

Coding Test: File Storage Web Applications

Backend Task: Develop a RESTful API for a file storage application. The API should support the following functionalities:

1. Upload a file: Allow users to upload files. Store these files on the server's filesystem.
2. List uploaded files: Provide an endpoint to list all files that have been uploaded, including their names and upload dates.
3. Download a file: Allow users to download a file by clicking on the file name in the list of uploaded files.
4. Delete a file: Allow users to delete a file

Backend Requirements:

- Use Node.js.
- Ensure the application handles different file types (e.g., images, PDFs, text files).
- Implement basic error handling, especially for file size limits and unsupported file types.
- Include comments in the code to explain critical sections.
- Generate a Dockerfile for this backend code

Frontend Task: Develop a frontend for the file storage application that interacts with the backend API developed. The interface should allow users to:

1. Upload files: Provide a form for users to select and upload files to the server.
2. View a list of uploaded files: Display a list of files that have been uploaded. Include the file name and the upload date.
3. Download files: Allow users to download files by clicking on an item in the list of uploaded files.
4. Delete files: Allow users to delete files

Frontend Requirements:

- Use React for the frontend development.
- Handle state management appropriately to reflect the file upload process and the listing of uploaded files.
- Ensure the user interface is intuitive and user-friendly.
- Consider responsiveness and usability in the design.
- **Design fidelity is crucial for evaluation.**
- Generate a Dockerfile for this backend code

Design Test: Creating the React Code based on the screenshot

Task Description:

The candidate is required to create a react page that closely resembles the following screenshot. **Design fidelity is crucial for evaluation.**

