Real-time/Field-Based Research Project Report On

ECOMMERCE - WEBSITE

A dissertation submitted to the Jawaharlal Nehru Technological University, Hyderabad in partial fulfillment of the requirement for the award of a degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted by

G. Jashwanth Reddy (23B81A0580)

B. Sreerama Chandra (23B81A05B2)

M. Vignay Reddy (23B81A05C4)

Department of Computer Science and Engineering

CVR COLLEGE OF ENGINEERING

(An UGC Autonomous Institution, Affiliated to JNTUH, Accredited by NBA, and NAAC)
Vastunagar, Mangalpalli (V), Ibrahimpatnam (M), Ranga Reddy (Dist.) - 501510,
Telangana State.

2024-25

CVR COLLEGE OF ENGINEERING

(An UGC Autonomous Institution, Affiliated to JNTUH,
Accredited by NBA, and NAAC)
Vastunagar, Mangalpalli (V), Ibrahimpatnam (M),
Ranga Reddy (Dist.) - 501510, Telangana State.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project work entitled **"E-Commerce Website"** is being submitted by G. Jashwanth Reddy (23B81A0580), B. Sreerama Chandra (23B81A05B2) and M. Vignay Reddy (23B81A05C4) in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering,** during the academic year 2024-2025.

Professor-in-charge RFP

(Swamy Das)

Professor and Head, CSE
(Dr. A. Vani Vathsala)

DECLARATION

I hereby declare that this project report titled **"E-Commerce Website"** submitted to the Department of Computer Science and Engineering, CVR College of Engineering, is a record of original work done by me. The information and data given in the report is authentic to the best of my knowledge. This Real Time/Field-Based Research Project report is not submitted to any other university or institution for the award of any degree or diploma or published at any time before.

G. Jashwanth Reddy (23B81A0580)

B. Sreerama Chandra (23B81A05B2)

M. Vignay Reddy (23B81A05C4)

Date:

Place:

TABLE OF CONTENTS

ABSTRACT

This project aims to create an e-commerce website designed to facilitate online buying and selling of products. The platform will feature a clean, user-friendly interface that makes navigation intuitive for users of all skill levels. By organizing products into various categories, customers will be able to easily find what they are looking for. The website will also include search functionality to enhance the shopping experience.

Security is a top priority for the e-commerce site. It will offer secure payment options, ensuring that customers' financial information is protected during transactions. The registration process will be straightforward, allowing users to quickly create accounts to manage their purchases, track orders, and save favorite items. Additionally, customer support will be available to assist with any questions or issues that may arise.

The website will also incorporate social features, such as user reviews and ratings, to help customers make informed purchasing decisions. Other functionalities will include wish lists, personalized recommendations, and order tracking. By providing a comprehensive and convenient shopping experience, the e-commerce website aims to connect buyers and sellers effectively in the growing online marketplace.

Keywords: React.js, Stripe, Node.js, Redis, Mongo Db, Tailwind CSS, JavaScript.

1.INTRODUCTION

Online shopping is growing faster than ever, and we want to build an E-Commerce store that makes buying and selling easy. Our project will include everything needed for a smooth shopping experience, from secure login and payments to product management and an admin dashboard.
Customers will be able to browse products, add them to their cart, apply discounts, and make secure payments. Store owners will have an easy-to-use dashboard to manage products, track orders, and see sales reports. The platform will also include a coupon system sales analytics improve performance.

We will use React.js for the front end, Node.js for the back end, and MongoDB for storing data. The design will be clean and modern, using Tailwind CSS ensuring a smooth experience on both desktop and mobile devices.

Our goal is to create a fast, secure, and user-friendly online store that can grow over time. Whether for small businesses or large ones, this platform will simplify online selling, improve efficiency, and provide valuable insights through analytics. With a strong focus on security, performance, and user experience, this project will be a complete solution for e-commerce success.

1.1 MOTIVATION

E-commerce is growing rapidly, with more consumers preferring to shop online due to convenience and variety. A multi-vendor platform is a powerful solution that benefits both sellers and buyers while creating a scalable business model.

Growing Online Shopping Demand - As more people turn to online shopping, a multi-vendor marketplace enables multiple sellers to list their products, increasing product variety and attracting a larger customer base. This scalability makes it a profitable and sustainable business opportunity.
Easy for Sellers - Many small businesses struggle to set up an online store due to technical barriers and high costs. A multi-vendor platform allows sellers to list their products easily without worrying about website development, hosting, or payment processing. The platform takes care of logistics, making online selling more accessible.

