**A PROJECT REPORT**

On

**SHOPEZ : One Stop Shop For Online Purchases**

by

##### UMMADI LAKSHMI ARAVIND REDDY (22FE1A4450)

##### TAMMISETTY LAKSHMI PRASANNA

##### (22FE1A4446)

##### SYAMALA UMA MAHESWARA REDDY

##### (22FE1A1253)

##### SURANGI GNANESH

##### (22FE1A1252)

Under the guidance of

**Ganesh M**

# TABLE OF CONTENTS

|  |  |
| --- | --- |
| **NAME OF THE CONTENT** | **PAGE NOs** |
| 1.ABSTRACT | 3 |
| 2.INTRODUCTION | 4 |
| 3.MODULE DESCRIPTION | 5-12 |
| 4.USER INTERFACE | 13-14 |
| 5.TESTING | 15 |
| 6.RESULT | 16-17 |
| 7.FUTURE ENHANCEMENT. | 18 |
| 8.CONCLUSION | 19 |

**ABSTRACT**

**ShopeZ** is an all-in-one e-commerce platform developed under the name **ALPHA Works,** designed to redefine the online shopping experience through convenience, variety, and user-centric design. Built using the MERN stack (MongoDB, Express.js, React.js, Node.js), ShopeZ enables seamless interaction between customers and sellers, offering a robust and scalable infrastructure for managing products, orders, and secure payments. Users can browse a wide selection of products across diverse categories—ranging from electronics to fashion and home essentials—using a clean, responsive interface tailored for both desktop and mobile users. The platform features intuitive navigation, real-time inventory updates, and a personalized shopping experience through intelligent product recommendations. Sellers can easily manage their product listings, track sales, and engage with customers through a dedicated seller dashboard. Customers benefit from features like secure checkout, order tracking, and a transparent return policy. An integrated admin panel ensures platform integrity by overseeing listings, transactions, and user activities. ShopeZ brings automation, analytics, and flexibility together in a modular architecture, empowering both sellers and buyers in a dynamic digital marketplace designed for growth, trust, and ease of use.

# INTRODUCTION

**ShopeZ** is a comprehensive e-commerce platform developed as a one-stop shop for online purchases, designed to transform the way consumers explore and buy products across multiple categories. With a focus on user experience, **ShopeZ** combines modern design, secure transactions, and dynamic product discovery to provide an engaging and seamless shopping journey. The platform enables users to browse, filter, and purchase products effortlessly—ranging from daily essentials to specialized accessories—while ensuring speed, convenience, and trust in every interaction.

Built using the powerful MERN stack—MongoDB, Express.js, React.js, and Node.js—**ShopeZ** delivers a scalable and responsive infrastructure that supports real-time inventory updates, secure user authentication, and personalized recommendations. Buyers can view detailed product pages, add items to wishlists or carts, and track orders in real-time, while sellers can manage product listings, monitor sales, and communicate with customers through an intuitive dashboard. Integrated payment gateways and order fulfillment modules ensure a complete end-to-end e-commerce experience.

**ShopeZ** is engineered to meet the demands of modern online retail with features like mobile-friendly design, intelligent search, role-based access control, and admin-level monitoring. Whether for individual shoppers or growing small businesses, the platform is equipped to handle high-volume traffic, promotional campaigns, and seasonal spikes. With automation-enhanced processes, notification systems, and robust backend support, ShopeZ aims to redefine digital commerce by making online shopping more accessible, efficient, and enjoyable for all.

# MODULE DESCRIPTION

## Project Overview

#### Purpose:

#### The purpose of ShopeZ is to deliver a user-friendly, secure, and scalable e-commerce platform that bridges the gap between buyers and sellers by offering a seamless online shopping experience. ShopeZ aims to address common challenges faced in traditional and digital marketplaces, such as limited product visibility, poor customer service, and inefficient order management. The project's goal is to empower businesses of all sizes to showcase their products while providing customers with a reliable platform to discover, compare, and purchase items across a wide range of categories. ShopeZ is committed to fostering trust, convenience, and satisfaction for both vendors and consumers in the digital retail ecosystem.

#### Features:

* **User Registration & Authentication**: Secure sign-up and login for customers and sellers with JWT-based session management.
* **Product Listing & Management :** Sellers can list products with images, descriptions, prices, and inventory tracking. Includes categories and tags for better organization.
* **Customer Profiles:** Customers can manage personal details, view order history, save wishlists, and track shipments.
* **Shopping Cart & Checkout:** Seamless cart functionality with dynamic updates, discount codes, and secure multi-step checkout process.
* **Order Management:** Real-time order tracking, status updates, and purchase history for both buyers and sellers.
* **Payment Integration:** Secure payment gateway integration supporting cards, UPI, wallets, and COD options.
* **Admin Dashboard:** Admins oversee all platform activity, including user management, product approvals, order disputes, and analytics.
* **Review & Rating System:** Customers can rate products and leave reviews, helping future buyers make informed decisions.
* **Search & Filter:** Advanced search with filters by price, category, ratings, and availability to enhance product discovery.
* **Responsive UI:** Mobile-first design using Bootstrap and Material UI for a smooth shopping experience across all devices.
* **Notification System:** Real-time alerts for order updates, delivery status, offers, and promotional campaigns.
* **Promotions & Discounts:** Sellers and admins can create limited-time offers, coupon codes, and seasonal discounts.

## Architecture

#### Frontend:

The frontend of **ShopeZ** is developed using **React.js**, a powerful component-based library that enables a fast, interactive, and modular user experience. The UI is crafted with a combination of **Tailwind CSS** and **Material UI**, ensuring a modern, responsive layout that works seamlessly across desktop and mobile devices..State management is implemented through React's built-in hooks like useState, useEffect, and useContext, with optional integration of **Redux Toolkit** for more complex data flows such as user sessions, shopping cart states, and real-time updates. The frontend offers **role-specific interfaces** tailored to customers, vendors, and administrators, each with customized forms and interactive elements to match their workflows.

#### Backend:

The backend is developed using **Node.js** and the **Express.js** framework, which provides a lightweight, modular, and fast environment for server-side development. RESTful APIs are implemented to handle user authentication, project creation, bidding, messaging, and administrative actions. The backend includes middleware for token-based authentication (using JWT), request validation, error handling, and access control. Controllers manage the logic for each route, ensuring separation of concerns and clean code organization.

