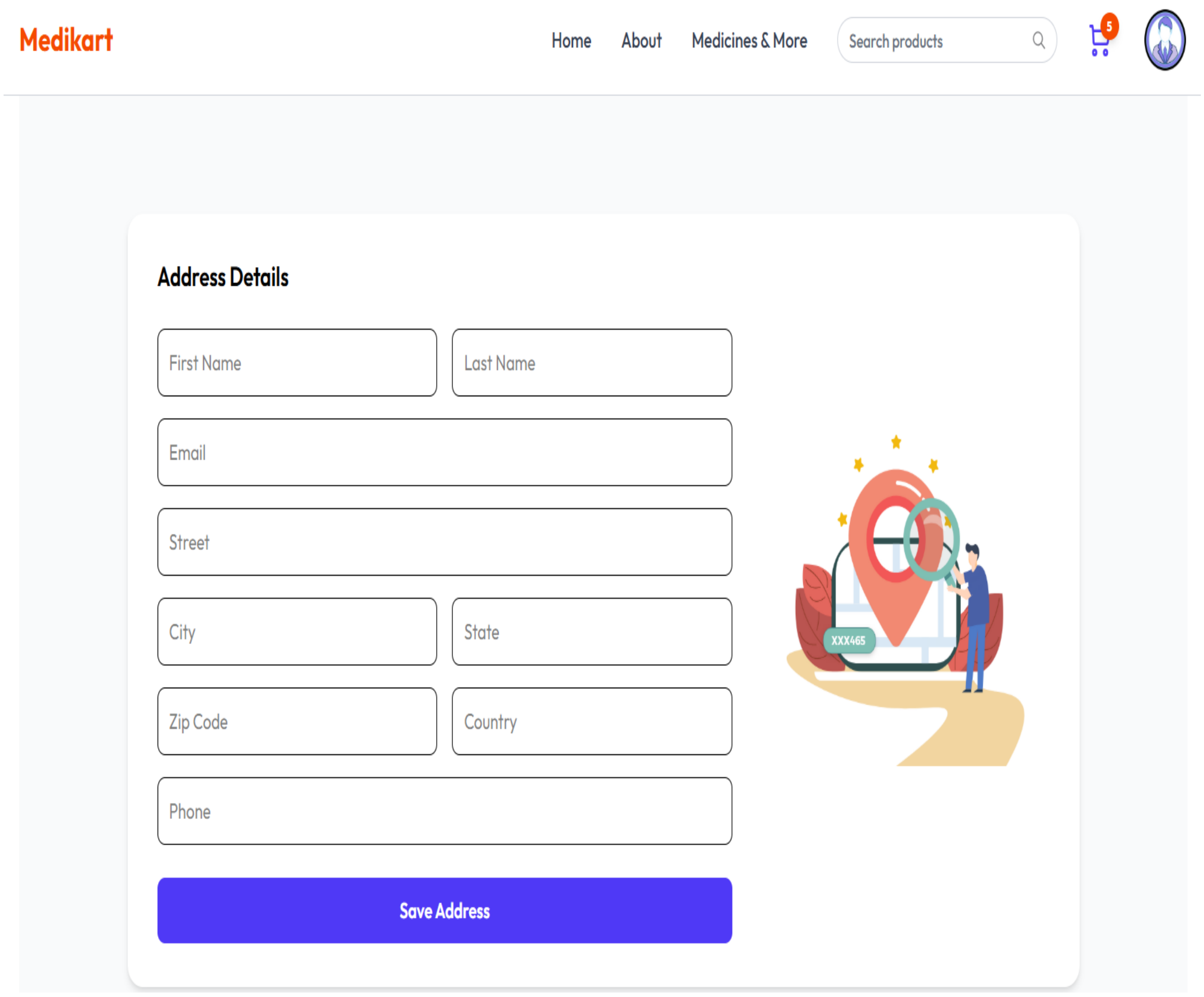
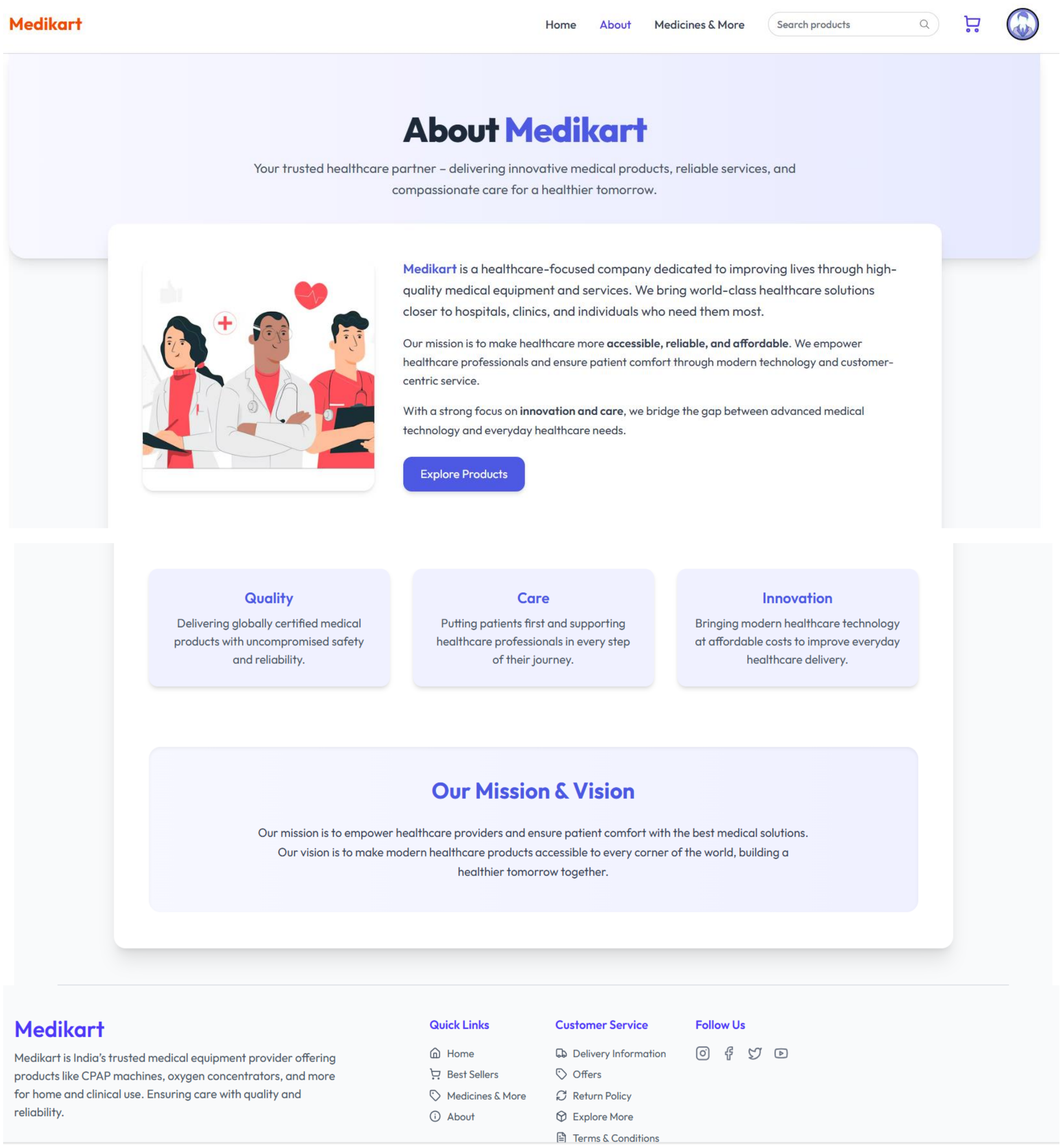


Figure 4.2: *Add address Page*

About Page

- Provides information about MediKart, its vision, services, and team.
- May include contact details and links to social media.