Market Advantage - Competing with major marketplaces like Amazon and eBay may seem challenging, but focusing on a specific niche (electronics, fashion, handmade items, etc.) can help build a loyal and engaged customer base. This specialization allows businesses to stand out in a crowded market.

Simplified Management - A multi-vendor platform offers centralized control, making it easy to manage products, orders, payments, and customer interactions from a single dashboard. This ensures a smooth operation for both administrators and sellers.
By combining these advantages, this project aims to create a feature-rich, scalable, and user-friendly e-commerce marketplace that benefits both business owners and consumers

1.2 PROBLEM STATEMENT

Selling products online is challenging for many businesses due to high costs, technical difficulties, and complex management systems. At the same time, customers often face slow, unreliable, and unsecure shopping experiences. Additionally, order, payment, and inventory management can be overwhelming for both sellers and administrators.

- By simplifying e-commerce, this platform empowers small businesses, improves customer experience, and creates a profitable and scalable business model
- This project aims to develop a multi-vendor e-commerce platform that makes online selling and shopping easier by providing:
- **Hassle-Free Selling** - Sellers can list products effortlessly, without needing technical expertise or their own website.
- **Seamless Shopping Experience** - Customers enjoy a fast, secure, and smooth shopping process with reliable payments and easy navigation.
- **Efficient Administration -** Admins can manage everything from a centralized dashboard, including user accounts, orders, and transactions.
- **Advanced Features** - Built-in functionalities like discounts, order tracking, analytics, and sales insights improve efficiency for both sellers and admins.

1.3 PROJECT OBJECTIVES

The main objective of this project is to develop a secure, user-friendly, and efficient multi-vendor e-commerce platform that benefits both sellers and buyers while ensuring smooth management for administrators.

- **Simplified Selling** - Enable sellers to list, manage, and sell products effortlessly without needing technical knowledge.
- **Enhanced Shopping Experience** - Provide customers with a fast, secure, and seamless shopping journey, ensuring smooth navigation and checkout.
- **Centralized Management** - Offer admins a powerful dashboard to efficiently handle products, orders, payments, and user management in one place.
- **Optimized Performance** - Implement Redis caching and an efficient database to improve speed, reduce load times, and ensure system stability.
- **Scalability for Growth** - Design a future-ready platform that can handle an increasing number of users, products, and transactions over time.

- By focusing on these objectives, this project will redefine online shopping, making it easier and more accessible for sellers, buyers, and administrators.

1.4 PROJECT REPORT ORGANISATION

1.4.1 Introduction

- **Overview of the Project:** This section introduces the idea of your eCommerce website, what it aims to do, and why it's important.
- **Problem Statement:** Describes the problem the website is solving, such as making online shopping easier or improving the shopping experience.
- **Scope of the Project:** Lists the main features and functions your website will have, like product listings, shopping carts, and secure payments.

1.4.2 Literature Review / Existing Work

- **Current eCommerce Trends:** Talks about how other eCommerce websites work and the latest trends in the industry.
- **Limitations of Existing System**s: Explains what other websites might be lacking and how your project will improve or fix those problems.

1.4.3 System Design

- **Architecture Design:** Describes how the system will be set up, including how the website and its backend will work together.
- **User Interface (UI) Design**: Discusses how the website will look and how easy it will be for users to navigate.
- **Database Design:** Explains how the data (like products, customer info, orders) will be stored and organized.

1.4.4 Requirement Analysis

User Requirements: Lists of what the users (shoppers and admins) need, like easy account management, simple checkout, and good customer support.
Functional Requirements: Describes the features the website must have, like searching products, adding to the cart, and processing payments.
Non-Functional Requirements: Talks about other important aspects like security (protecting customer info) and performance (fast website speed).

