File Upload & Metadata Extraction Service

Objective

Build a service that allows users to upload files, stores them in AWS S3, and triggers a process to extract metadata (e.g., file size, type, basic text content if a PDF).

Instructions

1. **Create a RESTful API endpoint**:
    - Implement a `/upload` endpoint that accepts file uploads (e.g., PDF, image) and user provided metadata, for example: author name, expiration date, etc.
    - Store each file in an S3 bucket and store the user metadata.
    - If the process is successful, return `file_id`, otherwise an error message.
2. **Metadata Extraction with AWS Lambda**:
    - Set up a Lambda function to be triggered when a new file is uploaded to the S3 bucket.
    - The Lambda function should extract additional metadata from the file (e.g., file type, size, number of pages if a PDF).
    - Store this metadata in a DynamoDB table with a unique connection to the file.
3. **Retrieve Metadata**:
    - Implement an additional API endpoint `/metadata/{file_id}` that allows users to retrieve metadata for a specific file by its unique identifier.

Requirements

- Use Node.js for the Lambda functions.
- Deployable on AWS.
- Clear, modular code with comments explaining each step.

Evaluation Criteria

- **Code Structure & Modularity**: Evaluate how well components are separated (API endpoint, S3 storage, Lambda processing).
- **AWS Integration**: Ensure correct usage of S3, DynamoDB, and Lambda.
- **Error Handling**: Look for comprehensive error handling for file uploads, Lambda triggers, and metadata retrieval.
- **Extra**: A README file explaining decisions, challenges faced, and any assumptions made.