**ShopEZ:One-Stop Shop for Online Purchases**

TEAM ID : LTVIP2025TMID59506

Team Members:

|  |  |
| --- | --- |
| **MEMBER** | **RESPONSIBILITY** |
| 1. Nancharla Vinaykumar | Frontend Development |
| 2. Mallam Parameswari | DevOps & Deployment |
| 3. Manchikanti Chandra S S Lakshmi Narayana | Backend & API Development |
| 4. Mata Nithin Kumar | Documentation & Testing |

**1. Introduction**

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

**2. Project Overview**

**Purpose:**  
ShopEZ is a full-stack e-commerce platform designed to empower small and local shopkeepers to take their businesses online with ease. The platform bridges the digital gap by providing an accessible marketplace where sellers can list products and customers can browse, order, and pay securely.

**Features:**

* Customer and seller registration/login
* Product listing, editing, and order tracking by sellers
* Product search, filtering, cart, and checkout for customers
* Admin dashboard to manage seller approvals and platform activity
* Secure payment integration using Stripe or Razorpay
* Email/SMS notifications for order updates
* Review and rating system for customer feedback

**3. Architecture**

**Frontend:**  
Built using React.js, featuring:

 Functional component-based architecture with hooks

 React Router v6 for page navigation

 Context API for global state (user, cart, auth)

 Tailwind CSS for responsive design

 Axios for REST API communication

**Backend:**  
Built using Node.js and Express.js:

 RESTful API structure

 Middleware: CORS, dotenv, morgan, validation

 JWT for secure user authentication

 Modular structure: Routes, Controllers, Services

**Database:**

MongoDB with Mongoose ORM:

* Collections: Users, Products, Orders, Reviews, Payments
* Hosted on MongoDB Atlas

**4. Setup Instructions**

**Prerequisites:**

* Node.js (v16 or higher)
* npm or yarn
* MongoDB Atlas account or local instance
* Git

**Installation Steps:**

* Clone the repository

git clone https://github.com/NARAY333/ShopEZ

cd shopez

* Set up environment variables:

Create .env file in the server directory

PORT=5000

MONGO\_URI=mongodb://localhost:27017/shopEZ

JWT\_SECRET=mysecretpassword

* Install dependencies:

**#Backend**

cd server

npm install

**#Frontend**

cd client

npm install

**Folder Structure**

**CLIENT:**

**client/**

├── .gitignore

├── package.json

├── README.md

├── build/

│ ├── asset-manifest.json

│ ├── favicon.ico

│ ├── index.html

│ ├── logo192.png

│ ├── logo512.png

│ ├── manifest.json

│ ├── robots.txt

│ └── static/

│ ├── css/

│ ├── js/

│ └── media/

├── public/

│ ├── favicon.ico

│ ├── index.html

│ ├── logo192.png

│ ├── logo512.png

│ ├── manifest.json

│ └── robots.txt

├── src/

│ ├── App.css

│ ├── App.js

│ ├── App.test.js

│ ├── index.css

│ ├── index.js

│ ├── logo.svg

│ ├── reportWebVitals.js

│ ├── setupTests.js

│ ├── components/

│ │ ├── FlashSale.jsx

│ │ ├── Footer.jsx

│ │ ├── Login.jsx

│ │ ├── Navbar.jsx

│ │ ├── Products.jsx

│ │ └── Register.jsx

│ ├── context/

│ ├── images/

│ ├── pages/

│ └── styles/

SERVER:

server/

├── index.js

├── package.json

└── Schema.js

**6. Running the Application**

To start both servers:

Frontend:

Navigate to the client directory:

cd client

Start the development server:

npm start

The frontend application runs on: http://localhost:3000

Backend:

Navigate to the server directory:

cd server

Start the backend server:

node index.js

The backend application runs on: http://localhost:6001

**7. API Documentation**

**Base URL: http://localhost:3000/**

**Auth Routes:**

* POST /auth/register – Register customer/seller
* POST /auth/login – Login and get JWT token

**Product Routes:**

* GET /products – List all products
* POST /products – Add product (Seller only)
* PUT /products/:id – Edit product
* DELETE /products/:id – Delete product

