MERN stack powered by MongoDB
Naan Mudhalvan – Project Documentation

Introduction

Project Title: ShopEZ (E-Commerce Platform)College: Adhi College of Engineering and Technology.

Department: B.Tech Artificial Intelligence and Data Science.

Team Members:

S.no	Student Name	Register No	Naan Muthalvan ID	Role
1	Pradeep kumar R	410121243039	015B9CD E6CAFB4 7B08348 E753FFE 48F2	Backend
2	Nathaniel Felix L	410121243034	7CAB704A432ABDDA14B4D565BB78C3AE	Frontend
3	Prakash A	410121243040	9781781EF182D9585324C73E8596D9C3	Frontend
4	Mohan Krishna	410121243036	DF9F05F6AD2E679818A43ED84537549	Backend

Project Overview

Purpose: Our E-Commerce Platform is designed to provide a seamless and engaging shopping experience for users, along with a comprehensive and efficient management interface for administrators. It offers customers a wide range of products, while enabling administrators to manageinventory, track orders, and analyze sales data.

Features:

Customer/User:

Signup/Login:

- User registration with email.
- Login with email and password, with a “Remember Me” option.

Product Browsing and Filtering:

- View all available products with filter, and sorting capabilities by category, price, popularity.
- View detailed product information, including images, descriptions, specifications, and reviews.

Wishlist and Cart Management:

- Add products to the wishlist for future reference.
- Add products to the cart, view, and modify quantities.
- Save items in the cart for later, with cart persistence across sessions.

Order Placement and Tracking:

- Secure checkout process with address input, order summary, and delivery options.
- Order tracking with status updates and estimated delivery times.

User Profile:

- Editable profile information except for email (name, contact, address, etc.).
- Order history view with downloadable receipts.

Admin:

User and Product Management:

- View, add, edit, and delete products and categories.
- Manage users, with the ability to view user profiles and delete users if needed.

Order Management:

- View and update order statuses, including processing, shipping, and delivery.
- Generate invoices and send order notifications.

Inventory and Sales Analytics:

- Track inventory levels, set low-stock alerts, and restock products as necessary.
- View sales data, including sales volume, top products, revenue trends, and customer purchase behavior.

Promotions and Discounts:

- Create, update, and manage discount codes, sales events, and promotional offers.

Architecture

Frontend:

- **Developed using React** with responsive design and user-friendly interface, optimized for smooth navigation.
- **Bootstrap for styling** and layout, enhancing mobile and tablet responsiveness.

Backend:

- **Node.js and Express.js** to handle business logic and manage API endpoints for product listing,user actions, and order processing.

Middleware includes:

- **Bcrypt.js** for secure password hashing.
- **jsonwebtoken** for user authentication and authorization.
- **Multer** for managing image uploads and handling product image files.

Database:

- **MongoDB** serves as the database solution for storing all data, including user profiles, productlistings, orders, and reviews.
- **Mongoose** ORM (Object-Relational Mapping) is used for defining schemas and handling datainteractions efficiently.

This architecture and feature set ensure a robust, flexible, and scalable platform that caters to customersneeds and streamlines administrative tasks, with a responsive, high-performance user experience.

Setup Instructions

Prerequisites:

- Node.js
- MongoDB Atlas Account.

Installation:

Clone the repository:

- git clone cd online-learning-platform

Install dependencies:

Frontend:

```
cd client
npm install
```

Backend:

```
cd server
npm install
```

Set up environment variables:

- Create **.env** file in the **server** folder with the following:

MONGODB_URI=’ mongodb://localhost:27017/shopEZ’

JWT_SECRET='t9843yt8hg0h8y834th893hy89h'

- Create .**env.local** file in the **client** folder with the following:

BACKEND_BASEURL=http://localhost:5000

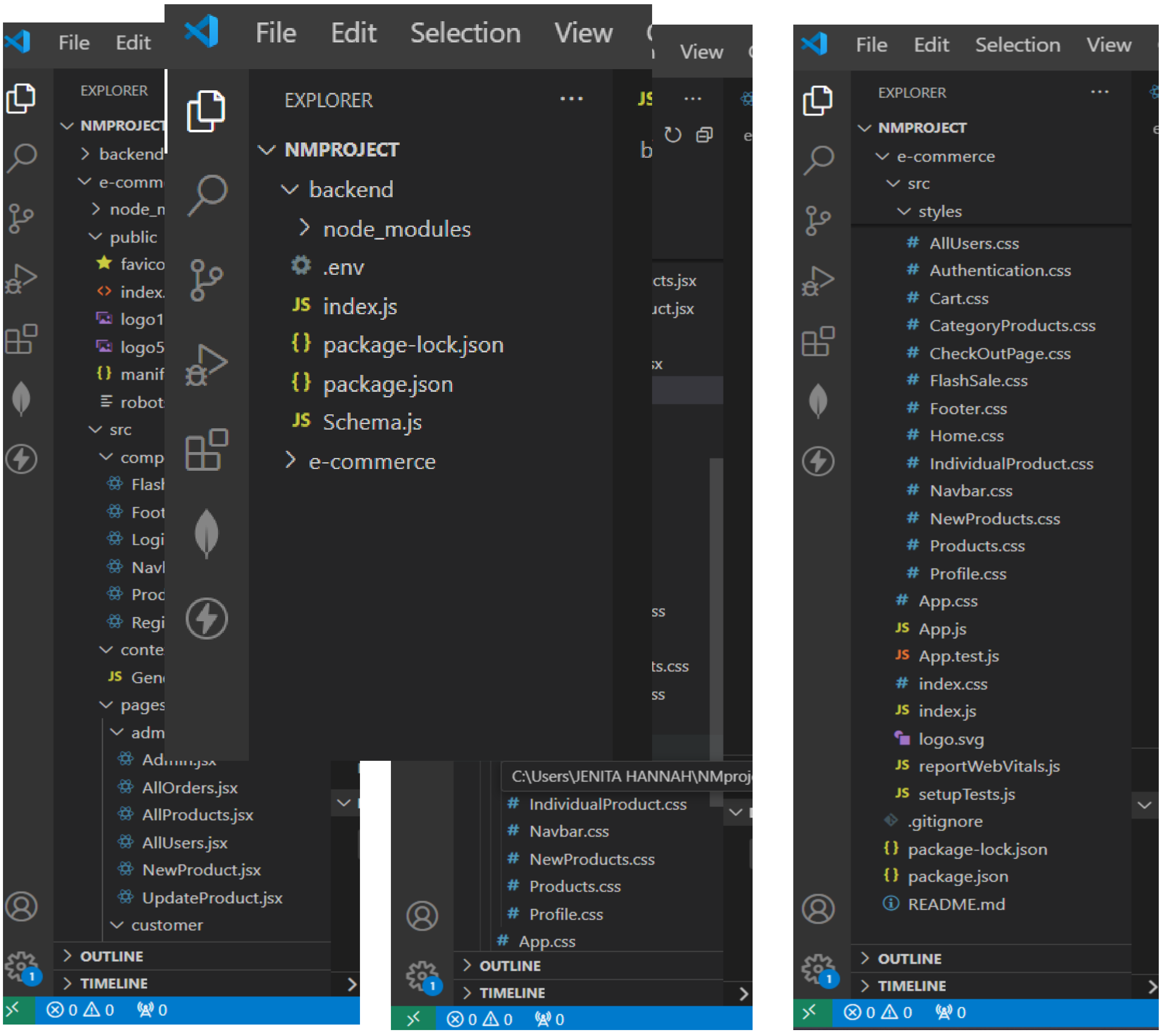
Folder Structure

Client (Frontend):

- **src/:** Contains all React components, pages, and assets.

