ShopEZ: E-commerce Applications

1. Introduction

Project Title: ShopEZ: One-Stop Shop For Online Purchases

Step into the future of online shopping with ShopEz: Your One-Stop Shop for Every Purchase, a smart and user-friendly e-commerce platform designed to simplify and enhance your shopping experience. Whether it's fashion, electronics, groceries, or daily essentials—ShopEz brings it all to your fingertips.

Effortlessly browse a wide range of products, enjoy personalized recommendations based on your interests, and experience secure, smooth transactions. With features like real-time order tracking, easy returns, and responsive design, ShopEz ensures convenience at every step.

We prioritize customer satisfaction and data security. Our platform provides a seamless interface with encrypted access, ensuring a safe and enjoyable shopping journey. From discovery to delivery, ShopEz makes online shopping easier, faster, and smarter—for everyone.

Team Members:

- Sai Dinesh Kumar Vema
- Keerthi Narendra Babu
- Bandireddy Dusyanta Venkata Sai Gopinadh
- Ravipati Lokesh Sai Kumar
- Chimakurthy Charan Srinivas

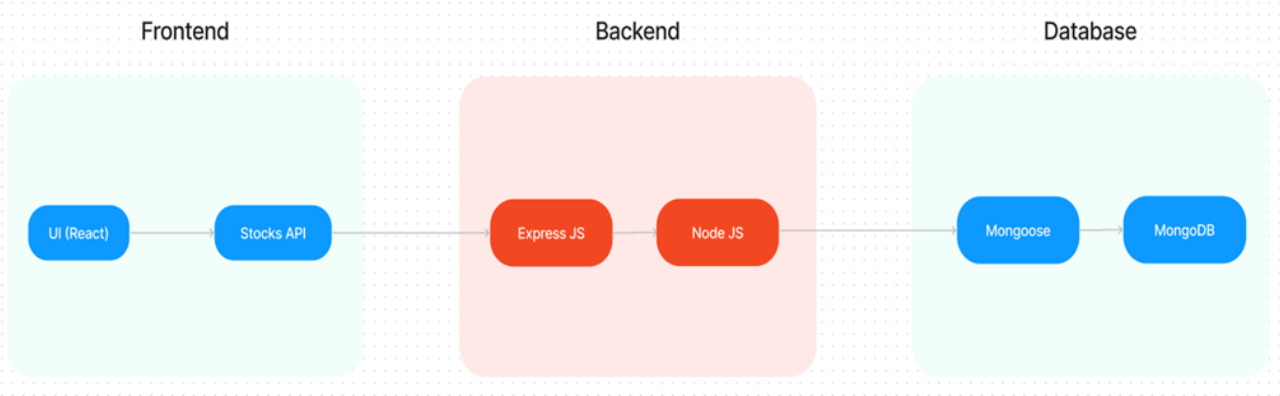
2. Project Overview

Purpose:
ShopEZ is designed to cater to the growing need for efficient and user-friendly e-commerce platforms. It bridges the gap between complex online shopping systems and user convenience by providing streamlined navigation, secure transactions, and personalized product recommendations. Additionally, the platform empowers sellers with a dashboard for managing their inventory, processing orders, and accessing analytics to drive growth.

Features:

1. **Seamless Checkout:** Secure and smooth payment process with instant order confirmations and email notifications.
2. **Effortless Product Discovery:** Advanced search capabilities, intuitive category navigation, and powerful filters to help users find exactly what they need.
3. **Personalized Recommendations:** AI-driven algorithms analyze user behavior to provide curated product suggestions.
4. **Seller Dashboard:** Comprehensive tools for inventory tracking, order processing, and analytics to monitor performance metrics.
5. **Real-time Analytics:** Data-driven insights for sellers, highlighting sales trends, customer preferences, and product performance

Architecture



Frontend:

- Developed using **React.js** for its component-based architecture and state management capabilities.
- Features dynamic components like:
  - **Product Listings**: Displays products with sorting and filtering options.
  - **Cart Management**: Allows users to add, update, or remove items in their cart.
  - **User Authentication**: Login and registration pages with secure validation.
  - **Admin Panel**: Provides sellers with tools to manage inventory and view analytics.

Backend:

- Built with **Node.js** and **Express.js**, ensuring scalability and high performance.
- Features include:
  - **API Endpoints**:
    - /productsfor fetching product data.
    - /ordersfor processing customer orders.
    - /usersfor managing user authentication.
  - **Middleware** for error handling and authentication using **JWT**.

Database:

- **MongoDB** serves as the database, storing collections for:
  - **Users**: Authentication credentials, profiles, and purchase history.
  - **Products**: Information on inventory, prices, categories, and descriptions.
  - **Orders**: Details about placed orders, delivery status, and payment.

This architecture ensures modularity, scalability, and efficient data management.

3. Setup Instructions

Prerequisites:

- **Node.js**: v14 or later
- **MongoDB**: Installed locally or set up using a cloud provider like MongoDB Atlas
- **npm**: Package manager for installing dependencies

Installation guide

1. Clone the Repository:

Use the following command to clone the repository from GitHub:

bash

CopyEdit

git clone https://github.com/stinastanley/stina.git

2. Navigate to the Project Directory:

After cloning, change into the main project folder:

bash

CopyEdit

cd stina

cd shopEZ

3. Install Project Dependencies:

ShopEZ has both frontend and backend components. Install the required dependencies separately for each:

4.Backend Setup:

bash

CopyEdit

cd server

npm install

This installs all necessary Node.js modules required for the backend server to function properly, including Express, body-parser, and other dependencies.

5. Frontend Setup:

Open a new terminal tab/window or return to the root project folder, then:

bash

CopyEdit

cd client

npm install

This installs all the React app dependencies, such as React, React Router, and Axios.

6. Run the Application:

You'll need to run both the backend and frontend servers simultaneously.

7.Start the Backend Server:

From the server directory:

bash

CopyEdit

node index.js

The backend will typically run on http://localhost:5000 unless otherwise configured.

8.Start the Frontend Server:

From the client directory:

bash

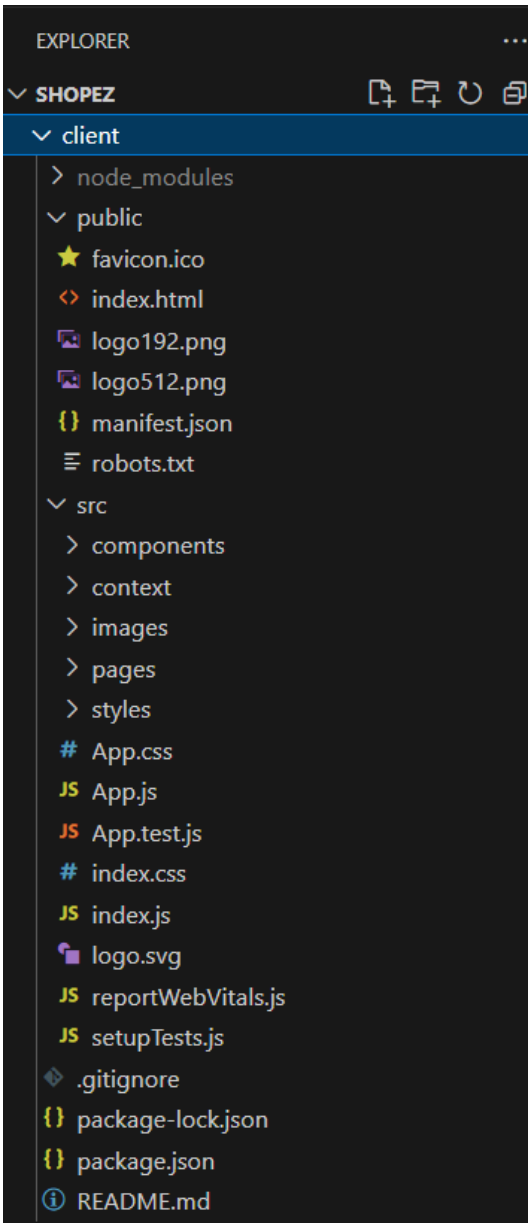
CopyEdit

npm start

The frontend will open in your default browser at http://localhost:3000.

