

**SAVEETHA SCHOOL OF ENGINEERING**

**SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES**

**CHENNAI-602105**

**Conventional support system for on-line retail services**

**A CAPSTONE PROJECT REPORT**

*Submitted in the partial fulfillment for the completion of the course*

**CSA4307 INTERNET PROGRAMMING FOR CLIENT SERVER MODEL**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**Submitted by**

**B. Hevanthkumar**

**192373032**

**Under the Supervision of**

**L.REETHA**

**MAY 2025**

**DECLARATION**

I, **B.Hevanthkumar**, students of **Bachelor of Engineering in the Department** of Computer Science and Engineering [Data science] Saveetha Institute of Medical and Technical Sciences, Saveetha School of Engineering, Chennai, hereby declare that the work presented in this Capstone Project Work entitled **Conventional support system for on-line retail services** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics.

B.Hevanthkumar

192373032

Date:

Place:

**CERTIFICATE**

This is to certify that the project entitled Conventional support system for on-line retail servicessubmitted by **B. Hevanthkumar** has been carried out under my supervision. The project has been submitted as per the requirements in the current semester of B.E. Computer Science and Engineering.

Supervisor

L.Reetha

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TOPICS** | **PAGE NO.** |
| 1 | **Abstract** | **4** |
| 2 | **Introduction** | **5** |
| 3 | **Project Description**  Key Features and Functionalities  Product Browsing and Checkout SystemReal-time Order Tracking  Admin Dashboard | **6** |
| 4 | **Problem Description**  Program to build a simple Software for < > | **7** |
| 5 | **Tool Description**  Customer Interface  Vendor Admin Interfac | **8** |
| 6 | **Operations**  Store Order Information  Store Vendor (Retailer) Information  Store Customer Information | **11** |
| 7 | **Approach / Module Description / Functionalities**   1. User Authentication Module 2. Customer Interface Module 3. Vendor Management Module 4. Order Processing Modul | **13** |
| 8 | **Implementation**  Coding | **16** |
| 9 | **Result**  Output with Screenshots | **17** |
| 10 | **Conclusion**  Future Enhancement  **References** | **18** |

**ABSTRACT**

This capstone project presents the design and development of a conventional support system for online retail services, utilizing the MERN stack—MongoDB, Express.js, React, and Node.js—to create an efficient, secure, and user-centric platform. The system serves as a digital bridge between consumers and vendors, streamlining product browsing, order placement, real-time tracking, and secure payment processing. Designed with scalability and responsiveness in mind, the platform ensures seamless interaction across both desktop and mobile devices.

The project addresses key challenges associated with online retail platforms, including managing large-scale data operations, maintaining transactional security, and ensuring optimal performance during peak usage. Vendors are empowered to register, manage inventory, and process customer orders, while consumers benefit from an intuitive interface for exploring products, adding them to a virtual cart, and completing purchases with ease.

This initiative showcases the practical application of full-stack development skills gained during the course of study and highlights the MERN stack’s effectiveness in building dynamic and responsive web applications. The outcome is a robust and scalable solution that reflects real-world demands in the online retail sector and lays the groundwork for future enhancements and broader service integration

**INTRODUCTION**

The rapid advancement of technology and widespread internet accessibility have transformed the landscape of online retail services, reshaping how consumers engage with products and vendors. This capstone project focuses on developing a conventional support system for online retail services using the MERN stack—MongoDB, Express.js, React, and Node.js—a robust and modern framework for full-stack web development. Leveraging the capabilities of the MERN stack enables the creation of a dynamic, efficient, and scalable platform capable of managing various operations typical of online retail environments.

The primary goal of the project is to provide a seamless and user-friendly platform that connects consumers with local vendors, allowing them to browse products, place orders, and track them in real time. Core functionalities include secure user authentication, product and inventory management, order tracking, and integrated payment solutions, showcasing the practical application of full-stack web development in addressing real-world retail challenges.