#### Database:

The platform uses **MongoDB** as the database, a NoSQL system suitable for flexible and scalable data storage. Mongoose is used as the ODM (Object Data Modeling) library to define schemas and manage interactions. Key collections include:

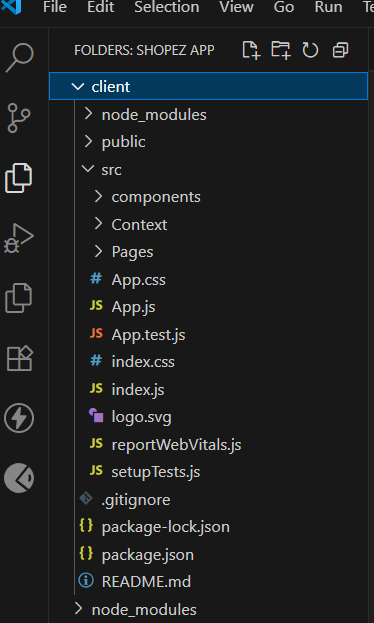
* users: Stores client and freelancer profiles
* projects: Contains project details, statuses, and assigned users
* bids: Tracks proposals and offers from freelancers
* messages: Stores chat histories between users
* reviews: Captures feedback and ratings for completed projects

The database is optimized for quick access and supports relational referencing where needed (e.g., linking users to projects or reviews). Environment variables securely store the MongoDB connection URI.

## Folder Structure

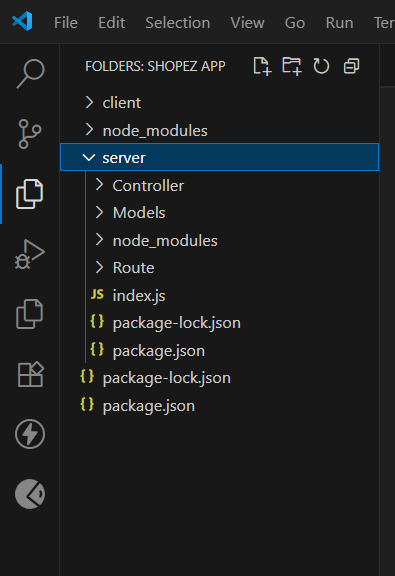
#### Client (React Frontend):

The client folder contains all the source code related to the frontend built with React. It follows a modular structure to separate UI components, pages, services, and static assets.



#### Server (Node.js Backend):

The server folder contains backend logic, route handlers, controllers, models, and middleware necessary to power the API.



## Setup Instructions

#### Prerequisites:

Before setting up the project, ensure the following software is installed on your system:

* [Node.js](https://nodejs.org/) (v18 or higher)
* [MongoDB](https://www.mongodb.com/) (local or MongoDB Atlas cloud setup)
* npm (comes with Node.js) or yarn
* Git (for cloning the repository)
* Code editor (e.g., VS Code)

**Installation:**

Follow the steps below to set up the project locally:

### Clone the Repository:

git clone <https://github.com/TammisettyLakshmiPrasannaKumar/shopez-one-stop-shop-for-online-purchases/tree/main>

**Navigate to the Project Directory:**

cd SHOPEZ

### Install Frontend Dependencies:

npm install

### Install Backend Dependencies:

cd server npm install

### Setup Environment Variables:

* + Create a **.env** file in the server directory and add the following: MONGO\_URI=your\_mongodb\_connection\_string JWT\_SECRET=your\_jwt\_secret\_key

PORT=5000

### Start the Backend Server:

cd server

### Start the Frontend Server:

cd client npm start

Once both servers are running, open [http://localhost:3000](http://localhost:3000/) in your browser to access the application.

## API Documentation

This section outlines the key RESTful API endpoints exposed by the backend of **ShopeZ**. All protected routes require a valid JWT token passed via the Authorization header.

## Authentication & Authorization

SHOPEZ uses JWT (JSON Web Token) for secure, stateless authentication and authorization across the platform. The implementation ensures that only authenticated users can access protected routes and perform specific actions based on their assigned roles.

#### Key Concepts:

##### JWT (JSON Web Token)

Used for securing API endpoints. Tokens are generated upon successful login and include user identity and role in the payload.

##### TOKEN STORAGE

Once the user logs in, the JWT token is securely stored in the browser’s localStorage on the client side for subsequent requests.

##### Protected Routes

Middleware on the backend validates incoming tokens for all protected endpoints. Unauthorized requests without valid tokens are denied access.

##### Role-Based Access Control (RBAC)

Users are assigned specific roles during registration (client,admin). Each role has access to specific features and routes:

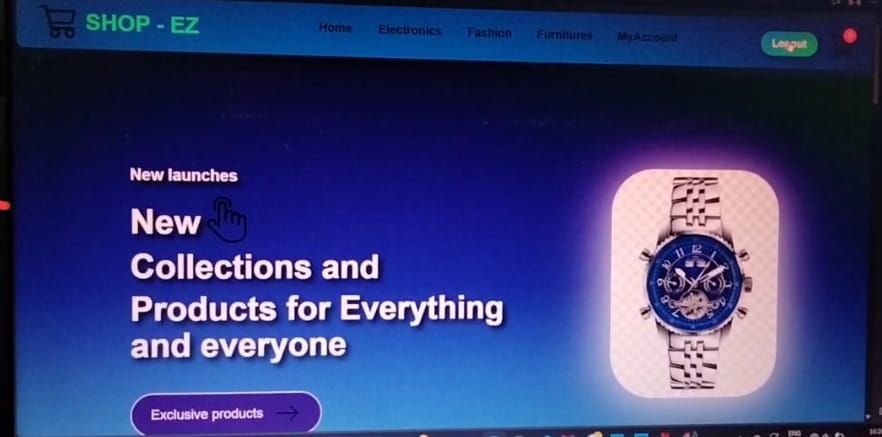
* + - Clients: Can Order products, review products,track order details.
    - Admins: Can oversee the entire platform, moderate content, and handle issues

