The backend code is designed to securely handle file uploads (such as lecture materials and audio) from users. It supports specific file types like PDF, DOCX, PPTX, MP3, WAV, and now OPUS. The code ensures that only safe and expected files are accepted by verifying their file extensions and MIME types.

* This dictionary maps each file extension to a list of MIME types that are valid for that extension.
* For example, if a user uploads a .pdf file, the backend checks that its MIME type is 'application/pdf'.
* The same applies to audio files like .mp3, which may be encoded in various ways (audio/mpeg, audio/opus, etc.).

**How File Upload Works**

1. User uploads a file using Postman or a frontend form.
2. The backend receives the file using Flask's request.files.
3. The backend extracts:
   * The file extension (e.g., mp3)
   * The MIME type (e.g., audio/opus)
4. It then checks:
   * Is the extension in ALLOWED\_EXTENSIONS?
   * Does the MIME type match any in the list for that extension?
5. If both checks pass → the file is accepted and saved.
6. If not → an error message is returned.

Example error response if invalid:

{

"error": "File type not allowed. Supported formats: PDF, DOCX, PPTX, MP3, WAV",

"success": false

}

**Using Postman**

On Day 2, tested this upload system using Postman, a tool that helps test APIs without writing frontend code. You:

* Opened Postman and created a new request.
* Set the method to POST.
* Entered the upload endpoint URL
* Went to the "Body" tab → selected form-data.
* Added a key as file, and selected your file from your computer (e.g., .opus file).
* Clicked Send to submit the file to your backend.

Postman then shows the server response based on whether the file was accepted or rejected.