Key features of the system include an intuitive customer interface, a vendor dashboard for managing inventory and processing orders, and robust data handling to track user activity and vendor performance. Beyond addressing standard issues such as high data volume and security, the project offers a scalable foundation for future enhancements, including personalized recommendations, advanced analytics, and multi-language support—making it a comprehensive solution for modern online reta

**PROJECT DESCRIPTION**

This project involves the design and development of a full-stack web application that serves as a conventional support system for online retail services, built using the MERN stack—MongoDB, Express.js, React, and Node.js. The platform offers an interactive and efficient solution that connects consumers with a variety of vendors, enabling seamless browsing, purchasing, and real-time order tracking. It provides a comprehensive digital infrastructure that addresses the needs of both end-users and retail service providers.

**Key Features and Functionalities:**

User Authentication and Profile Management:

Consumers can securely register, log in, and manage their profiles. The authentication system ensures user data privacy and secure access across the platform.

**Product Browsing and Checkout System:**

Users can explore products offered by various vendors, add items to a shopping cart, and proceed through an optimized checkout process. The system supports integrated payment gateways to ensure smooth and secure transactions.

**Real-time Order Tracking:**

After placing an order, customers can track its progress in real-time. This functionality improves transparency and enhances customer satisfaction by providing updates at every stage of order fulfillment.

**Admin Dashboard:**

An administrative interface allows system administrators to manage users, vendors, products, and orders efficiently. This dashboard provides centralized control for overseeing platform operations and maintaining service quality.

**PROBLEM DESCRIPTION**

The primary goal of this project is to develop a functional software application that supports online retail services by connecting consumers with local vendors. The application addresses common challenges in modern e-commerce platforms, including efficient order management, secure user authentication, real-time order tracking, and streamlined payment processing.

With the rising demand for accessible and efficient online retail experiences, many small and medium-sized vendors struggle to adopt sophisticated systems due to cost and technical complexity. This project seeks to bridge that gap by delivering a cost-effective and scalable solution using the MERN stack—MongoDB, Express.js, React, and Node.js.

**Project Objectives:**

**Simplify User Experience:**

Design an intuitive and responsive user interface that enables consumers to easily browse vendors, view product offerings, and place orders with minimal friction.

**Efficient Vendor Management:**

Allow vendors to register on the platform, manage product listings, and monitor customer orders via a user-friendly administrative dashboard.

**Seamless Order Processing and Tracking:**

Enable real-time order placement and status tracking to ensure transparency and improve customer satisfaction.

**Secure Payment Integration:**

Implement reliable and secure payment gateways to provide a trustworthy transaction experience for both consumers and vendors.

**Development Tools:**

This online retail platform is built using the MERN stack:

MongoDB for flexible and scalable database management.

Express.js as a backend framework for handling API requests.

React for building a dynamic and responsive user interface.

Node.js to enable scalable, high-performance server-side logic.

Each component plays a vital role in creating a seamless experience for both end-users and administrative users.

**User Interface Design:**

The UI is structured to be clean, responsive, and accessible, supporting a wide range of users and devices. Built with React, it features two main portals: the Customer Interface and the Vendor Admin Interface.

**1. Customer Interface:**

Home Page:

Displays a list of available vendors with filters and search options to enhance browsing.

Product Page:

Presents product listings for each vendor, complete with descriptions, pricing, and an “Add to Cart” feature.

Cart:

Enables users to review selected items, adjust quantities, and proceed to checkout.

Checkout and Payment:

Offers a secure and streamlined process for confirming orders and making payments.

Order Tracking:

Provides real-time status updates (e.g., “Processing,” “Shipped,” “Delivered”) for transparency and convenience.