- **Components/:** Reusable UI components such as Footer, Header, HeaderStudent and SideBar

- **Pages/:** Folders such as Admin, User, Instructor and Navs containing Pages for different aspects of the application.


```
File  Edit
EXPLORER
NMPROJECT
  > backend
  v e-comm
    > node_n
    v public
        favico
        index.
        logo1
        logo5
        manif
        robot
    v src
      v comp
          Flasl
          Foot
          Logi
          Navl
          Proc
          Regi
      v conte
          Gen
      v pages
        v adm
            Admin.jsx
            AllOrders.jsx
            AllProducts.jsx
            AllUsers.jsx
            NewProduct.jsx
            UpdateProduct.jsx
        v customer
> OUTLINE
> TIMELINE
⊗ 0 ⚠ 0   0
```

```
File  Edit  Selection  View
EXPLORER  ...
v NMPROJECT
  v backend
    > node_modules
      .env
      index.js
      package-lock.json
      package.json
      Schema.js
  > e-commerce
```

```
C:\Users\JENITA HANNAH\NMproj
  IndividualProduct.css
  Navbar.css
  NewProducts.css
  Products.css
  Profile.css
  App.css
> OUTLINE
> TIMELINE
⊗ 0 ⚠ 0   0
```

```
File  Edit  Selection  View
EXPLORER  ...
v NMPROJECT
  v e-commerce
    v src
      v styles
          AllUsers.css
          Authentication.css
          Cart.css
          CategoryProducts.css
          CheckOutPage.css
          FlashSale.css
          Footer.css
          Home.css
          IndividualProduct.css
          Navbar.css
          NewProducts.css
          Products.css
          Profile.css
        App.css
        App.js
        App.test.js
        index.css
        index.js
        logo.svg
        reportWebVitals.js
        setupTests.js
      .gitignore
      package-lock.json
      package.json
      README.md
> OUTLINE
> TIMELINE
⊗ 0 ⚠ 0   0
```

Server (Backend):

- **routes/:** Defines API routes for users, instructors, forums, and admin.
- **models/:** MongoDB schemas for Users, Courses, Instructor, Thread and OTP.
- **middleware/:** Authentication using jwt token.

Running the Application

- **Frontend:**
  cd client
  npm start

- **Backend:**
-
  cd server
  node index.js

User Routes:

POST /register: Register a new user.
POST /login: Login an existing user.
GET /fetch-users: Fetch all registered users.

Admin Routes:

GET /fetch-banner: Fetch the banner content.
POST /update-banner: Update the banner content.
GET /fetch-categories: Fetch all categories.
POST /add-new-product: Add a new product.
PUT /update-product/:id: Update an existing product by ID.

Product Routes:

GET /fetch-products: Fetch all products.
GET /fetch-product-details/:id: Fetch details of a specific product by ID.

Order Routes:

GET /fetch-orders: Fetch all orders.
POST /buy-product: Place an order for a specific product.PUT /cancel-order: Cancel an order.
PUT /update-order-status: Update the status of an order. POST /place-cart-order: Place orders for all items in the cart.

Cart Routes:

GET /fetch-cart: Fetch all items in the cart.
POST /add-to-cart: Add a new item to the cart.

PUT /increase-cart-quantity: Increase the quantity of a cart item by ID.
PUT /decrease-cart-quantity: Decrease the quantity of a cart item by ID.
PUT /remove-item: Remove a specific item from the cart by ID.

Authentication:

JWT (JSON Web Tokens):

- **Token Generation**: Tokens are generated upon successful login, allowing secure access to user-specificfeatures.
- **Middleware Validation**: Middleware checks token validity for protected routes, ensuring secure accesscontrol.

Email Verification:

- During signup, a verification link is sent to the user’s email using **Nodemailer** to verify the account.

User Interface

Customer Dashboard:

- A personalized dashboard where users can view their profile, order history, and wishlist.
- Accessible options to update profile details (excluding email) and manage orders.
- View cart items, track orders, and add products to the wishlist for future purchases.

Admin Dashboard:

- A streamlined interface that enables admin users to manage all products, orders, categories, and users.
- Tools to add, edit, and delete products and categories, monitor stock levels, and view user profiles.
- Admin-specific reports on sales, inventory, and user activity.

Testing

Manual Testing:

- Verified all major functionalities, including user signup/login, product search and filtering, cartmanagement, order placement, and order tracking.

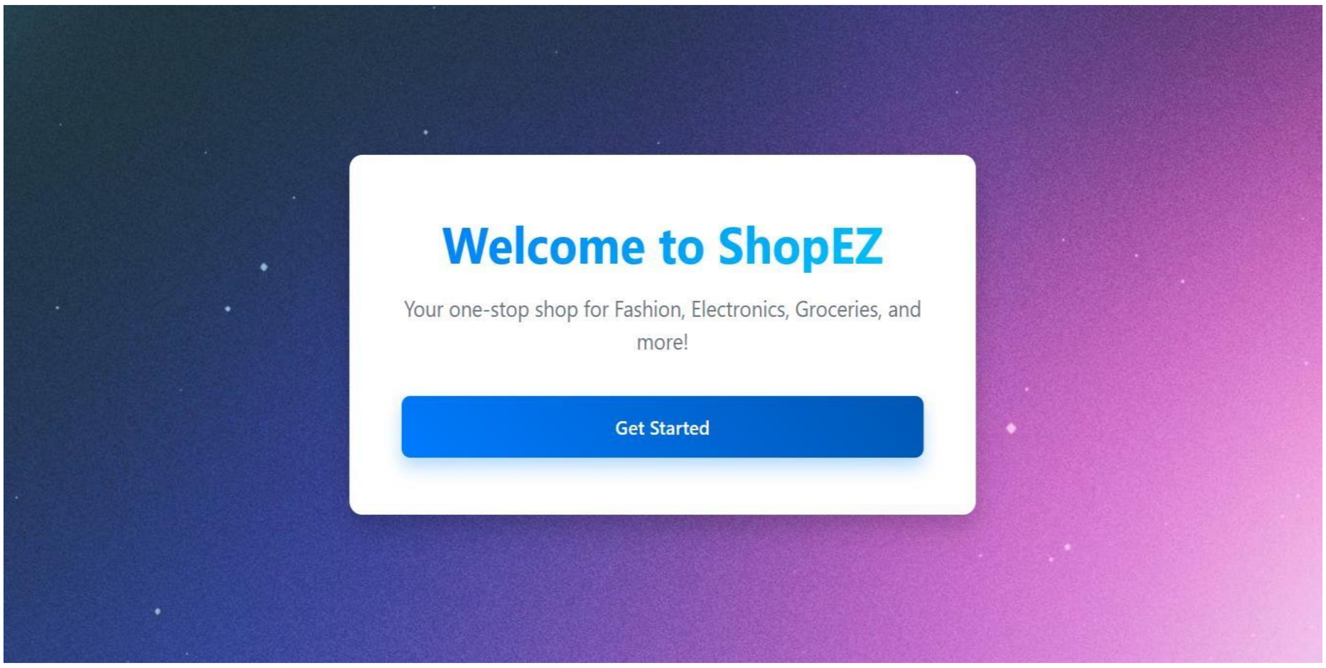
Postman Testing:

- Used for API endpoint testing, covering essential operations such as signup/login, product CRUDoperations, cart updates, and order handling.

Screenshots or Demo

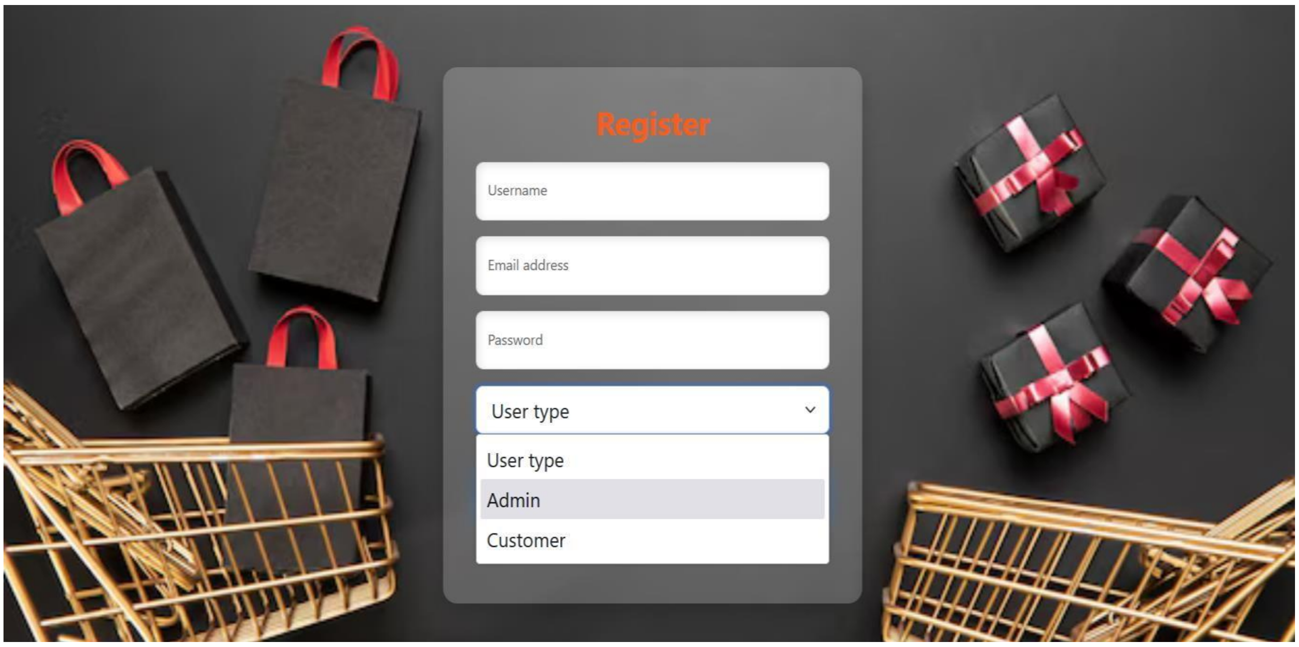
- **Demo Link: https://drive.google.com/file/d/1jSy3fhvGZlBvtGzUKvIxKWKfIg1bcagT/view?usp=sharing**
- **Screenshots:**

1. **Landing Page**

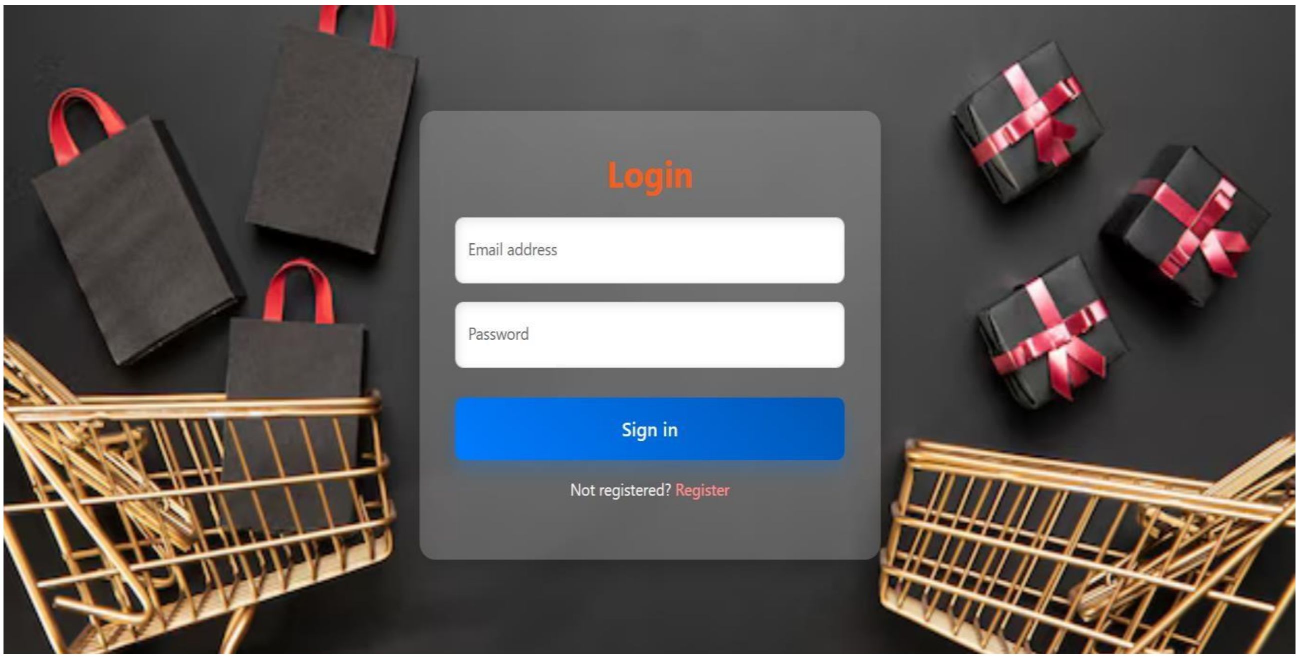

Welcome to ShopEZ

Your one-stop shop for Fashion, Electronics, Groceries, and more!