2.LITERATURE SURVEY

E-commerce platforms have made online shopping easier, but excessive costs, complex setups, and security concerns make it difficult for small businesses to compete. Research highlights key areas for building an efficient and scalable marketplace:

- Traditional Platforms - Marketplaces like Amazon and Shopify require high setup costs and technical skills, making them less accessible for small sellers.
- **Secure Payments** - Gateways like Stripe and PayPal improve trust and prevent fraud, leading to higher conversion rates.
- **Authentication & Security** – JWT and OAuth provide secure login systems, preventing unauthorized access and protecting user data.
- **Database & Performance** – MongoDB and Redis ensure fast and efficient data management, reducing server load and improving user experience.
- **Admin Dashboards & Analytics** – Real-time tracking and insights help businesses manage orders, optimize pricing, and improve sales.
- This project integrates these best practices to build a secure, high-performance, and user-friendly multi-vendor e-commerce platform

2.1 Existing Work

- This section looks at what eCommerce websites already offer and the common features that most websites have today:

Technologies Used:

- **Website Platforms:** Many eCommerce websites use platforms like Shopify, WooCommerce, or Magento to build and manage online stores.
- **Website Technologies:** Websites are built using technologies like HTML, CSS, JavaScript (for looks and behavior), and back-end languages like PHP or Python (for functionality).
- **Payment Options:** Sites offer payment gateways like PayPal or Stripe for easy transactions.

Key Features:

- **Product Listings:** Products are organized into categories and can be searched and filtered by customers.
- **Shopping Cart & Checkout:** Customers can add items to their cart, proceed to checkout, and make payments.
- **User Accounts**: Customers can sign up, log in, track their orders, and save preferences.
- **Mobile-Friendly:** Many websites are designed to work well on mobile phones too.
- **Customer Support:** Options like live chats, FAQs, and contact forms are available for assistance.

Marketing & SEO: Features like promotions, discounts, and email marketing are used to attract more customers.

Security: Websites use encryption (SSL certificates) to keep customer data safe.

- Analytics: Tools like Google Analytics track how customers use the site and which products are popular.
-

2.2 Limitations of Existing Work

- Here, we talk about the problems or shortcomings that exist in current eCommerce websites:
- User Experience (UX) Issues:

Hard to Navigate: Some websites have complicated menus or too much information, making it difficult for users to find what they want.

- **Slow Loading:** Websites with too many images or scripts can take too long to load, frustrating customers.
- **Long Checkout Process:** Some websites require many steps to complete a purchase, which can cause customers to leave without buying.
- **Poor Mobile Experience:** Even though many sites are mobile-friendly, some don't work as smoothly on smartphones or tablets.
- **Lack of Personalization:** Some websites don't suggest products based on customers' previous actions or interests, missing chances to make more sales.
- **Security Concerns:**
- **Vulnerable to Attacks:** Some eCommerce sites don't have strong enough security, which can make them easy targets for hackers.
- **Privacy Issues:** Some websites don't fully protect user data, which can lead to privacy breaches or violations of data protection laws (like GDPR)

3.REQUIREMENT ANALYSIS

Functional Requirements

- This project focuses on building a multi-vendor e-commerce platform with an admin dashboard, supporting secure transactions, real-time product management, and advanced analytics.

Functional requirements include:

- **User Authentication & Authorization** - Secure login/signup
- **Product & Category Management** - Vendors can list, update, and remove products, categorized for better organization.
- **Shopping Cart & Checkout** - Users can add items to their cart, apply coupon codes, and complete purchases via Stripe.
- **Order Management** - Users can track their orders, and sellers can manage order fulfillment.

E) **Admin Dashboard** - Centralized control for managing users, vendors, products, and analytics.

F) **Payment Integration** - Secure Stripe API integration for payments.

G) Real-time Notifications - Email and dashboard alerts for order updates.

H) Sales Analytics & Reports - Generate reports on sales, revenue, and product performance.

Role-based Access Control - Admin, vendor, and customer roles for secure operations.

Non-Functional Requirements

The system must ensure high performance, security, scalability, and reliability for optimal functionality.

Portability

The application runs across multiple devices, including desktops, tablets, and mobile phones.
It is compatible with major browsers and operating systems without additional customization.

•**Security**

- All transactions are encrypted to ensure data security and integrity.
- Secure authentication and authorization mechanisms.
- Role-based access to limit unauthorized operations.

•**Maintainability**

Modular architecture allows easy updates and debugging. B) Well-documented codebase ensures smooth development and scalability.

•**Reliability**

The system handles multiple concurrent users without crashes. B) Error handling and fallback mechanisms ensure smooth operations.

•**Performance**

Optimized database queries for fast data retrieval. B) Redis caching reduces server load and speeds up response time.

