AVERROES

Technical Test Assignment

Deadline: Complete the assignment within **3 days** from the date of receipt.

Introduction

Welcome to our technical assessment! This assignment is designed to evaluate your skills in frontend development using **Next.js App Router**, **Material UI (MUI)**, and **React Query**. You will build a web application that manages images, categories, and annotations, providing functionalities like uploading, viewing, deleting images, and annotating images using drawing tools.

Objective

Build a frontend application that allows users to:

- **Upload**, **view**, and **delete** images.
- **Manage categories** (create, read, update, delete).
- **Filter images** by name and metadata.
- **Annotate images** using drawing tools (e.g., **React Konva**).
- Utilize UI elements like **forms** and **modals**.

Requirements

Technologies to Use

- **Framework:** Next.js (App Router)
- **UI Library:** Material UI (MUI)
- **Data Fetching & State Management:** React Query
- **Drawing Library:** React Konva (for image annotation)
- **Virtualization Library**: React Window (for image gallery)

API Endpoints

Use the following API endpoints for data operations:

Base URL: `https://my-json-server.typicode.com/MostafaKMilly/demo`

Categories

- **Get all categories**
  - `GET /categories`
- **Get a category by ID**
  - `GET /categories/{id}`

- **Create a new category**
  - `POST /categories`
- **Update a category**
  - `PUT /categories/{id}`
- **Delete a category**
  - `DELETE /categories/{id}`

Images

- **Get all images**
  - `GET /images`
- **Get an image by ID**
  - `GET /images/{id}`
- **Upload a new image**
  - `POST /images`
- **Update an image**
  - `PUT /images/{id}`
- **Delete an image**
  - `DELETE /images/{id}`

Annotations

- **Get all annotations**
  - `GET /annotations`
- **Get annotations by Image ID**
  - `GET /images/{imageId}/annotations`
- **Create a new annotation**
  - `POST /annotations`
- **Update an annotation**
  - `PUT /annotations/{id}`
- **Delete an annotation**
  - `DELETE /annotations/{id}`

Note: Since this is a JSON placeholder API, the data will not persist. You can simulate the API responses as needed for the purpose of this assignment.

AVERROES

Features to Implement

1. Image Management

- **Upload Images**
  - Allow users to upload images (simulate the upload process).
  - Associate images with categories.
- **View Images**
  - Display a gallery of images fetched from the API.
  - Show image details like name, upload date, metadata, and category.
- **Delete Images**
  - Enable users to delete images.
  - Confirm deletion using a modal.

2. Category Management

- **Create Categories**
  - Allow users to add new categories using a form.
- **View Categories**
  - Display a list of categories.
- **Update Categories**
  - Enable users to edit existing categories.
- **Delete Categories**
  - Allow users to delete categories.
  - Confirm deletion using a modal.

3. Filtering

- **Filter by Name**
  - Implement a search functionality to filter images by name.
- **Filter by Metadata**
  - Allow filtering images based on metadata (e.g., size, resolution).
  - Enable users to filter images by category.

AVERROES

4. Image Annotation

- **Implement Drawing Tool**
    - Use **React Konva** for image annotation.
- **Annotation Features**
    - Allow users to draw rectangles on images.
    - Choose colors for annotations.
- **Save Annotations**
    - Save annotations to the API.
    - Fetch and display annotations when viewing an image.
- **Annotation Management**
    - Enable users to delete annotations

Additional Guidelines

- **Code Quality**
    - Write clean, maintainable, and well-documented code.
- **Responsive Design**
    - Ensure the application is responsive and user-friendly on all devices.
- **Error Handling**
    - Implement proper error handling and display user-friendly messages.
- **Performance**
    - Optimize for performance where possible (e.g., efficient data fetching).
- **State Management**
    - Use React Query for data fetching and caching.

Deliverables

- **Source Code**
    - A link to your GitHub repository containing the complete source code.
- **README**
    - Instructions on how to set up and run the application.

    - Any notes or considerations you'd like to share.
- **Demo (Optional)**
    - If possible, deploy the application and provide a link.

Submission

Please email the link to your GitHub repository (and demo link if available) by the deadline.

Evaluation Criteria

- **Functionality**
    - Completeness of the required features.
- **Code Quality**
    - Cleanliness, structure, and documentation.
- **User Experience**
    - Intuitive and responsive UI/UX design.
- **Technical Skills**
    - Proper use of Next.js, MUI, React Query, and React Konva.
- **Bonus Implementation**
    - Additional points for implementing the image annotation feature effectively.

Final Notes

- **Data Persistence**
    - Since the API is a JSON placeholder and doesn't persist data, you can simulate API responses or use local state management for annotations.
- **Creativity**
    - Feel free to add any additional features or enhancements that showcase your skills.
- **Questions**
    - If you have any questions or need clarifications, don't hesitate to reach out.

Good luck, and we look forward to reviewing your work!