

FULL STACK DEVELOPMENT WITH MERN

FRONTEND DEVELOPMENT REPORT

Date	17.07.2024
Team ID	SWTID1720001202
Project Name	Shop-EZ (E-commerce Website)
Maximum Marks	

Objective

The objective of this report is to document the frontend development progress and key aspects of the user interface implementation for the Shop-EZ project.

Technologies Used

- **Frontend Framework:** React.js
- **State Management:** Context API is used for managing state across the application.
- **UI Framework/Libraries:** React is used for the frontend, with Bootstrap for UI components and styling.

Project Structure:

Screenshot 1: Landing Page

The landing page of the Shop-EZ website features a navigation bar for quick access, making it convenient for customers to explore various sections.

Screenshot 2: Popular Deals and Footer

This section displays popular deals and includes the footer at the end, which is present on every page of the website.

Screenshot 3: Electronics Category Page

This page showcases the electronics category, highlighting various products available for purchase.

Screenshot 4: Product Display

This page displays individual products. When a product is clicked, the product's detailed page will be shown.

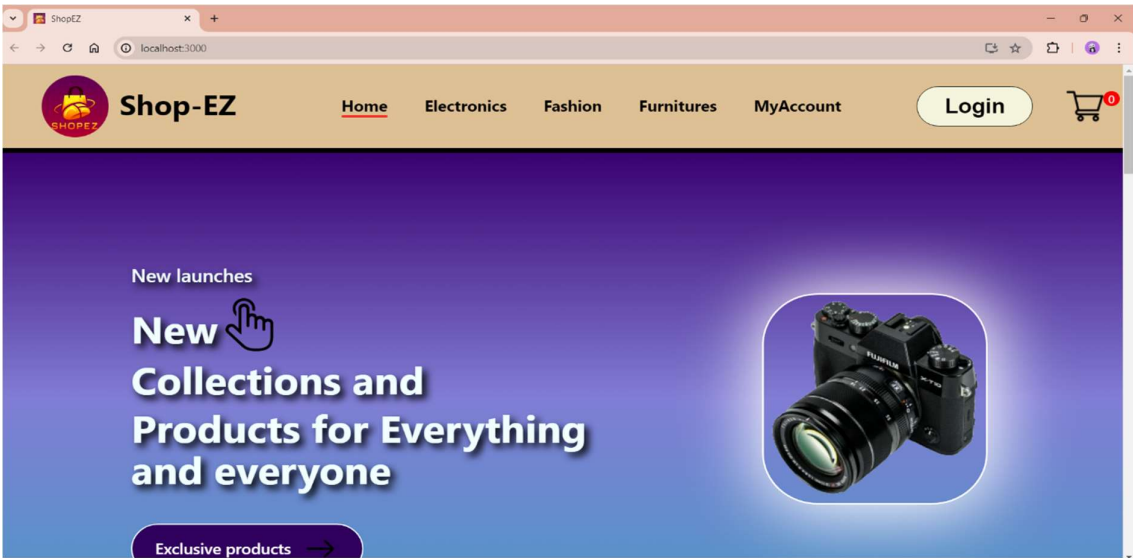
Screenshot 5: Cart Page

The cart page allows users to view and manage the items they have added to their shopping cart.

