

Full Stack Development with MERN

Project Documentation

E-Commerce Web Application (ShopEZ)

DHAANISH AHMED COLLEGE OF ENGINEERING

Department of Artificial Intelligence And Data Science

Team Members:

- | | |
|-------------------|-------------------------|
| 1. R. Anusha | UI/UX Developer |
| 2. I. D. Puvishaa | Frontend Developer |
| 3. K.Valli | Backend Developer |
| 4. C.Raisa | Manual/Automated Tester |

Submitted to Naan Mudhalavan Team

ANNA UNIVERTSITY.

UNDER THE GUIDANCE OF

PROF Dr.A.Thenmozhi

1.INTRODUCTION

PROJECT TITLE : E-Commerce Site using MERN STACK

- ShopEZ is a contemporary e-commerce application designed to meet the growing demand for online shopping platforms in the digital era. With a focus on user experience, personalization,

and business scalability, ShopEZ provides businesses with a reliable and innovative solution to connect with their customers globally.

2.PROJECT OVERVIEW

PURPOSE

ShopEZ is an e-commerce platform designed to revolutionize online shopping by combining a seamless user experience with powerful business tools. Built with the MERN stack, the platform caters to businesses of all sizes, enabling them to showcase products, process orders efficiently, and interact with customers in real-time.

The key objectives of ShopEZ are:

- To create a secure and scalable platform for online shopping.
- To provide a dynamic and responsive interface for users.
- To offer businesses tools for inventory and order management.
- To integrate modern technologies for a competitive edge.

FEATURES

Dynamic Product Catalog

- Displays products with real-time updates and customizable filters.
- Supports multiple categories and product variations.

Shopping Cart Management

- Allows users to add, update, and remove products from the cart.
- Offers real-time calculation of totals, including discounts and taxes.

Admin Dashboard

- Enables product and order management with analytics and reporting features.

Mobile-Friendly Design

- Optimized for seamless navigation across devices.

3. ARCHITECTURE

Frontend:

- The frontend is built using React.js, featuring a component-based architecture that promotes reusability and maintainability. It utilizes React Router for navigation and Redux for state management.

Backend:

- The backend is developed using Node.js and Express.js, providing a RESTful API that handles client requests, business logic, and data processing.

Database:

- MongoDB is used as the database to store user profiles, task details, and collaborative project information. Mongoose is employed for object modeling and schema validation.

4. SETUP INSTRUCTIONS

- **Prerequisites:**
 -
 - Node.js
 -
 - MongoDB
 -
- **Clone the repository:**
 -
 - `git clone https://github.com/MSivananthan/NM-E-commerce-Site.git`
 -
- **Navigate to the client directory:**
 -
 - `cd chicmart/client`
 -
- **Install frontend dependencies:**
 -
 - `npm install`
 -
- **Set up environment variables (create a .env file):**
 - - `MONGODB_URI=your_mongodb_uri`
 -
 - `JWT_SECRET=your_jwt_secret`
 -
- **Navigate to the server directory:**
 -
 - `cd ../server`
 -
- **Install backend dependencies:**
 -
 - `npm install`

5. FOLDER STRUCTURE

Client Directory Structure:

```
text
/client
├── /public      # Static files like index.html
├── /src         # Source files for React components
│   ├── /components  # Reusable components (e.g., Header, Footer)
│   ├── /pages      # Page components (e.g., Home, Dashboard)
│   ├── /hooks      # Custom hooks for state management
│   └── App.js      # Main application file
```

Server Directory Structure:

```
text
/server
├─ /config          # Configuration files (e.g., database
connection)
├─ /controllers     # Business logic for handling requests
├─ /models          # Mongoose models for MongoDB collections
├─ /routes          # API route definitions
└─ server.js        # Entry point for the Node.js application
```

6. RUNNING THE APPLICATION

- Provide commands to start the frontend and backend servers locally.
 - **Frontend:** `npm start` in the client directory.
 - **Backend:** `npm start` in the server directory.

7. API DOCUMENTATION

- **GET /api/tasks:** Retrieve all tasks for the authenticated user.
- **POST /api/users/register:** Register a new user.
- **POST /api/users/login:** Authenticate a user.
- **POST /api/tasks:** Create a new task.
- **PUT /api/tasks/:id:** Update an existing task.
- **DELETE /api/tasks/:id:** Delete a task.

8. AUTHENTICATION

- Authentication is handled using JWT (JSON Web Tokens) for secure user login and session management, where tokens are stored in HTTP-only cookies or local storage. Authorization is managed by verifying user roles and permissions based on the JWT claims to control access to protected routes and resources.

