

FullStack Development with MERN

Project Documentation : ShopEZ E- One-Stop Shop For Online Purchases

1. Introduction

Project Title :

Team Members:

Kolanti Manogna–Full Stack Developer

Andugulapati Prasanna – Frontend

Tata Bhavya Sri – Backend Developer

Puppalla Rama Krishna – Database Administrator

2. Project Overview

Purpose:

The **ShopEZ** project is designed to deliver a modern and efficient **e-commerce platform** using the **MERN stack (MongoDB, Express.js, React.js, Node.js)**. It aims to provide:

- A seamless **shopping experience for customers**.
- An intuitive and powerful **dashboard for sellers/admins** to manage products and orders.
- Secure and scalable infrastructure to support real-world online shopping needs.

Features:

- Comprehensive Product Catalog
- Effortless Product Discovery
- Personalized Recommendations
- Shop Now Functionality
- Secure & Seamless Checkout
- Order Confirmation and Tracking
- User Profile Management
- Cart Management

3. Architecture

Frontend:

The frontend of **ShopEZ** is built using **React.js**, a popular JavaScript library for building fast, interactive, and component-based user interfaces. It ensures users have a smooth and responsive shopping experience.

Backend:

The backend of ShopEZ is developed using **Node.js** with the **Express.js** framework. It serves as the backbone of the application, managing data flow, business logic, and secure communication between the client and the database.

Database:

uses **MongoDB**, a NoSQL, document-based database, as its data storage solution. It stores data in collections (similar to tables in relational databases), using **Mongoose** as an Object Data Modeling (ODM) library in Node.js to define schemas and manage interactions.

4. Setup Instructions

Prerequisites:

- Node.js
- npm
- MongoDB
- Vite
- Express.js

5. FolderStructure

```
shopEZ--e-commerce-MERN/
├── client/
│   ├── public
│   └── src/
│       ├── components/
│       ├── pages/
│       ├── App.js
│       └── index.js
├── server/
│   ├── controllers/
│   ├── routes/
│   ├── .env
│   ├── index.js
│   └── package.json
├── .gitignore
├── package.json
├── README.md
└── .env
```

6. Running the Application

Frontend:

npm run dev

Backend:

npm start

Access URL:

Default port for frontend: **3000**

Default port for backend: **5000**

7.API Documentation

Example Endpoints:

- User Authentication APIs
- Product API
- Cart Management API
- Order Management API
- Admin Control API

8. Authentication

Method Used:

JSON Web Tokens(JWT)

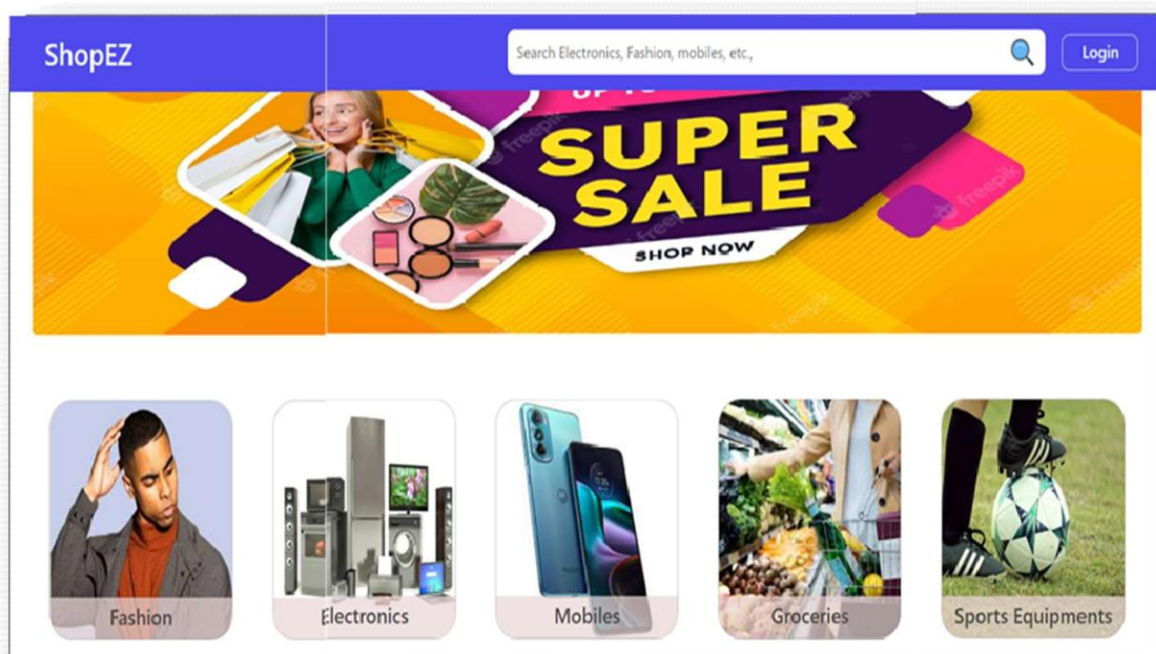
How it Works:

- Users register and login to get JWT tokens.
- Protected routes use authentication middleware.
- User roles(Admin,User(Customer)) are checked at the API level.

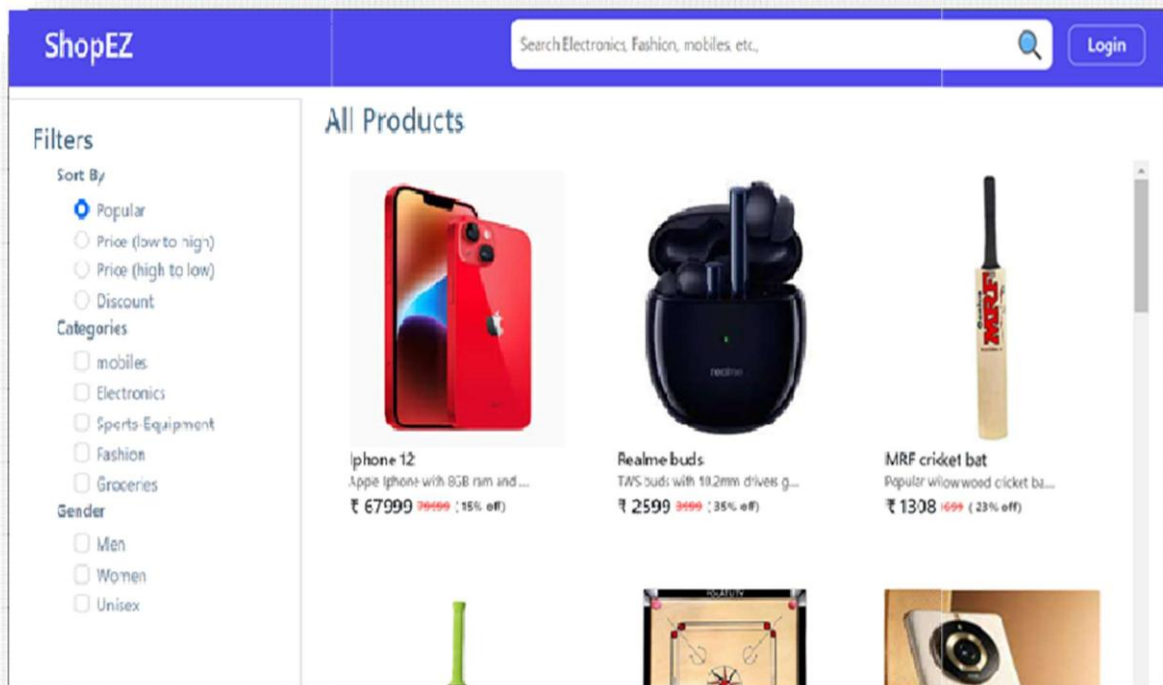
9.User Interface

Screens Included:

Landing Page



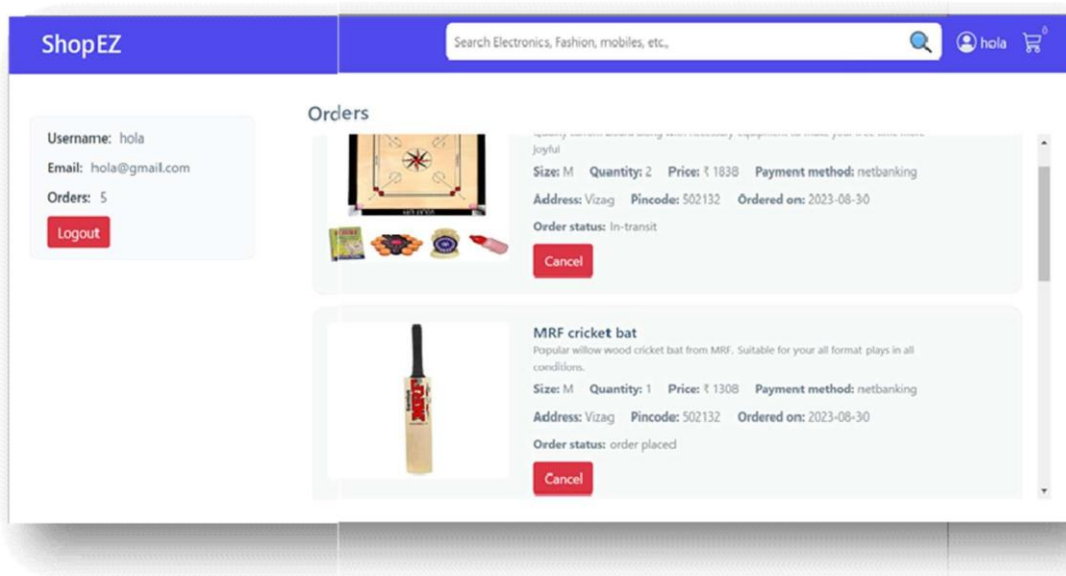
Products:



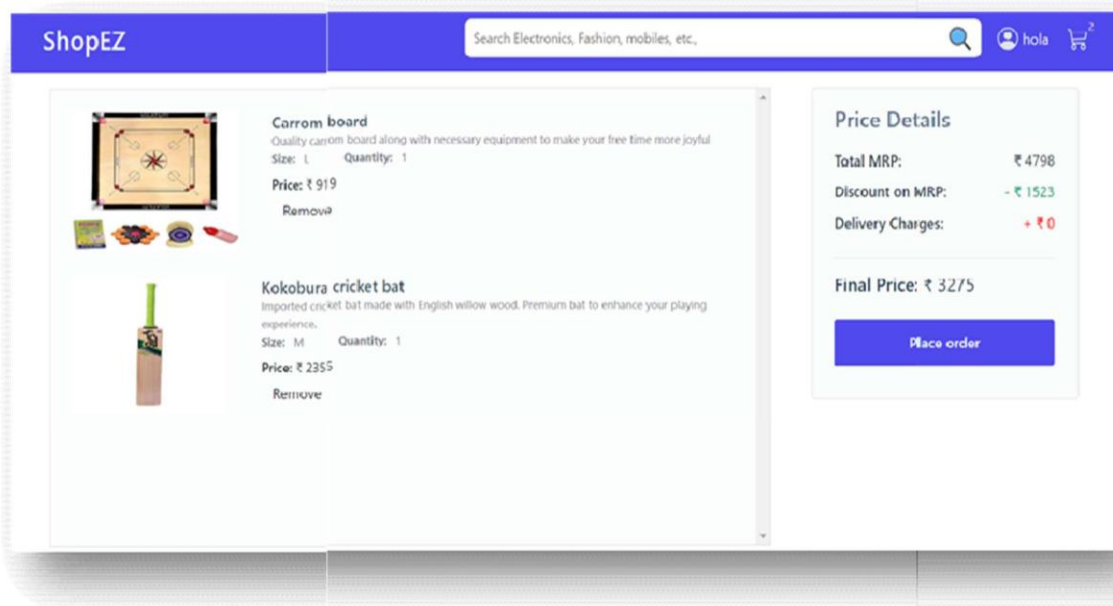
Authentication:

The screenshot shows the 'Register' form on the ShopEZ website. The header is identical to the previous page. The form contains the following fields: 'Username', 'Email address', 'Password', and a 'User type' dropdown menu. The dropdown menu is open, showing 'Admin' and 'Customer' as options. Below the form, there is a link that says 'Already registered? Login'.