**2. Vendor Admin Interface:**

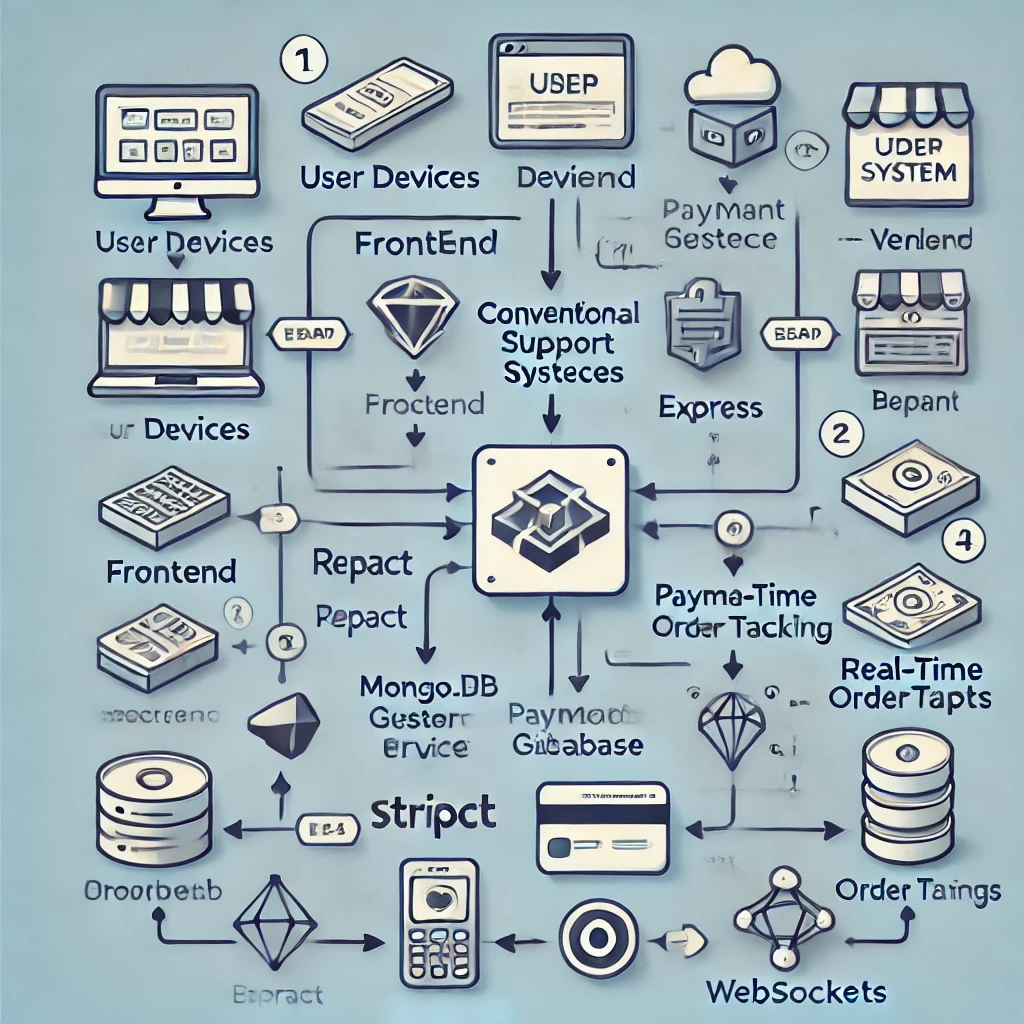
Dashboard:

Gives an overview of current orders, sales performance, and customer activity.

Product Management:

Allows vendors to add, edit, or remove product listings with relevant details such as descriptions, pricing, and availability.

**BLOCK DIAGRAM**



**Operations:**

In this project, a full-stack software application is developed to support online retail services by managing users, vendors, and orders. The platform handles a range of essential operations necessary for effective retail service delivery. Below are the primary operations and corresponding data storage requirements:

**1. Store Customer Information**

First Name and Last Name: Used for personalization and identification across the platform.

Unique User ID: Assigned to each customer to enable secure access and traceable transactions.

Contact Details: Stores email, phone number, and delivery address to facilitate communication and order fulfillment.

**2. Store Vendor (Retailer) Information:**

Vendor Name: Displayed to users for identification and brand visibility.

Unique Vendor ID: Each vendor is given a unique ID for backend management and order association.

Location and Contact Information: Includes address, contact number, and email for support and logistics.

Product Catalog Details: Stores product information such as item names, descriptions, pricing, and stock status.

**3. Store Order Information**

Order ID: Each transaction is assigned a unique identifier for tracking and processing.

Customer ID: Links orders to specific customers to support order history and personalization.

Vendor ID: Associates each order with a specific vendor for accurate routing and fulfillment.

**Module Description:**

This support system is designed using a modular architecture, where every core function is separated into well-defined, maintainable components. This approach ensures scalability, simplifies debugging, and supports future upgrades. Below are the key modules of the system:

**1. User Authentication Module**

Functionality:

Manages user access and security protocols.

**Features:**

Secure registration with encrypted password storage.

JWT-based authentication for secure login sessions.

Profile editing and password recovery options.