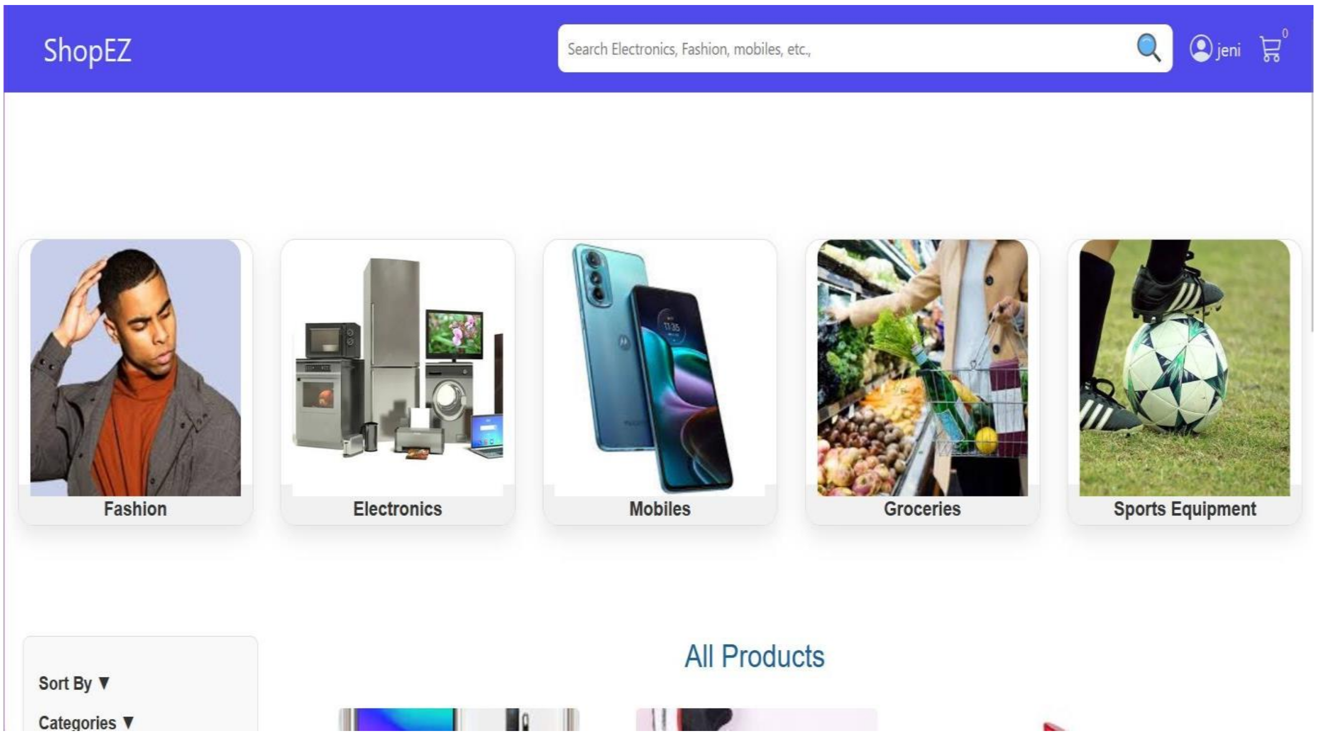
2. Register (customer and admin)



