# React.js Frontend Developer Internship Task Guide

## 1. Introduction

Welcome to the React.js Frontend Developer Internship Task Guide. This guide aims to provide you with clear and detailed instructions for completing the assignment, which involves creating an Admin Dashboard for a social media application using React.js or Next.js. The primary objective of this task is to evaluate your skills in developing user interfaces, handling data, and implementing state management in a React environment.

### Assignment Objectives

- Develop an Admin Dashboard for a social media application.

- Implement navigation, KPIs, and data listings for users and posts.

- Use React or Next.js frameworks.

- Ensure the application is user-friendly and follows best practices in React development.

## 2. Step-by-Step Instructions

### Step 1: Setting Up Your Development Environment

1. **Install Node.js and npm**: Ensure you have Node.js and npm installed. These are necessary for managing packages and dependencies.

- [Download Node.js and npm](https://nodejs.org/)

2. **Create a New React or Next.js Project**:

- **React**: Run `npx create-react-app admin-dashboard`

- **Next.js**: Run `npx create-next-app admin-dashboard`

3. **Navigate to Project Directory**:

- Use the command `cd admin-dashboard`

### Step 2: Project Structure and Setup

1. **Create Components**: Set up folders and files for your components. Suggested structure:

- `/src/components/Navbar`

- `/src/components/Home`

- `/src/components/UserListing`

- `/src/components/PostListing`

2. **Install Required Libraries**:

- For navigation and UI components you may use libraries such as `react-router-dom`, `material-ui`, or `bootstrap`.

- Install using `npm install` or `yarn add`.

### Step 3: Creating the Login Page

1. **Create a Login Page**:

- Add a simple form with email and password fields.

- Allow login with any credentials by setting a dummy authentication.

- On successful login, redirect to the admin dashboard home page.

### Step 4: Implementing the Admin Dashboard Layout

1. **Create the Navigation Bar**:

- Add a sidebar with links to Home, User Listing, and Post Listing pages.

2. **Develop the Home Page**:

- Display 4 Key Performance Indicators (KPIs): Total Users, Total Posts, Users Active in the Last 24 Hours, and Posts Published in the Last 24 Hours.

- Each KPI should be presented in a large rectangular box.

### Step 5: User Listing Page

1. **Create User Listing Page**:

- At the top, display the total users and users active in the last 24 hours KPIs.

- Below, create a paginated list view of users with columns: `User\_id`, `username`, `name`, and `email`.

2. **Add Control Buttons**:

- Each user row should have "ban" and "edit" buttons for admin actions.

### Step 6: Post Listing Page

1. **Create Post Listing Page**:

- Structure similar to User Listing Page.

- Columns should include `post\_id`, `post\_caption`, and `media\_url`.

2. **Add Control Buttons**:

- Each post row should have "delete" and "hide" buttons.

### Step 7: Dummy Data Integration

1. **Generate Dummy Data**:

- Use libraries like `faker.js` to create a dataset for users and posts.

- Store this data in state (e.g., React useState or Context API).

### Step 8: Finalizing and Testing

1. **Test the Application**: Ensure all pages and functionalities work as expected.

2. **Add Comments and Documentation**: Add comments in your code explaining the logic. Document any significant design choices.

## 3. Best Practices

- **Code Organisation**: Keep code modular and organized. Use meaningful component names.

- **User Experience**: Ensure the UI is clean, responsive, and intuitive.

- **Performance**: Optimize listing pages for performance (e.g., pagination).

- **Error Handling**: Gracefully handle any potential errors.

## 4. Submission Guidelines

1. **Code Submission**:

- Share the complete code (excluding the `node\_modules` folder).

- Include a `README` file with setup and testing instructions.

2. **README File**:

- Instructions to run the application, including dependencies, installation, and usage.

3. **Video Recording**:

- Provide a 1-2 minute video demonstrating the working of the dashboard.

- Include some screenshots of the dashboard.

4. **ZIP File**:

- Package all the deliverables (code, `README` file, video recording) into a ZIP file.

- Upload the ZIP file to Google Drive.

- Share the public access link of the ZIP file.

## 5. Frequently Asked Questions (FAQ)

**Q1: What frameworks can be used for this assignment?**

A: You can use either React.js or Next.js.

**Q2: How should I handle form submissions in the login page?**

A: You can use dummy authentication, where any set of credentials will allow access to the admin dashboard.

**Q3: What is the expected design of the navigation bar?**

A: The navigation bar should be a sidebar on the left side of all pages, allowing navigation between different views (Home, User Listing, Post Listing).

**Q4: How to handle user and post data?**

A: Use dummy data generated by libraries like `faker.js`. Store and manage this data using React state or Context API.

**Q5: What if I encounter a problem or have questions?**

A: Ensure to follow best practices, refer to official React or Next.js documentation, and seek assistance from available resources or communities.

Thank you for participating in this assignment. Best of luck!