Project Overview

You will create a simple dashboard in React with a single navigation link: Compliance. On this page, there will be two accordion cards labeled KYC and AML. Each accordion will include:

1. A short explanation of why the documents are required.
2. A list of required documents (e.g., “Government-issued ID,” “Proof of Address,” etc. for KYC).
3. A list of requirements (e.g., must be valid for 3 months, must be unaltered).
4. An inline Upload button next to each required document (so the user can upload the correct file).
5. A Submit button at the bottom of each accordion. When clicked, a modal should appear confirming that the documents have been submitted successfully.

On the backend, you will build a Django application connected to a PostgreSQL (or similar) database. You will:

- Create a model (or set of models) to store the uploaded documents and their metadata.
- Provide an API endpoint (or multiple endpoints) to handle the file uploads and any status updates.
- Implement basic error handling (e.g., show an error message if the file cannot be saved).

This project tests your ability to implement a small feature end-to-end, from front-end UI/UX to back-end data persistence.

Detailed Requirements

1. Frontend (React)

1. Project Structure & Navigation
    a. Use a modern React setup (e.g., Create React App or Vite).
    b. A simple top-level navigation or sidebar with a single link labeled Compliance.
    c. When that link is clicked, load the Compliance page with the two accordion sections.
2. Accordion Layout

a. KYC Accordion (closed by default):
  i. Description: “Personal identification verification is required to comply with regulatory requirements and prevent fraud.”
  ii. Required Documents:
    1. Government-issued ID
    2. Proof of Address
    3. Selfie with ID
  iii. Requirements:
    1. Valid for at least 3 months
    2. Clearly visible photo and text
    3. No digital alterations
  iv. Inline Upload Button: each of the above documents should have its own button to upload the relevant file.
  v. Submit Button at the bottom:
    1. On click, open a modal (e.g., “Your KYC documents have been submitted!”).
      a. Just a simple success message is fine.
b. AML Accordion (closed by default):
  i. Description: “Anti-money laundering checks are required to ensure the legal source of funds and compliance with financial regulations.”
  ii. Required Documents:
    1. Bank Statement
    2. Source of Funds Declaration
  iii. Requirements:
    1. Documents must be less than 3 months old
    2. Must show full name and address
    3. Must be from a recognised financial institution
  iv. Inline Upload Button for each required doc.
  v. Submit Button with a similar modal.

3. UI/UX Notes
  a. You have freedom in the design (use any React UI library or plain CSS).
  b. Make it clear which documents have been uploaded (e.g., show a small confirmation label or icon).
  c. The modal can be minimal, just a quick message and an “OK” button.
4. File Upload Mechanics
  a. Each inline button should open up a file picker.

b. On selecting a file, send it to the backend via a REST API endpoint (or you can store it temporarily in state until the user hits “Submit,” whichever approach you prefer).

c. Show basic error handling (e.g., display a message if upload fails).

This is due on Monday 17th March at 3pm. Upload it to a private GitHub repository and invite me to the repo. My email address is nana.ossei@tokunize.com