**Order Routes:**

* POST /orders – Place order (Customer)
* GET /orders – View user/seller orders
* PUT /orders/:id/status – Update order status

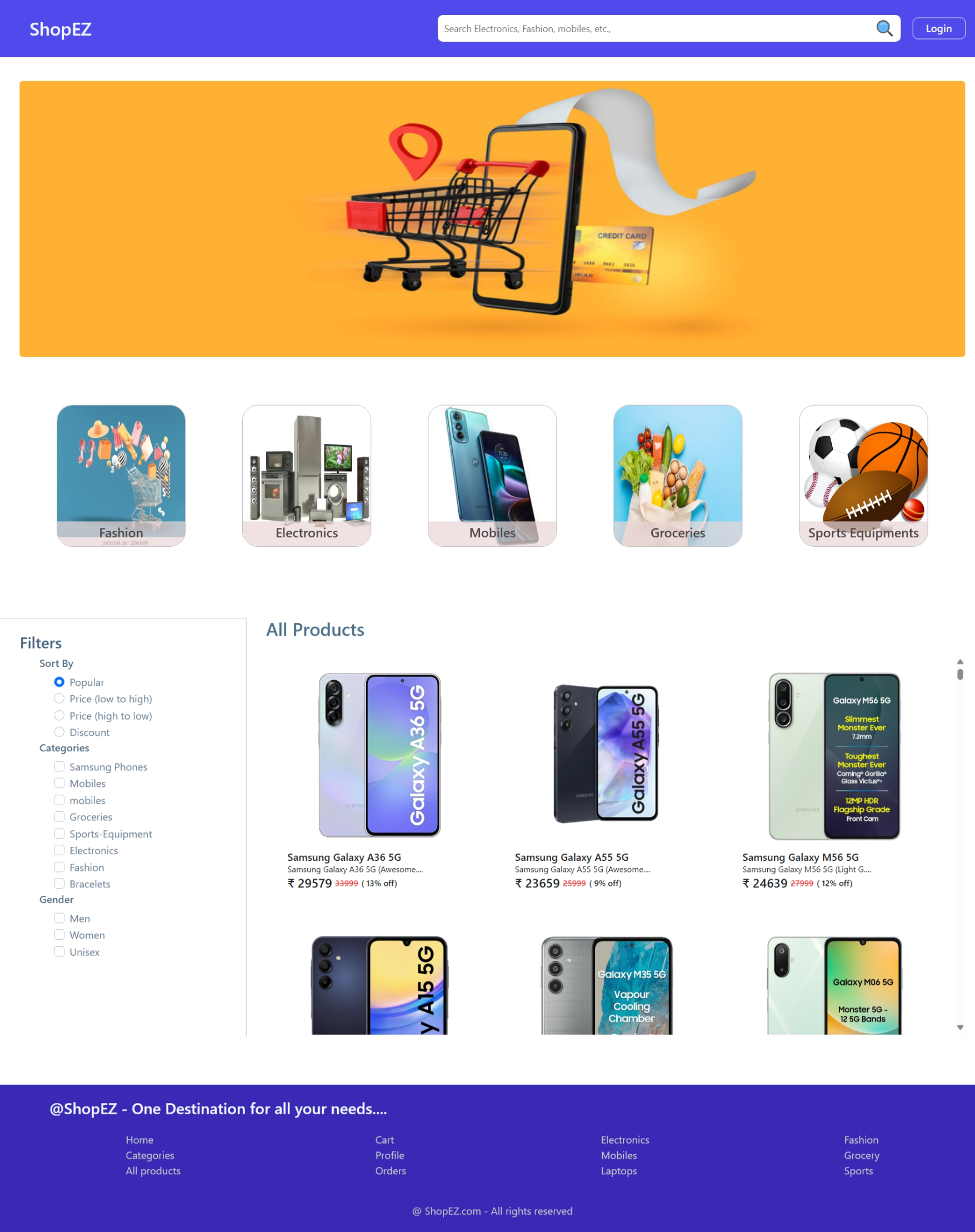
**8. Authentication**

Handled using JWT:

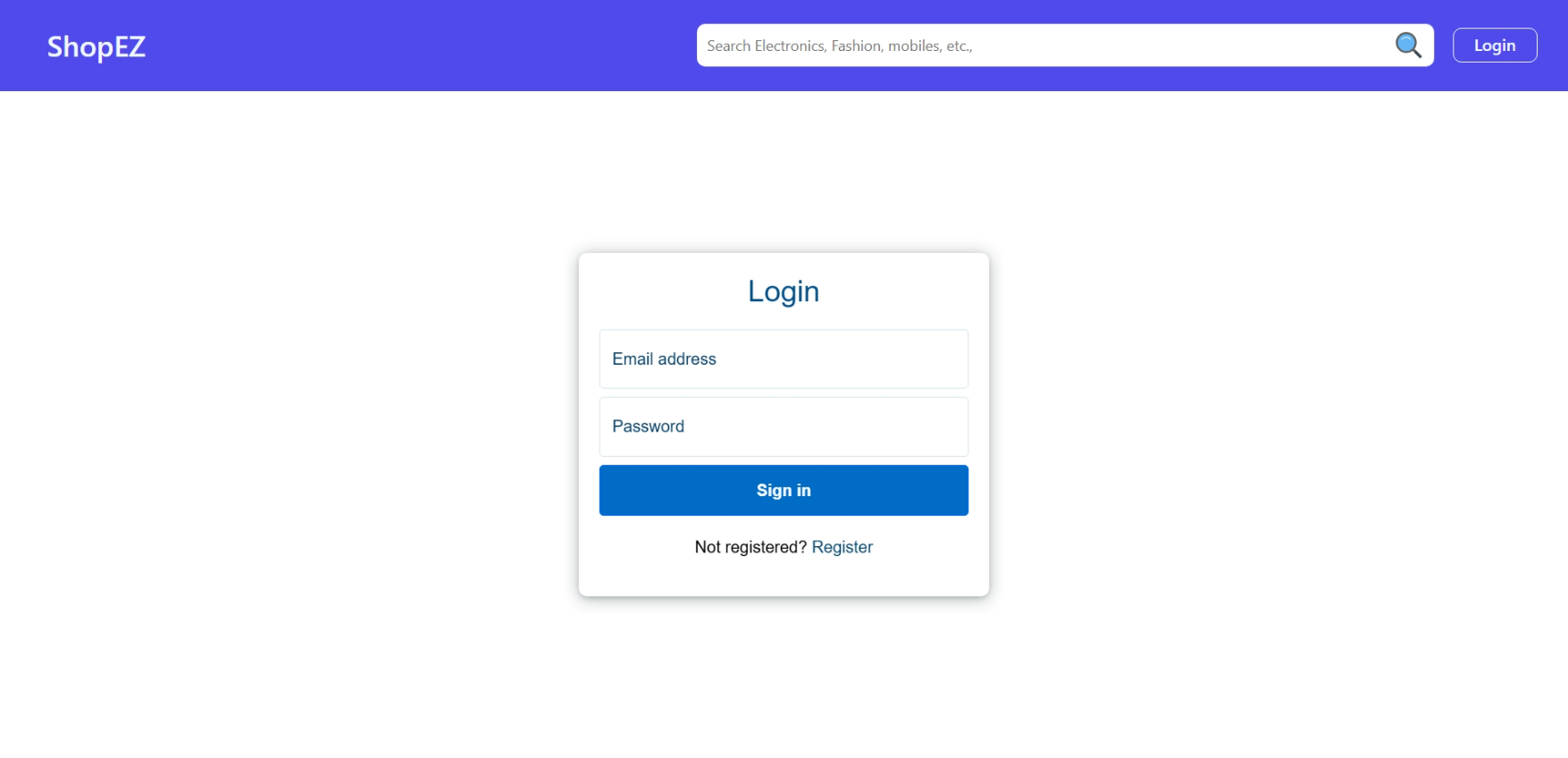
* Token issued on login and stored in localStorage
* Protected routes verified via middleware
* Role-based access for Customer, Seller, and Admin