Access control to limit feature visibility based on user roles (e.g., customer, vendor, admin).

**Objective:**

To ensure that only authorized users can access relevant parts of the system securely.

**2. Customer Interface Module**

Functionality:

Provides end-users with a seamless and intuitive shopping experience.

**Features:**

Browse, search, and filter vendor product listings.

Detailed product views with pricing, descriptions, and ratings.

Dynamic shopping cart with quantity adjustments and price calculation.

Order history display and real-time tracking of current orders.

**Objective:**

To enhance user engagement and streamline the shopping and ordering experience.

**3. Vendor Management Module**

Functionality:

Empowers vendors to control their online storefronts, product listings, and incoming orders.

**Features:**

Vendor profile management including contact details and branding.

Product management to add, update, or remove items, and control availability.

Order dashboard to view and update order statuses (e.g., “Processing,” “Dispatched”).

**Objective:**

To give vendors full control over their product listings and operational wrkflow.

**4. Order Processing Module**

Functionality:

Handles all aspects of the order lifecycle from placement to delivery.

**Features:**

Real-time order placement and confirmation from the customer’s cart.

Live tracking of order status updates for both customers and vendors.

Notification system to inform users of order progress or issues.

**Objective:**

To streamline the order management process and enhance transparency for all stakeholders.

**IMPLEMENTATION**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<title>Welcome </title>

<style>

body {

font-family: 'Segoe UI', sans-serif;

background: linear-gradient(135deg, #f5f7fa, #c3cfe2);

margin: 0;

padding: 0;

display: flex;

align-items: center;

justify-content: center;

min-height: 100vh;

}

.container {

background: white;

padding: 40px;

border-radius: 12px;

box-shadow: 0 10px 25px rgba(0,0,0,0.1);

width: 100%;

max-width: 400px;

}

section { display: none; }

section.active { display: block; }

h2 {

text-align: center;

color: #333;

}

input, button {

width: 100%;

padding: 12px;

margin: 12px 0;

border-radius: 8px;

border: 1px solid #ccc;

font-size: 16px;

}

input:focus {

outline: none;

border-color: #6a82fb;

}

button {

background: linear-gradient(to right, #6a82fb, #fc5c7d);

color: white;

font-weight: bold;

border: none;

cursor: pointer;

transition: background 0.3s;

}

button:hover {

background: linear-gradient(to right, #fc5c7d, #6a82fb);

}

ul {

list-style: none;

padding: 0;

}

ul li {

background: #f0f0f0;

margin: 8px 0;

padding: 10px;

border-radius: 6px;

}

</style>

</head>

<body>

<div class="container">

<!-- REGISTER -->

<section id="register" class="active">

<h2>Register</h2>

<input type="text" id="regName" placeholder="Full Name" required />

<input type="email" id="regEmail" placeholder="Email" required />

<input type="password" id="regPassword" placeholder="Password" required />

<button onclick="handleRegister()">Register</button>

</section>

<!-- LOGIN -->

<section id="login">

<h2>Login</h2>

<input type="email" id="loginEmail" placeholder="Email" required />

<input type="password" id="loginPassword" placeholder="Password" required />

<button onclick="handleLogin()">Login</button>

</section>

<!-- HOME -->

<section id="home">

<h2>Welcome, Shopper!</h2>

<p style="text-align:center;">Enjoy shopping the best products online!</p>

<button onclick="goTo('search')">Next: Search Products</button>

</section>

<!-- SEARCH -->

<section id="search">

<h2>Search Products</h2>

<input type="text" id="searchBox" placeholder="Search for items..." />

<button onclick="handleSearch()">Search</button>

<div id="searchResults"></div>

<button onclick="goTo('payment')">Next: Payment</button>

</section>

<!-- PAYMENT -->

<section id="payment">

<h2>Payment Details</h2>

<input type="text" placeholder="Card Number" required />

<input type="text" placeholder="Cardholder Name" required />

<input type="text" placeholder="Expiry Date" required />

<input type="text" placeholder="CVV" required />

<button onclick="goTo('orders')">Pay & Continue</button>

</section>

<!-- ORDER LIST -->

<section id="orders">

<h2>Your Orders</h2>

<ul>

<li>Order #1234 - Smartphone - Delivered</li>

<li>Order #5678 - Shoes - In Transit</li>

</ul>

<p style="text-align:center;">Thank you for shopping with us! 😊</p>

</section>

</div>

<script>

let registeredUsers = [];

function goTo(step) {

document.querySelectorAll('section').forEach(s => s.classList.remove('active'));

document.getElementById(step).classList.add('active');