About Medikart

Your trusted healthcare partner – delivering innovative medical products, reliable services, and compassionate care for a healthier tomorrow.

Medikart is a healthcare-focused company dedicated to improving lives through high-quality medical equipment and services. We bring world-class healthcare solutions closer to hospitals, clinics, and individuals who need them most.

Our mission is to make healthcare more **accessible, reliable, and affordable**. We empower healthcare professionals and ensure patient comfort through modern technology and customer-centric service.

With a strong focus on **innovation and care**, we bridge the gap between advanced medical technology and everyday healthcare needs.

Explore Products

Quality

Delivering globally certified medical products with uncompromised safety and reliability.

Care

Putting patients first and supporting healthcare professionals in every step of their journey.

Innovation

Bringing modern healthcare technology at affordable costs to improve everyday healthcare delivery.

Our Mission & Vision

Our mission is to empower healthcare providers and ensure patient comfort with the best medical solutions. Our vision is to make modern healthcare products accessible to every corner of the world, building a healthier tomorrow together.

Medikart

Medikart is India's trusted medical equipment provider offering products like CPAP machines, oxygen concentrators, and more for home and clinical use. Ensuring care with quality and reliability.

Figure 5.1- *About page*

Medicines & More

- Displays all medicines in categories like Medicines, Skin Care, Health Care and Personal Care etc.

- Users can search and filter products by name, type, or price.
- Clicking a product opens its details page for more information.

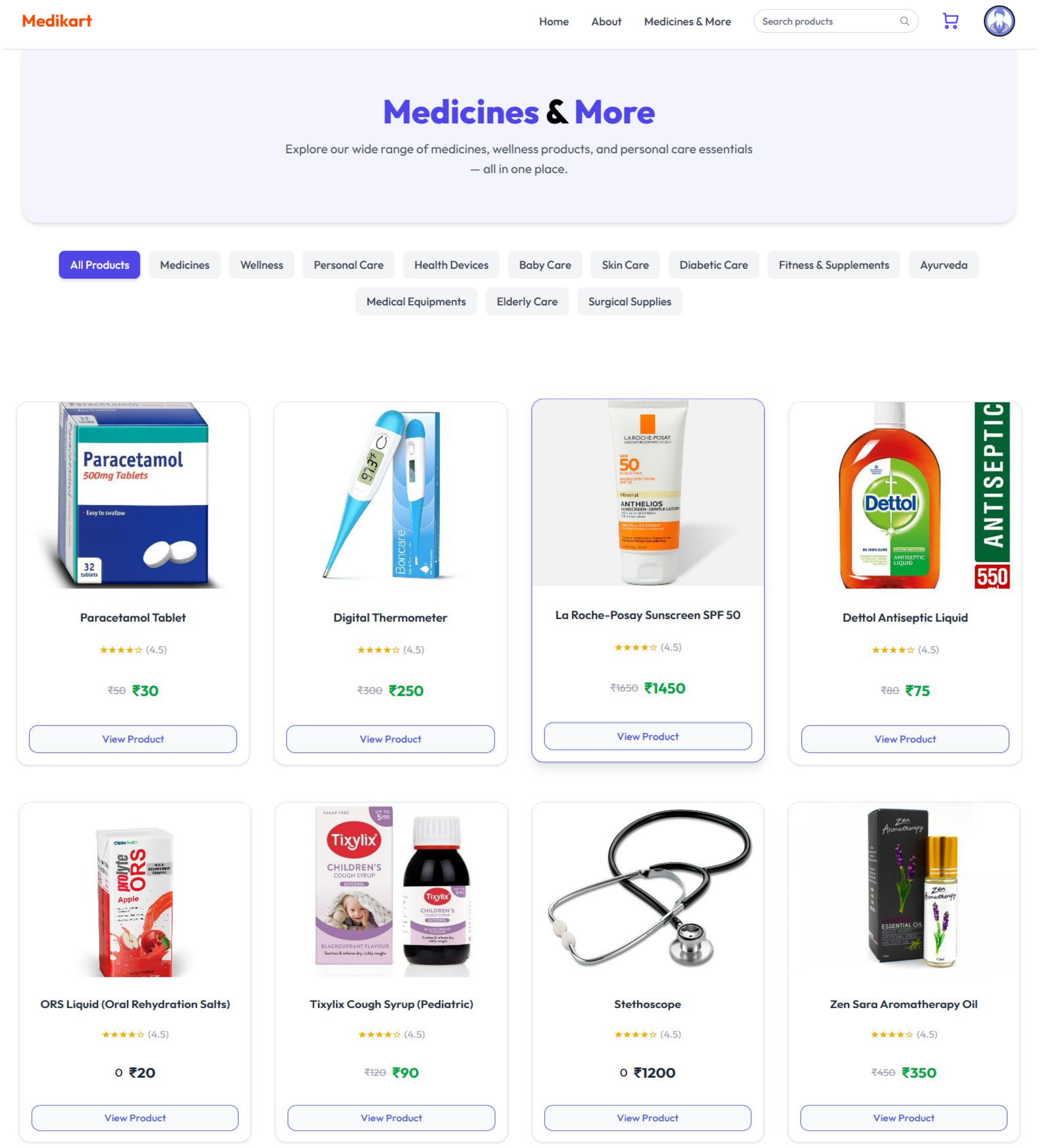
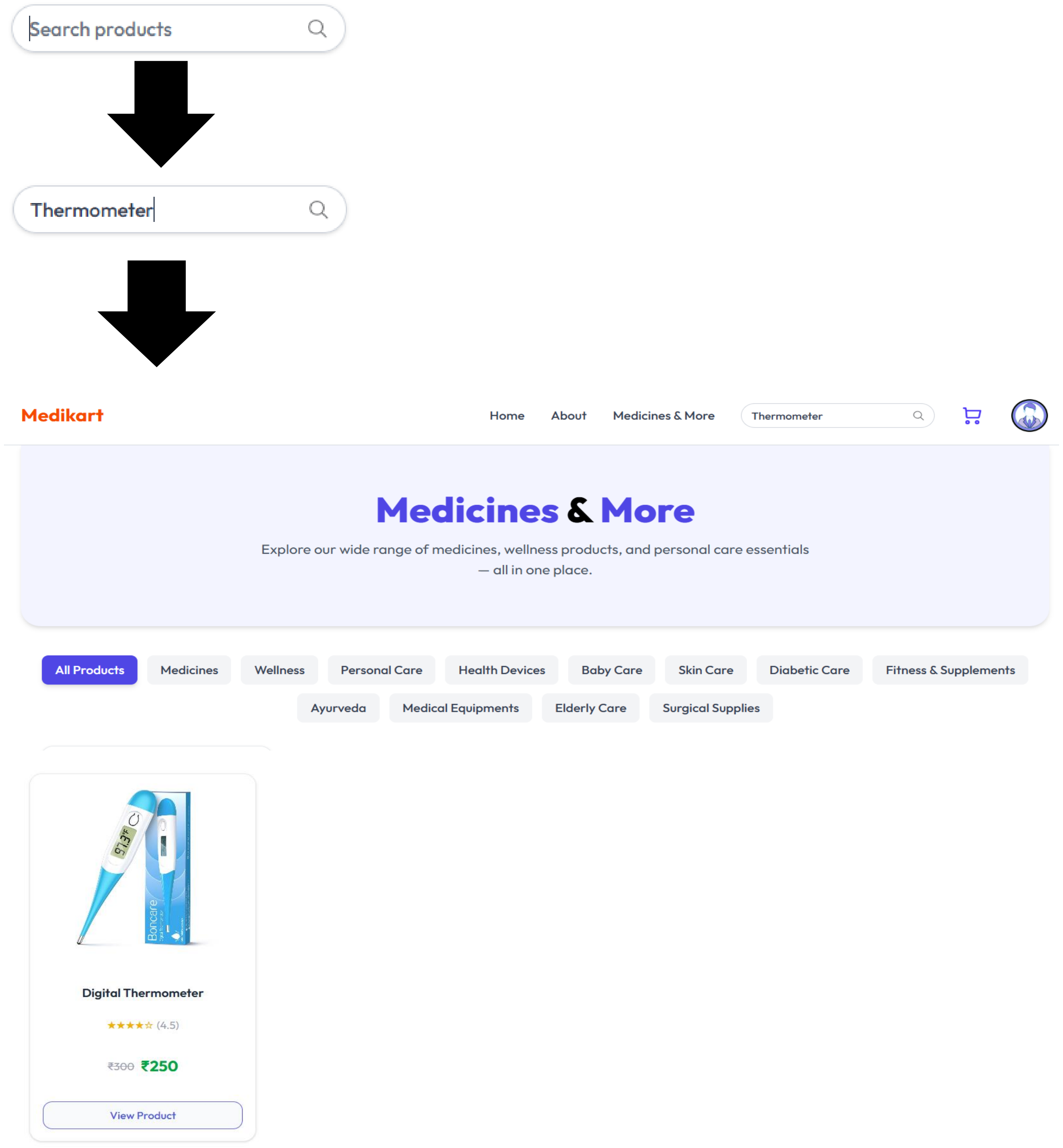