9. USER INTERFACE

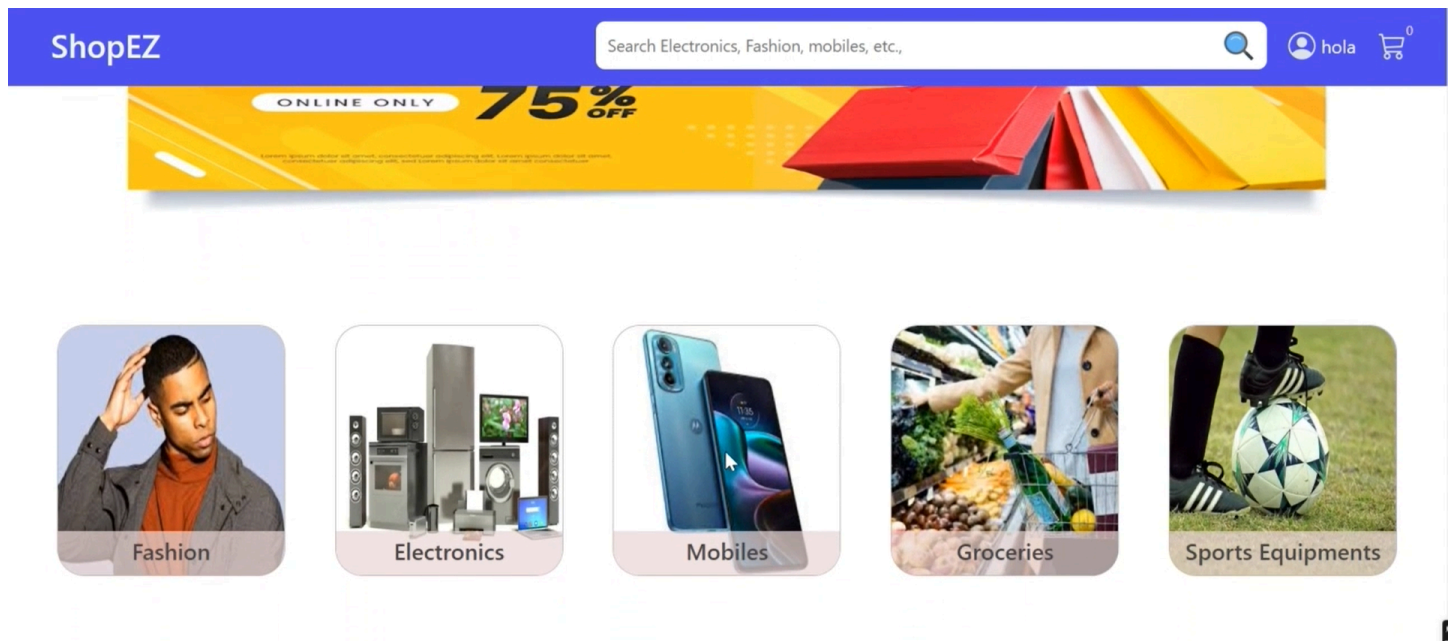
- Provide screenshots or GIFs showcasing different UI features.
- Responsive Design
- Intuitive Navigation
- Product Display
- Product Filtering and Sorting
- Product Detail Page
- Shopping Cart Interface
- Checkout Process

10. TESTING:

- **Exploratory Testing**
 - Conduct unscripted testing sessions to discover unexpected issues and assess user experience.
- **User Acceptance Testing (UAT)**
 - Engage end-users to validate that the application meets their expectations and requirements before launch.
- **Cross-Browser Compatibility Testing**
 - Verify that the site functions correctly across different web browsers (e.g., Chrome, Firefox, Safari) to ensure a consistent user experience.
- **Device and Responsive Testing**
 - Test the application on various devices (desktops, tablets, smartphones) to ensure it is fully responsive and functional.
- **Performance Testing**
 - Assess the application's performance under various conditions, including load times and responsiveness during peak usage scenarios.

11. SCREENSHOTS OR DEMO

Home Page:



User Login Page:

Login

Email address
hola@gmail.com

Password
I

Sign in

Not registered? Register

Admin login page:

Login

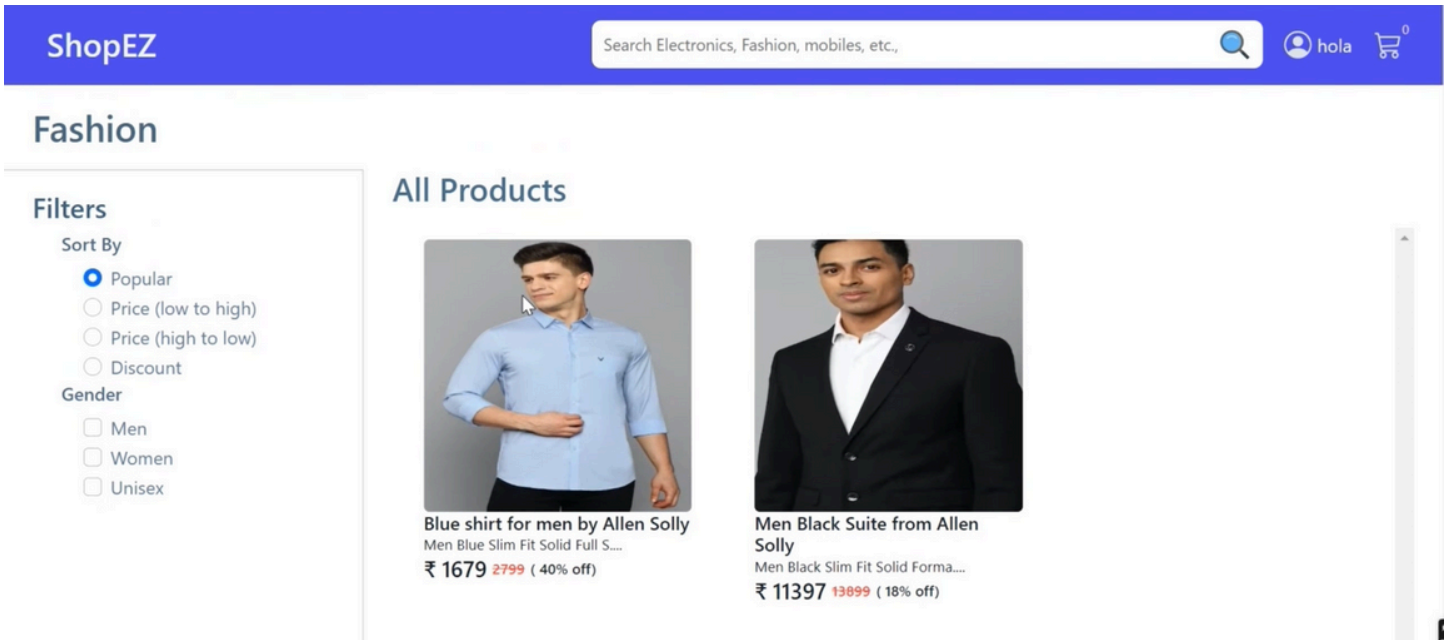
Email address
admin@gmail.com

Password
I

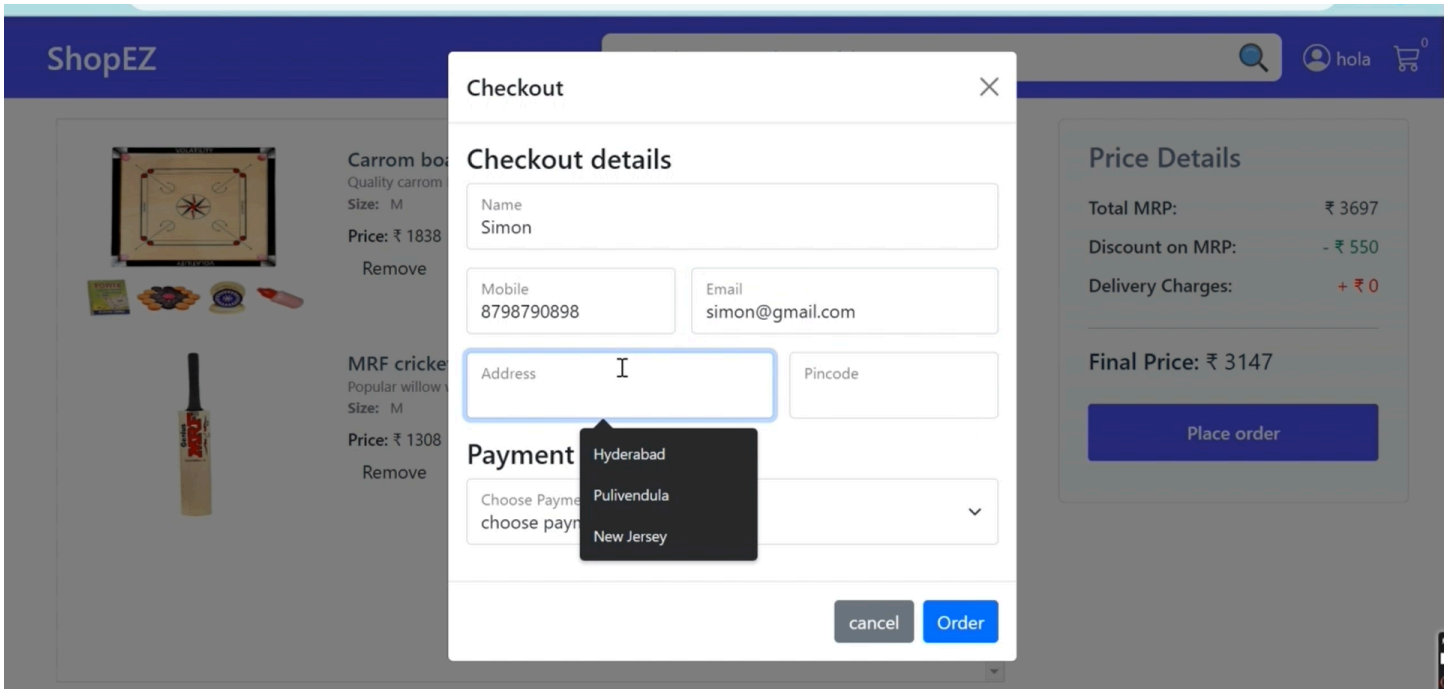
Sign in

Not registered? Register

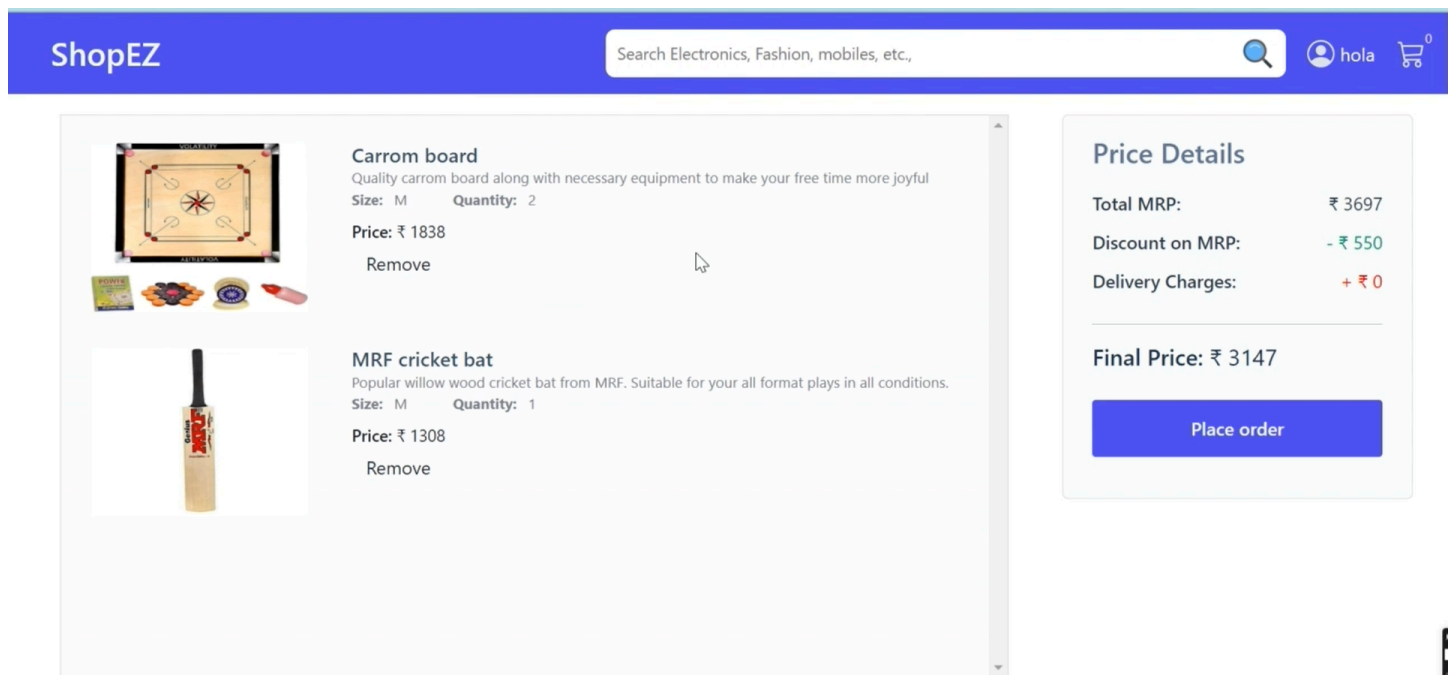
Product Page:



Contact Page :



Cart Page:



12. KNOWN ISSUES

- In the context of software development, particularly for applications built using the MERN stack, the inability to explore the application through automated testing can present significant challenge for us.

13. FUTURE ENHANCEMENTS

“ Develop voice-activated shopping capabilities through virtual assistants (e.g., Amazon Alexa, Google Assistant) to facilitate hands-free browsing and purchasing.”

GITHUB LINK:

<https://github.com/Anusha4522/naan-mudalvan-teamTMID18249>

DEMO LINK:

https://drive.google.com/file/d/1OEB5g_zb2-DjdNm6dfd_hxD_U0HWFtdw/view?usp=drivesdk