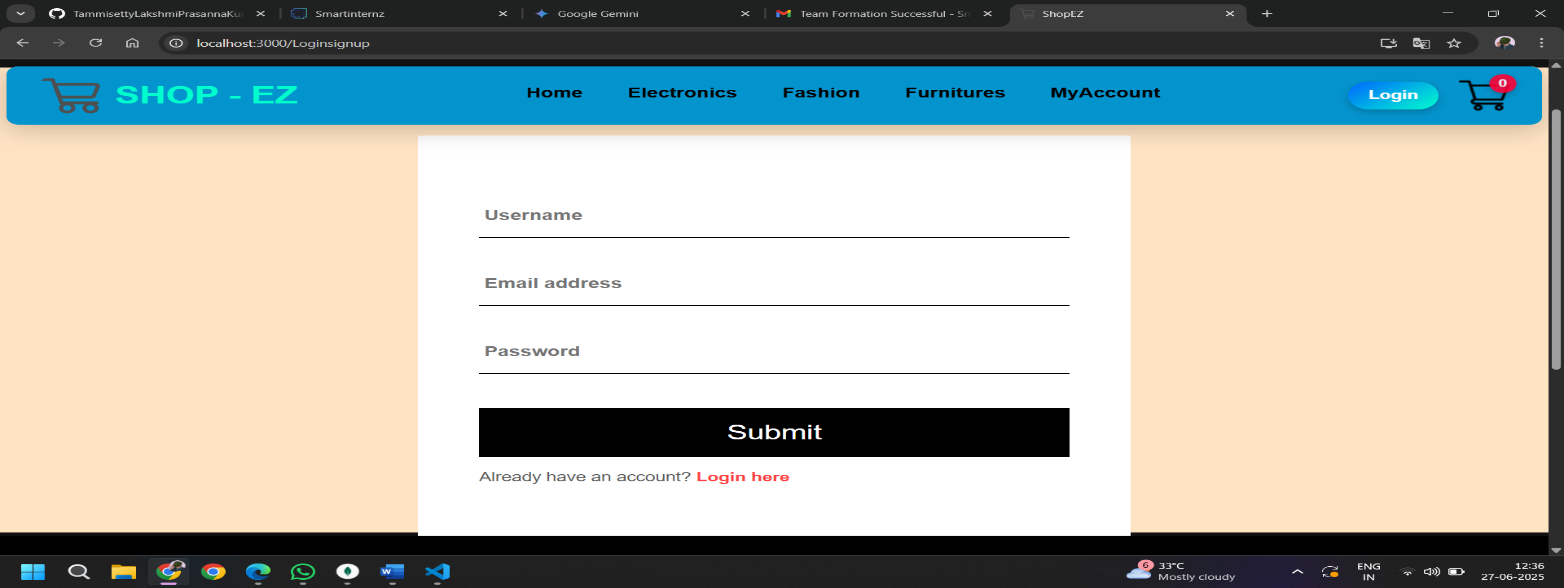
##### Authorization Headers:

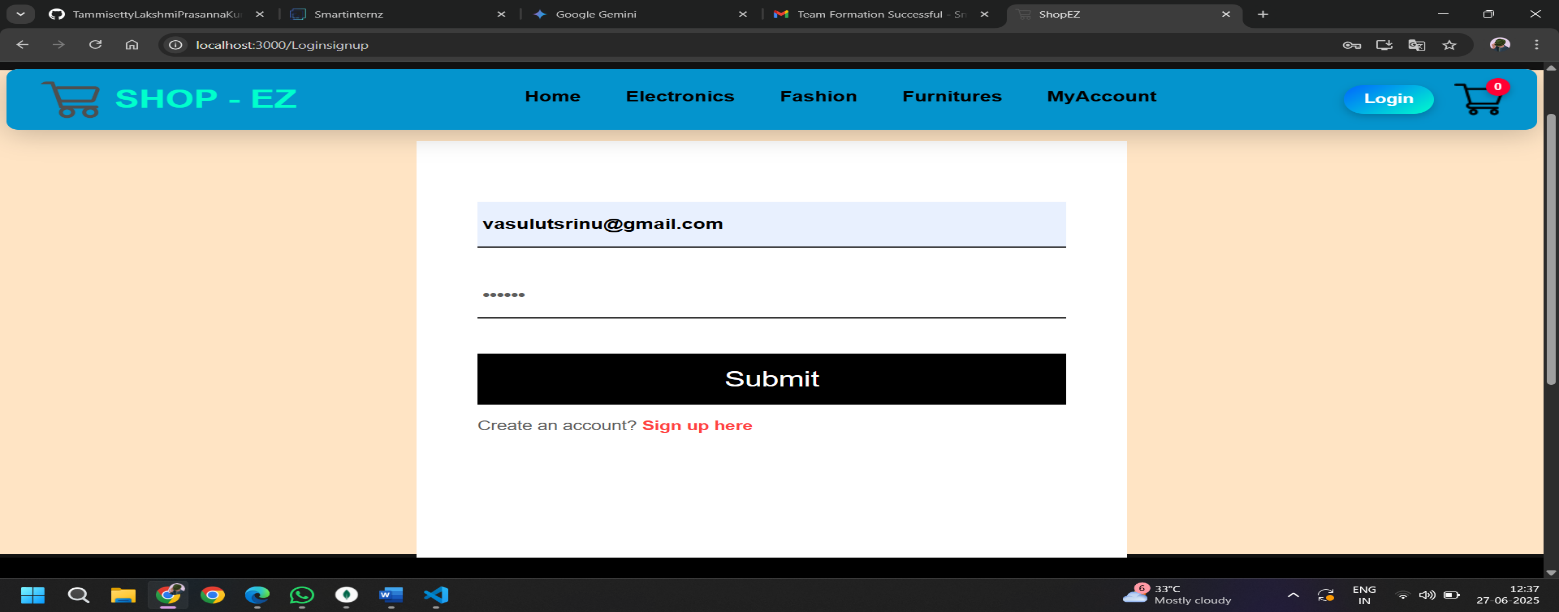
This token-based system ensures a secure, scalable, and flexible way to manage user sessions and permissions throughout the application.