•**Scalability**

Designed to handle increased traffic with horizontal scaling capabilities. B) Supports additional vendor and product onboarding without performance degradation.

•**Flexibility**

Easily integrates with third-party services such as planning, analytics, and marketing tools.

- Adaptable for future enhancements like AI-driven recommendations.

3.1 SOFTWARE REQUIREMENTS

- **Backend:** Node.js, Express.js
- **Database:** MongoDB
- **Frontend:** React.js, Tailwind CSS
- **Development Environment:** Visual Studio Code
- **Version Control:** GitHub
- **UML Design Tools:** StarUML

3.2 HARDWARE REQUIREMENTS

- **Processor:** Intel i5 or higher
- **RAM:** 8GB
- **Storage:** 512GB SSD
- **Internet Connection:** High-speed broadband for seamless performance
- This SRS document outlines the technical and functional requirements to ensure a robust, scalable, and secure e-commerce platform

3.3 USER REQUIREMENTS

Customer (Buyer) Requirements:

a. Easy Account Management: Customers should be able to easily create, update, and manage their accounts.

b. Browsing and Searching: Customers should be able to search for and browse products quickly using filters and categories.

c. Shopping Cart and Checkout: Customers should be able to add products to the cart, proceed to checkout, and make payments smoothly.

d. Order Management: Customers should be able to track, manage, and view their past orders.

e. Customer Support: Customers should have easy access to support for help with orders or issues.

- Admin (Store Owner or Manager) Requirements:
- **a. Product Management:** Admins should be able to add, update, and remove products from the store.
- **b. Order Management**: Admins should be able to manage and track customer orders from purchase to delivery.
- **c. Customer Management:** Admins should have access to manage customer information and account details.

- **Business/Marketing User Requirements:**
- **a. Sales and Marketing:** Marketing users should be able to create and manage promotions, discounts, and track sales performance.

4.SYSTEM DESIGN

4.0 PROPOSED SYSTEM ARCHITECTURE

1. Frontend Layer (Client-Side)

This is where users interact with the website. The focus is on delivering a good user experience.

- **Web Application**:
    - Built with technologies such as **HTML5**, **JavaScript** (React.js, Angular).
    - **Mobile Applications** (optional): Native or hybrid apps for iOS and Android (React Native, Flutter).
- **UI/UX**:
    - Responsive design to ensure compatibility across various devices (mobile, tablet, desktop).
    - Search bars, product displays, filtering options, and checkout process.
- **User Authentication**:
    - Login/Sign-up screens, social media login, etc.

2. Database Layer

This is where all data is stored. It should be scalable, reliable, and secure.

- **NoSQL Database** (e.g., **MongoDB**): For unstructured data like product reviews, browsing history, and session data.
- **Caching**:
  Redis or **Memcached** for caching frequently accessed data, improving performance.

3. Payment Gateway Layer

This is an essential part of any eCommerce platform for handling transactions.

- **Payment Processing**:
  - Integrates with third-party services like **Stripe**, etc., for secure payment processing.
- **Encryption**:
  - **SSL/TLS** protocols for secure payment processing.

4. Admin Dashboard

The administrative side for managing the eCommerce platform.

Admin Panel:

- Built using a framework such as **React**
- Features: Product catalog management, order management, user management, inventory control, and financial reporting.

Analytics:

- Integration with Google Analytics or custom dashboards for tracking traffic, sales, and other KPIs.

4.1 PROPOSED METHODS/ ALGORITHM

Proposed Methods:

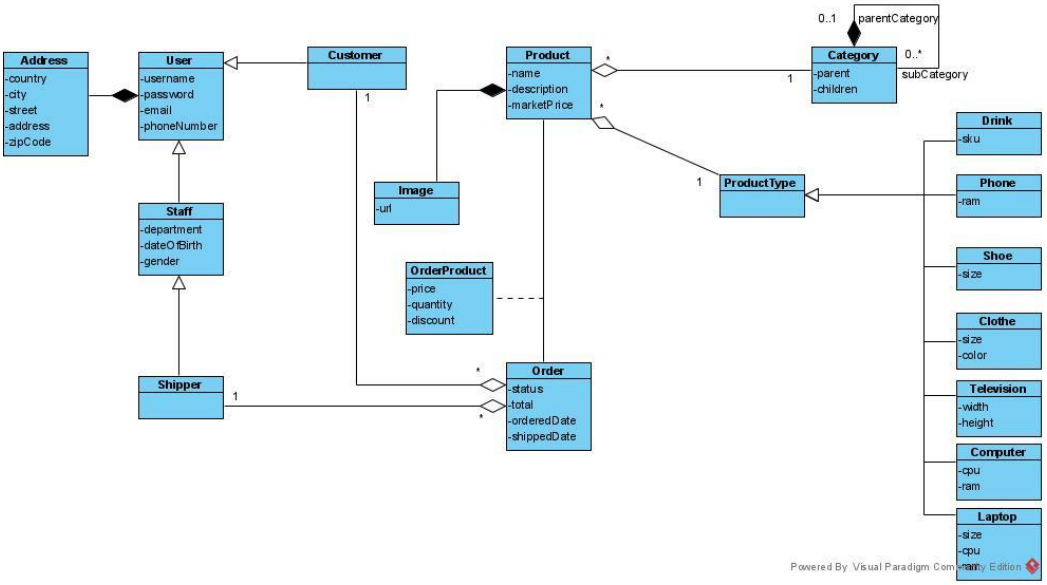
- Product Search and Filtering
- **Method**: Use of tools like Elasticsearch.
- **Explanation**: This helps customers quickly search for products and apply filters (like price, category, etc.) to find exactly what they're looking for, even if there are thousands of products.

- **Recommendation System**
- **Algorithm**: Collaborative Filtering or Content-Based Filtering.
- **Explanation**: This suggests products to users based on things they've bought or looked at before or based on what other customers with similar interests have bought.

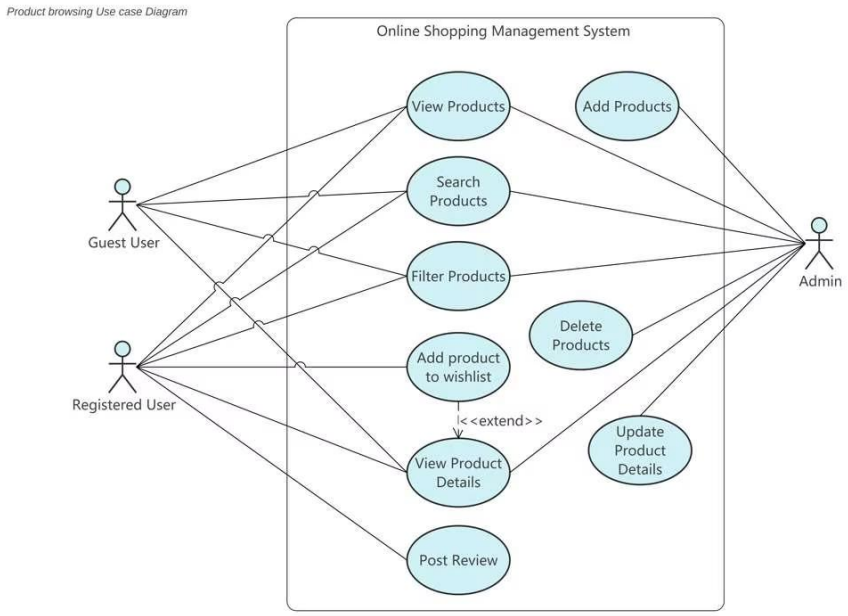
- **Payment Security**
- **Method**: Secure Payment Algorithms like SSL/TLS encryption.
- **Explanation**: These algorithms protect customer payment details (like credit card info) by encrypting it, ensuring that it's secure and safe from hackers.