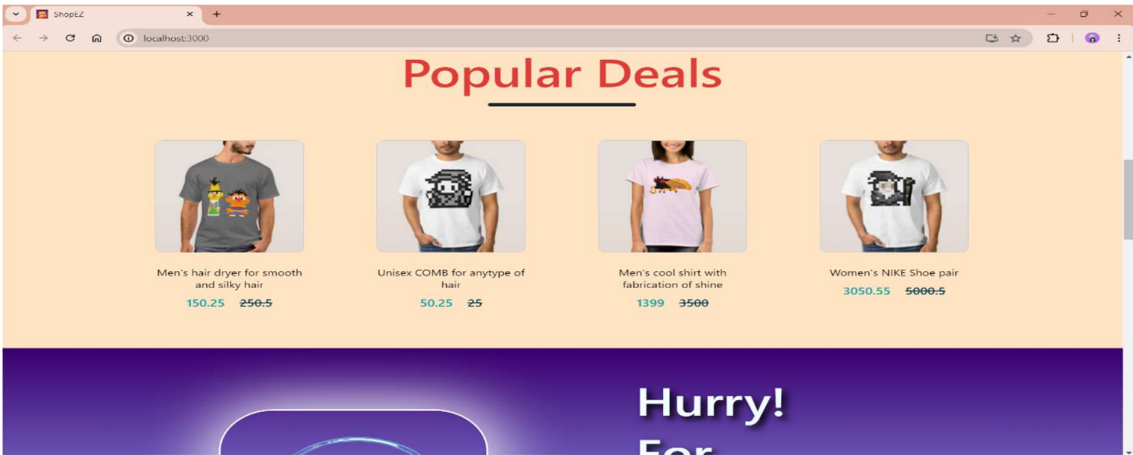
Screenshot 6: Login Page

This page facilitates user authentication, allowing customers to log in to their accounts.

