Assignment Title: Building a RESTful API with Flask - Error Handling, Authentication, and File Handling with Public and Admin Routes

Introduction:
In this assignment, you will be building a RESTful API using Flask that covers error handling, authentication, and file handling. The API will have two types of routes - public routes that can be accessed without authentication and protected admin routes that require authentication. The purpose of this assignment is to help you understand how to build a robust API that can handle errors, authenticate users, and handle file uploads.

Task 1: Setting up the Flask application

- Create a new Flask application and set up the necessary packages and modules.
- Create a virtual environment for the application.
- Connect your Flask application with the Database (MySQL preferably.)

Task 2: Error Handling

- Implement error handling for your API to ensure that it returns proper error messages and status codes.
- Create error handlers for ex. 400, 401, 404, 500, and any other errors that you feel are necessary.
- Make sure that error messages are returned in a consistent format.

Task 3: Authentication

- Implement authentication for your API using JWT Authentication.
- Create a user model with username and password fields.
- Implement a login endpoint that authenticates the user and returns a JWT token.
- Implement a protected endpoint that requires a valid JWT token to access.

Task 4: File Handling

- Implement file handling for your API to allow users to upload files.
- Create an endpoint that allows users to upload files.
- Implement file validation to ensure that only certain file types are allowed.
- Implement file size validation to ensure that files are uploaded within the allowed file size limit.
- Store uploaded files in a secure location. (A folder in your project’s folder structure.)

Task 5: Public Route

- Create a public route that allows users to view public information.
- Implement an endpoint that returns a list of items that can be viewed publicly.
- Ensure that this endpoint does not require authentication.

Submission Requirements:

On Canvas, submit a link to the Git repository that can be publicly accessed and the video. **Each student must submit it on Canvas.**

- Your submission should include a Git repository with all the code for the Flask application.
- Include a README file that explains how to set up and run the application and **names of the team members (2 - 3 members per team).**
- Include a requirements.txt file with all the necessary packages.
- Include one screenshot including all the endpoints listed in POSTMAN.
- **Your submission should include a video demonstrating the working of authentication and file handling endpoints.**

Grading Rubric:

- Proper implementation of error handling (20%)
- Proper implementation of authentication (20%)
- Proper implementation of file handling (20%)
- Proper implementation of public and admin routes (20%)
- Naming convention, formatting, comments. (20%)