Workflow Management System - Assignment Documentation

Overview

This document provides guidance for completing the Front-End Software Engineer assessment assignment. The assignment requires building a workflow management application with specific features outlined in user stories.

Assignment Goals

The primary objective is to assess your:

- Front-end development skills using React
- UI implementation abilities
- Data management approach
- Understanding of software engineering concepts

Requirements

Technology Stack

- **Frontend Framework**: React (required)
- **Data Storage**: Your choice (Firebase, custom backend, etc.)
- **Deployment**: Required (Hostinger, Vercel, Netlify, or similar platforms)

Feature Requirements

Authentication

- User login with email and password

Workflow List View

- Display list of workflows with status
- Search functionality by name or ID
- Status indicators (passed/failed)
- Single-click execution with warning modal
- Edit existing workflows
- Create new workflows

Workflow Creator

- Clear starting point for new workflows
- Define start and end points
- Add multiple step types (API calls, emails, etc.)
- Connect steps in logical sequence
- Delete workflow steps
- Save progress functionality
- Flowchart visualization
- Zoom in/out capability
- Canvas movement/panning

Example Workflow

A user should be able to create a workflow that includes an API component and an email component. When executed, the API should trigger, and the response should be emailed to the email added.

Resources

- **Figma Design File**: Design Link
- **Figma Prototype**: Prototype Link
    - Press "R" to reset the prototype if stuck
- **Mock API for Testing**: Beeceptor Sample API

Evaluation Criteria

Your submission will be evaluated on the following milestones:

1. Workflow Editor UI implementation
2. Complete UI implementation
3. UI implementation with functional data storage and retrieval

Submission Guidelines

- **Deadline**: Friday EOD
- **Submission Method**: Submit through the provided form link (do not reply via email)
- **Required Deliverable**: Deployed application link

Development Guidelines

Approach

1. Begin by studying the Figma designs carefully
2. Plan your component structure
3. Implement the UI components based on the designs
4. Add functionality and state management
5. Implement data persistence
6. Test thoroughly, focusing on user stories
7. Deploy the application

Best Practices

- Use component-based architecture
- Implement responsive design
- Follow React best practices
- Write clean, maintainable code
- Add appropriate error handling
- Consider accessibility

Important Notes

- This assignment is for assessment purposes only and will not be used for any business purposes
- The use of AI tools like Cursor, ChatGPT, etc. is expected and encouraged
- Understanding of implementation details is crucial for further rounds

Technical Implementation Tips

Authentication

- Consider using Firebase Authentication or a simple mock authentication system
- Store authentication state using context or a state management library

Workflow Management

- Use a state management solution (Context API, Redux, etc.)
- Consider using a library like react-flow for the workflow canvas functionality
- Implement drag-and-drop for better user experience

Data Persistence

- Firebase Firestore is a good option for quick implementation
- Consider implementing your own backend if comfortable

UI Implementation

- Leverage existing component libraries to speed up development
- Closely follow the Figma designs for accurate UI representation

Testing

- Test the application thoroughly against all user stories
- Ensure all features work on different screen sizes

Conclusion

This assignment evaluates your ability to interpret requirements, implement complex UI functionality, and manage data effectively. Focus on delivering a polished, functional application that meets all the specified user stories.