}

function handleRegister() {

const name = document.getElementById('regName').value.trim();

const email = document.getElementById('regEmail').value.trim();

const password = document.getElementById('regPassword').value;

if (!name || !email || !password) {

alert('Please fill all fields');

return;

}

registeredUsers.push({ email, password });

alert('Registration successful! Now please log in.');

goTo('login');

}

function handleLogin() {

const email = document.getElementById('loginEmail').value.trim();

const password = document.getElementById('loginPassword').value;

const user = registeredUsers.find(u => u.email === email && u.password === password);

if (user) {

goTo('home');

} else {

alert('Invalid login. Please try again.');

}

}

function handleSearch() {

const query = document.getElementById('searchBox').value;

document.getElementById('searchResults').innerHTML = `<p style="color:#555">Showing results for "<strong>${query}</strong>"...</p>`;

}

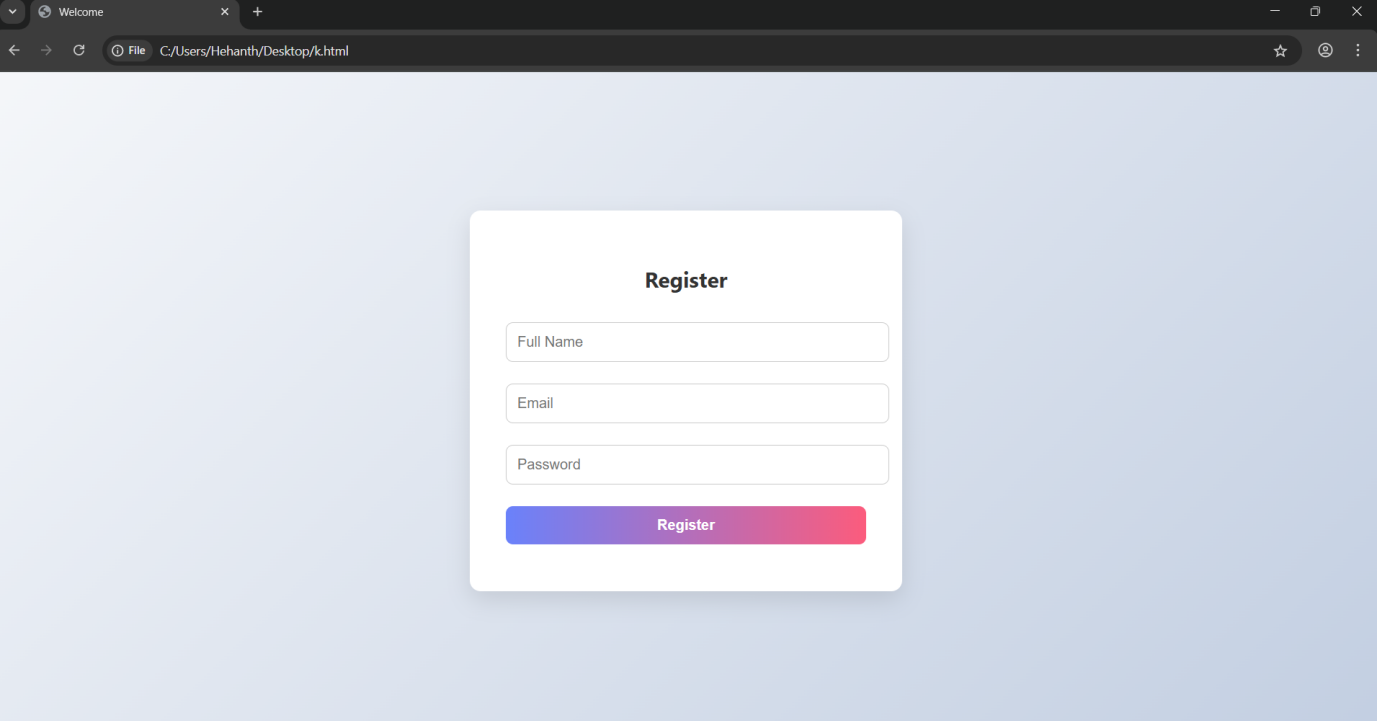
</script>

</body>

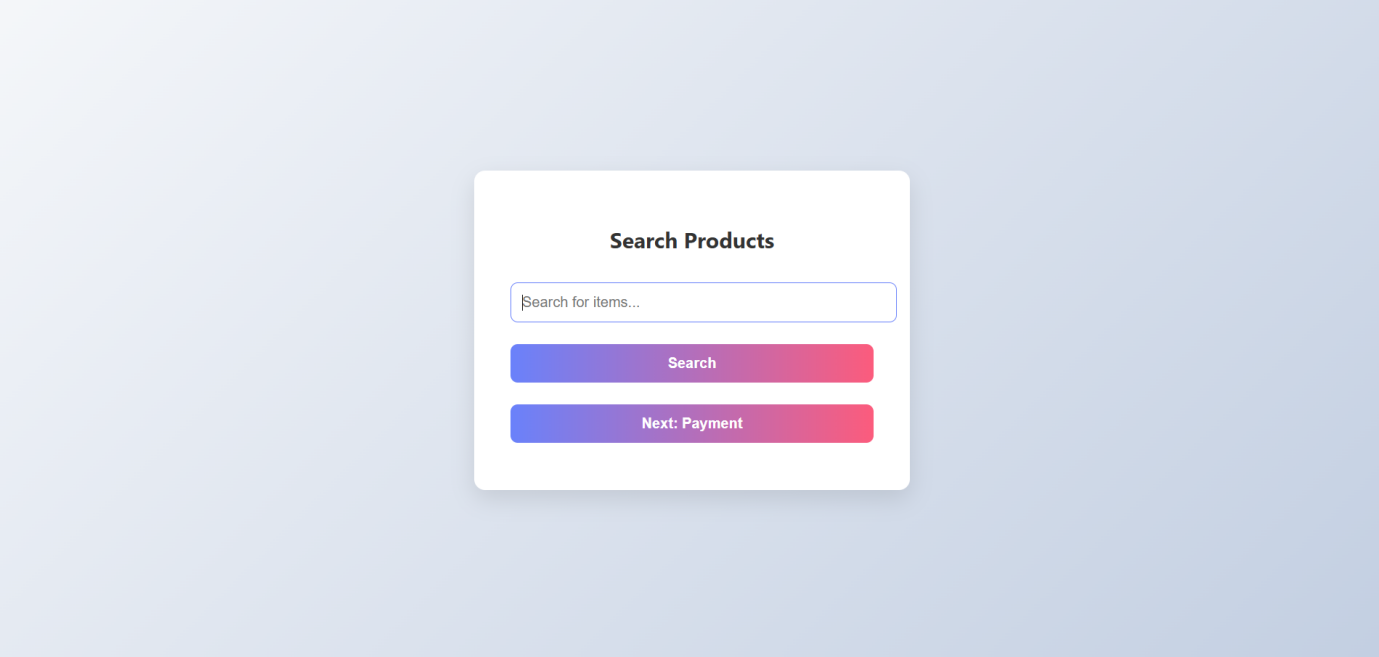
</html>

**RESULT**

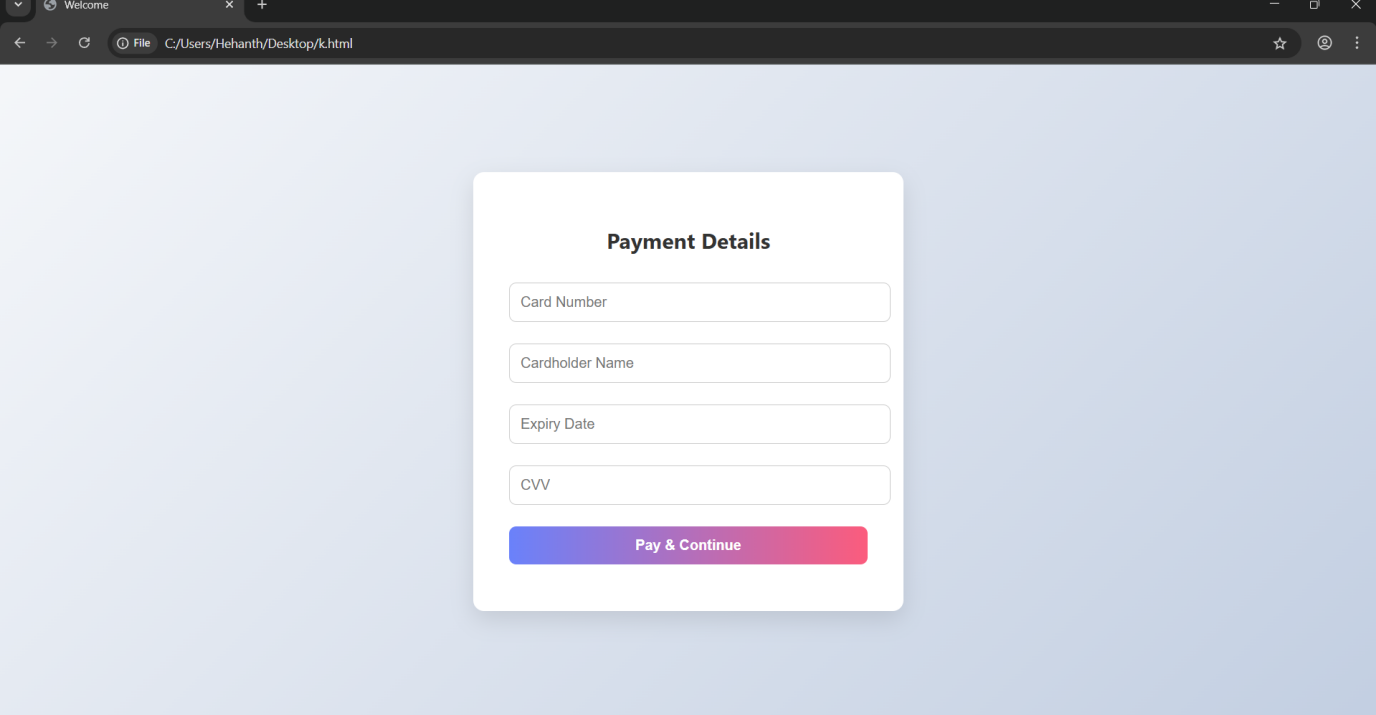