4. Folder Structure

Client:

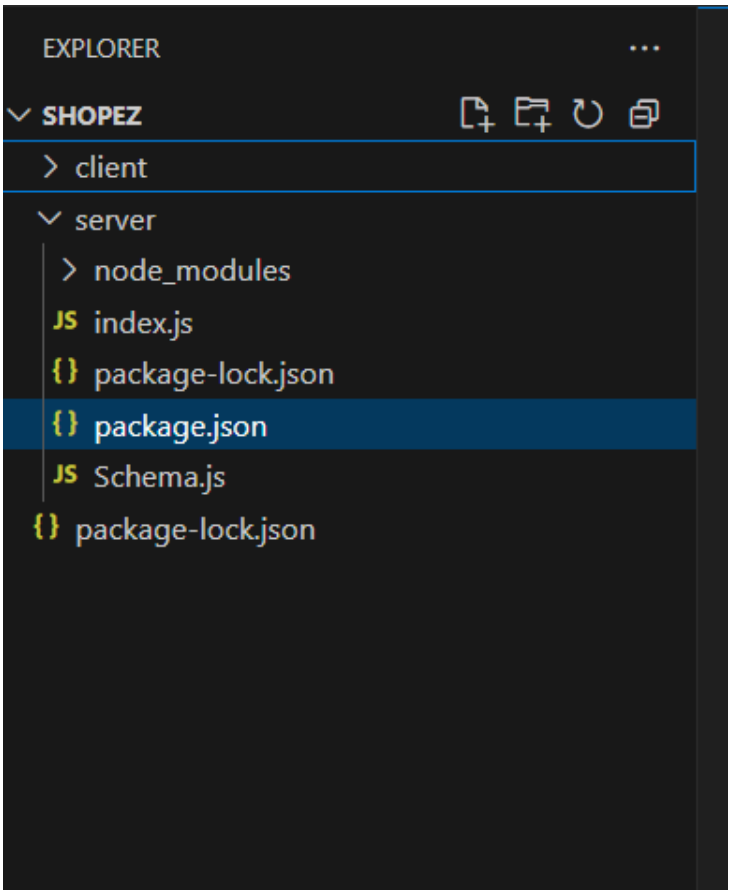


- src/components: Contains reusable components like **Navbar**, **ProductCard**, **CartItem**, etc.
- src/pages: Holds page components such as **Home**, **Cart**, **Checkout**, and

AdminPanel.

- src/services: Manages API interactions for fetching and posting data.
- src/redux: Implements state management for cart items, user authentication, and order status.

Server:



- routes: Defines all RESTful API routes for users, products, and orders.
- controllers: Contains the logic for handling API requests and responses.
- models: Defines database schemas for **Users**, **Products**, and **Orders**.
- middleware: Handles authentication (JWT) and error management.

5. Running the Application

Frontend:

Navigate to the client directory:

```
cd client
```

Start the React application:

```
npm start
```

Backend:

Navigate to the server directory:

```
cd server
```

Start the Node.js server:

```
node index.js
```

6. API Documentation

Endpoints:

1. **GET /products**
   - Fetches the list of all products.
   - Parameters: Optional category and price filters.

Response: json

```
[
  {
    "id": "123",
    "name": "Gold Bracelet", "price": 50,
    "category": "Accessories"
  }
]
```

2. **POST /orders**
   - Places a new order.
   - Parameters: User ID, product details, and quantity.

Response: json

```
{
  "message": "Order placed successfully", "orderId": "456"
}
```

7. Authentication

JWT-based Authentication:

- **Login**: Issues a JWT token upon successful authentication.
- **Token Validation**: Protects private routes like /orders and /admin.
- **Logout**: Invalidates the token on the client side.

8. User Interface

Screens:

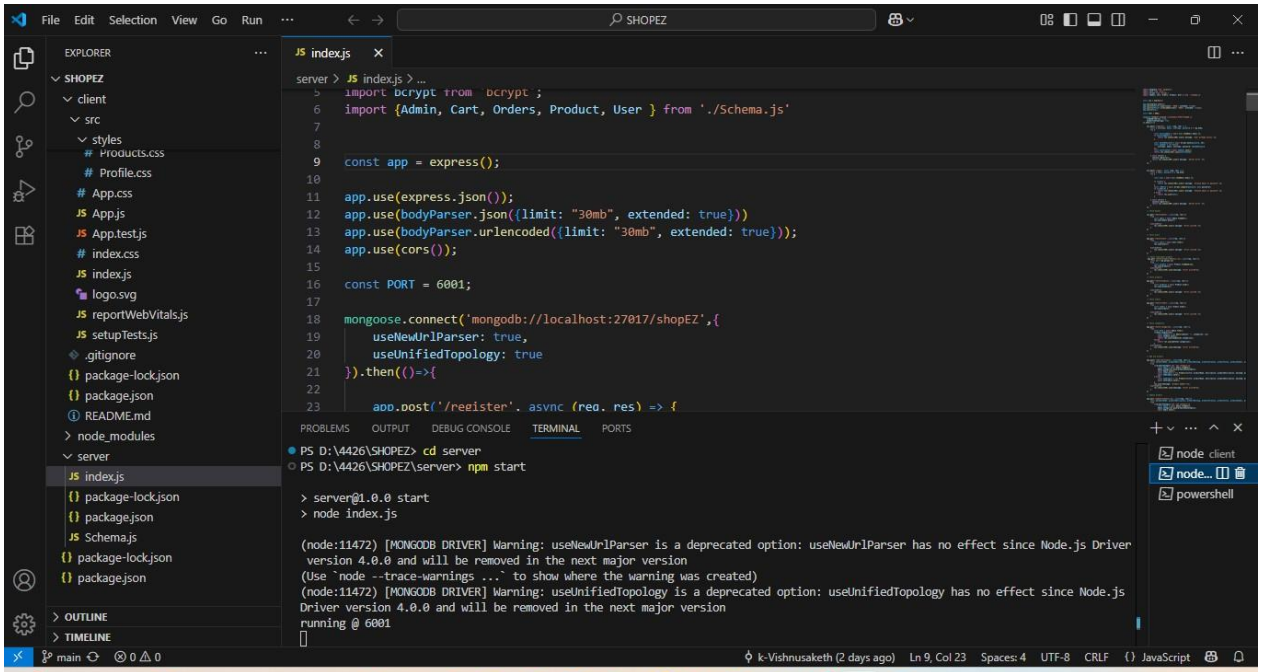
- **Home:** Displays trending products and categories.
- **Product Details:** Shows detailed information about a selected item.
- **Cart:** Summarizes selected products and their quantities.
- **Admin Dashboard:** Offers order status updates and analytics for sellers.