4.2 CLASS / USE CASE / ACTIVITY / SEQUENCE DIAGRAMS

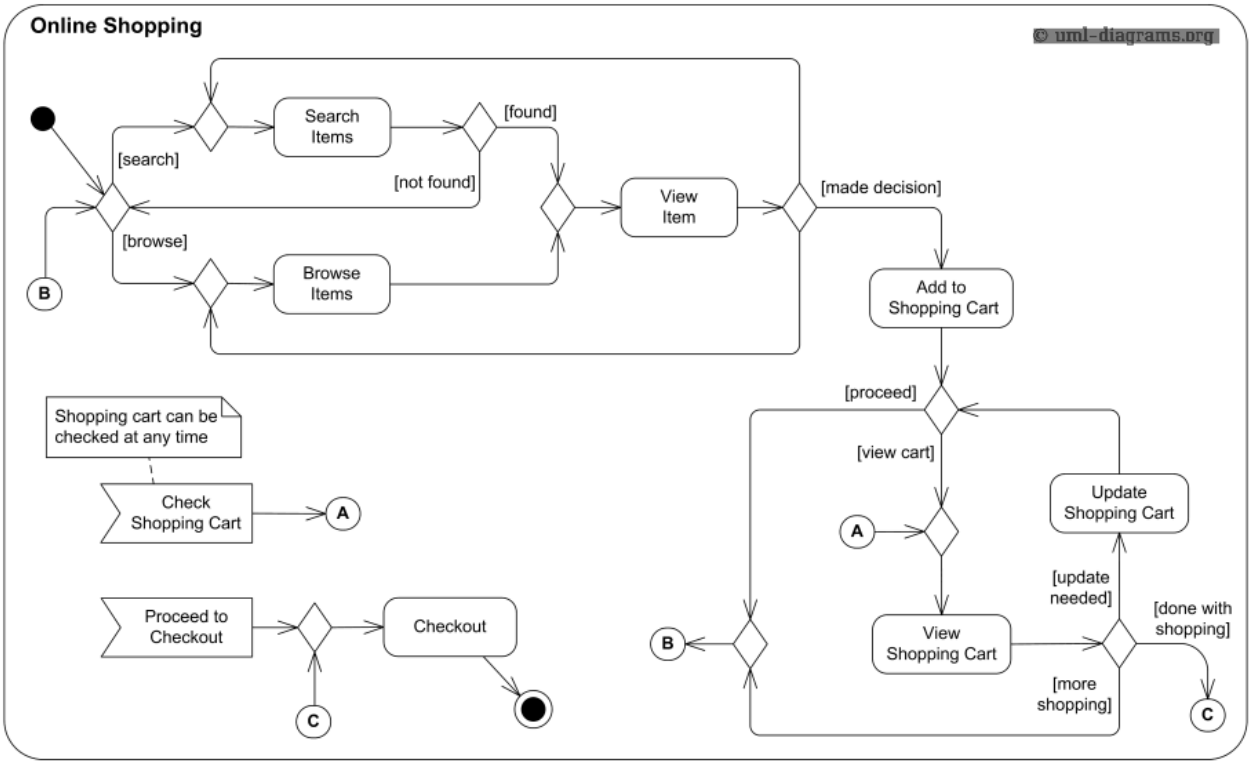
Class Diagram



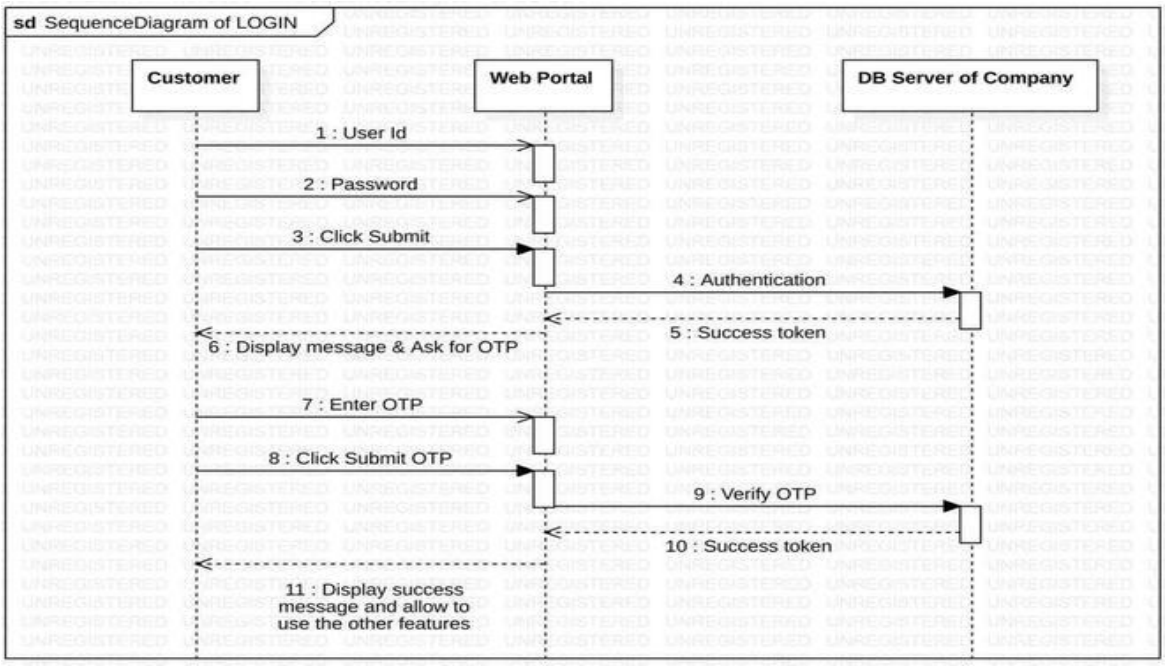
Use Case Diagram



Activity Diagram



Sequence Diagram



4.3 DATASETS AND TECHNOLOGY STACK

Frontend (Client-side):

- **HTML:** Structural foundation of the website.
- **CSS:** Styling elements for visual appearance.