**Figure 1: Register page**



**Figure 2: Search page**



**Figure 3: Payment Page**



**CONCLUSION**

This project, Conventional Support System for On-line Retail, successfully demonstrates the development of a robust and user-friendly platform designed to streamline customer support operations for online retail services. By utilizing a modular architecture and established web technologies, we have created a scalable, responsive, and secure application that addresses the core challenges faced by both customers and support staff in the online retail industry.

Throughout this project, we achieved several key objectives:

Implemented a secure and efficient user authentication system to protect user data and ensure authorized access.

Developed a comprehensive ticketing system that enables customers to easily raise and track support requests.

Designed an intuitive staff interface, empowering support teams to manage, prioritize, and resolve tickets efficiently.

Integrated communication features, such as threaded comments and status updates, to enhance transparency and collaboration between customers and support personnel.

The platform not only meets the current requirements for online retail support but also establishes a solid foundation for future enhancements. Features such as real-time chat, AI-driven ticket routing, advanced analytics, and multi-channel support can be incorporated seamlessly due to the system’s modular design.

Overall, this project provides valuable experience in full-stack application development and practical insight into solving real-world challenges in the e-commerce support domain. It highlights our ability to deliver a well-structured, functional, and reliable web application, with a strong emphasis on scalability, security, and user-centered design.

**REFERENCE**

1. MongoDB Documentation: https://www.mongodb.com/docs
2. Express.js Documentation: https://expressjs.com/
3. React Documentation: https://reactjs.org/docs/getting-started.html
4. Node.js Documentation: https://nodejs.org/en/docs/
5. Building a Full-Stack App: https://www.freecodecamp.org/news/mern-stack-tutorial/
6. JWT Authentication: https://auth0.com/learn/json-web-tokens/
7. Stripe Documentation: https://stripe.com/docs
8. W3Schools Web Development: https://www.w3schools.com/
9. MDN Web Docs: https://developer.mozilla.org/en-US/