Figure 6.1- Medicines & More Page

Search Functionality

- Allows users to quickly find products by typing keywords.
- Integrated with backend APIs to fetch live product data.
- Improves user experience by making product discovery faster.



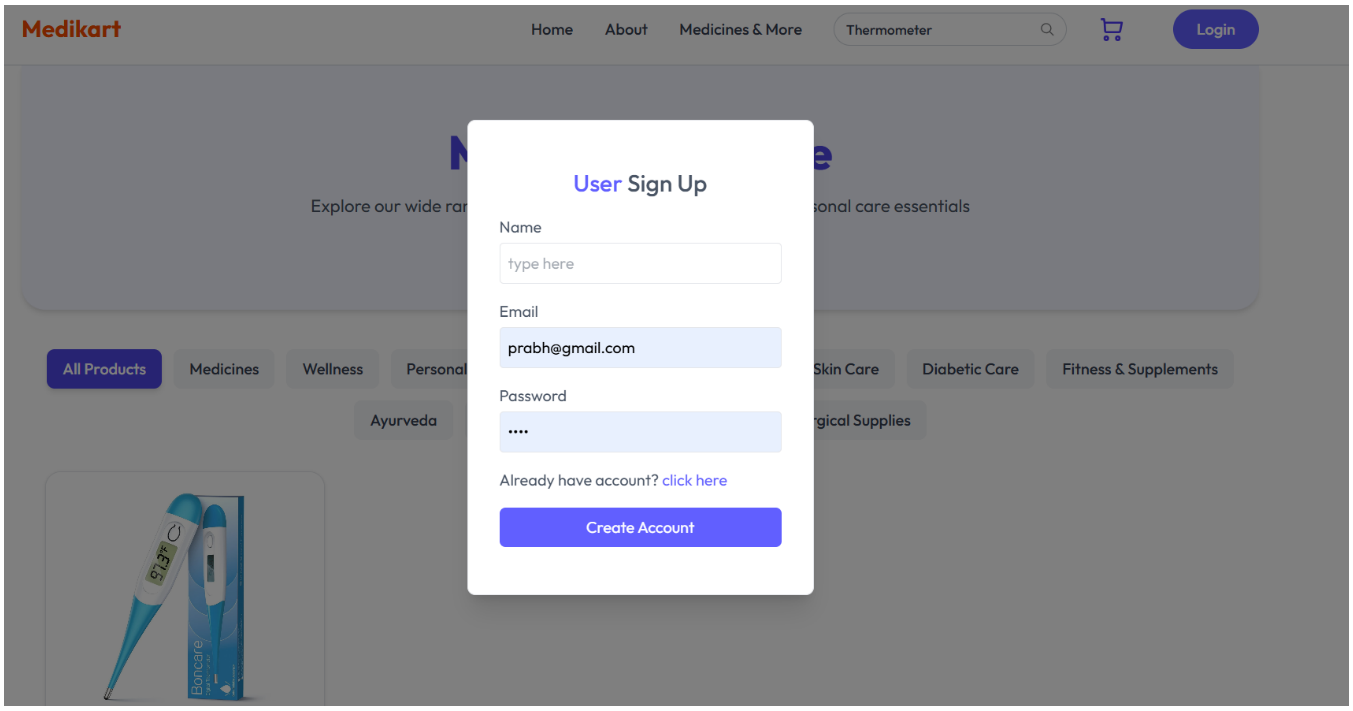
Explore our wide range of medicines, wellness products, and personal care essentials — all in one place.

***Figure 7.1-** Searching product*

Login and Register User

- Login allows existing users to access their accounts.
- Register lets new users create an account with email and password.

Register:



User Sign Up

Name

type here

Email

prabh@gmail.com

Password

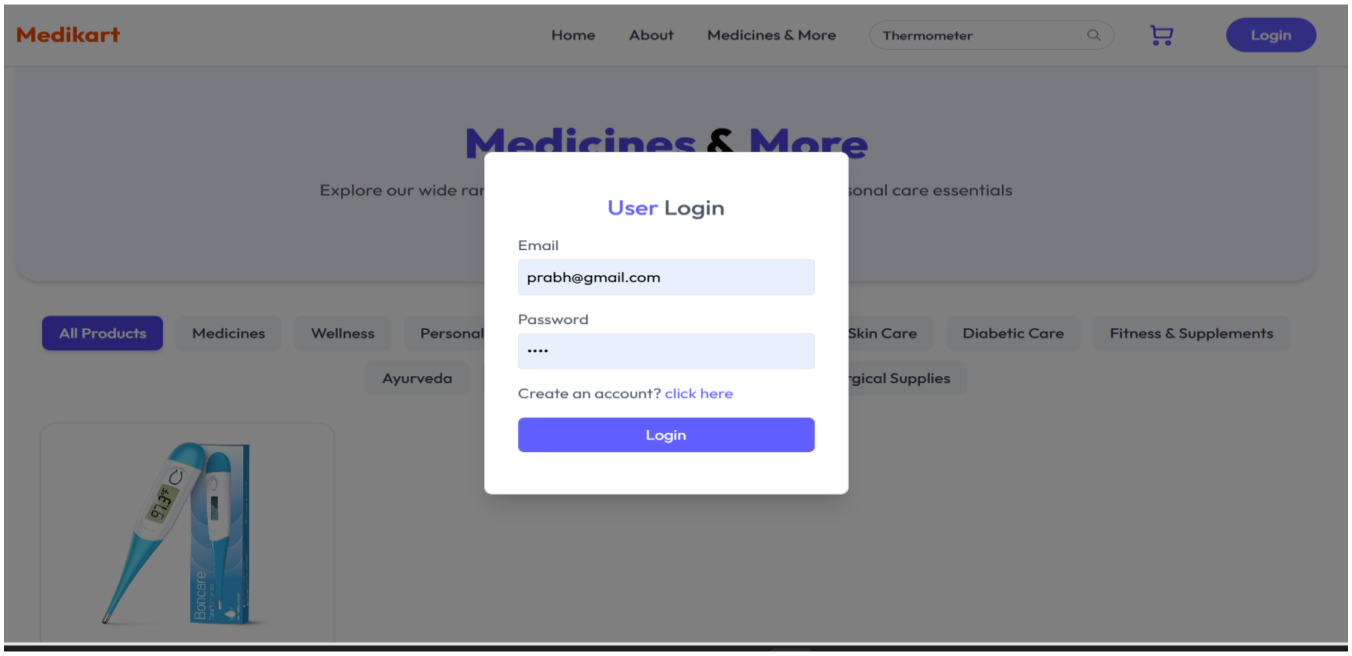
••••

Already have account? click here

Create Account

Figure 8.1- *Register user Page*

Login:



User Login

Email

prabh@gmail.com

Password

••••

Create an account? click here

Login

Figure 8.2 *– Login User Page*

5. Seller Dashboard

The **Seller Dashboard** enables sellers to manage their inventory and track customer orders.

- **Add Product Page:**
  - Sellers can upload new medicines, including **name, price, description, and image**.
  - Data is sent to the backend and stored in MongoDB.

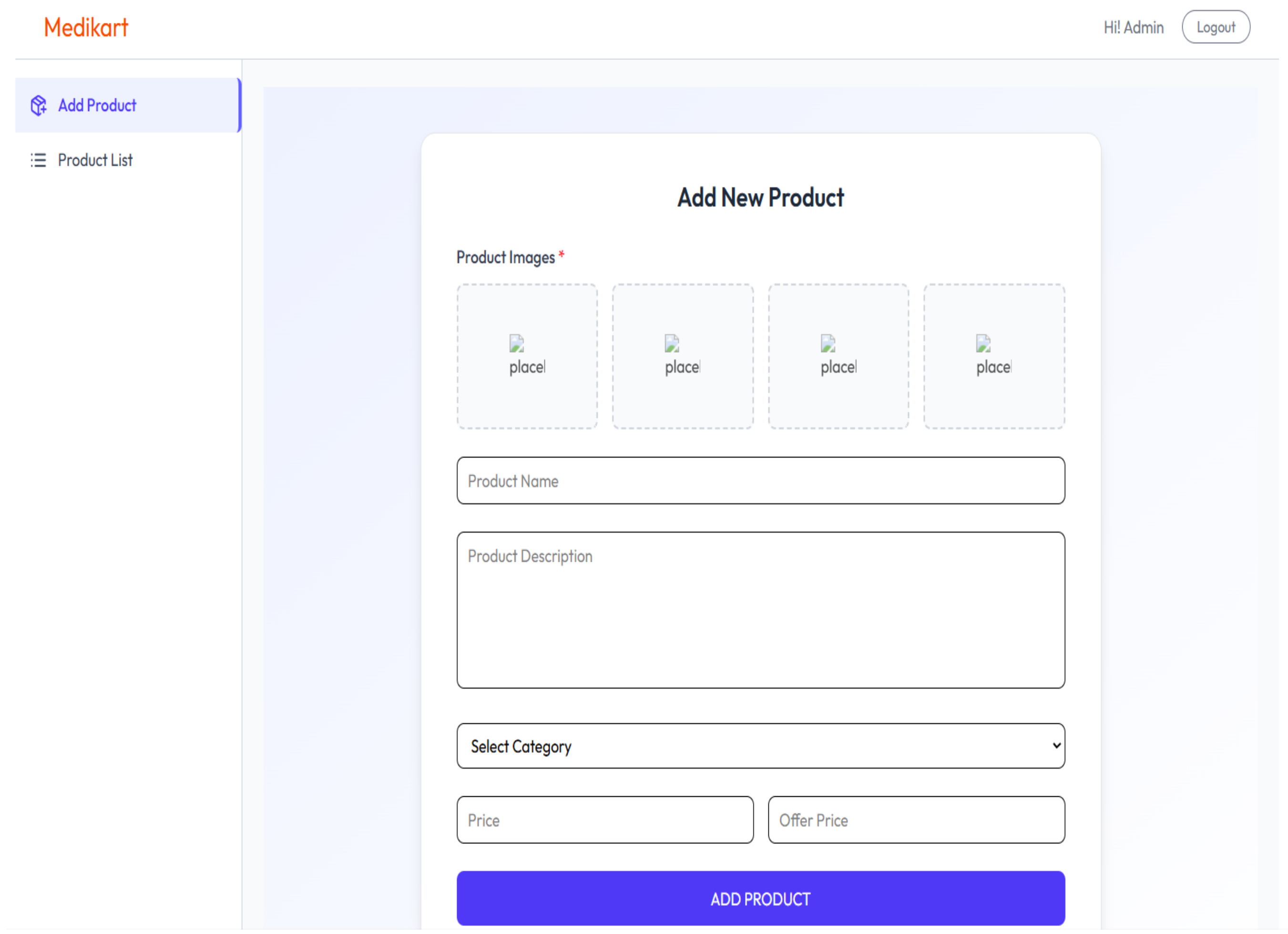
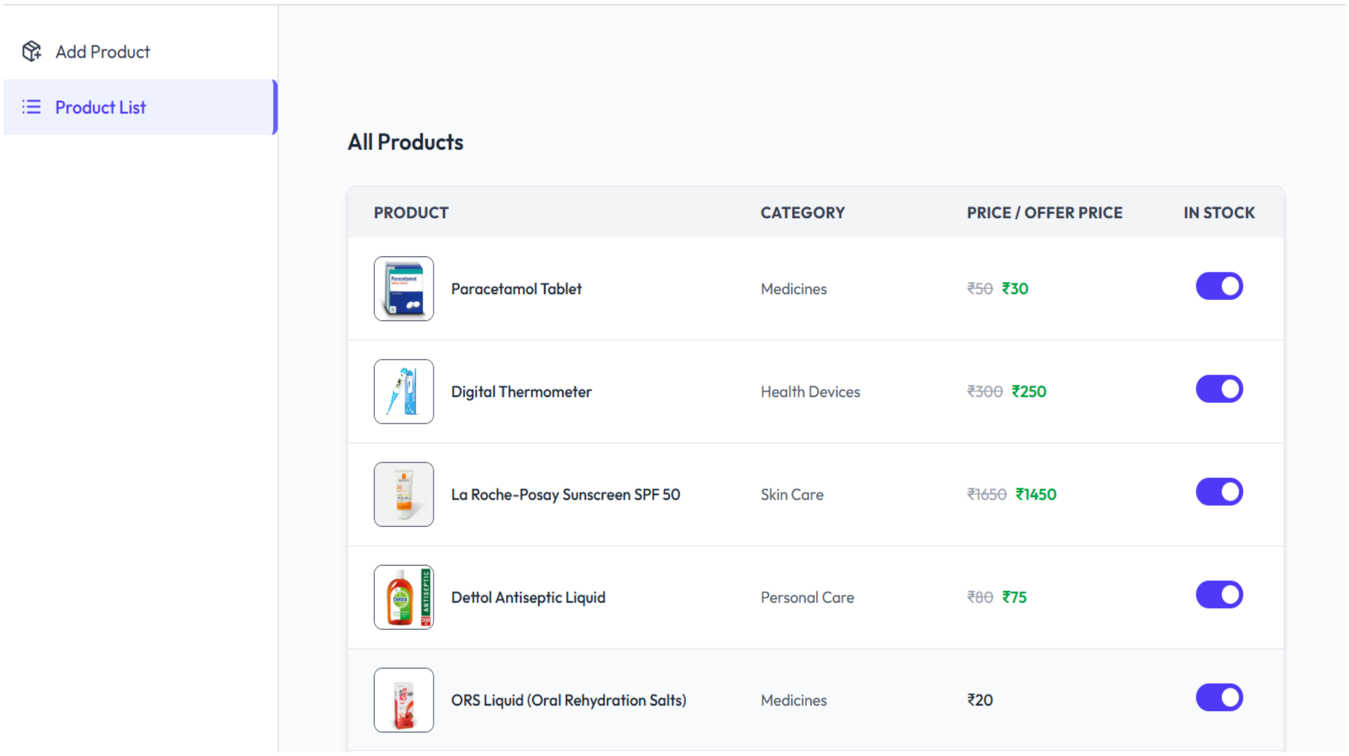


