**Full Stack Development with MERN**

**Frontend Development Report**

|  |  |
| --- | --- |
| Date | 18 July 2024 |
| Team ID | SWTID1720195916 |
| Project Name | Stock Trading Web App |
| Maximum Marks |  |

**Project Title:** Stock Trading Web App

Date: 18th July 2024

Prepared by: Anish R A, Mukundan S, Adithya Hosur, Kirthick Easwar

**Objective**

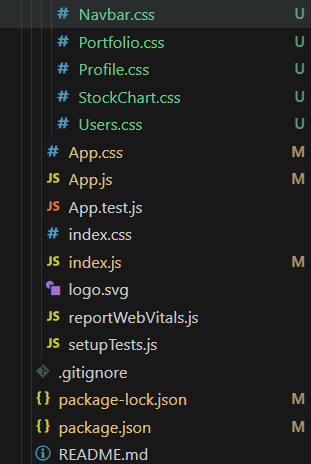
The objective of this report is to document the frontend development progress and key aspects of the user interface implementation for the Stock Trading Web App project.

**Technologies Used**

* **Frontend Framework:** React.js
* **State Management:** React Hooks, React Context API
* **UI Framework/Libraries:** Bootstrap
* **API Libraries:** Axios, JSONWebToken, CORS, RapidAPI – Mboum Finance, RapidAPI – TwelveData, bcrypt

**Project Structure**

Give Screenshot of Frontend Project with explanation

**** ****

**Key Components**

1. **App.js**
   * Responsible for routing and main application layout.
2. **/components**
   * Contains reusable UI components used across the application.
   * Login, SignUp, and Navbar page
3. **/pages**
   * Includes different pages for Web App.
4. /context
   * Manages global state and shared logic using React Context API.
5. /RouteProtectors

* Implements access control for routes, ensuring proper user authentication and authorization.

1. /Styles

* Contains styling components throughout the page

1. /assets

* contains static assets such as images, fonts, and other media files

**Routing**

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

* **/home** - Landing page of the application.
* **/portfolio** - Dashboard displaying user data and statistics.
* **/profile** - User profile management.
* **/history**- User transaction history with status and details of each transaction
* **/stock/:id-** Contains information about a specific stock mentioned with ‘id’
* **/admin-** Admin dashboard, the main control panel of administrators
* **/users-** Allows administrators to manage users, including viewing user details
* **/all-orders-** provides administrators with access to all buy and sell orders across the platform
* **/all-transactions-** allows administrators to view and manage all financial transactions
* **/admin-stock/:id-** provides detailed information about a specific stock, but tailored for administrative purposes

**State Management (If Applicable)**

State management is achieved using [Redux/Context API].

**Integration with Backend**

The frontend communicates with the backend APIs hosted on [backend URL]. Key endpoints include:

* **GET /api/data** - Retrieves data for display.
* **POST /api/user/login** - Handles user authentication.
* **POST /api/user/register** – Handles user registration
* **POST /api/deposit** – Displays the updated balance
* **GET /api/users** – Retrieves users for display.
* **GET /api/transactions** – Retrieves user transaction history
* **POST /api/buyStock** – Handles Stock being bought
* **POST /api/sellStock** – Handles Stocks being sold.

**User Interface (UI) Design**

* The UI design follows a minimalistic dark mode approach.
* Implemented using bootstrap and css

**Third-Party Integrations (If any)**

* React Apex Charts.
* React Icons