**Full Stack Development with MERN**

**Frontend Development Report**

|  |  |
| --- | --- |
| Date | 15-07-2024 |
| Team ID | SWTID1720010107 |
| Project Name | HireConnect |
| Maximum Marks | 10 |

**Project Title: Hire Connect**

Date: 15-07-2024

Prepared by: Hire Connect

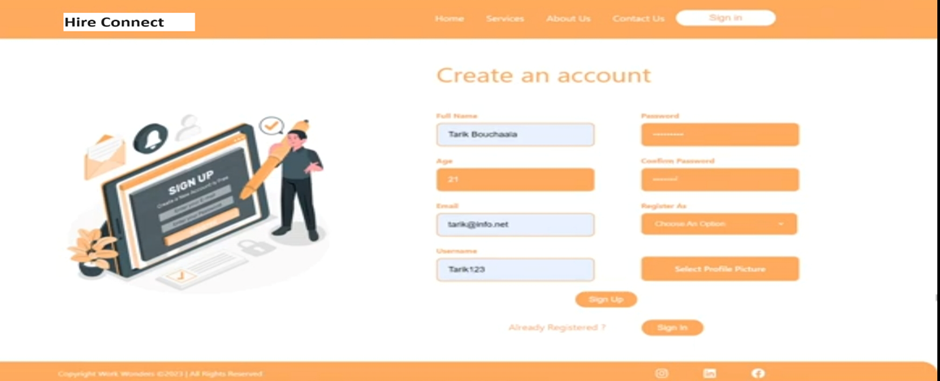
**Objective**

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

**Technologies Used**

* **Frontend Framework:** React.js
* **State Management:** Redux.js
* **UI Framework/Libraries:**Bootstrap, React Libraries
* **API Libraries:** Axios
* **Frontend Packages** : Moment, React Icons, toolkit, react router/HashLink, Spinners, Toastify, Socket.io Client, Swiper

**Project Structure**

****

**Goal :**We have developed the website just like user will register for their freelancing work

**Frontend Contains all frontend-related files and folders.**

**public/:** Contains static files like images and other assets**.**

**src/:** Main source directory for the React application**.**

**Components:** Reusable React components like Filter Component, Freelancer Card, Navbar,

 and Footer.

**Pages**: React components representing different pages, such as Homepage,

             FreelancerListPage, FreelancerDetailPage, and Search Page**.**

**Context:** Context API for state management, such as FreelancerContext**.js.**

**Services:** Contains services for API calls, like freelancerService.js.

**Styles**: CSS modules for styling specific components.

**App.js:** Main application component**.**

**index.js:** Entry point for the React application.

**.env:** Environment variables for the frontend.

**.gitignore:** Specifies which files and directories to ignore in version control.

**package.json:** Lists frontend dependencies and scripts.

**README.md:** Contains information about the frontend project.

**yarn.lock:** Keeps track of the exact versions of dependencies**.**

**Key Components**

1. **App.js**
   * Responsible for routing and main application layout.
2. **/components**
   * Contains reusable UI components used across the application.
3. **/pages**
   * Includes different pages for Web App.

**Routing**

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

* **/home** - Landing page of the application.
* **/dashboard** - Dashboard displaying user data and statistics.
* **/profile** - User profile management.

**State Management (If Applicable)**

State management is achieved using Redux.js

**Frontend (React)**

* **src**
  + **index.js** (or **main.js**)
    - Set up the Redux Provider to wrap the app.
  + **App.js**
    - Main application component.
  + **store**
    - **index.js**
      * Configure and export the Redux store.
  + **features**
    - **freelancers**
      * **freelancerSlice.js**
        + Define the Redux slice for freelancer data.
  + **components**
    - **FreelancerList.js**
      * Component to display the list of freelancers using Redux state.

**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.
* **Backend Structure:**
* **API Directory (api/)**:
* **Express.js**: Sets up the server and middleware.
* **Routes**: Handles API endpoints (e.g., freelancers, clients).
* **Models**: Defines Mongoose schemas for MongoDB (e.g., Freelancer.js).
* **Frontend Integration (client/):**
* **Redux Store**:
* Configures state management (src/store/).
* Uses createSlice for actions and reducers (src/features/).
* **Components**:
* Fetches and displays data using Redux (useSelector, useDispatch).
* **Real-time Communication (socket/)**:
* **Socket.io**:
* Manages real-time events for features like chat.

**User Interface (UI) Design**

**Styling Libraries**:

* React Icons for consistent iconography.
* React Spinners for loading indicators.
* React Toastify for notifications and feedback messages.
* Swiper for interactive carousels and sliders.

**Socket.io Client**:

* Real-time communication for chat and instant notifications.