3. User Login



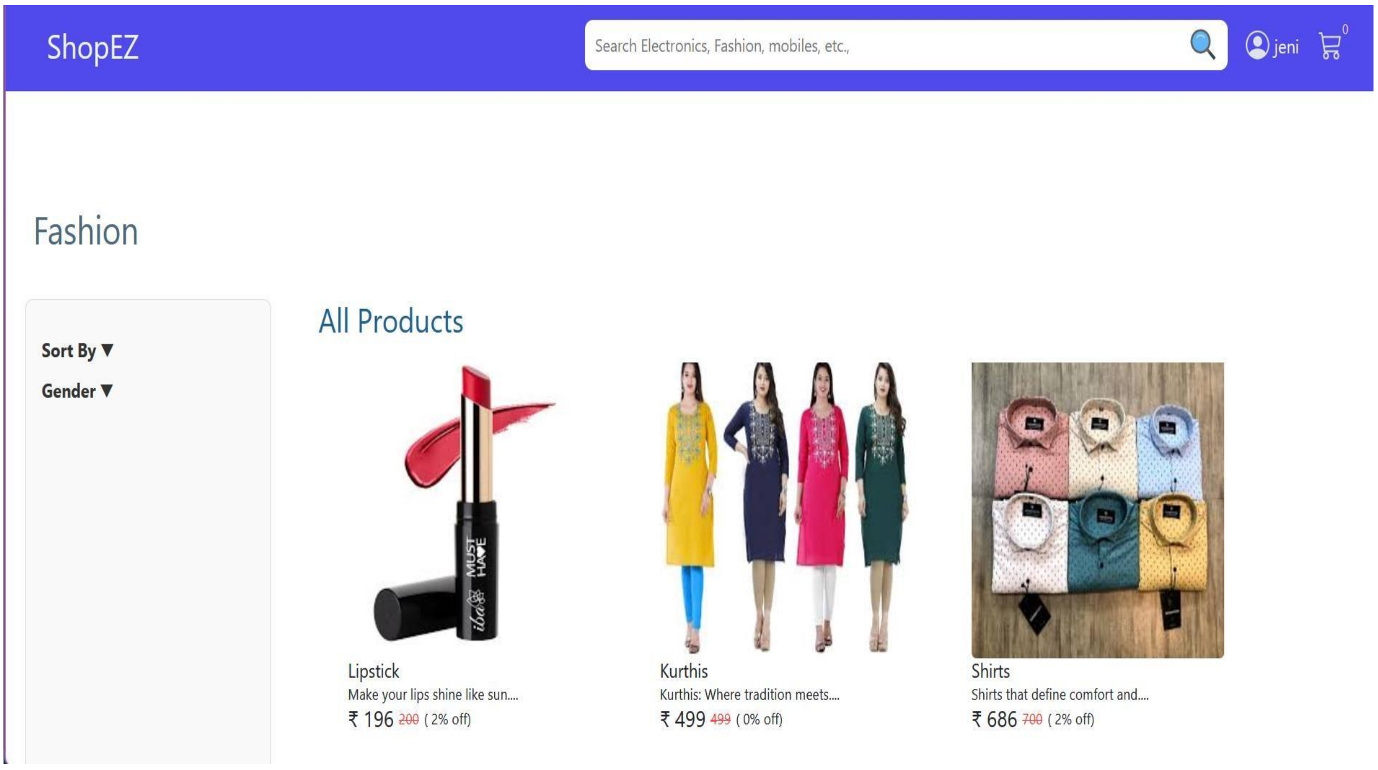
4. User Dashboard


ShopEZ

Search Electronics, Fashion, mobiles, etc.,

jeni

5. Product categorization


ShopEZ

Search Electronics, Fashion, mobiles, etc.,

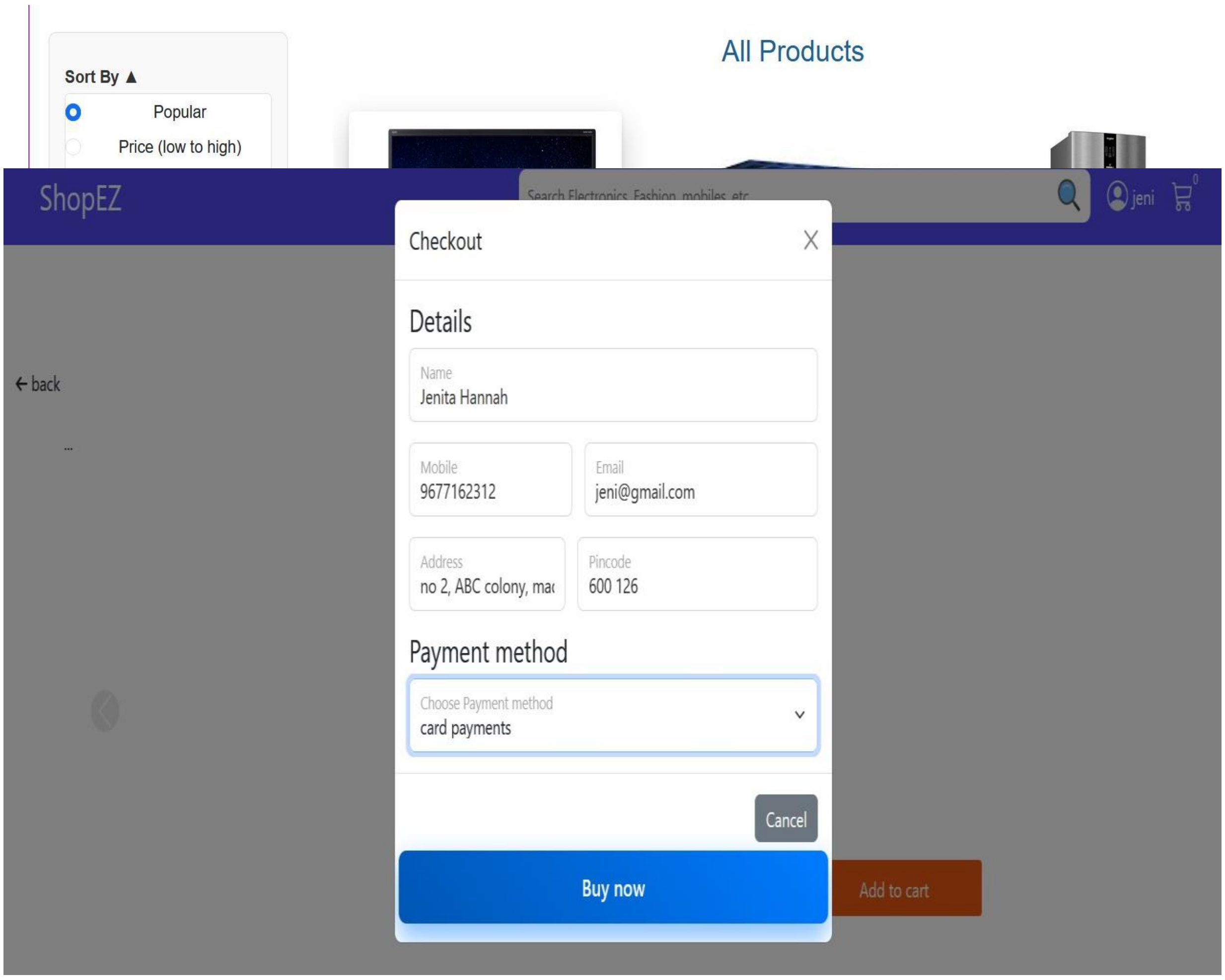
jeni

Lipstick
Make your lips shine like sun....
₹ 196 ~~200~~ (2% off)

Kurthis
Kurthis: Where tradition meets....
₹ 499 ~~499~~ (0% off)

Shirts
Shirts that define comfort and....
₹ 686 ~~700~~ (2% off)