SCREENSHOT:1



SCREENSHOT:2



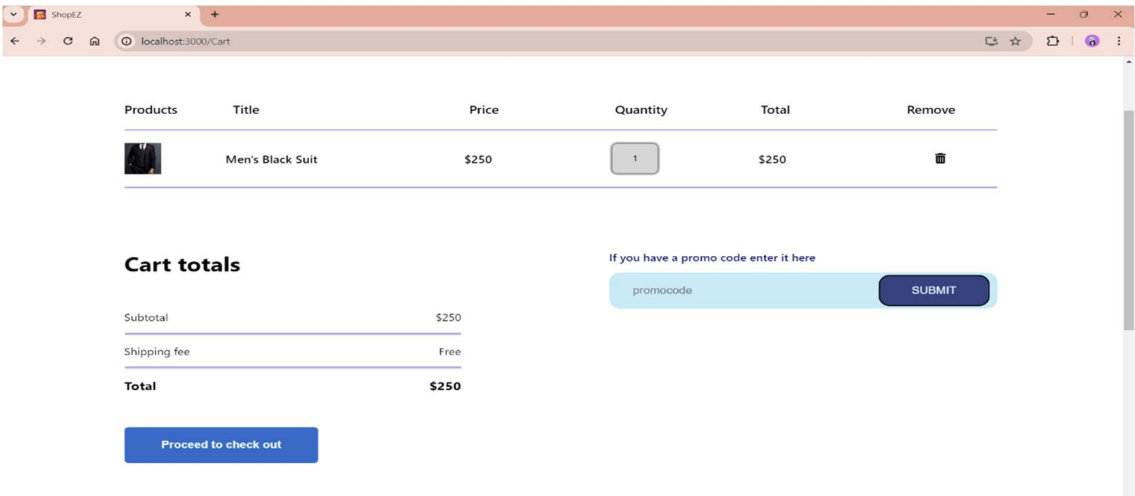
SCREENSHOT:3



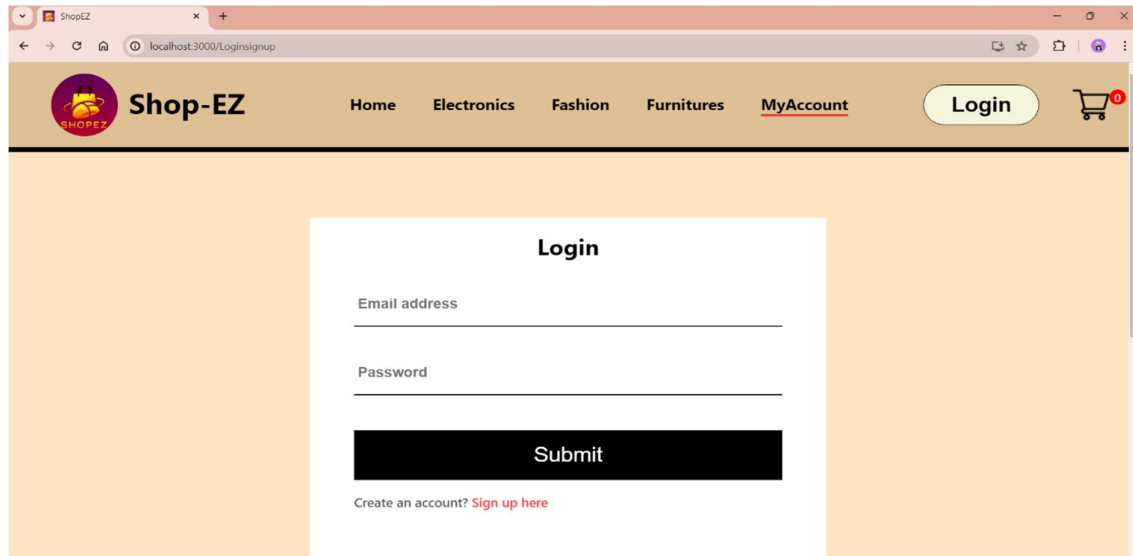
SCREENSHOT:4



SCREENSHOT:5



SCREENSHOT:6



Key Components

1. **App.js**
 - Responsible for routing and main application layout.
2. **/components**
 - Contains reusable UI components used across the application.
 - In the components folder there are many folders like assets, breadcrumbs, cart items, description box, home comp, product display, and related products.
3. **/pages**
 - Includes different pages for the E-commerce website.

Routing

Routing is managed using React Router. Here are the main routes:

- /home - Landing page of the application.
- /Electronics - Shows the Electronics category page.
- /Fashion/Mens - Shows the Men's Fashion category page.
- /Fashion/Womens - Shows the Women's Fashion category page.
- /Fashion/Kids - Shows the Kids' Fashion category page.
- /Homeandfurnitures - Shows the Home and Furniture category page.
- /product/:productId - Shows the specific product page.
- /Cart - Shows the Cart page.
- /Loginsignup - Shows the Login and Signup page.

State Management

Context API is used for state management across the application, with the `useState`, `useContext`, `createContext`, and `useRef` hooks.

- **useState:** Creates state variables within functional components.
- **createContext:** Creates a Context object to hold the state that can be accessed by any component in the component tree.
- **useContext:** Accesses the context value within functional components, enabling subscription to context changes.
- **useRef:** Creates a mutable object which persists across re-renders and can hold a reference to a DOM element or any other value that should persist.

Integration with Backend

The frontend communicates with the backend APIs hosted on `http://localhost:18000`. Key endpoints include:

- **GET /user** - Retrieves all users.
- **POST /user** - Creates a new user.
- **PUT /user/** - Updates an existing user by ID.
- **DELETE /user/** - Deletes an existing user by ID.
- **GET /user/-** Retrieves a user by ID.

User Interface (UI) Design

- **The UI design follows a clean and minimalist design principle**, focusing on simplicity and ease of use. The interface aims to be intuitive, reducing clutter and ensuring that users can easily find what they need.
- **Implemented using Bootstrap**, which offers a wide range of pre-styled components and utilities to speed up the development process. This framework helps create a consistent look and feel across the application, though the current layout does not fully adapt to different screen sizes, indicating areas for improvement in responsive design.

Third-Party Integrations (If any)

Backend:

- **Express:** Web framework.
- **Mongoose:** MongoDB object modeling.
- **jsonwebtoken:** JWT authentication.
- **cors:** Cross-Origin Resource Sharing.
- **multer:** File upload handling.

Frontend:

- **React:** User interface library.
- **react-dom:** DOM rendering for React.
- **react-router-dom:** Routing for React applications.
- **bootstrap:** UI components and styling.
- **@testing-library:** Testing utilities.
- **web-vitals:** Web performance metrics.