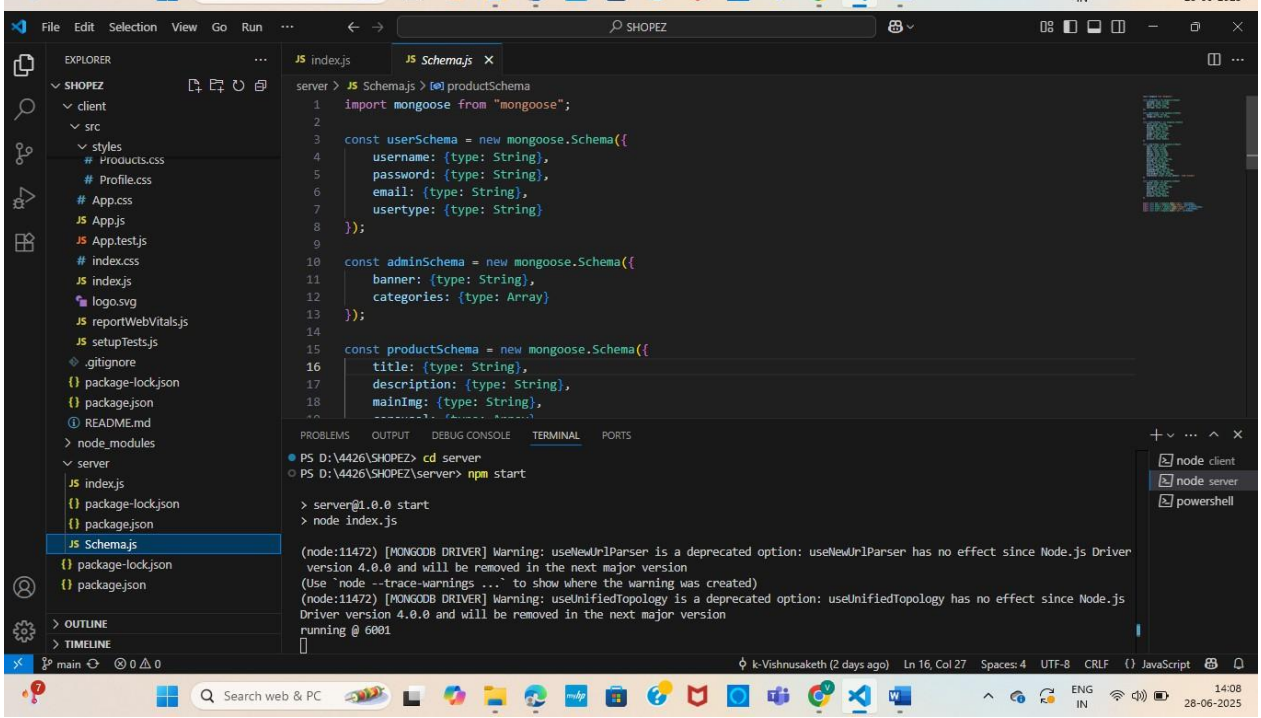
9. Testing

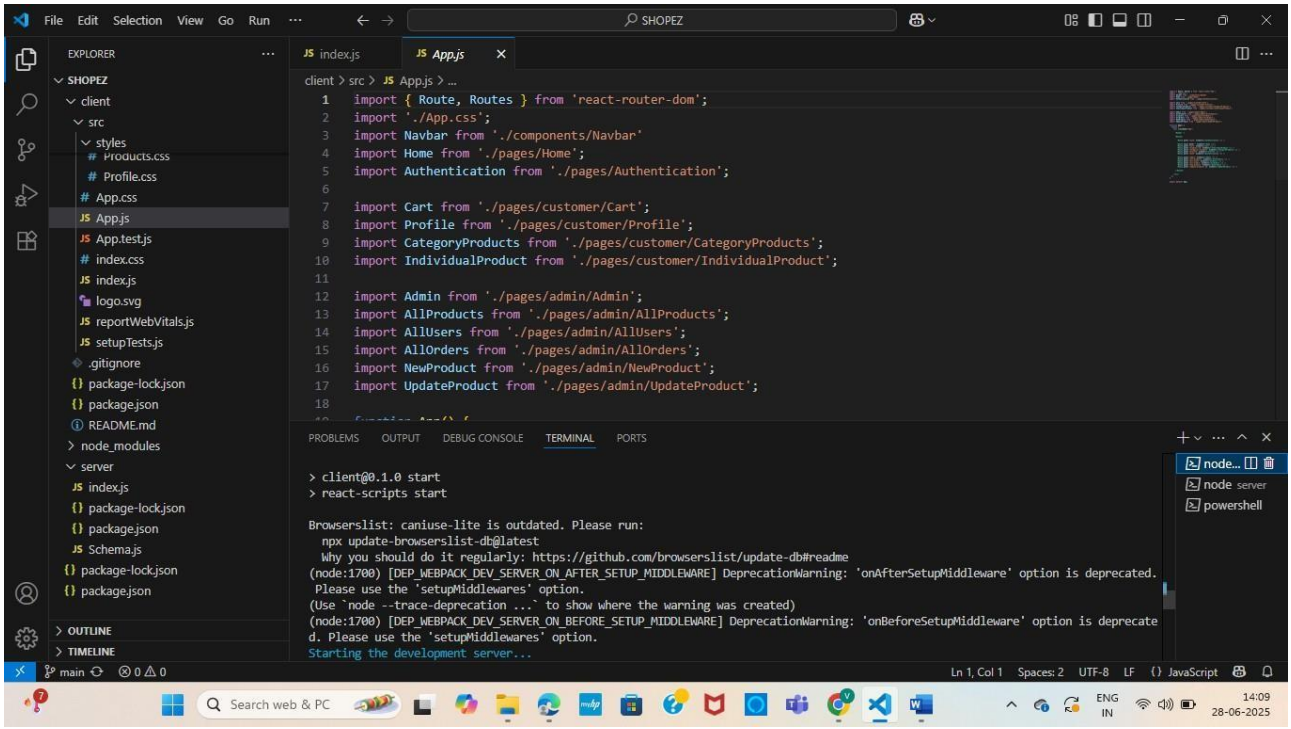
- **Unit Testing:** Conducted using **Jest** for components and backend logic.
- **API Testing:** Performed using **Postman** to validate endpoints.