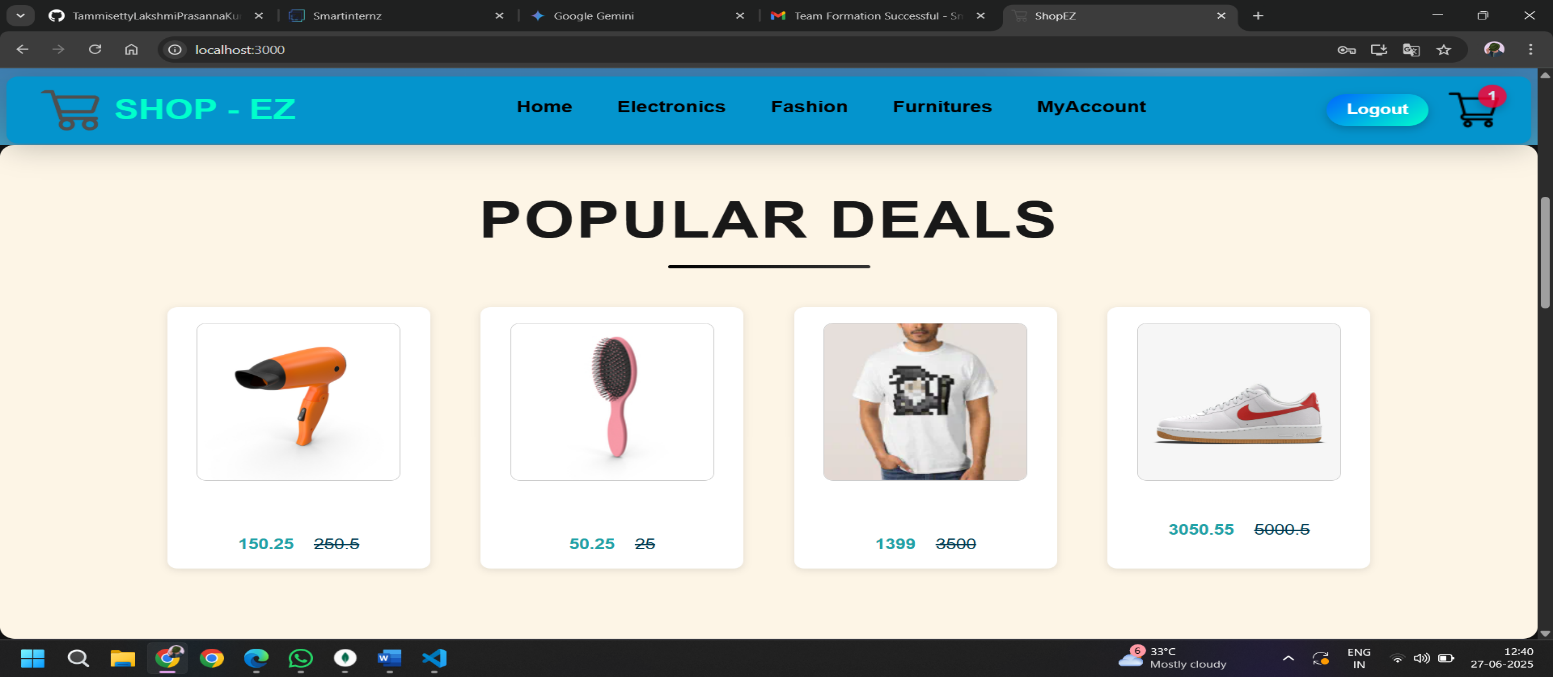
# USER INTERFACE

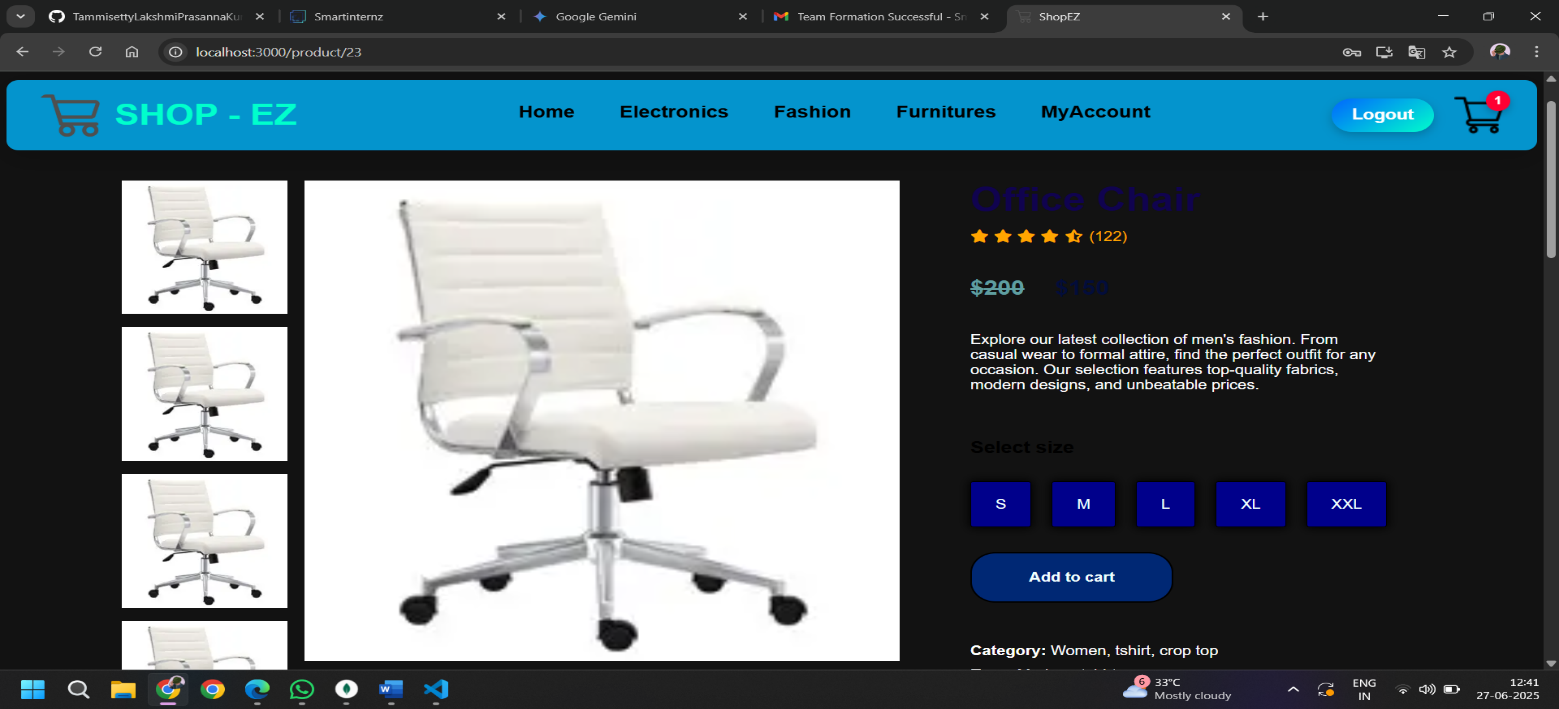
USER VIEW: -

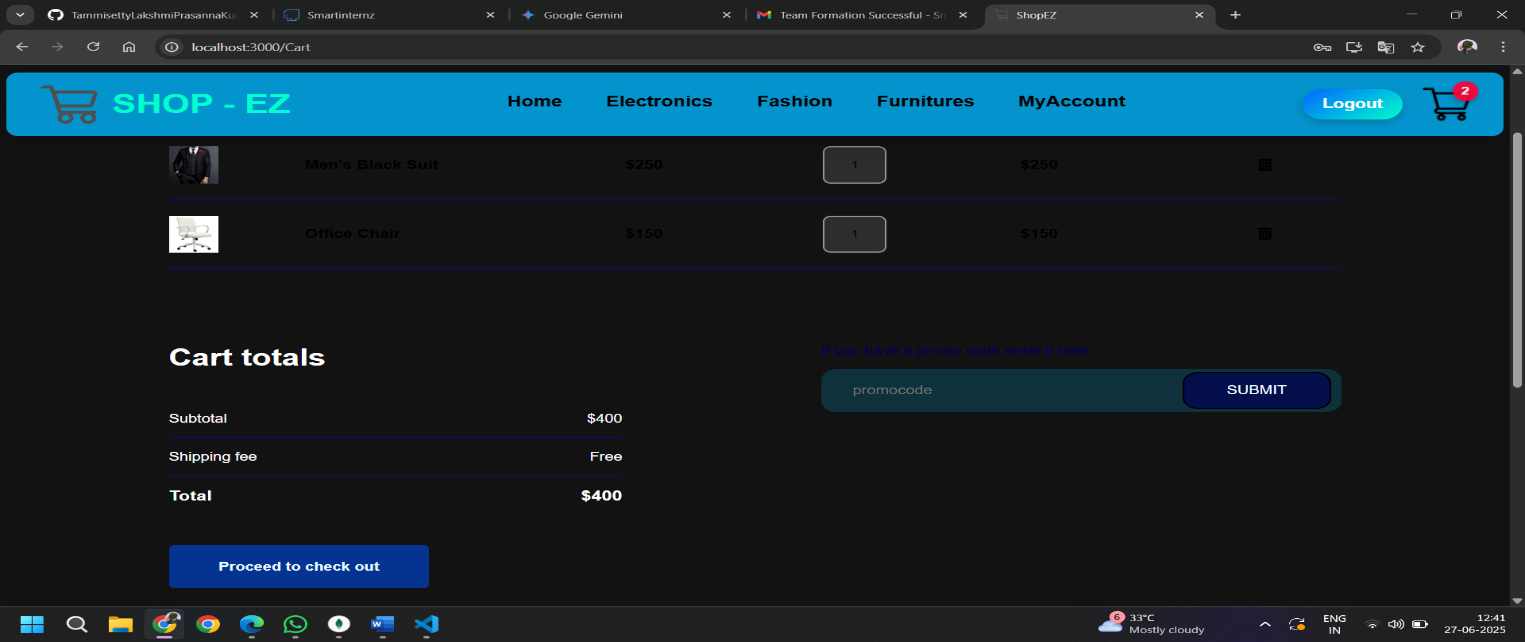












# TESTING

Testing is a crucial phase of the development process to ensure the functionality, performance, and reliability of the ShopEZ platform. Both manual and automated testing methods were employed to validate the frontend and backend components.

##### Manual Testing:

* **Frontend:**

Conducted using various user roles (freelancer, client, admin) to verify correct rendering of pages, input validation, and user interactions (login, project bidding, message exchange, etc.).

##### Backend:

Used **Postman** to test all API endpoints with valid and invalid inputs to ensure proper request/response handling and error management.

##### Responsiveness:

Manually tested the UI across different screen sizes (mobile, tablet, desktop) to ensure a consistent and responsive user experience.

##### Automated Testing (Planned / Partial):

* **Frontend(React):**

Integration of **Jest** and **React Testing Library** was initiated to test components like forms, buttons, and API responses.

##### Backend(Node.js+Express):

Planned to integrate **Mocha**, **Chai**, and **Supertest** to automate endpoint testing and simulate various edge cases.

