

Day 24 – File Uploads & Real-Time Communication

Theme:

Adding file upload capability and integrating live chat for student-instructor interactions.

Challenge 1 (Easy): File Upload

User Story:

As an instructor, I want to upload course materials.

Problem Statement:

Use **multer** to upload PDF files and store them in /uploads folder.

Expected Outcome:

POST /upload → "File uploaded successfully: module1.pdf"

Challenge 2 (Average): Serve Static Files

User Story:

As a student, I want to download course materials.

Problem Statement:

Serve uploaded files using `express.static()` from /uploads folder.

Expected Outcome:

Visiting /materials/module1.pdf downloads the file.

Challenge 3 (Difficult): Real-Time Chat

User Story:

As a student, I want to chat with my instructor during a live session.

Problem Statement:

Integrate **Socket.io** to enable real-time bi-directional communication between client and server.

Expected Outcome:

Messages appear instantly for all connected users.

Best Practices for Day 24

- Always sanitize file names and restrict file types.
 - Store large files on cloud (S3, Azure Blob).
 - Use namespaces or rooms in Socket.io for scalability.
 - Secure WebSocket connections with HTTPS in production.
-