6. ‘All Products’ section with categorizing options



Checkout

Details

Name
Jenita Hannah

Mobile
9677162312

Email
jeni@gmail.com

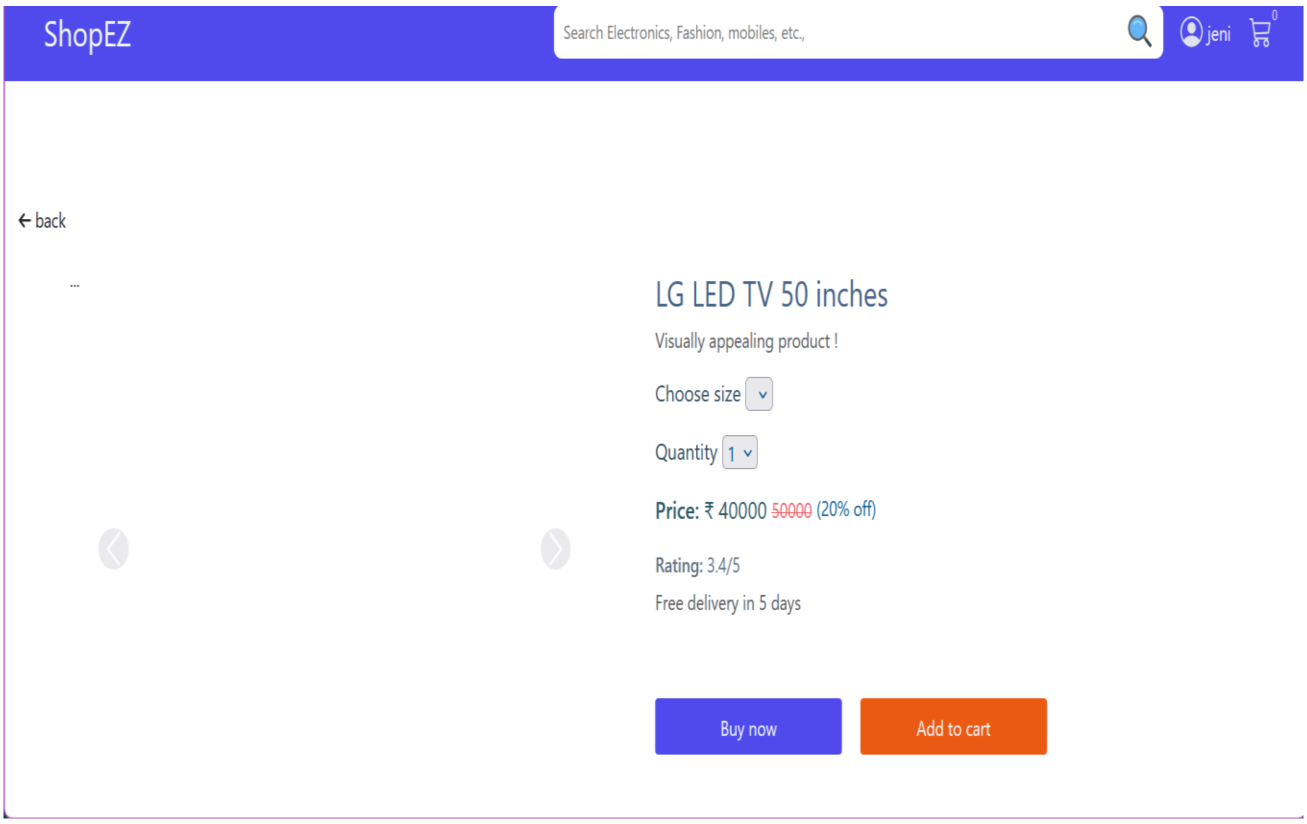
Address
no 2, ABC colony, mac

Pincode
600 126

Payment method

Choose Payment method
card payments

7. Placing Order



LG LED TV 50 inches

Visually appealing product !

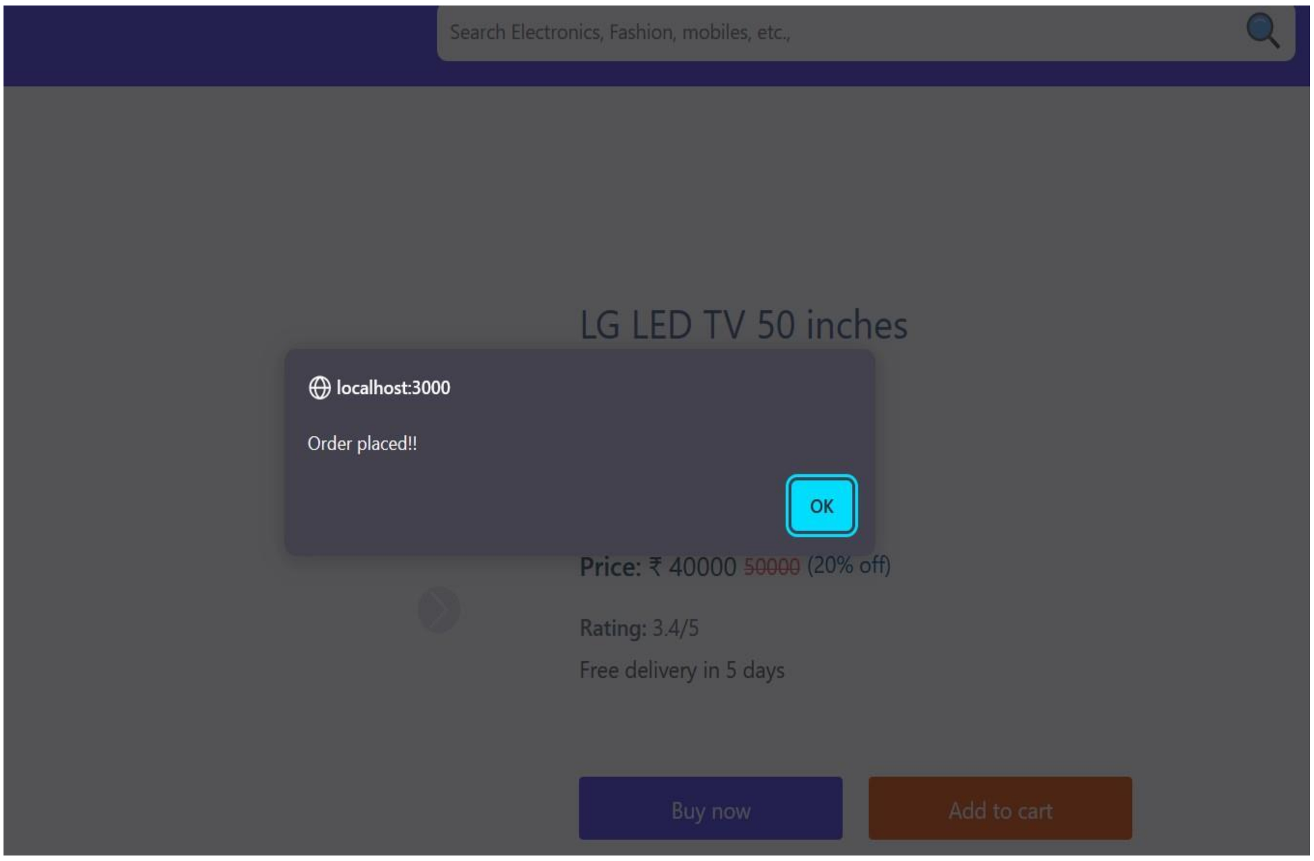
Choose size

Quantity 1

Price: ₹ 40000 ~~50000~~ (20% off)