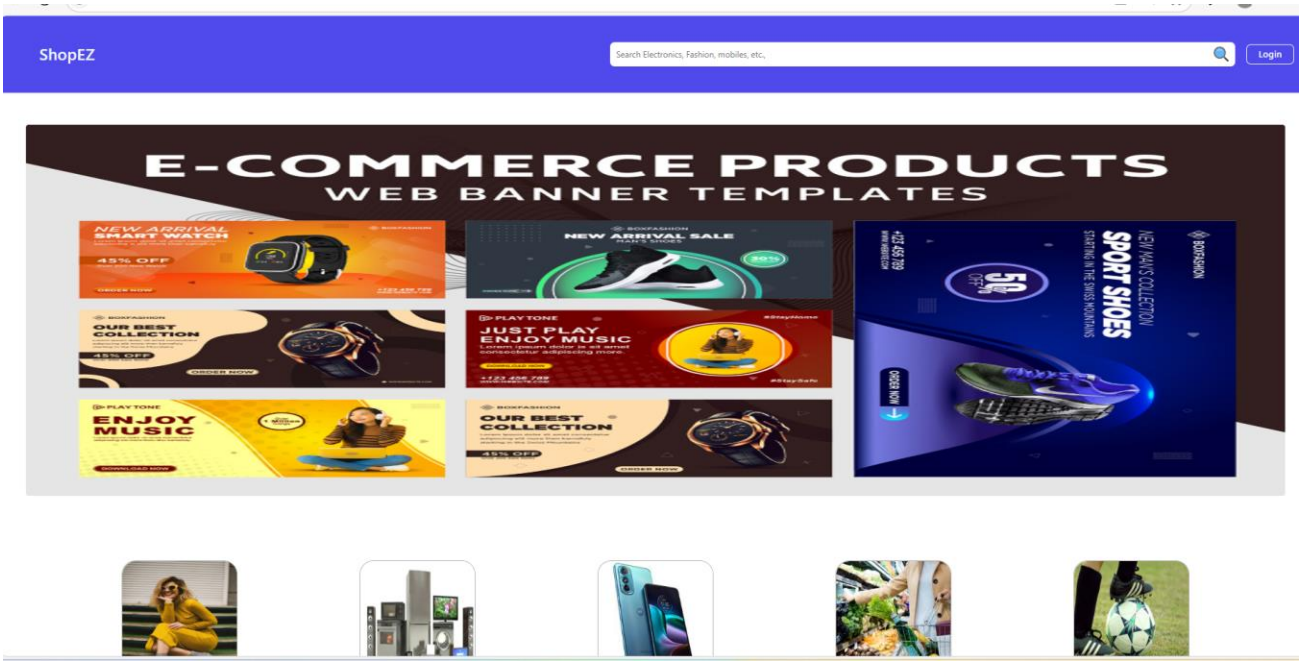
10. Screenshots or Demo

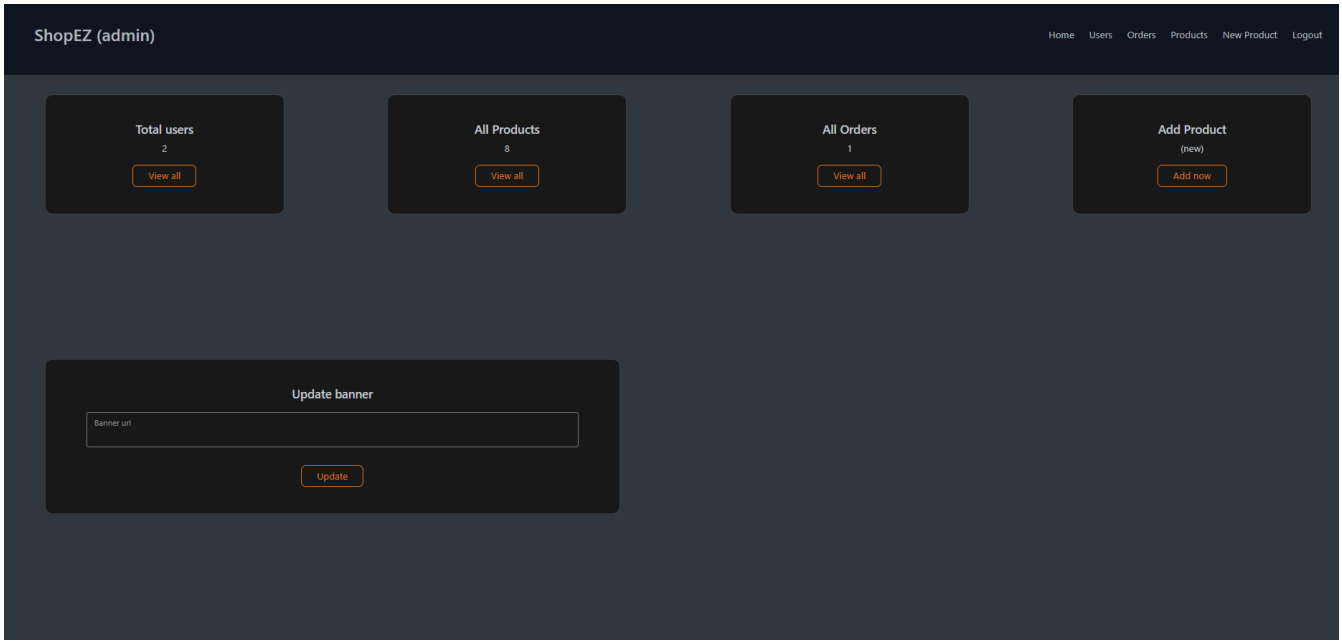
Screenshots of key UI components and admin panel.

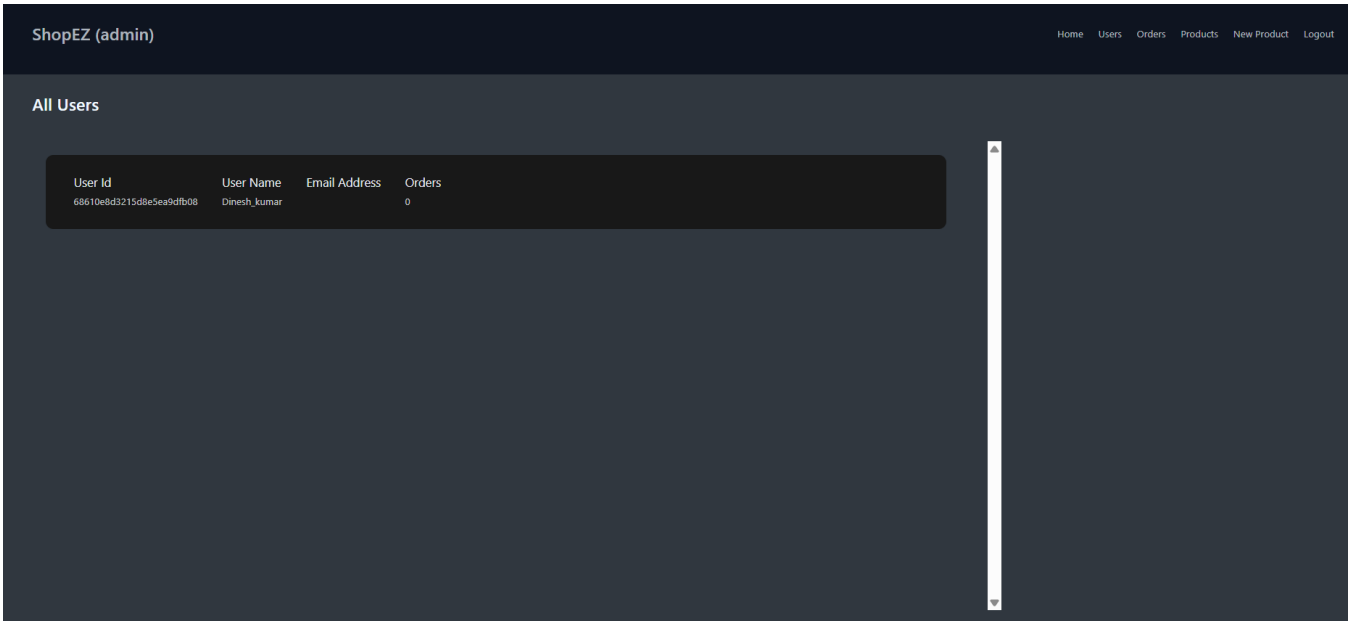

File Edit Selection View Go Run
SHOPEZ
EXPLORER
index.js
server > index.js > ...
import bcrypt from 'bcrypt';
import {Admin, Cart, Orders, Product, User } from './Schema.js'
const app = express();
app.use(express.json());
app.use(bodyParser.json({limit: "30mb", extended: true}))
app.use(bodyParser.urlencoded({limit: "30mb", extended: true}));
app.use(cors());
const PORT = 6001;
mongoose.connect('mongodb://localhost:27017/shopEZ',{
useNewUrlParser: true,
useUnifiedTopology: true
}).then(()=>{
app.post('/register', async (req, res) => {
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\4426\SHOPEZ> cd server
PS D:\4426\SHOPEZ\server> npm start
> server@1.0.0 start
> node index.js
(node:11472) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:11472) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
running @ 6001
node client
node...
powershell
OUTLINE
TIMELINE
main
k-Vishnusaketh (2 days ago) Ln 9, Col 23 Spaces: 4 UTF-8 CRLF {} JavaScript
Search web & PC
ENG IN
14:07
28-06-2025

