Full-Stack Development Test Scenario: Image Upload and Display

Objective:

Assess the candidate's skills in full-stack development, including backend API creation and frontend application development for image upload and display.

Technologies:

- Backend: Nodejs
- Frontend: Any programming language or framework

Backend Tasks:

1. Create a Backend API:

1.1 Image Upload Endpoint

- Develop a backend API endpoint allowing users to upload images.
- Save the uploaded images on the server or cloud storage.

1.2 List Images API

- Implement an API endpoint to fetch a list of uploaded images.
- The response should include necessary details such as file name, upload date, etc.

Frontend Tasks:

2. Create a Frontend Application:

- Set up a basic front-end application using any preferred programming language or framework.

3. Image Upload Component:

3.1 Basic Implementation

- Create a component in the frontend application allowing users to select and upload images.
- Display a preview of the selected image before uploading.

3.2 Filter Options Extension

- Extend the image upload component to include filter options such as grayscale and sepia.
- Allow users to apply these filters to the image before finalizing the upload.

4. List Images Component:

- Develop a component in the frontend application to fetch and display the list of uploaded images.
- Each image in the list should display its details like file name, upload date, etc.
- Include information about applied filters, if any.

5. Integration:

- Integrate the backend API with the frontend application to handle image uploads and fetching the list.

6. Styling:

- Apply basic styling to the application to make it user-friendly.
- Ensure the filter options are visually appealing and seamlessly integrated into the user interface.

Evaluation Criteria:

Backend:

- Successful implementation of the image upload API.
- Implementation of the API to fetch the list of uploaded images.

Frontend:

- Successful implementation of the extended image upload component with filter options.
- Proper integration of backend APIs for uploading and listing images.
- User-friendly interface and basic styling.

Submission Instructions:

Candidates should provide a link to their code repository (e.g., GitHub) containing both backend and frontend code.

Include clear instructions on how to run the application, emphasizing any new features related to the extended image upload component with filter options.