Figure 9.1- *Add Product page in seller dashboard*

- **Product List Page:**
  - Displays all products uploaded by the seller.
  - Sellers can **edit or delete** products as needed.



All Products

PRODUCT	CATEGORY	PRICE / OFFER PRICE	IN STOCK
Paracetamol Tablet	Medicines	~~₹50~~ ₹30	●
Digital Thermometer	Health Devices	~~₹300~~ ₹250	●
La Roche-Posay Sunscreen SPF 50	Skin Care	~~₹1650~~ ₹1450	●
Dettol Antiseptic Liquid	Personal Care	~~₹80~~ ₹75	●
ORS Liquid (Oral Rehydration Salts)	Medicines	₹20	●

***Figure 9.2-** Product List Page In Seller Dashboard*

7. Summary

The MediKart UI provides a **responsive, modular, and user-friendly interface**, designed for both end-users and sellers. Key highlights include:

- **Modular Components:** Each React component is reusable and linked to live backend data.
- **Dynamic Interaction:** Changes made in the UI (e.g., cart updates, order placements) are immediately reflected in the database.
- **Seamless Navigation:** React Router DOM ensures fast page transitions without reloading the app.
- **Cross-Device Compatibility:** Bootstrap ensures consistent rendering on desktops, tablets, and mobile devices.

Overall Result:

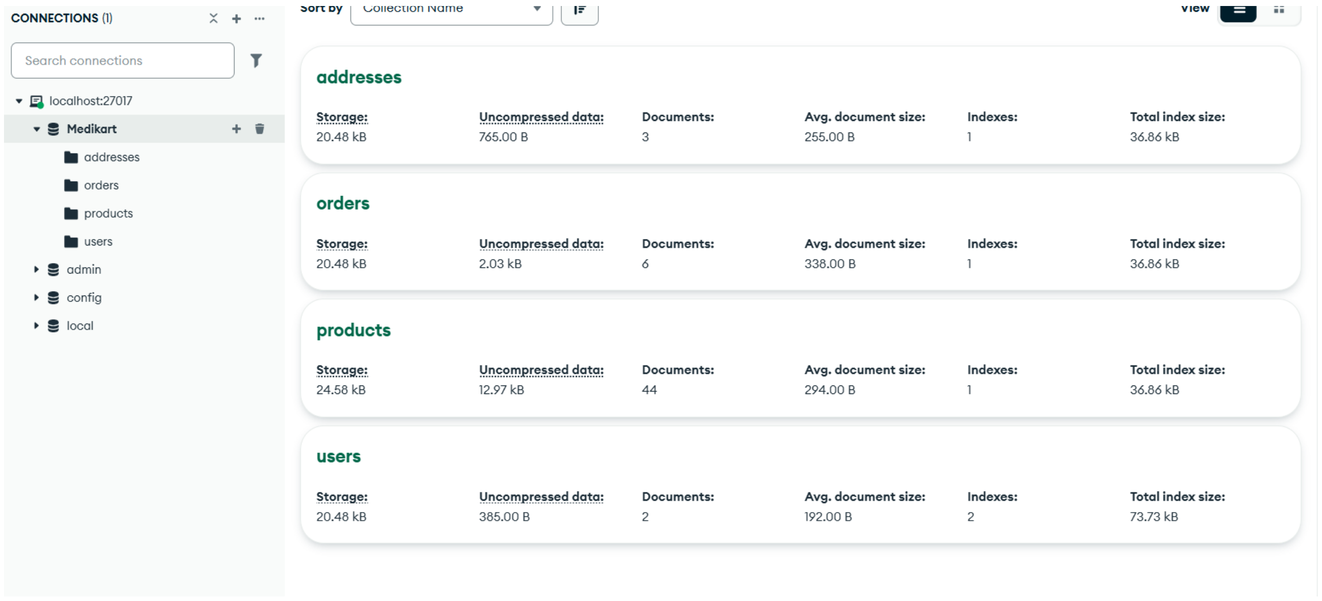
The frontend successfully integrates with the backend, providing a smooth and efficient shopping experience while enabling sellers to manage their products and orders effectively. Screenshots of each page should be inserted in the sequence described above to visually support the report.

Backend Integration

The MediKart backend is developed using **Node.js and Express.js**, which handle all server-side logic and RESTful API requests. **MongoDB** is used as the database to store all application data, including users, sellers, products, carts, orders, and addresses.