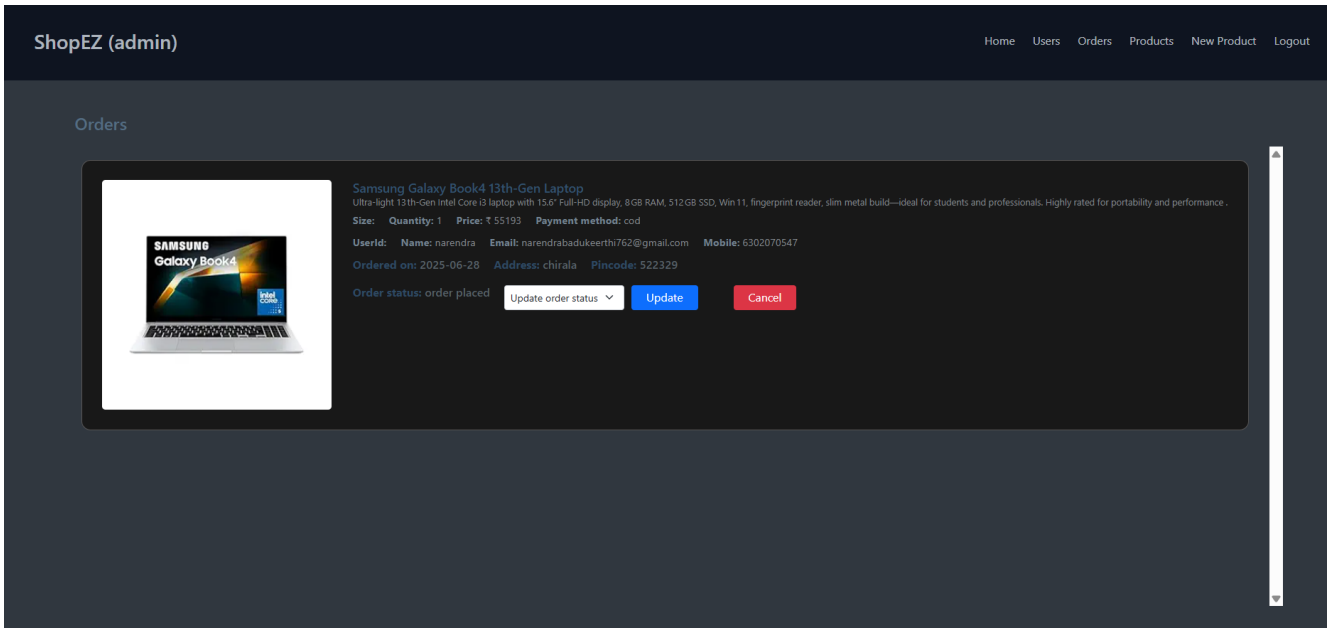

File Edit Selection View Go Run
SHOPEZ
EXPLORER
index.js
Schema.js
server > Schema.js > productSchema
import mongoose from "mongoose";
const userSchema = new mongoose.Schema({
username: {type: String},
password: {type: String},
email: {type: String},
usertype: {type: String}
});
const adminSchema = new mongoose.Schema({
banner: {type: String},
categories: {type: Array}
});
const productSchema = new mongoose.Schema({
title: {type: String},
description: {type: String},
mainImg: {type: String},
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\4426\SHOPEZ> cd server
PS D:\4426\SHOPEZ\server> npm start
> server@1.0.0 start
> node index.js
(node:11472) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:11472) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
running @ 6001
node client
node server
powershell
OUTLINE
TIMELINE
main
k-Vishnusaketh (2 days ago) Ln 16, Col 27 Spaces: 4 UTF-8 CRLF {} JavaScript
Search web & PC
ENG IN
14:08
28-06-2025

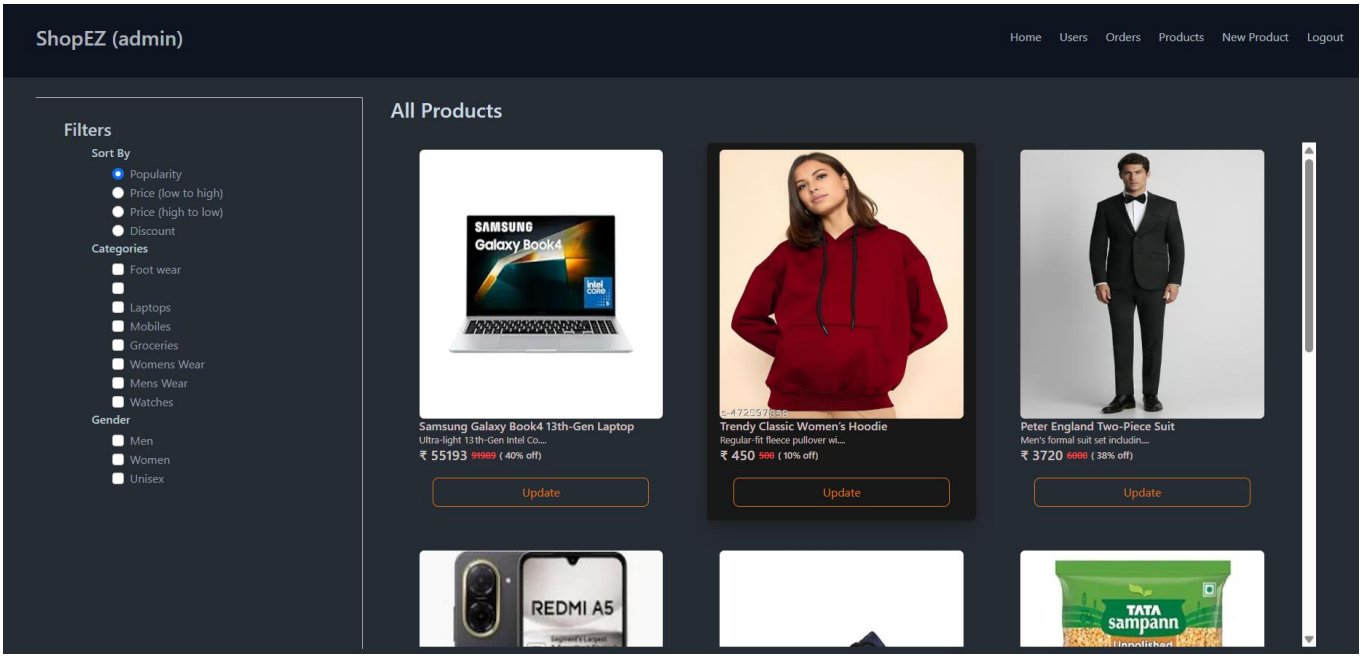

File Edit Selection View Go Run
SHOPEZ
EXPLORER
index.js
App.js
client > src > App.js > ...
import { Route, Routes } from 'react-router-dom';
import './App.css';
import Navbar from './components/Navbar'
import Home from './pages/Home';
import Authentication from './pages/Authentication';
import Cart from './pages/customer/Cart';
import Profile from './pages/customer/Profile';
import CategoryProducts from './pages/customer/CategoryProducts';
import IndividualProduct from './pages/customer/IndividualProduct';
import Admin from './pages/admin/Admin';
import AllProducts from './pages/admin/AllProducts';
import AllUsers from './pages/admin/AllUsers';
import AllOrders from './pages/admin/AllOrders';
import NewProduct from './pages/admin/NewProduct';
import UpdateProduct from './pages/admin/UpdateProduct';
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
> client@0.1.0 start
> react-scripts start
Browserslist: caniuse-lite is outdated. Please run:
npx update-browserslist-db@latest
Why you should do it regularly: https://github.com/browserslist/update-db#readme
(node:1700) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is deprecated.
Please use the 'setupMiddlewares' option.
(Use `node --trace-deprecation ...` to show where the warning was created)
(node:1700) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
Starting the development server...
node... node server powershell
OUTLINE
TIMELINE
main
Ln 1, Col 1 Spaces: 2 UTF-8 LF JavaScript
Search web & PC
ENG IN
14:09 28-06-2025

