# Reference Document for ReactJS Frontend Developer Internship Assignment

## Table of Contents

1. **Introduction**

2. **Step-by-Step Instructions**

3. **Best Practices**

4. **Submission Guidelines**

5. **Frequently Asked Questions (FAQ)**

---

## 1. Introduction

### Overview

This document is designed to assist you in completing the ReactJS frontend developer internship assignment. The assignment aims to evaluate your ability to create a functional and visually appealing admin dashboard for a social media application using ReactJS/NextJS.

### Objectives

- Develop a user-friendly admin dashboard.

- Display key metrics and data in an organized manner.

- Implement basic user and post management features.

- Provide a seamless user experience.

---

## 2. Step-by-Step Instructions

### 2.1. Setting up the Project

1. **Create a New React/NextJS Project:**

- Use `create-react-app` for ReactJS or `create-next-app` for NextJS to set up a new project.

- Make sure Node.js and npm are installed on your machine.

2. **Install Necessary Dependencies:**

- ReactJS (if using NextJS, this is included)

- React Router (if using ReactJS)

- Material-UI or another component library for UI elements

- Axios for data fetching (optional)

### 2.2. Building the Admin Dashboard

3. **Set Up the Project Structure:**

- Create folders for `components`, `pages`, `services`, `assets`, etc.

- Ensure a clean directory structure that separates concerns.

4. **Create a Dummy Dataset:**

- Generate sample data for users and posts.

- Store this data locally in a JSON file or a JavaScript object.

5. **Implement the Login Page:**

- Create a simple login form with email and password fields.

- Implement a fake authentication mechanism that grants access with any credentials.

6. **Set Up Navigation:**

- Create a sidebar or top navigation bar for easy access to different pages of the dashboard.

### 2.3. Building Specific Dashboard Pages

7. **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.

- Use large rectangular boxes for each KPI.

8. **Users Listing Page:**

- Display two KPI boxes for Total Users and Users Active in the Last 24 Hours.

- Create a table listing each user's details: User ID, Username, Name, Email.

- Add pagination for better user experience.

- Include control buttons for each user to ban or edit details.

9. **Posts Listing Page:**

- Mirror the structure of the Users Listing Page, focusing on posts.

- Display data columns: Post ID, Post Caption, Media URL.

- Include control buttons for each post to delete or hide the post.

### 2.4. Final Touches

10. **Add Interaction:**

- Ensure all control buttons have functionality or placeholder actions.

11. **Testing and Debugging:**

- Test the application thoroughly to ensure all features work as expected.

- Fix any bugs or UI issues.

---

## 3. Best Practices

1. **Code Quality:**

- Use clean and modular code.

- Comment your code to explain functionality.

2. **UI/UX:**

- Ensure the dashboard is responsive and user-friendly.

- Follow consistent styling and layout conventions.

3. **Performance:**

- Optimize rendering and data fetching if applicable.

- Minimize unnecessary re-renders.

4. **Documentation:**

- Write comprehensive and clear documentation.

- Make sure the README file includes all necessary setup instructions.

---

## 4. Submission Guidelines

1. **Prepare Deliverables:**

- Code: Ensure the complete codebase is organized.

- README File: Include setup and testing instructions.

- Video Recording: Create a 1-2 minute video demonstrating the dashboard.

- Screenshots: Take clear screenshots of the dashboard's key pages.

2. **ZIP File:**

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

- Upload the ZIP file to Google Drive or a similar service.

- Share the public access link.

3. **Submission:**

- Submit the Google Drive link with the public access setting enabled to the provided submission portal or email address.

---

## 5. Frequently Asked Questions (FAQ)

### Q1. What frameworks or libraries can I use?

You can use ReactJS or NextJS for the frontend, and any component library like Material-UI for UI elements.

### Q2. How should I manage the dummy data?

You can use a static JSON file or JavaScript object to hold the dummy data for users and posts.

### Q3. Can I use any styling method?

Yes, you can use CSS, CSS-in-JS libraries (like styled-components), or any styling method you are comfortable with.

### Q4. What if I encounter issues while developing?

Refer to online documentation, forums, or official repositories for troubleshooting. You can also reach out for help through the provided support channels if available.

### Q5. How important is the video recording?

The video recording is crucial as it demonstrates the functionality and visual aspects of your dashboard, providing a quick overview for reviewers.

---

By following this reference document, you will be able to successfully complete the ReactJS frontend developer internship assignment. Good luck!