Key Backend Functionalities:

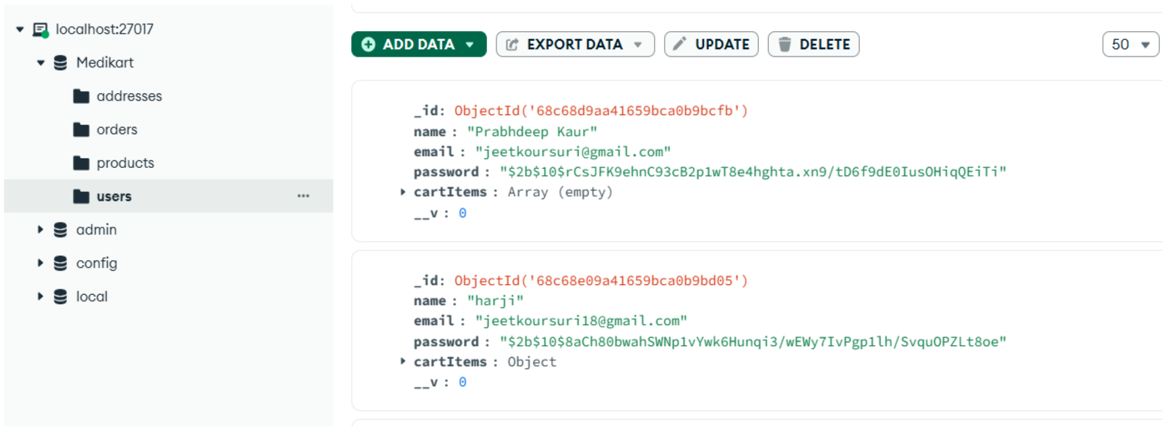
- **User and Seller Authentication:**
  - Secure login and registration using **JWT tokens**.
  - Ensures that only authenticated users can access protected routes.
- **Product Management:**
  - Sellers can add, update, or delete products.
  - Product data is stored in MongoDB collections.
- **Order and Cart Management:**
  - User carts and orders are saved in the database.
  - Orders can be tracked and retrieved in real-time.
- **Address Management:**
  - Users can add multiple addresses.
  - All addresses are stored in a dedicated MongoDB collection.
- **API Communication:**
  - The frontend uses **Axios** to fetch and update data via Express.js APIs.
  - All CRUD operations reflect instantly in MongoDB.



Collection	Storage:	Uncompressed data:	Documents:	Avg. document size:	Indexes:	Total index size:
addresses	20.48 kB	765.00 B	3	255.00 B	1	36.86 kB
orders	20.48 kB	2.03 kB	6	338.00 B	1	36.86 kB
products	24.58 kB	12.97 kB	44	294.00 B	1	36.86 kB
users	20.48 kB	385.00 B	2	192.00 B	2	73.73 kB

***Figure 10.1-** Backend Data*

1. **Users collection** –

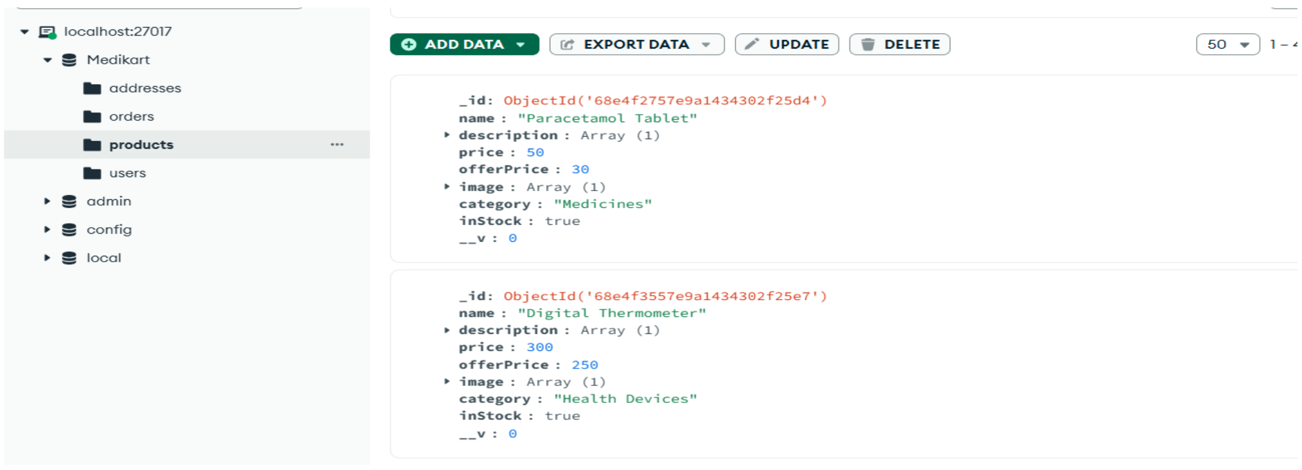


```
_id: ObjectId('68c68d9aa41659bca0b9bcfb')
name : "Prabhdeep Kaur"
email : "jeetkoursuri@gmail.com"
password : "$2b$10$rCsJFK9ehnC93cB2p1wT8e4hghta.xn9/tD6f9dE0IusOHiqQEiTi"
cartItems : Array (empty)
__v : 0

_id: ObjectId('68c68e09a41659bca0b9bd05')
name : "harji"
email : "jeetkoursuri18@gmail.com"
password : "$2b$10$8aCh80bwahSWNp1vYwk6Hunqi3/wEWy7IvPgp1lh/SvquOPZLt8oe"
cartItems : Object
__v : 0
```

***Figure 10.2** – User Data*

2. **Products collection** –

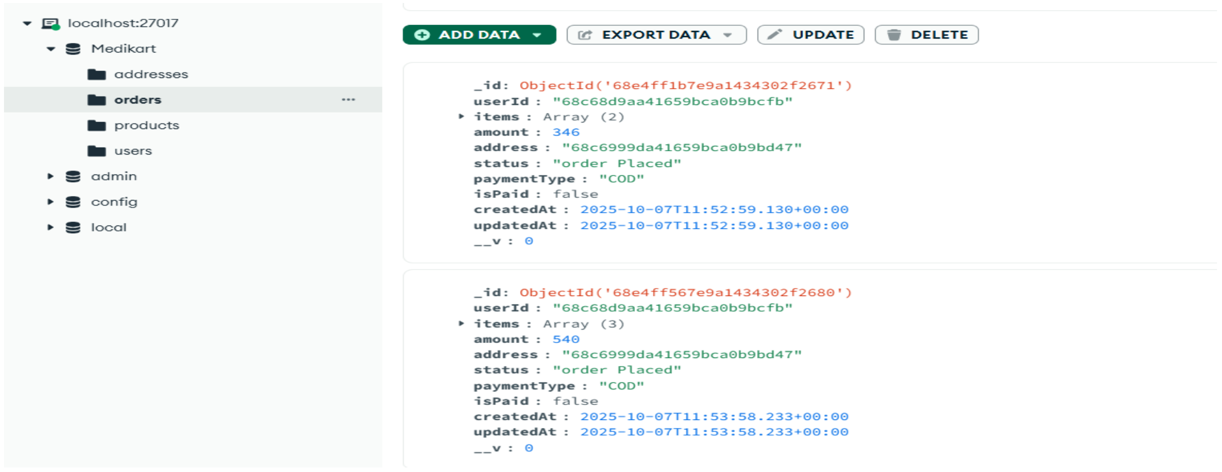


```
_id: ObjectId('68e4f2757e9a1434302f25d4')
name : "Paracetamol Tablet"
description : Array (1)
price : 50
offerPrice : 30
image : Array (1)
category : "Medicines"
inStock : true
__v : 0

_id: ObjectId('68e4f3557e9a1434302f25e7')
name : "Digital Thermometer"
description : Array (1)
price : 300
offerPrice : 250
image : Array (1)
category : "Health Devices"
inStock : true
__v : 0
```