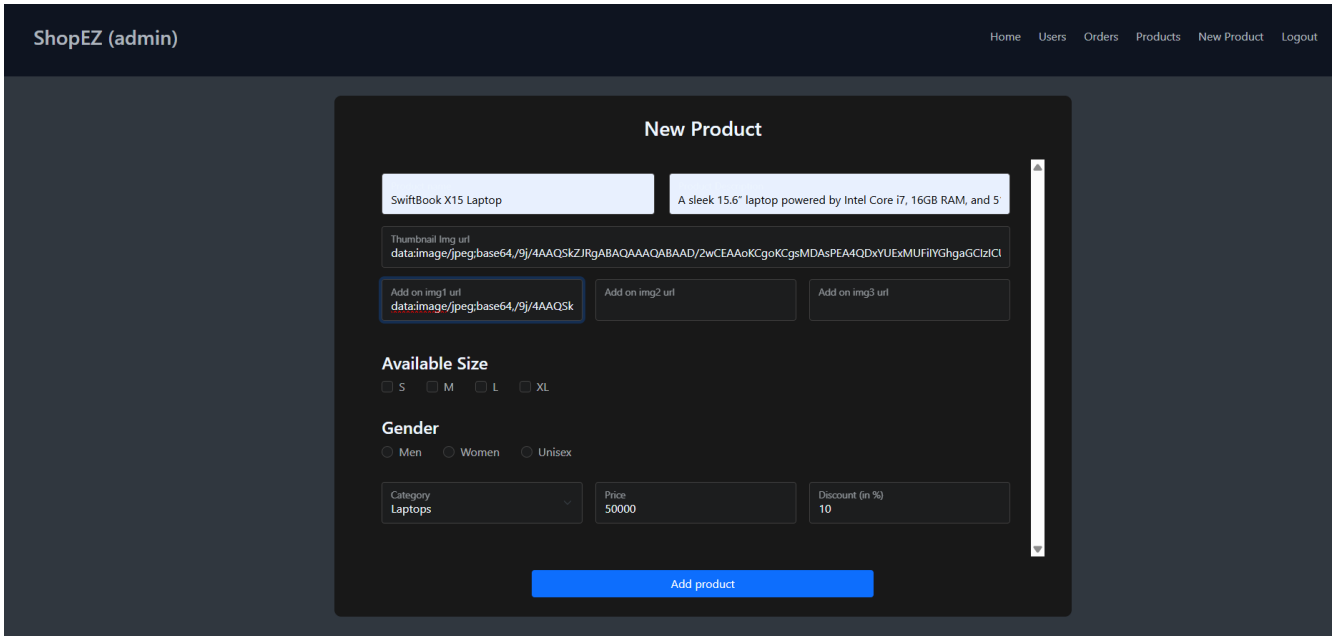

ShopEZ
Search Electronics, Fashion, mobiles, etc.
Login
E-COMMERCE PRODUCTS
WEB BANNER TEMPLATES
NEW ARRIVAL SMART WATCH
45% OFF
NEW ARRIVAL SALE
OUR BEST COLLECTION
JUST PLAY ENJOY MUSIC
ENJOY MUSIC
SPORT SHOES
ORDER NOW


ShopEZ (admin)
Home
Users
Orders
Products
New Product
Logout
Total users
2
View all
All Products
8
View all
All Orders
1
View all
Add Product
(new)
Add now
Update banner
Banner url
Update


ShopEZ (admin)
Home
Users
Orders
Products
New Product
Logout
All Users
User Id
68610e8d3215d8e5ea9dfb08
User Name
Dinesh_kumar
Email Address
Orders
0

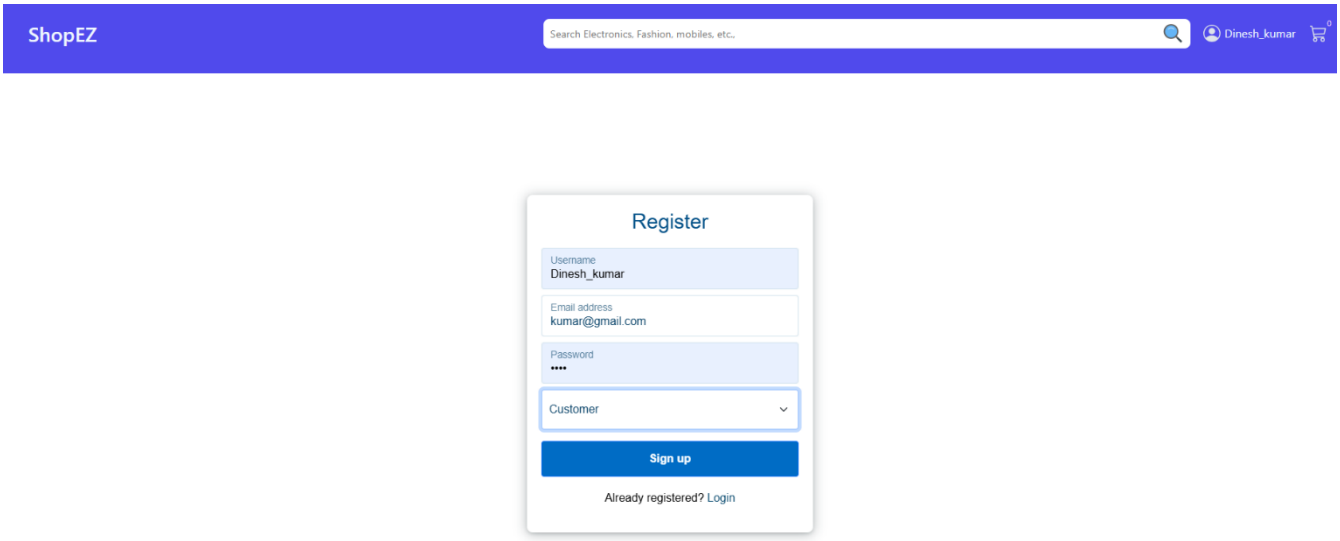

ShopEZ (admin)
Home
Users
Orders
Products
New Product
Logout
Orders
Samsung Galaxy Book4 13th-Gen Laptop
Ultra-light 13th-Gen Intel Core i3 laptop with 15.6" Full-HD display, 8GB RAM, 512GB SSD, Win 11, fingerprint reader, slim metal build—ideal for students and professionals. Highly rated for portability and performance .
Size: Quantity: 1 Price: ₹ 55193 Payment method: cod
UserId: Name: narendra Email: narendrabadukeerthi762@gmail.com Mobile: 6302070547
Ordered on: 2025-06-28 Address: chirala Pincode: 522329
Order status: order placed
Update order status
Update
Cancel

