**Full Stack Development with MERN**

**Frontend Development Report**

| Date | 18/07/2024 |
| --- | --- |
| Team ID | SWTID1719999205 |
| Project Name | Darshan Ease |
| Maximum Marks |  |

**Project Title: [**Darshan Ease**]**

Date: [18/07/2024]

Prepared by: [Sashidhar Reddy, Adhvik Sai, Bharath A, Sajja Yuva Sai Rithvik]

**Objective**

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

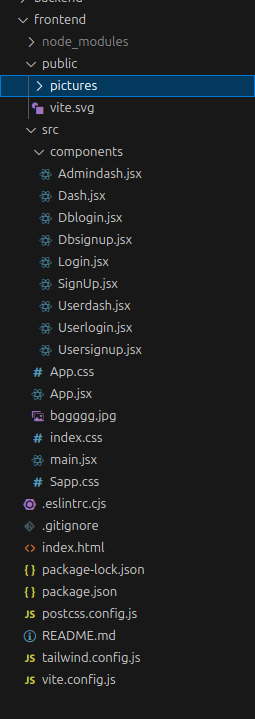
**Technologies Used**

* **Frontend Framework:** React.js
* **UI Framework/Libraries:** [Bootstrap,Tailwind CSS]
* **API Libraries:** [Axios]

**Project Structure**

The frontend folder consists of public>pictures(from where we imported all the images)

And the components folder consists of all the components of the page and the src folder consists of the app.js and the vite.config.js which is created while installing vite to improve the performance of the page.

****

**Key Components**

**Key Components**

1. **App.js** 
   * **Responsible for routing and main application layout.**
   * **Uses React Router for navigation.**
   * **Defines routes for different pages/components.**
2. **/components** 
   * **Contains reusable UI components used across the application.**
   * **Includes:** 
     + **Dblogin**
     + **Dbsignup**
     + **Login**
     + **Signup**
     + **Userlogin**
     + **Usersignup**
     + **Dash (Database dashboard)**
     + **Userdash (User dashboard)**
3. **Routes** 
   * **/ - Landing page (Dblogin)**
   * **/db-login - Database login**
   * **/db-signup - Database signup**
   * **/login - General login**
   * **/signup - General signup**
   * **/user-login - User-specific login**
   * **/user-signup - User-specific signup**
   * **/user-dashboard - User dashboard**
   * **/db-dash - Database dashboard**

**Routing**

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

* **/** - Landing page, renders the Dblogin component
* **/db-login** - Database login page, renders the Dblogin component
* **/db-signup** - Database signup page, renders the Dbsignup component
* **/login** - General login page, renders the Login component
* **/signup** - General signup page, renders the Signup component
* **/user-login** - User-specific login page, renders the Userlogin component
* **/user-signup** - User-specific signup page, renders the Usersignup component
* **/user-dashboard** - User dashboard, renders the Userdash component
* **/db-dash** - Database dashboard, renders the Dash component

**State Management (If Applicable)**

State management is likely achieved using React's built-in hooks, particularly useState and useContext.

Local component state is used for managing UI-specific states, while global application state (such as user authentication status) is likely managed through context or a custom hook.

**Integration with Backend**

The frontend communicates with the backend APIs hosted on [http://localhost:5500]. Key endpoints include:

* **GET /api/data** - Retrieves data for display.
* **POST /api/user/login** - Handles user authentication.
* **PUT /api/user/{id}** - Updates user information for the specified user ID.
* **DELETE /api/post/{id}** - Deletes the post with the specified ID.

**User Interface (UI) Design**

* The UI design follows a component-based architecture, focusing on reusability and modularity.
* Implemented using React.js, a popular JavaScript library for building user interfaces.
* The design includes separate components for various user interactions such as login, signup, and dashboard views for both database administrators and regular users.
* Utilizes React Router for seamless navigation between different pages without full page reloads.
* The UI likely follows a responsive design principle to ensure compatibility across different devices and screen sizes.