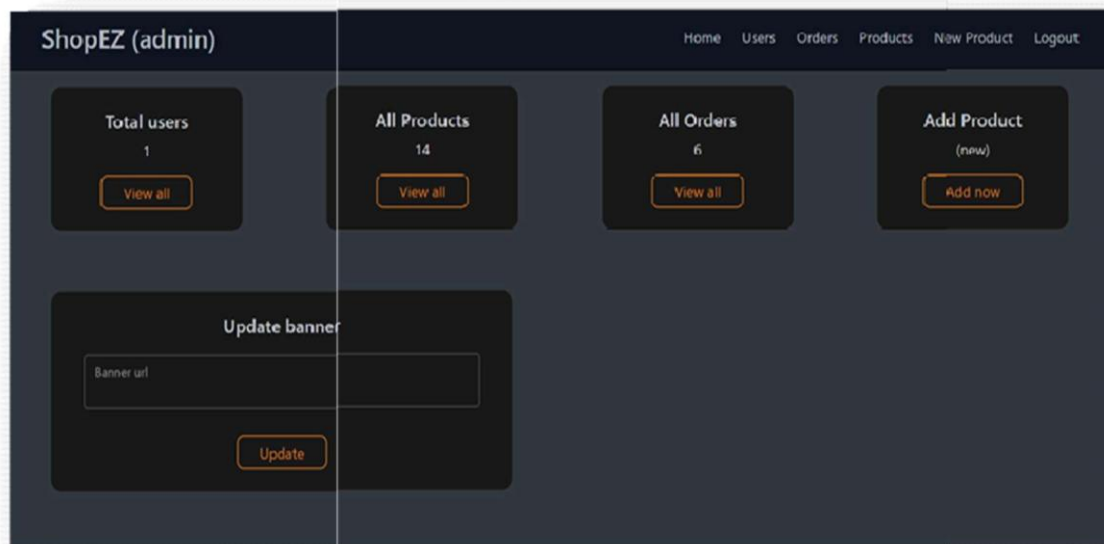
User Profile:



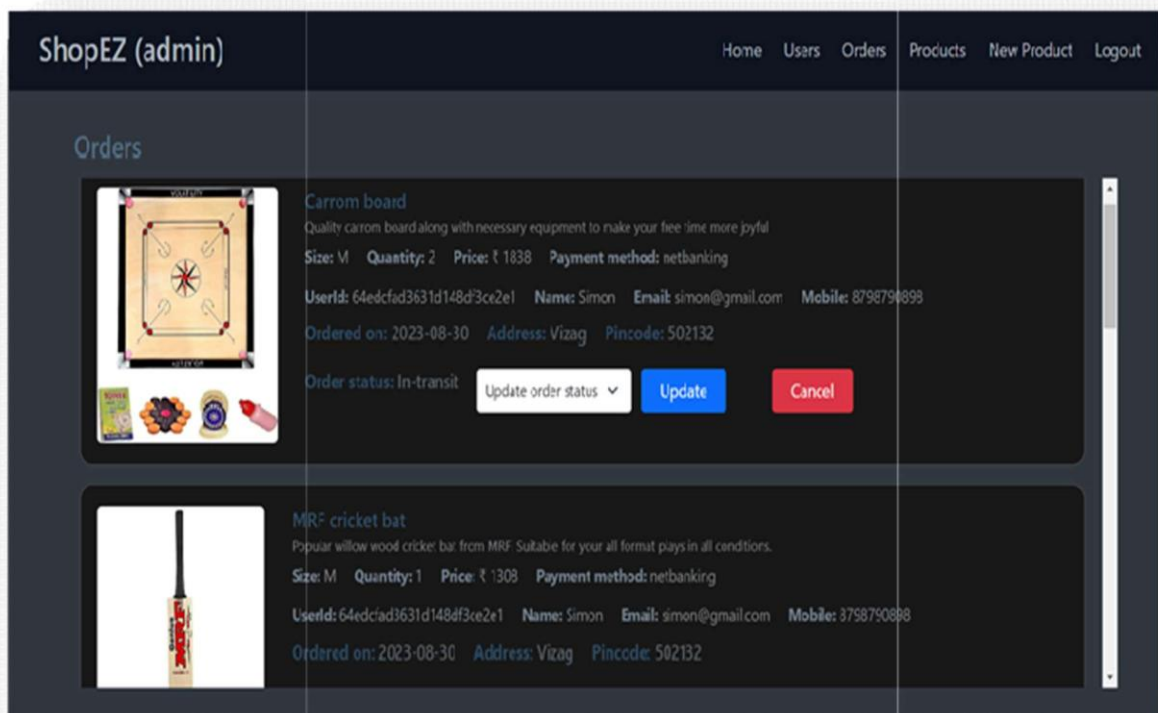
Cart:



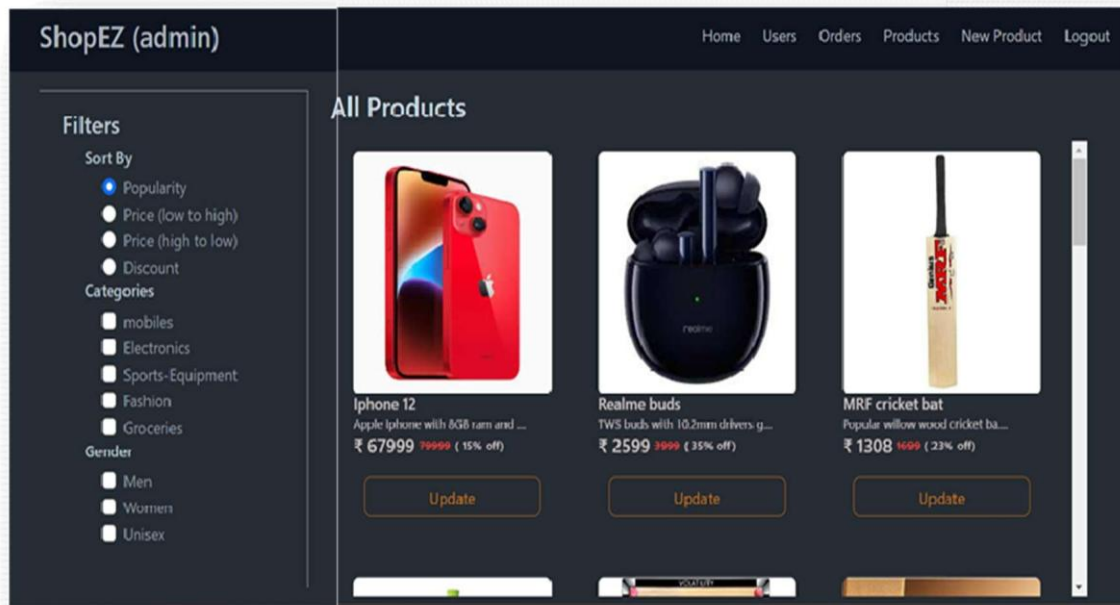
Admin Dashboard:



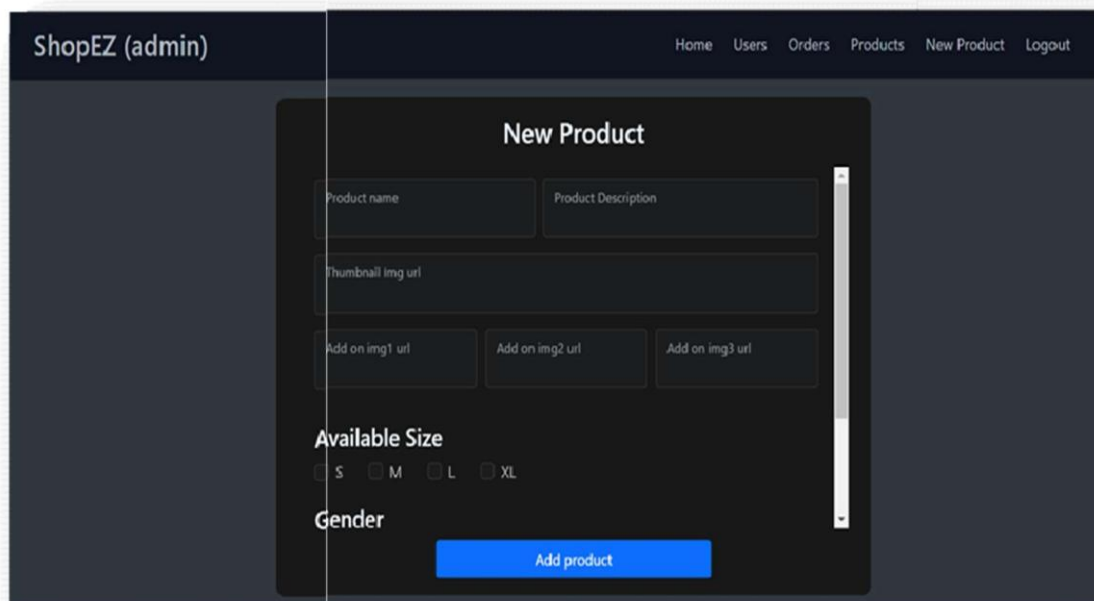
All Orders:



All Products:



New Product Page:



10. Testing

Testing Tools:

- Postman for API testing
- Manual frontend testing
- Basic unit tests on backend routes(optional)

11. Demo

Demo Video: [shopEZ MERN demo.mp4](#)

12. Known Issues

- Limited payment gateway features
- Minimal frontend form validation
- Error handling for API responses can be improved

13. Future Enhancements

- Mobile App Development
- Two-Factor Authentication (2FA)
- Live Chat Support / Chatbot
- Advanced Order Tracking System
- Admin Analytics Dashboard
- Seller Onboarding & Marketplace Support
- AI-Powered Recommendations