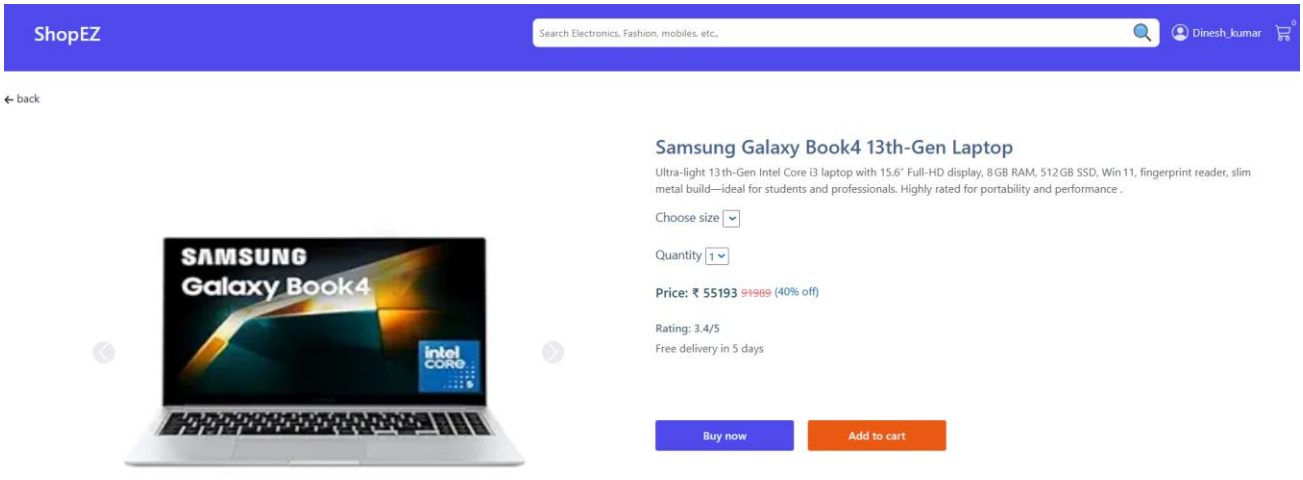

ShopEZ (admin)
Home
Users
Orders
Products
New Product
Logout
Filters
Sort By
Popularity
Price (low to high)
Price (high to low)
Discount
Categories
Foot wear
Laptops
Mobiles
Groceries
Womens Wear
Mens Wear
Watches
Gender
Men
Women
Unisex
All Products
Samsung Galaxy Book4 13th-Gen Laptop
Ultra-light 13th-Gen Intel Co....
₹ 55193 ₹1989 (40% off)
Update
Trendy Classic Women's Hoodie
Regular-fit fleece pullover wi....
₹ 450 ₹00 (10% off)
Update
Peter England Two-Piece Suit
Men's formal suit set includin....
₹ 3720 ₹000 (38% off)
Update
REDMI A5
TATA sampann

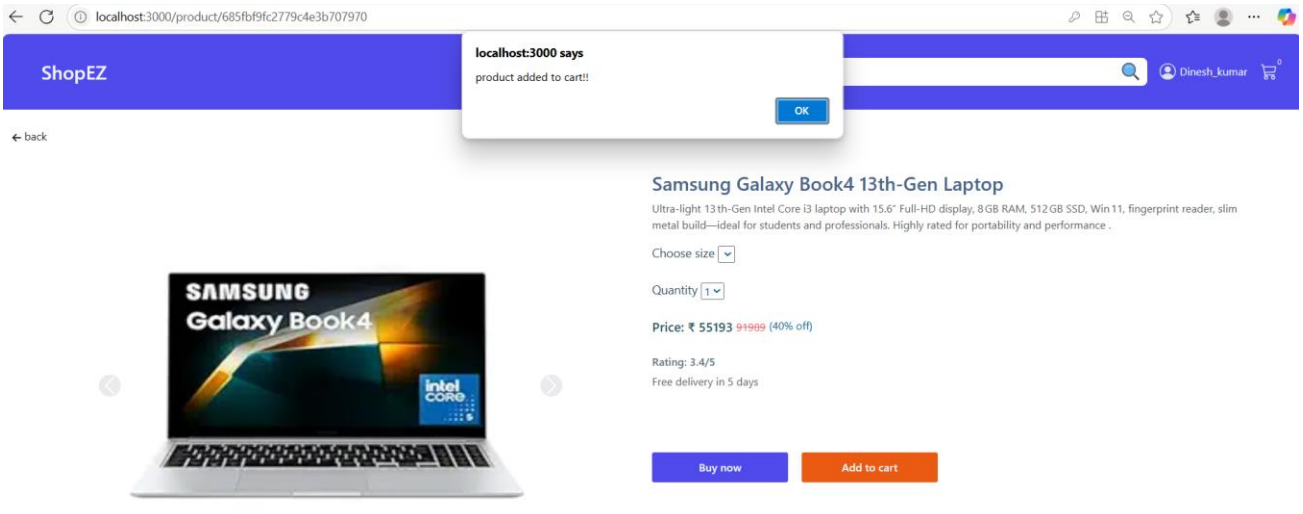

ShopEZ (admin)
Home
Users
Orders
Products
New Product
Logout
New Product
SwiftBook X15 Laptop
A sleek 15.6" laptop powered by Intel Core i7, 16GB RAM, and 5
Thumbnail Img url

