Assessment: File Upload Platform

🏗️ Overview

Build a secure, full-stack web app that allows authenticated users to upload, manage, and view files stored on **AWS S3**. This app should enforce security, support metadata tracking, and be cleanly designed with a React frontend.

🔧 Tech Stack (Mandatory)

Layer	Technology
Backend	Django, Django REST Framework, Boto3
Frontend	React, HTML, CSS, JS (Bootstrap/Tailwind optional)
Storage	AWS S3 (via Boto3 SDK)
Database	PostgreSQL
Auth	JWT-based

🚀 Functional Requirements

🔐 Authentication

- Register / Login with JWT tokens
- Only logged-in users can upload/view/delete files

📤 File Upload Module

- Upload **multiple files** (via drag & drop or file picker)
- Allowed types: `.pdf`, `.docx`, `.jpg`, `.png`, `.xlsx`
- Max file size: 15MB per file

- Uploads go to AWS S3 using Boto3 with a user-specific prefix (`s3://bucket/user123/filename.pdf`)
- Store metadata in DB:
    - Filename
    - S3 URL
    - Size
    - MIME type
    - Uploaded at
    - User ID

📁 File Management

- Show list of uploaded files (with download & delete buttons)
- Enable users to **delete files** (from S3 and DB)
- Admins can see all uploads by all users
- Generate **pre-signed URLs** for downloads (expire after 1 hour)

🌐 Frontend (React)

- Auth pages: Login / Register
- File upload interface:
    - Drag-and-drop or file selector
    - Progress bars for each file
    - Display list of files with metadata
    - Buttons for download (via pre-signed URL) and delete
- Responsive layout with a clean UI

🔒 Security Expectations

- Use environment variables for AWS keys
- Validate file types/sizes in both frontend & backend
- Prevent users from accessing or deleting other users' files
- Use IAM policies to lock S3 bucket access by key prefix (`user123/`)

📄 Optional Enhancements (Bonus)

- Add folder grouping support for uploads
- Paginate file list
- Tagging system for files (e.g., “Resume”, “Invoice”, etc.)
- Add thumbnail preview for images