# \*\*ReactJS Frontend Developer Internship Assignment Reference Document\*\*

## \*\*1. Introduction\*\*

### Overview

Welcome to the ReactJS Frontend Developer Internship assignment! This document is designed to help you understand the objectives, tasks, and steps necessary to successfully complete the assignment. By the end of this exercise, you will have created a responsive and functional admin dashboard for a social media application.

### Purpose and Goals

The primary goal of this assignment is to assess your skills in ReactJS/NextJS by developing an admin dashboard for a social media application. This includes implementing multiple pages with KPIs, user and post listings, and control buttons for administrative actions.

---

## \*\*2. Step-by-Step Instructions\*\*

### Prerequisites

- Basic knowledge of ReactJS or NextJS

- Familiarity with HTML, CSS, and JavaScript

- Node.js and npm installed on your computer

### Installation Setup

1. **Set up your development environment**

- Ensure Node.js and npm are installed.

- Create a new project if not already created using:

```

npx create-react-app admin-dashboard

cd admin-dashboard

```

### Step 1: Create Project Structure

2. **Initialize the project using ReactJS or NextJS**

- If using NextJS:

```

npx create-next-app admin-dashboard

cd admin-dashboard

```

### Step 2: Design and Develop the Admin Dashboard

3. **Create Navigation Bar Component**

- Develop a persistent navigation bar on the left side across all pages.

4. **Home Page**

- Render 4 KPIs in large rectangular boxes for Total Users, Total Posts, Users active in the last 24 hours, and Posts published in the last 24 hours.

5. **Users Listing Page**

- Include a KPI section with Total Users and Users active in the last 24 hours.

- Implement a table showing user\_id, username, name, and email with control buttons (ban, edit).

6. **Posts Listing Page**

- Similar to the user page, create a table listing post\_id, post caption, and media URL with control buttons (delete, hide).

7. **Dummy Dataset**

- Generate a dummy dataset for users and posts.

8. **Login Page**

- Create a simple login page with email and password fields, allowing any credentials to navigate to the admin dashboard.

### Step 3: Best Practices

9. **Code Quality**

- Ensure code is modular, reusable, and well-documented.

- Use consistent naming conventions and keep components small and single-purpose.

10. **Styling**

- Apply CSS or CSS-in-JS frameworks (like styled-components) for better scalability and maintainability.

11. **Testing**

- Include basic tests for components using Jest or similar testing library.

### Step 4: Prepare Deliverables

12. **README File**

- Include clear instructions to set up and run the application, describe dependencies, and include usage instructions.

13. **Video Recording**

- Record a 1-2 minute video demo showcasing the functionality of the admin dashboard.

14. **Screenshots**

- Capture screenshots of different sections of the admin dashboard.

### Step 5: Submission

15. **Package Deliverables**

- Package codebase, README file, video recording, and screenshots into a ZIP file.

- Upload the ZIP file to Google Drive and share the public access link.

---

## \*\*3. Best Practices\*\*

### Code Quality

- Follow DRY (Don’t Repeat Yourself) principles.

- Use ES6+ features for cleaner and more efficient code.

- Regularly commit changes with clear, descriptive commit messages.

### Development Tips

- Break down tasks into smaller, manageable chunks.

- Use version control (Git) to track progress.

- Regularly test components during development.

### UI/UX Design

- Ensure the dashboard is responsive and accessible.

- Use consistent color schemes and typography.

---

## \*\*4. Submission Guidelines\*\*

### Required Formats and Materials

1. **Complete Codebase**: Exclude the `node\_modules` folder.

2. **README File**: Detailed setup, installation, and usage instructions.

3. **Video Recording**: Short demo of the working dashboard.

4. **Screenshots**: Clear images of the dashboard UI.

5. **ZIP File**: Package all deliverables, upload to Google Drive, and share the link.

### Submission Process

- Assemble all deliverables.

- Create a ZIP file containing everything.

- Upload the ZIP file to Google Drive.

- Share the public access link of the ZIP file.

---

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

### Q1: Can I use any libraries in addition to ReactJS/NextJS?

**A1:** Yes, you're free to use any UI libraries like Material-UI, Bootstrap, etc., to enhance your UI design.

### Q2: How detailed should the dummy dataset be?

**A2:** The dummy dataset should be sufficiently detailed to show the functionality of your dashboard, but it doesn’t need to be exhaustive.

### Q3: How do I handle the navigation between different pages?

**A3:** Use React Router for ReactJS or the built-in routing features in NextJS.

### Q4: What should I include in the README file?

**A4:** The README should include instructions on:

- Installing dependencies

- Running the application

- Overview of the application structure

- Any additional setup steps

### Q5: What if I encounter issues or have questions during the assignment?

**A5:** If you run into problems, try searching online forums, official documentation, or reach out to your internship coordinator for guidance.

---

### Conclusion

By following this reference document, you should be able to complete the ReactJS frontend developer internship assignment with a high level of quality and professionalism. Stay organized, write clean code, and make sure your deliverables are well-packaged for submission. Good luck!