- **JavaScript:** Interactive features and dynamic content.

Frontend Frameworks:

- **React**: Popular for component-based UI, flexibility, and performance.

Backend (Server-side):

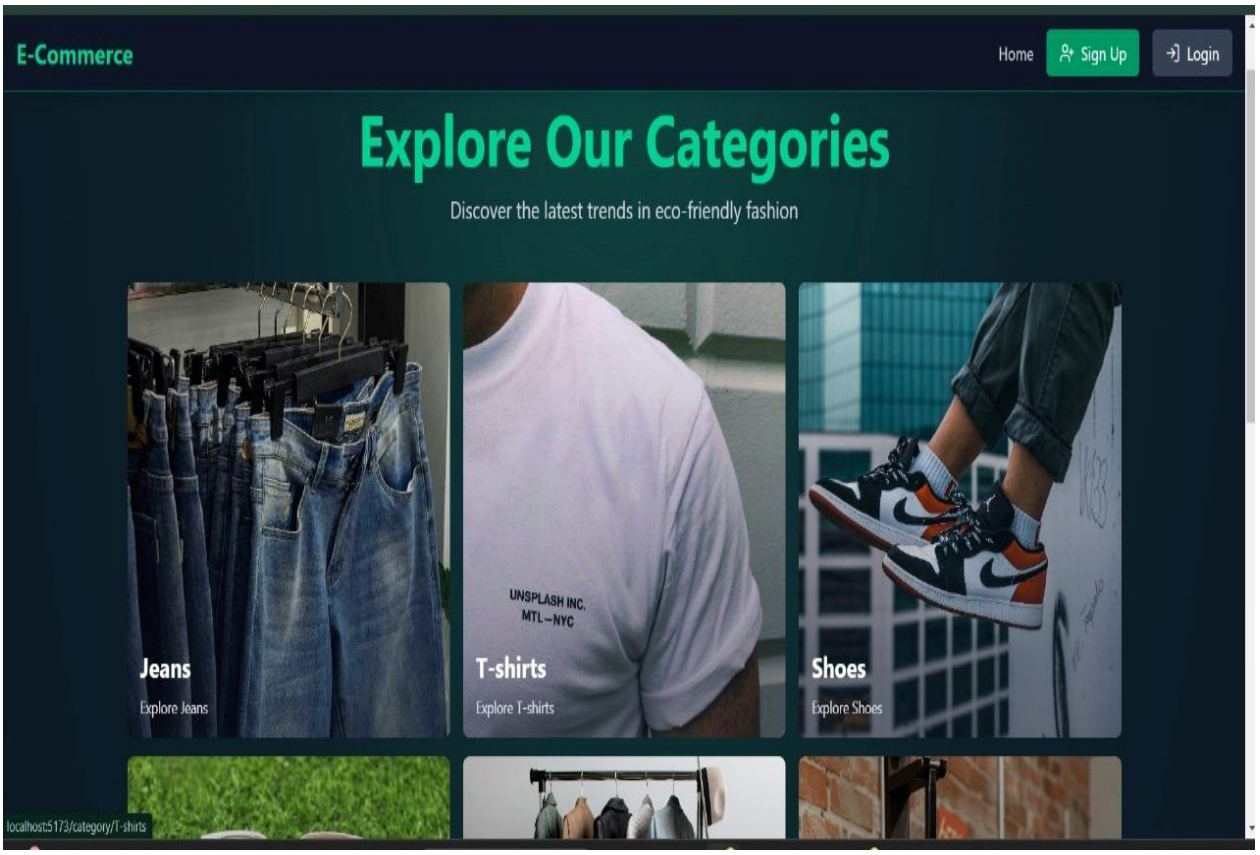
- Stripe
- MongoDB
- Redis
- Server-side Language:
- Node.js: JavaScript runtime for building scalable backend applications.