Rating: 3.4/5

Free delivery in 5 days



LG LED TV 50 inches

localhost:3000

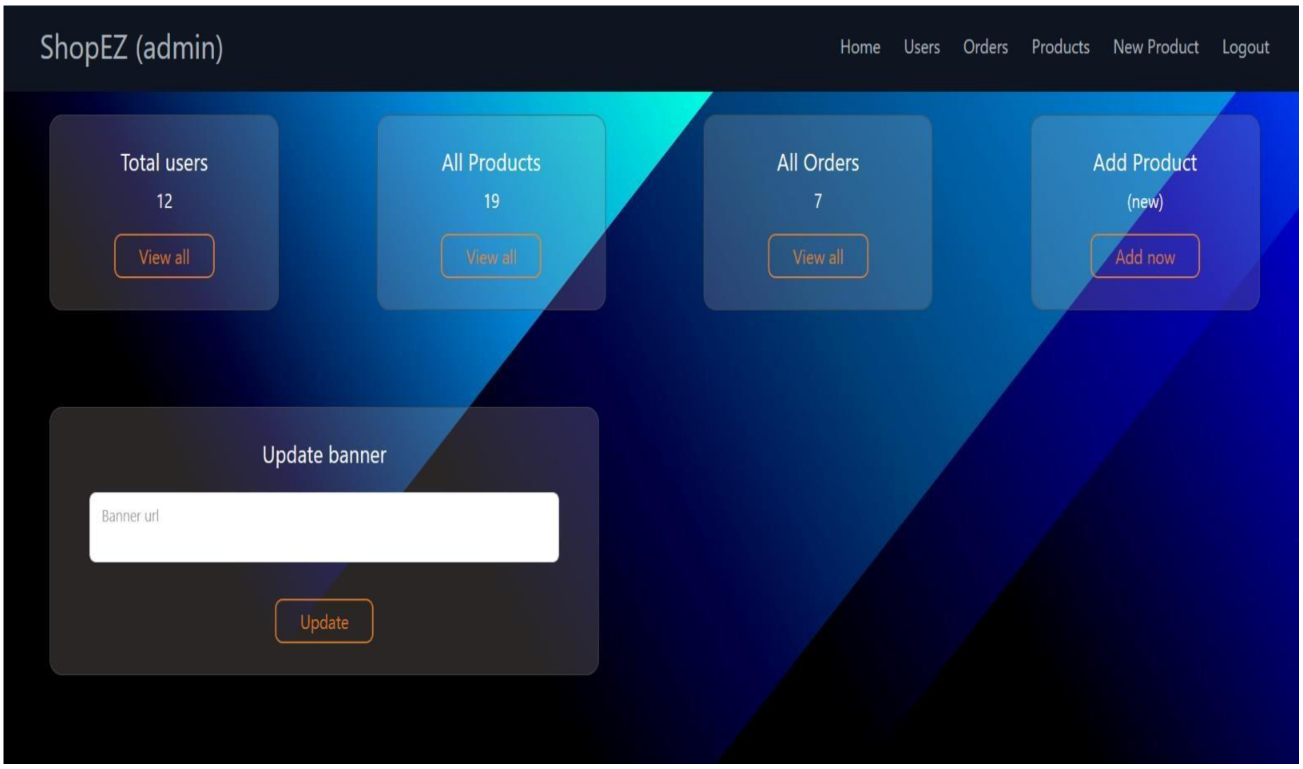
Order placed!!

Price: ₹ 40000 ~~50000~~ (20% off)