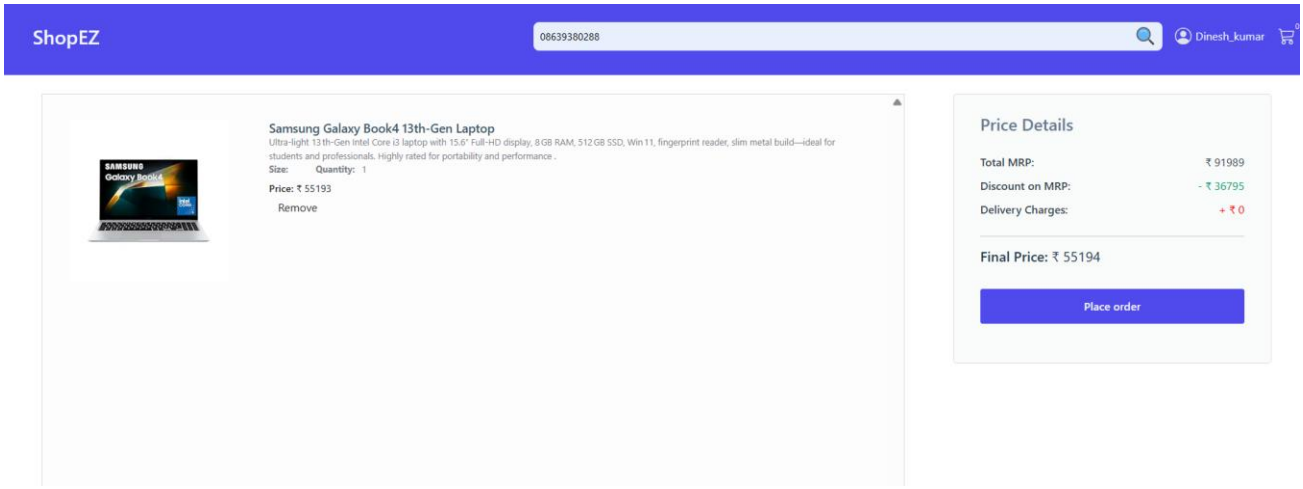
Add on img1 url

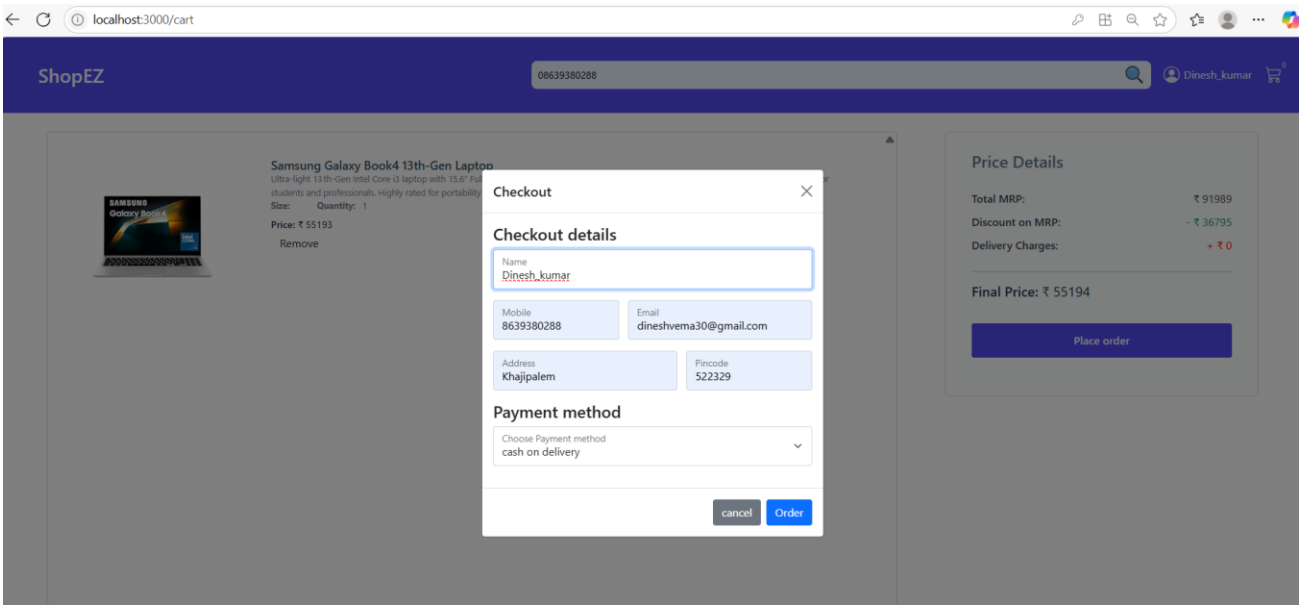
Add on img2 url
Add on img3 url
Available Size
S
M
L
XL
Gender
Men
Women
Unisex
Category
Laptops
Price
50000
Discount (in %)
10
Add product

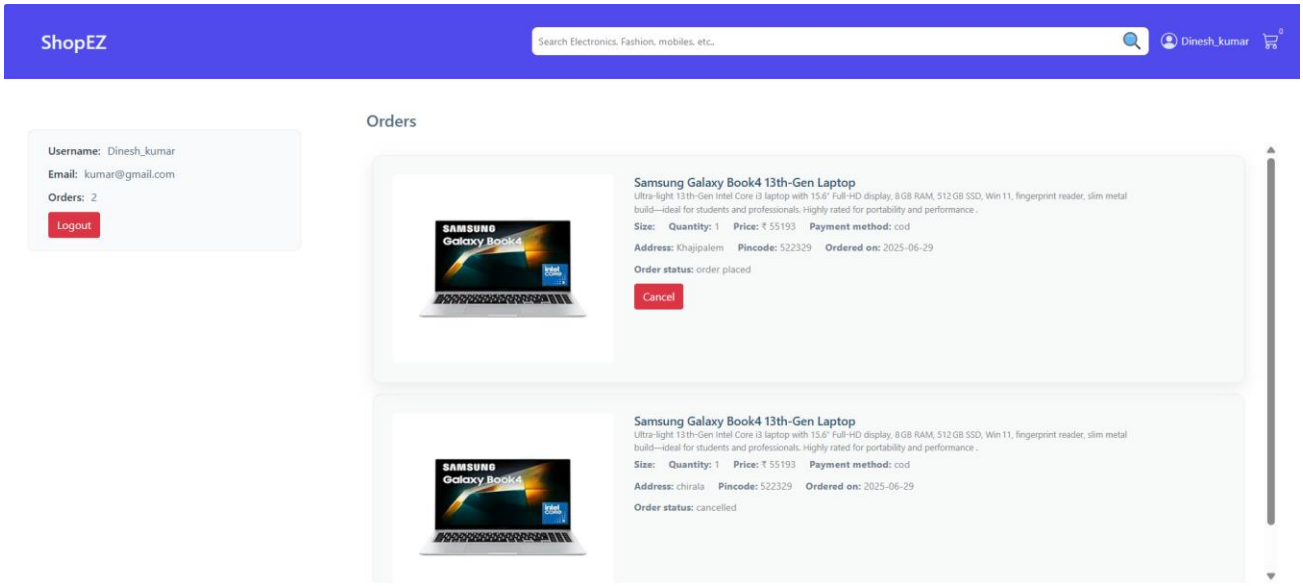

ShopEZ
Search Electronics, Fashion, mobiles, etc..
Dinesh_kumar
Register
Username
Dinesh_kumar
Email address
kumar@gmail.com
Password
••••
Customer
Sign up
Already registered? Login


ShopEZ
Search Electronics, Fashion, mobiles, etc.
Dinesh_kumar
back
SAMSUNG
Galaxy Book4
intel CORE
Samsung Galaxy Book4 13th-Gen Laptop
Ultra-light 13th-Gen Intel Core i3 laptop with 15.6" Full-HD display, 8 GB RAM, 512 GB SSD, Win 11, fingerprint reader, slim metal build—ideal for students and professionals. Highly rated for portability and performance .
Choose size
Quantity 1
Price: ₹ 55193 91989 (40% off)
Rating: 3.4/5
Free delivery in 5 days
Buy now
Add to cart


localhost:3000/product/685fbf9fc2779c4e3b707970
localhost:3000 says
product added to cart!!
OK
ShopEZ
Dinesh_kumar
back
SAMSUNG
Galaxy Book4
intel CORE
Samsung Galaxy Book4 13th-Gen Laptop
Ultra-light 13th-Gen Intel Core i3 laptop with 15.6" Full-HD display, 8 GB RAM, 512 GB SSD, Win 11, fingerprint reader, slim metal build—ideal for students and professionals. Highly rated for portability and performance .
Choose size
Quantity 1
Price: ₹ 55193 91989 (40% off)
Rating: 3.4/5
Free delivery in 5 days
Buy now
Add to cart


ShopEZ
08639380288
Dinesh_kumar
Samsung Galaxy Book4 13th-Gen Laptop
Ultra-light 13 th-Gen Intel Core i3 laptop with 15.6" Full-HD display, 8 GB RAM, 512 GB SSD, Win 11, fingerprint reader, slim metal build—ideal for students and professionals. Highly rated for portability and performance .
Size:
Quantity: 1
Price: ₹ 55193
Remove
Price Details
Total MRP: ₹ 91989
Discount on MRP: - ₹ 36795
Delivery Charges: + ₹ 0
Final Price: ₹ 55194
Place order


localhost:3000/cart
ShopEZ
08639380288
Dinesh_kumar
Samsung Galaxy Book4 13th-Gen Laptop
Size:
Quantity: 1
Price: ₹ 55193
Remove
Price Details
Total MRP: ₹ 91989
Discount on MRP: - ₹ 36795
Delivery Charges: + ₹ 0
Final Price: ₹ 55194
Place order
Checkout
Checkout details
Name
Dinesh_kumar
Mobile
8639380288
Email
dineshvema30@gmail.com
Address
Khajipalem
Pincode
522329
Payment method
Choose Payment method
cash on delivery
cancel
Order



Project Demo Link:

[Project Demo Link]

11. Known Issues

- **Slow Search Performance:** Optimization is required for large product datasets.
- **UI Bugs:** Minor alignment issues on smaller screens.

12. Future Enhancements

- **Voice Search:** Enable users to search using voice commands.
- **Mobile App:** Create a cross-platform app using React Native.
- **Multi-language Support:** Expand accessibility for global audiences.