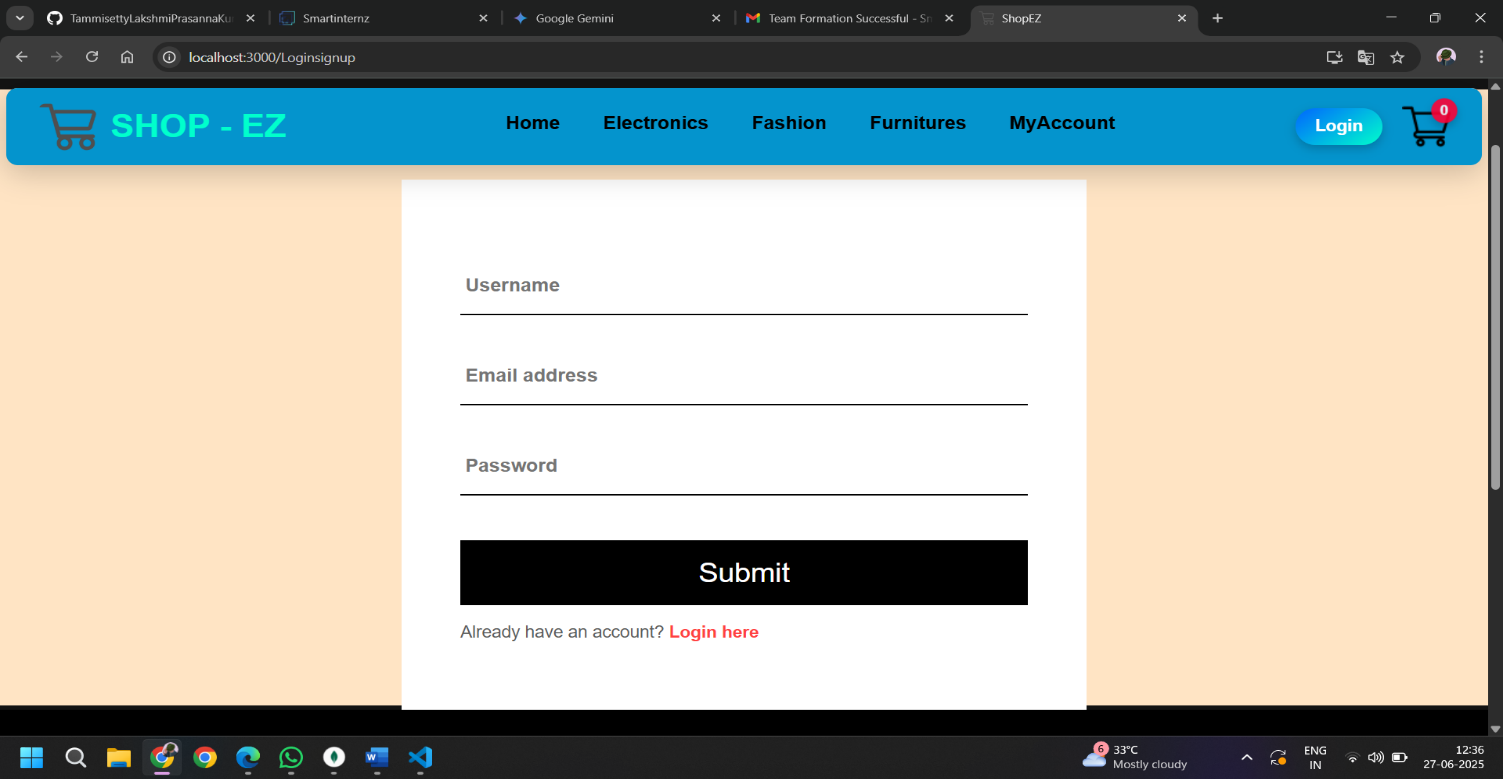
##### Key Areas Tested:

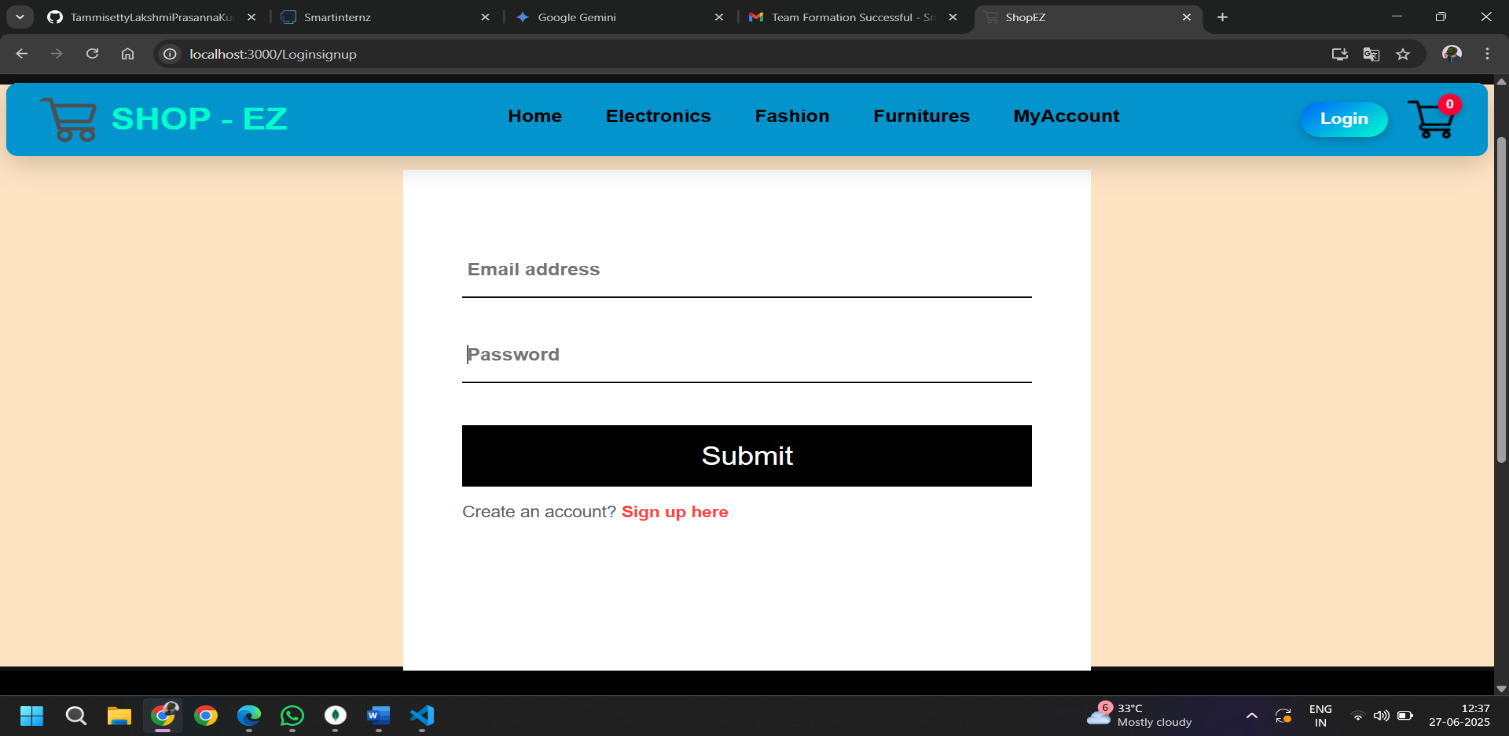
* + User registration and login with proper validation
  + Products, listing, filtering and price flows
  + Role-based access control for pages and APIs
  + Error and alert handling in both UI and server
  + Orders & Customer Reviews