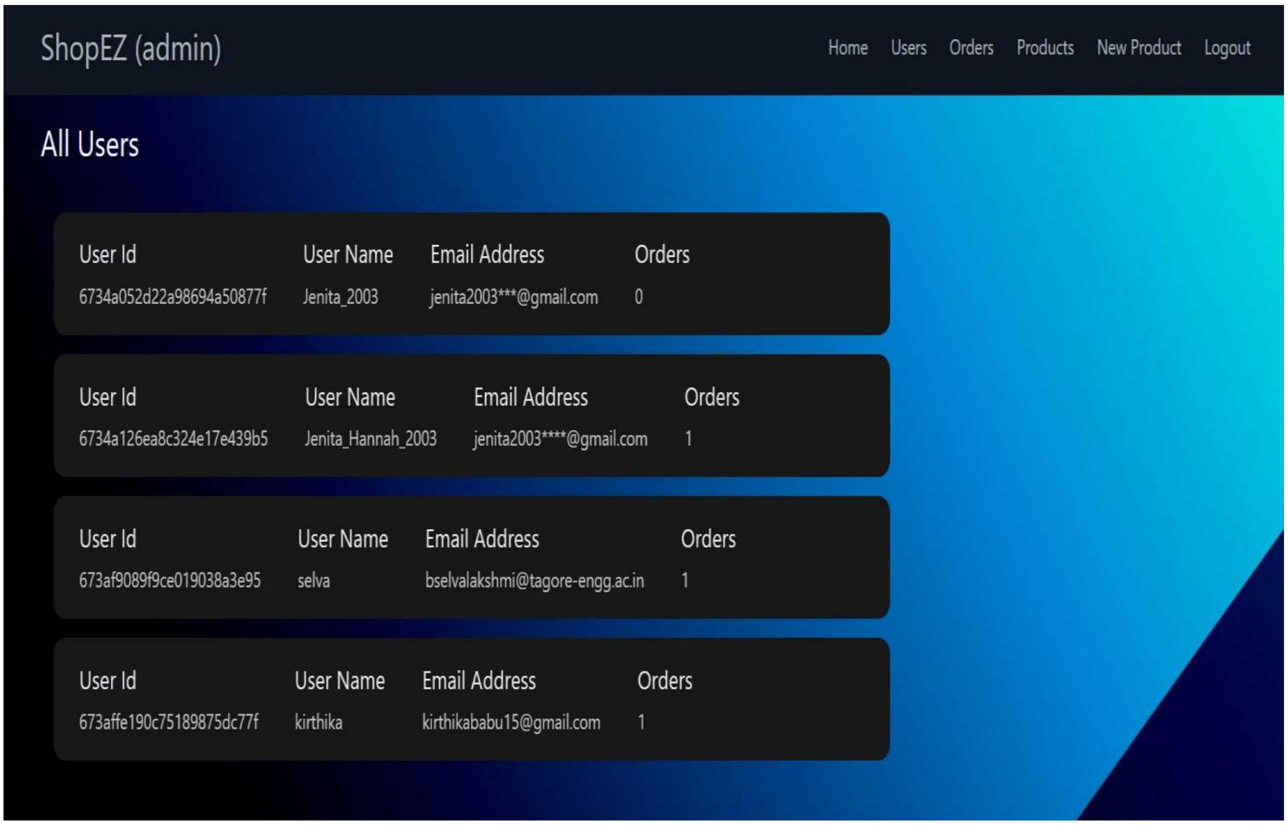
Rating: 3.4/5

Free delivery in 5 days

8. Admin Dashboard



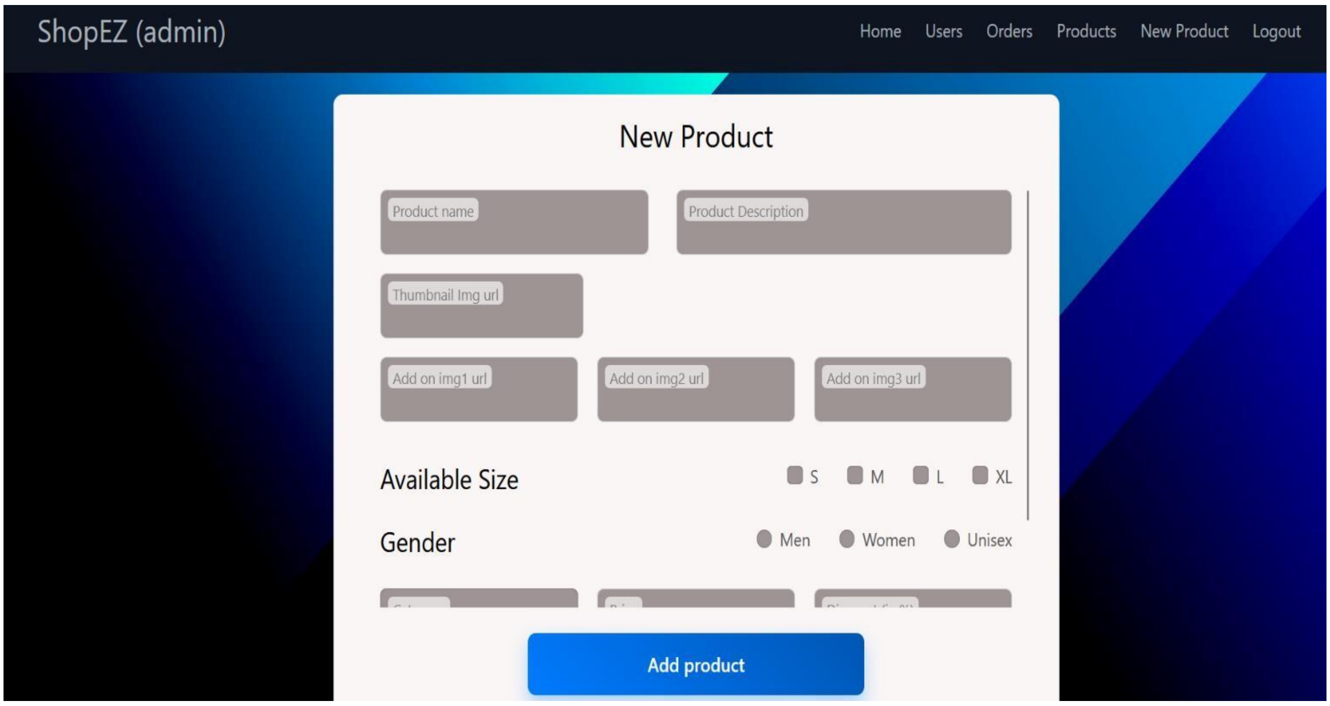
- **View of users' login details**

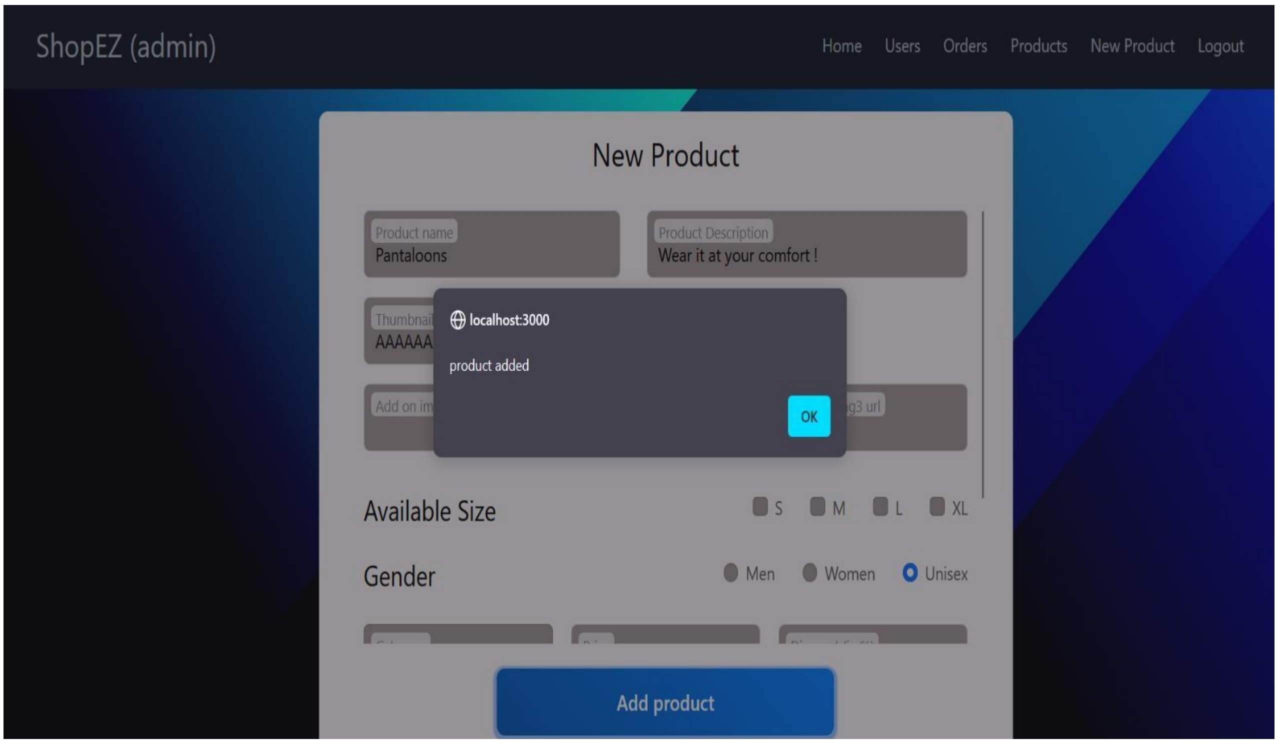


All Users

User Id	User Name	Email Address	Orders
6734a052d22a98694a50877f	Jenita_2003	jenita2003***@gmail.com	0
6734a126ea8c324e17e439b5	Jenita_Hannah_2003	jenita2003****@gmail.com	1
673af9089f9ce019038a3e95	selva	bselvalakshmi@tagore-engg.ac.in	1
673affe190c75189875dc77f	kirthika	kirthikababu15@gmail.com	1

- **Adding a new product from Admin side**





Product name: Pantaloons

Product Description: Wear it at your comfort !

Thumbnail: AAAAAA

localhost:3000

product added

Known Issues

- **Admin Login**: Currently, the admin panel lacks a dedicated login system, accessed only through a protectedroute.
- **Error Handling for File Uploads**: There is limited error handling for invalid product image or documentuploads.
- **No Payment Gateway**: Due to project requirements, the platform currently lacks a real payment gateway,and checkout is replaced with a mock payment process.
- **Profile and Product Editing Limitations**: When editing product details or user profiles, the platform doesn't automatically fetch the previously uploaded images or documents, requiring users to re-upload them.

Future Enhancements

- **Payment Gateway Integration**: Implement a secure payment gateway (e.g., Stripe, PayPal) to facilitateactual transactions during checkout.
- **Advanced Analytics for Admin**: Add in-depth analytics and reporting for the admin, including salesmetrics, product performance, and customer insights.
- **Recommendation Engine**: Incorporate machine learning algorithms to recommend products based on userpreferences, browsing history, and purchase patterns.
- **Live Customer Support via WebRTC**: Integrate WebRTC to enable live chat or video support forcustomers seeking assistance in real time.