Backend Framework:

- Express.js: Minimalist Node.js framework for building APIs.

5.IMPLEMENTATION

5.1 Front page Screenshot


E-Commerce
Home
Sign Up
Login
Explore Our Categories
Discover the latest trends in eco-friendly fashion
Jeans
Explore Jeans
UNSPLASH INC.
MTL—NYC
T-shirts
Explore T-shirts
Shoes
Explore Shoes
localhost:5173/category/T-shirts

Server.jsx :

```
import cookieParser from 'cookie-parser';
import cors from 'cors';

import authRoutes from "./routes/auth.route.js"
import productRoutes from "./routes/product.route.js";
import cartRoutes from "./routes/cart.route.js";
import couponRoutes from "./routes/coupon.route.js";
import paymentRoutes from "./routes/payment.route.js";
import analyticsRoutes from "./routes/analytics.route.js";

import { connectDB } from "./lib/db.js";

dotenv.config();
const app= express();
const PORT = process.env.PORT || 5000;

app.use(express.json()); // allows you to parse the body of the request
app.use(cookieParser());

// Configure CORS
app.use(cors({
  origin: 'http://localhost:5173',
  credentials: true
}));
app.use("/api/auth",authRoutes);
app.use("/api/products",productRoutes);
app.use("/api/cart",cartRoutes);
app.use("/api/coupons",couponRoutes);
app.use("/api/payments",paymentRoutes);
app.use("/api/analytics",analyticsRoutes);


app.listen(PORT,() =>
{
    console.log("Server is running on http://localhost:"+PORT);
    connectDB();
});

//pan18zL9H8GZRlpI
```

Main.jsx:

```
import { StrictMode } from "react";
import { createRoot } from "react-dom/client";
import App from "./App.jsx";
import "./index.css";

import { BrowserRouter } from "react-router-dom";

createRoot(document.getElementById("root")).render(
    <StrictMode>
        <BrowserRouter>
            <App />
        </BrowserRouter>
    </StrictMode>
);
```

GitHub Link For Entire CODE:

https://github.com/vignayreddy/Ecommerce-Service

6. CONCLUSIONS

6.1 CONCLUSIONS

- **User-Friendly Experience**: The site must be easy to use for customers, with simple navigation,

search, and checkout processes.

- **Effective Product and Order Management**: Admins need tools to manage products, stock,
- **Security**: The website should protect customer information, especially payment details.
- **Performance**: The site should be fast and handle high traffic smoothly, especially during busy times.
- **Personalization**: The site should offer personalized recommendations to keep customers engaged.

6.2 FUTURE SCOPE

AI and Smart Technology:

- **Personalized Shopping**: AI will help suggest products based on what customers like or buy.
- **Chatbots**: AI-powered chatbots will help answer questions and assist customers instantly
- **Faster Delivery:**
- **Same-Day Delivery**: Faster shipping options like same-day delivery will become more common.
- **Drones**: In the future, drones might deliver products to your door quickly.

Blockchain for Security:

- **Safe Payments**: Blockchain technology can make online payments more secure and transparent.

Eco-Friendly Practices:

- **Sustainability**: More stores will focus on using eco-friendly packaging and supporting environmentally friendly brands.

REFERENCES

Articles:

The Rise of E-Commerce: How Online Shopping is Transforming Retail

Explore the growth of e-commerce, the factors driving its popularity, and its impact on traditional retail.

Creating a User-Friendly E-Commerce Website: Best Practices

Discuss design principles and features that enhance user experience, such as navigation, layout, and accessibility.

Security in E-Commerce: Protecting Your Customers' Data

Examine the importance of data security, common threats, and strategies for implementing secure payment options.

Websites:

⇨ Commands – npm run dev for executing.
⇨ http://localhost:5173/ --Default