# Comprehensive Reference Document for ReactJS Frontend Developer Internship Assignment

## 1. Introduction

Welcome to the ReactJS Frontend Developer Internship assignment! This document provides a detailed guide to help you successfully complete the assignment. The purpose of this task is to create an admin dashboard for a social media application using ReactJS or NextJS. By completing this assignment, you will demonstrate your ability to build functional and visually appealing web applications.

### Objectives:

- Build an admin dashboard with navigation and KPI displays.

- List users and posts with actionable controls.

- Implement a dummy login page for authentication.

- Ensure the application is well-documented and follows best practices.

## 2. Step-by-Step Instructions

### Step 1: Set Up the Development Environment

1. **Install Node.js and npm**: Make sure you have Node.js and npm installed. You can download and install them from [Node.js official website](https://nodejs.org/).

2. **Create a New Project**: Initialize a new React project using Create React App or Next.js.

```bash

npx create-react-app admin-dashboard

cd admin-dashboard

```

or for Next.js:

```bash

npx create-next-app admin-dashboard

cd admin-dashboard

```

### Step 2: Implement the Dummy Login Page

1. **Create a Login Component**: Build a simple login form with email and password fields.

2. **Route the User Upon Login**: Implement a navigation redirection upon successful login to the admin dashboard.

### Step 3: Design the Layout

1. **Navigation Bar**: Create a sidebar navigation with links to the home page, users listing page, and post listing page.

2. **Home Page**: Display four KPIs: Total Users, Total Posts, Users active in the last 24 hours, and Posts published in the last 24 hours.

### Step 4: Develop the Users and Posts Listing Pages

1. **Users Listing Page**:

- KPIs for total users and users active in the last 24 hours.

- List of users with columns: User\_id, username, name, email.

- Control buttons for each user: 'ban' and 'edit'.

2. **Posts Listing Page**:

- KPIs for total posts and posts published in the last 24 hours.

- List of posts with columns: post\_id, post caption, media url.

- Control buttons for each post: 'delete' and 'hide'.

### Step 5: Create Dummy Data

1. **Generate Data**: Create mock datasets for users and posts.

2. **Display Data**: Use the dummy data to populate the tables and KPIs.

### Step 6: Style the Application

1. **Use CSS/Styling Libraries**: Apply styles to make the dashboard visually appealing.

2. **Responsive Design**: Ensure the application is usable on different screen sizes.

### Step 7: Add Final Touches

1. **Documentation and Comments**: Add comments and documentation within your code to explain your logic.

2. **README File**: Draft a README file with setup and usage instructions.

## 3. Best Practices

- **Code Quality**: Write clean, readable, and maintainable code. Follow naming conventions and organize your components logically.

- **Comments and Documentation**: Provide meaningful comments and documentation within the code to explain the purpose and functionality.

- **Responsive Design**: Ensure your application is responsive and works well on both desktop and mobile devices.

- **Testing**: Test your application thoroughly to identify and fix any bugs.

- **Version Control**: Use Git for version control to keep track of your changes and collaborate with others.

## 4. Submission Guidelines

1. **Package the Deliverables**:

- **Code**: Ensure to include all the files except the `node\_modules` folder.

- **README File**: Provide complete instructions for setting up and testing the application.

- **Video Recording**: Record a 1-2 minute video demonstrating the working dashboard.

- **Screenshots**: Include screenshots of the application.

2. **Create a ZIP File**:

- Combine the code, README file, video recording, and screenshots into a single ZIP file.

- Upload the ZIP file to Google Drive.

3. **Share the Link**:

- Set the Google Drive link to public access.

- Share the link as part of your submission.

## 5. Frequently Asked Questions (FAQ)

### Q1: How detailed should the dummy data be?

- **A1**: The dummy data should be sufficiently detailed to demonstrate the functionality of the dashboard, including realistic values for user and post attributes.

### Q2: What if I encounter an error during development?

- **A2**: Refer to the error messages and documentation. Utilize online resources like Stack Overflow, GitHub Issues, and official React/NextJS documentation.

### Q3: Can I use additional libraries for styling and functionality?

- **A3**: Yes, you are free to use additional libraries like Material-UI, Bootstrap, or any other CSS/JS libraries to enhance the application.

### Q4: How should the video demonstration be structured?

- **A4**: The video should briefly cover the login process, navigation, KPI displays, user and post listings, and the functionality of control buttons.

### Q5: How can I ensure my application follows best practices?

- **A5**: Follow ReactJS best practices, code conventions, and principles of software design. Ensure your code is modular, reusable, and well-documented.

By following these guidelines and instructions, you will create a comprehensive and professional admin dashboard for a social media application. Good luck with your assignment!