**9. Screenshots or Demo**

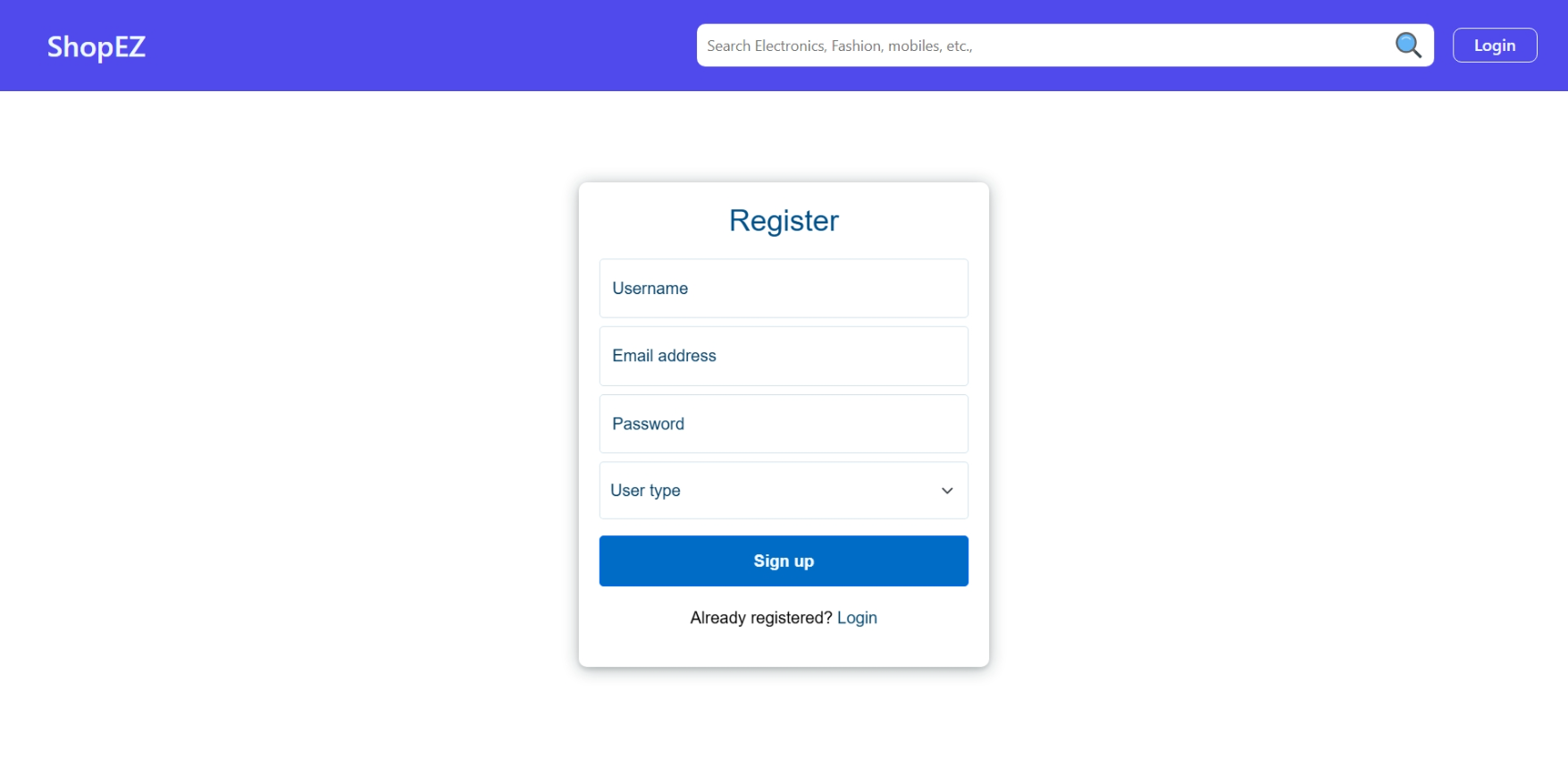
**HOME PAGE:**



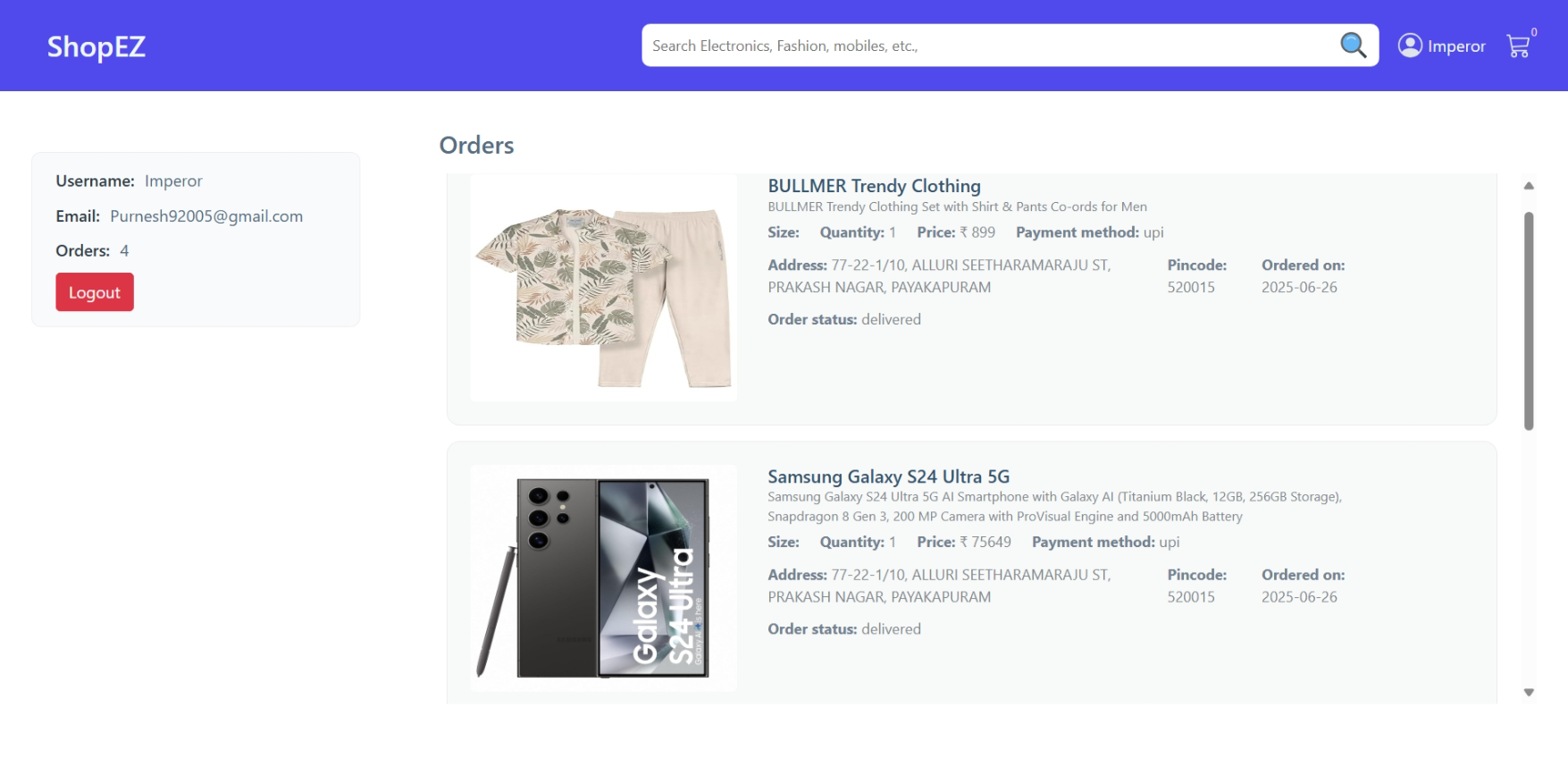
**LOGIN PAGE:**



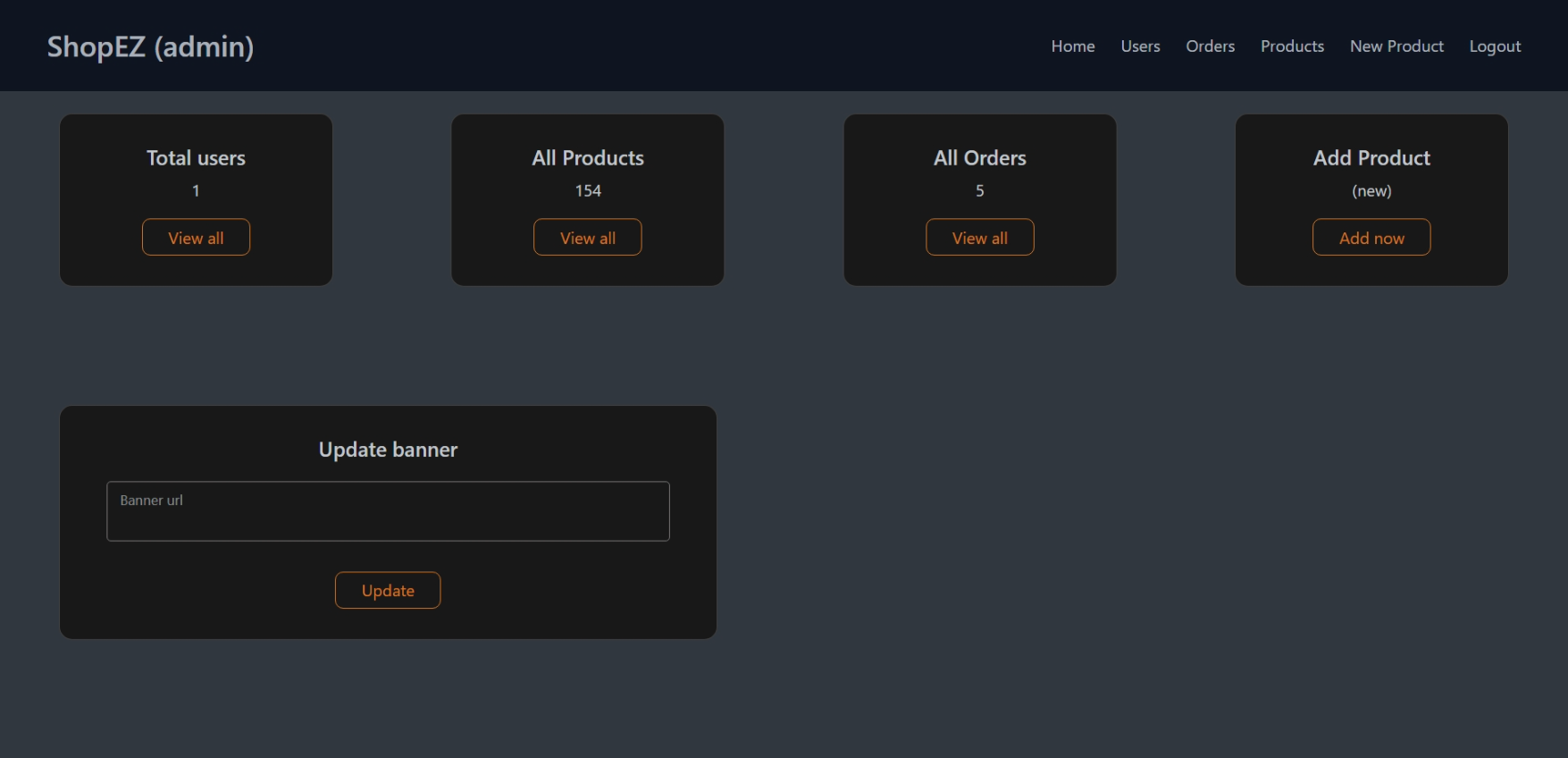
**REGISTER PAGE:**



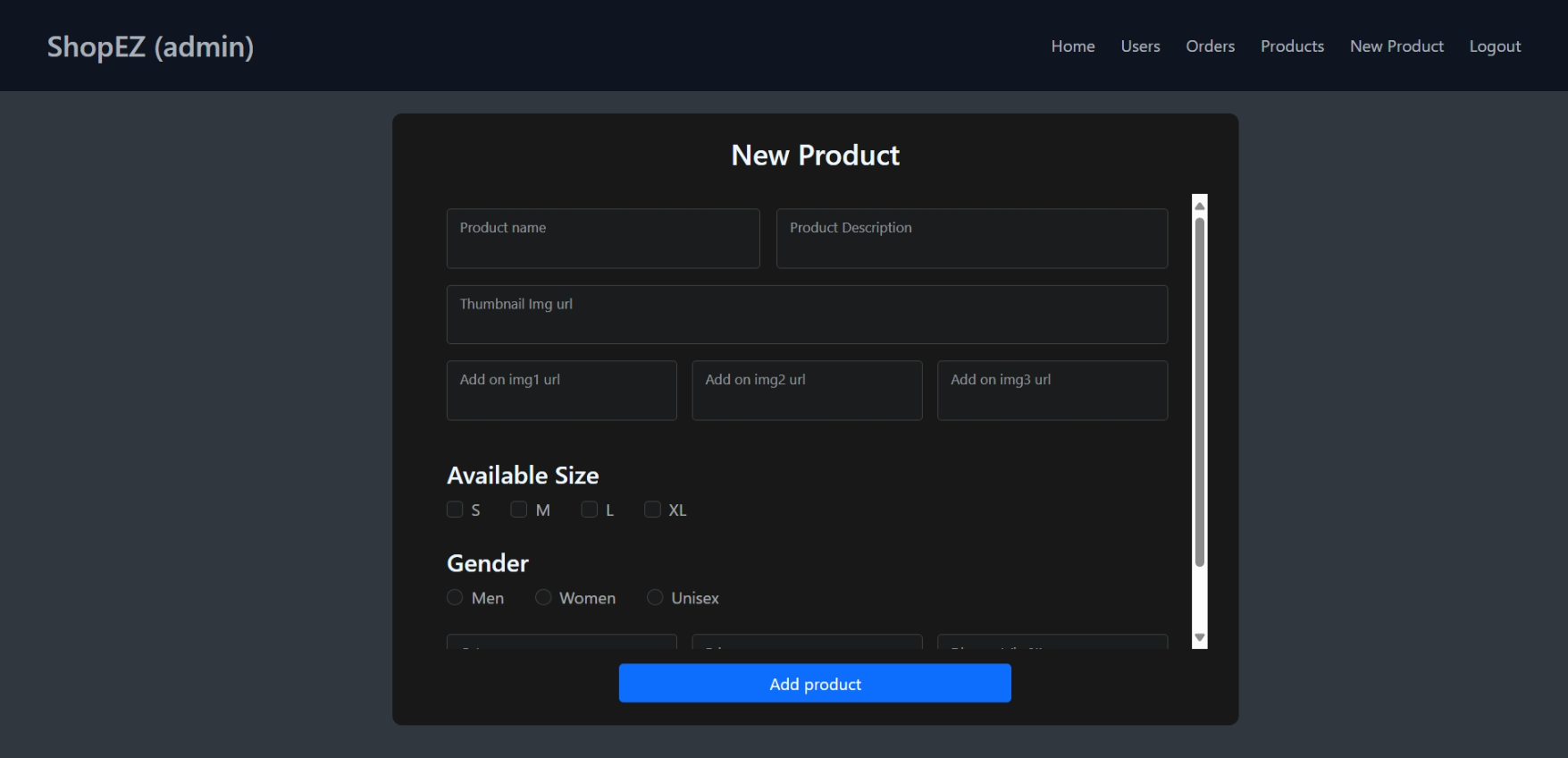
**PROFILE PAGE(USER):**



**ADMIN PAGE:**

****

**PRODUCT ADD PAGE:**

****

**10. Testing**

* **Frontend:** Tested with React Testing Library
* **Backend:** Tested using Postman and Jest
* **E2E Testing:** Simulated with Cypress

**11. Known Issues**

**Product images load slowly on low bandwidth connections**

* Large image files delay page rendering for some users.

**Order status does not auto-refresh on the dashboard**

* Users must manually refresh the page to view order updates.

**Admin approval system lacks real-time notifications**

* Sellers are not immediately informed when their accounts are approved or rejected.

**Cart state is lost on browser refresh for unauthenticated users**

* Items in the cart are not stored locally before login.

**Limited responsiveness on older mobile devices**

* Some dashboard components do not scale properly on small screen sizes.

**12. Future Enhancements**

Add voice search and product recommendations

Launch mobile app version using React Native

Enable multi-vendor management and seller analytics

Integrate delivery tracking system

Support regional languages and accessibility features