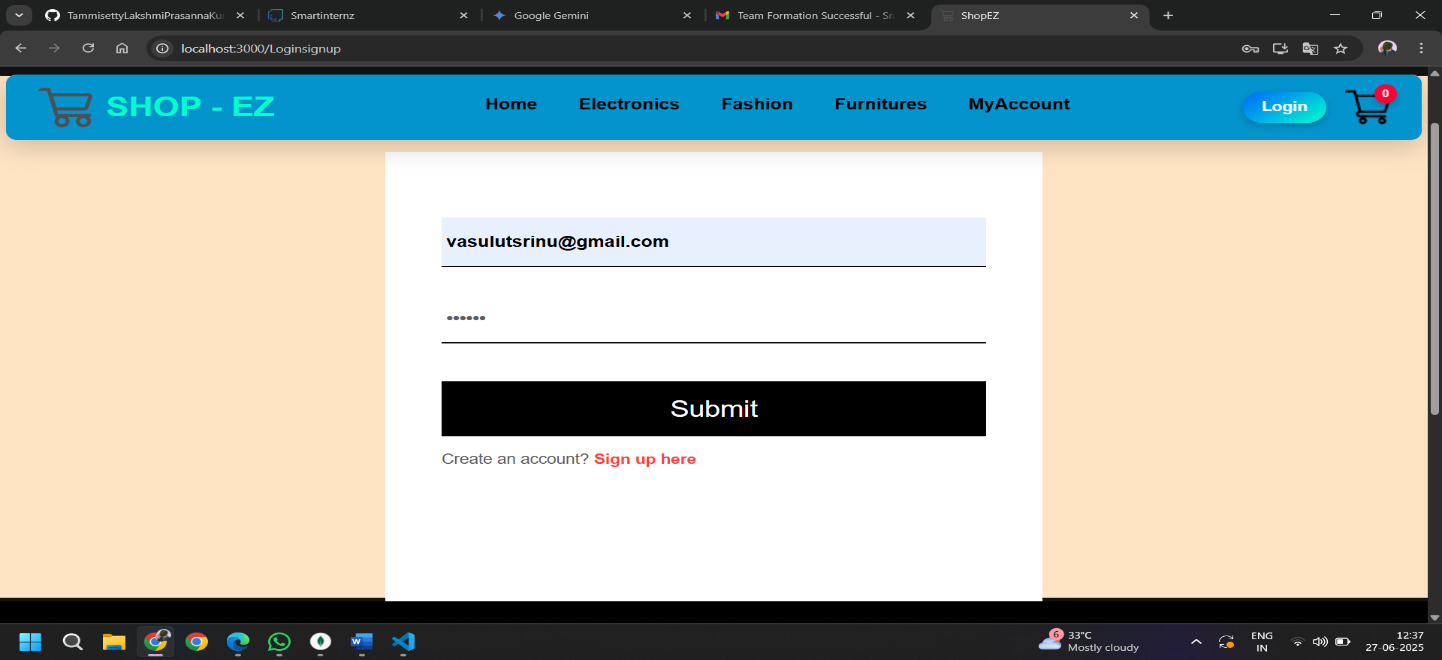
# RESULT

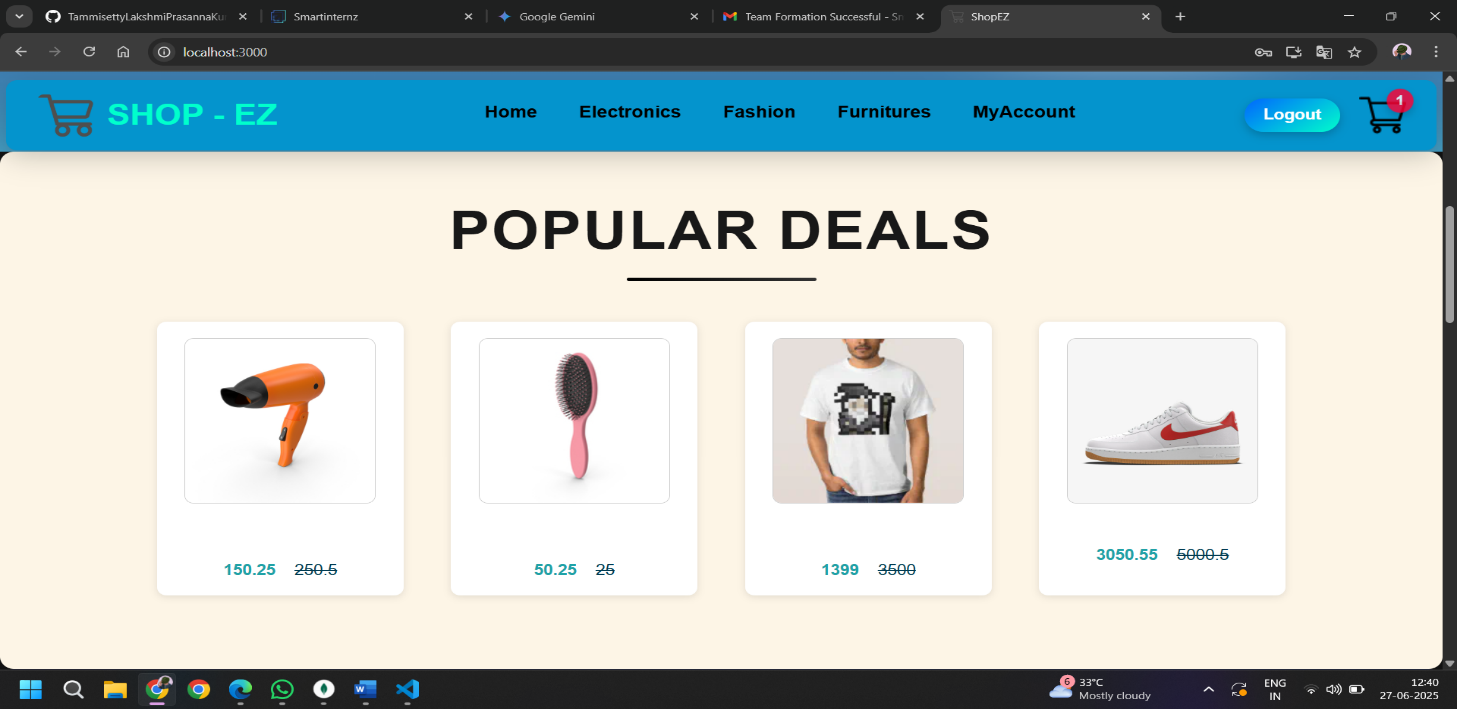
To demonstrate the functionality and user experience of **SHOPEZ**, a working demo and key UI screenshots have been provided:

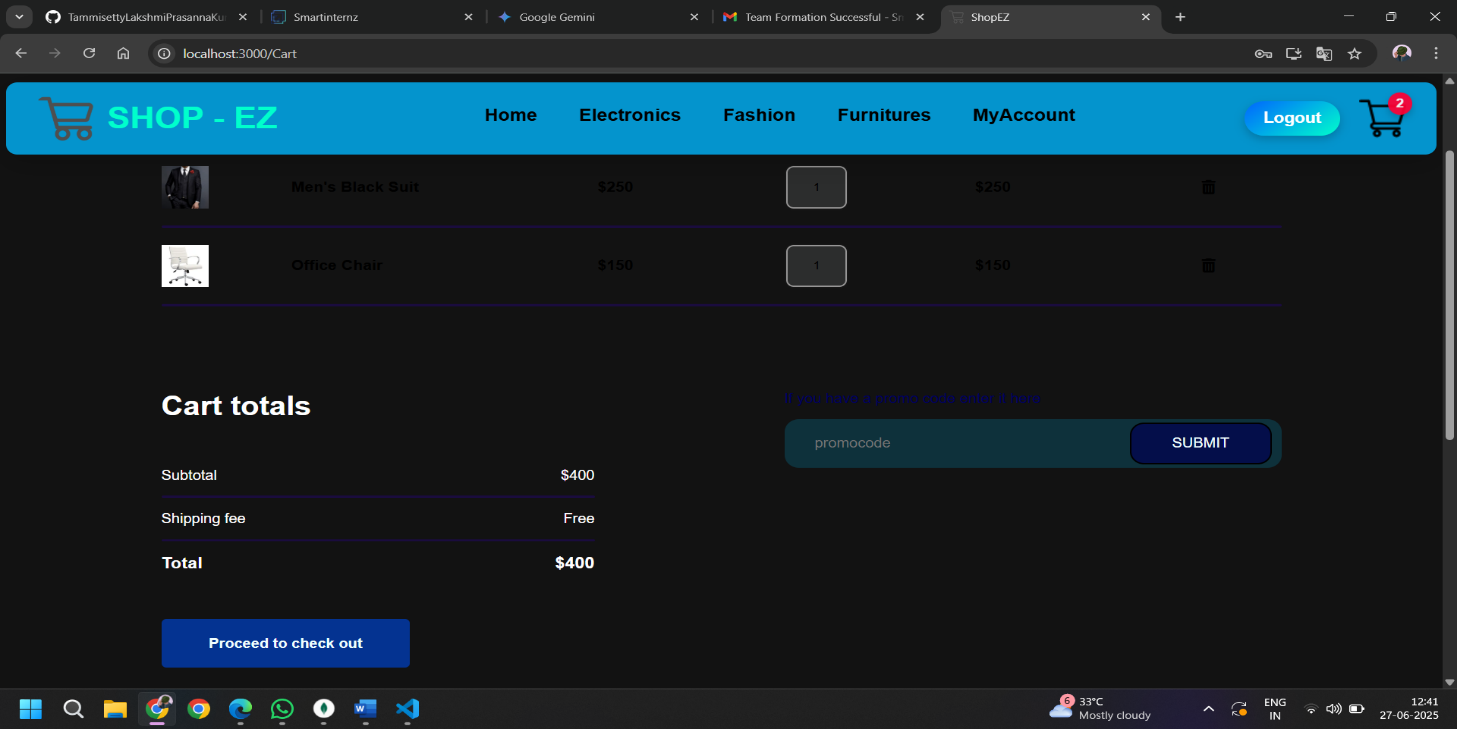
* **Live Demo:** https://github.com/TammisettyLakshmiPrasannaKumar/shopez-one-stop-shop-for-online-purchases/tree/main/Video%20Demo
* **Sample Screenshots:**

****

****



****

****

# FUTURE ENHANCEMENTS

To enhance the performance, usability, and scalability of SHOPEZ, the following improvements are planned:

1. **Implement real-time WebSocket-based chat** for faster and smoother communication between clients and freelancers.
2. **Integrate secure payment gateways** like Stripe and PayPal for streamlined transactions and escrow functionality.
3. **Enable project milestone tracking** to help clients and admins manage deadlines and deliverables efficiently.
4. **Build a cross-platform mobile application** using React Native for improved accessibility and mobile usability.
5. **Develop an AI-powered project recommendation engine** to match orders with relevant products.
6. **Introduce in-app notifications** for real-time alerts on prices, orders, product updates, and reviews.
7. **Add admin-level analytics and reporting tools** to monitor platform activity and performance.
8. **Implement a robust freelancer verification system** to ensure quality and trust across the platform.
9. **Create a dynamic reputation scoring algorithm** based on reviews, ratings, and order success history.
10. **Enhance form validation and error handling** across all modules to improve data accuracy and user experience.

# CONCLUSION

**ShopeZ** is a robust full-stack eCommerce web application built to streamline the online shopping experience for both customers and sellers. Utilizing the powerful MERN stack—MongoDB, Express.js, React.js, and Node.js—the platform delivers essential features such as product listings, user authentication, dynamic cart and wishlist management, and secure order processing.

The system effectively tackles key challenges faced by modern online marketplaces by promoting a smooth, responsive, and intuitive shopping environment. With separate interfaces for customers, sellers, and administrators, ShopeZ ensures role-based access and management, fostering a trustworthy and scalable digital storefront.

As the platform continues to grow, planned features like integrated payment gateways, real-time inventory tracking, and mobile application support are set to enhance the user experience even further. ShopeZ stands as a strong example of full-stack development excellence, thoughtfully crafted to meet the practical demands of today’s digital commerce landscape.