***Figure 10.3-** Products Data*

3. **Orders collection** –

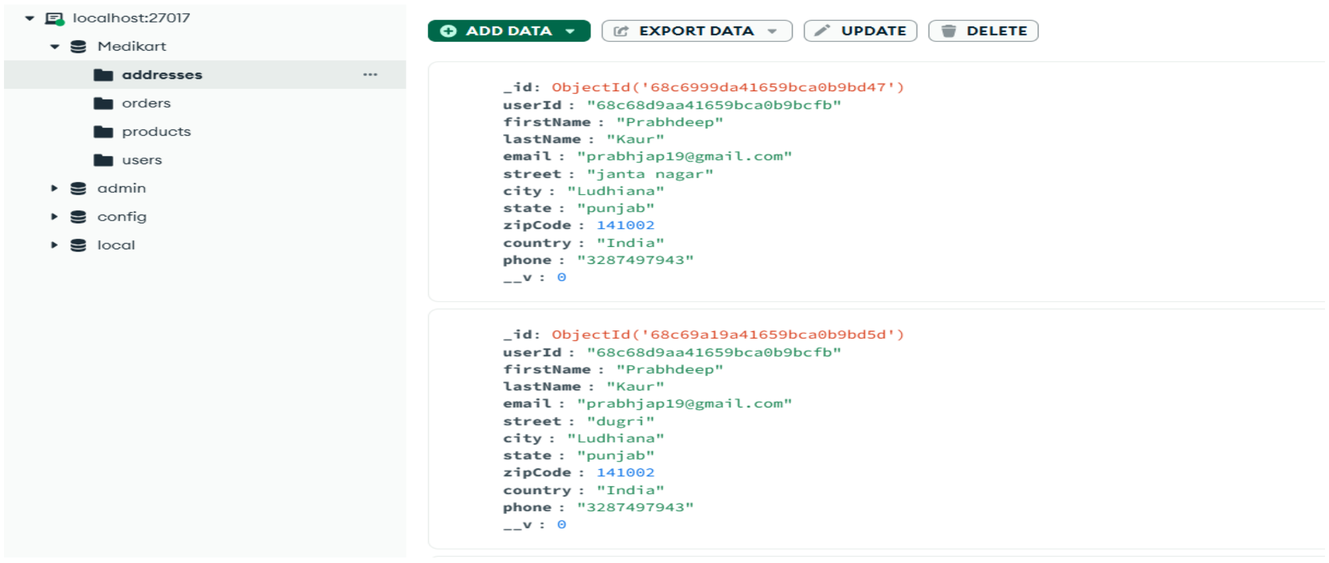


```
_id: ObjectId('68e4ff1b7e9a1434302f2671')
userId : "68c68d9aa41659bca0b9bcfb"
items : Array (2)
amount : 346
address : "68c6999da41659bca0b9bd47"
status : "order Placed"
paymentType : "COD"
isPaid : false
createdAt : 2025-10-07T11:52:59.130+00:00
updatedAt : 2025-10-07T11:52:59.130+00:00
__v : 0

_id: ObjectId('68e4ff567e9a1434302f2680')
userId : "68c68d9aa41659bca0b9bcfb"
items : Array (3)
amount : 540
address : "68c6999da41659bca0b9bd47"
status : "order Placed"
paymentType : "COD"
isPaid : false
createdAt : 2025-10-07T11:53:58.233+00:00
updatedAt : 2025-10-07T11:53:58.233+00:00
__v : 0
```

***Figure 10.4** – Orders Data*

4. **Addresses collection** –



```
_id: ObjectId('68c6999da41659bca0b9bd47')
userId : "68c68d9aa41659bca0b9bcfb"
firstName : "Prabhdeep"
lastName : "Kaur"
email : "prabhjap19@gmail.com"
street : "janta nagar"
city : "Ludhiana"
state : "punjab"
zipCode : 141002
country : "India"
phone : "3287497943"
__v : 0
```

```
_id: ObjectId('68c69a19a41659bca0b9bd5d')
userId : "68c68d9aa41659bca0b9bcfb"
firstName : "Prabhdeep"
lastName : "Kaur"
email : "prabhjap19@gmail.com"
street : "dugri"
city : "Ludhiana"
state : "punjab"
zipCode : 141002
country : "India"
phone : "3287497943"
__v : 0
```

Figure 10.5- *Address Data*

Summary:

The backend ensures that all user, seller, product, and order data is **securely stored and managed** in MongoDB. The frontend interacts smoothly with these collections via APIs, providing **real-time updates** for the application.

3.3 Technical Results and Backend Integration

The MediKart project is built on a **full-stack MERN architecture**, where the frontend React application communicates seamlessly with a Node.js and Express backend, and data is stored in **MongoDB**. The backend ensures that all user, seller, product, cart, order, and address data is processed, stored, and retrieved efficiently.

Backend Structure

The backend is organized into several modules for better scalability and maintainability:

1. **User Module**
   - Handles **signup, login, and logout** using **JWT tokens**.
   - Manages user profiles and authentication to secure private routes.
2. **Seller Module**
   - Sellers can **add, edit, or delete products**.
   - Allows sellers to view all **customer orders** for their products.

3. **Product Module**
    - Provides APIs to **fetch product lists**, filter by category, and get product details.
    - Products are stored in **MongoDB**, allowing dynamic display on the frontend.
4. **Cart Module**
    - Manages **additions, deletions, and updates** to the user's cart.
    - Ensures cart persistence across user sessions.
5. **Order Module**
    - Handles **order placement, tracking, and history**.
    - Stores order details like products, quantity, total amount, and delivery address.
6. **Address Module**
    - Enables users to **add multiple delivery addresses**.
    - Stores addresses securely and links them to user profiles.

Database Integration

- The project uses **MongoDB** as the database.
- Collections include: **users, sellers, products, carts, orders, and addresses**.
- **MongoDB Compass** can be used to visualize the database structure. Screenshots of these collections show real-time data inserted during development.
- Data operations (create, read, update, delete) are handled via **Express.js RESTful APIs**.

API Communication

- The frontend communicates with backend APIs using **Axios**.
- API endpoints are protected using **JWT-based authentication**, ensuring only authorized access.
- Responses from the backend are dynamically rendered in the frontend UI.

Performance and Security

- All APIs were tested using **Postman** to ensure correct request and response behavior.
- JWT authentication secures sensitive routes such as **cart, order, and product management**.
- Error handling in the backend returns meaningful messages to guide users in case of invalid actions.

Result

- Backend successfully integrates with the frontend, providing **real-time data updates**.
- MongoDB efficiently stores and retrieves all necessary data.
- APIs handle all CRUD operations without errors, ensuring smooth application functionality.

- JWT authentication ensures secure access for users and sellers.

3.4 Discussion of Learning Outcomes

Developing the MediKart project provided a **hands-on experience in full-stack web development**, covering both frontend and backend integration. The training allowed me to understand how all components of a web application work together in a real-world scenario.

Technical Learnings

1. **Frontend Development**
    - Learned to create **responsive and interactive UI components** using **React.js** and **Bootstrap**.
    - Applied **React Hooks** (useState, useEffect, and Context API) for **state management** across the application.
    - Implemented **dynamic navigation** using React Router to allow smooth page transitions without reloading.
2. **Backend Development**
    - Developed **RESTful APIs** with **Node.js** and **Express.js** to handle CRUD operations for users, products, carts, orders, and addresses.
    - Implemented **JWT-based authentication** for secure login and route protection.
    - Learned to structure backend code with **controllers, models, and middlewares** for better maintainability.
3. **Database Integration**
    - Designed **MongoDB schemas** for users, products, carts, orders, and addresses.
    - Connected the backend with MongoDB to store and retrieve data dynamically.
    - Learned how to **visualize and validate database collections** using MongoDB Compass (screenshots can be added here).
4. **API Integration**
    - Connected frontend and backend using **Axios** for seamless communication.
    - Learned to handle **API responses, errors, and validations** effectively.

Professional and Analytical Skills

- Gained **experience in project planning**, breaking down tasks into frontend, backend, and database modules.

- Improved **debugging skills** by fixing real-time issues with API calls, state updates, and authentication errors.
- Learned **version control** and collaboration using **Git and GitHub**.
- Understood the importance of **testing and documentation** in software development.

Result

By completing this project, I was able to:

- Build a **fully functional MERN stack application** with real-world e-commerce features.
- Understand the **end-to-end development process**, from frontend UI to backend API and database.
- Gain confidence in **full-stack web development**, problem-solving, and software project management.

3.5 Summary

In conclusion, the **MediKart project** successfully achieved all its **technical and functional objectives**. The system provides a **fully integrated MERN stack solution**, allowing users to browse, purchase, and track medicines, while enabling sellers to manage their products and orders efficiently.

Key points from this project:

- **Frontend Implementation:** Developed a responsive and user-friendly interface using **React.js** and **Bootstrap**, with dynamic navigation via **React Router**.
- **Backend Implementation:** Built secure and structured APIs using **Node.js** and **Express.js**, with **JWT authentication** for user and seller access.
- **Database Integration:** Designed and connected **MongoDB collections** for users, products, orders, carts, and addresses, ensuring smooth data storage and retrieval.
- **API Communication:** Successfully integrated **frontend and backend** using **Axios**, handling data dynamically and validating inputs.
- **Learning Outcomes:** Gained **hands-on experience in full-stack development**, including debugging, testing, version control with Git/GitHub, and real-world project management.

Overall, the project bridged the gap between **academic knowledge and industry-level web development**. It provided practical insights into building scalable applications, managing databases, securing user data, and integrating all components to create a functional e-commerce platform.

CHAPTER 4:
CONCLUSION & FUTURE SCOPE

4.1 Conclusion

The MediKart project marks the successful completion of my Full Stack Web Development training using the MERN Stack (MongoDB, Express.js, React.js, Node.js). It demonstrates how modern web technologies can be combined to build a dynamic, secure, and scalable online medicine ordering platform connecting users and sellers seamlessly.

During this project, I gained hands-on experience in both frontend and backend development, including designing user interfaces, managing API routes, implementing authentication, and integrating databases. The system was developed with a focus on usability, reliability, and efficiency, ensuring a smooth user experience and a well-organized backend structure.

Key Outcomes:

1. **Functional Success:** Implemented user and seller authentication using JWT tokens; developed modules for product display, cart management, address handling, and order placement; integrated frontend and backend APIs using Axios.
2. **Technical Achievements:** Used MongoDB Atlas for cloud database storage; applied React Hooks and Context API for state management; ensured responsive design with CSS and Bootstrap.
3. **Skill Development:** Enhanced understanding of client-server architecture and RESTful API design; learned practical debugging and testing using Postman and browser developer tools; improved Git and GitHub version control skills.
4. **Professional Growth:** Gained experience in project structuring, deployment readiness, time management, code organization, and documentation for industry-level projects.

Overall, MediKart not only fulfilled the goal of creating an online medical store but also served as a comprehensive learning experience, bridging theory and real-world application. The skills acquired — from API development to authentication and database integration — provide a strong foundation for a career in full-stack development.

4.2 Future Scope

The current version of MediKart provides essential e-commerce functionalities such as user login, product listing, cart management, and order handling. To make the platform more advanced, scalable, and competitive, the following enhancements are proposed:

1. **Integration of Online Payment System:** Incorporate payment gateways like Razorpay, Stripe, or PayPal to support multiple payment methods including UPI, debit/credit cards, and net banking for secure transactions.
2. **Admin Dashboard:** Introduce an Admin Panel to manage sellers, products, users, and analytics, including sales reports and product performance visualization.
3. **AI-Powered Recommendations:** Implement machine learning algorithms to suggest medicines and healthcare products based on purchase history, enhancing personalization and user engagement.
4. **Real-Time Stock and Inventory Management:** Allow sellers to update stock automatically, receive low-stock alerts, and maintain real-time synchronization between the database and UI.
5. **Mobile Application Version:** Develop a cross-platform mobile app using React Native or Flutter, synchronized with the web platform.
6. **Multi-Language and Accessibility Features:** Support regional languages and enhance accessibility with voice search, dark mode, and screen reader compatibility.
7. **Deployment and Cloud Optimization:** Deploy on AWS, Render, or Vercel for scalability and global access, with caching and load balancing for performance optimization.
8. **Email and SMS Notifications:** Integrate real-time notifications for order confirmations and delivery updates using services like Twilio or Nodemailer.
9. **Data Analytics and Insights:** Track user behavior and optimize strategies using Google Analytics or custom dashboards.
10. **Enhanced Security:** Implement Two-Factor Authentication (2FA) and role-based access control to strengthen security.

These enhancements will improve usability, scalability, and competitiveness, ensuring MediKart remains a modern and user-centric online medicine platform.

REFERENCES

The development of MediKart was supported by several official documentations, online tutorials, and development tools. The following references were consulted for conceptual understanding, coding guidance, and debugging:

[1] MongoDB Documentation. (n.d.). Available: https://www.mongodb.com/docs/

[2] Express.js Documentation. (n.d.). Available: https://expressjs.com/

[3] React.js Official Documentation. (n.d.). Available: https://react.dev/

[4] Node.js Documentation. (n.d.). Available: https://nodejs.org/en/docs

[5] Axios HTTP Client. (n.d.). Available: https://axios-http.com/

[6] JWT Authentication. (n.d.). Available: https://jwt.io/introduction/

[7] Bootstrap Official Guide. (n.d.). Available: https://getbootstrap.com/docs/

[8] Vite Build Tool. (n.d.). Available: https://vitejs.dev/

[9] MDN Web Docs. (n.d.). Available: https://developer.mozilla.org/

[10] GitHub Guides for Version Control. (n.d.). Available: https://guides.github.com/

[11] YouTube Tutorials:

• Codevolution – MERN Stack Tutorials. Available: https://www.youtube.com/c/Codevolution

• freeCodeCamp.org – Full MERN Stack Course. Available: https://www.youtube.com/@freecodecamp

• Programming with Mosh – React & Node.js Tutorials. Available: https://www.youtube.com/@programmingwithmosh

[12] Tutorials and technical blogs on GeeksforGeeks, W3Schools, and Stack Overflow for practical implementation guidance.