**Main Goal:**

Implement a backend system that can take lecture text and return a concise summary using an NLP model.

**What Building**

Adding a **text summarization feature** to your Flask backend. This allows students or teachers to upload lecture content and receive an AI-generated summary via an API.

**Team Task Breakdown**

**Person A: Backend API Developer**

* Works in app.py
* Creates an API endpoint /summarize that:
  + Accepts text via POST
  + Calls the summary generator function
  + Returns the summary as JSON
* Handles errors like empty input or long text

**Person B: NLP Model Integrator**

* Works in summarizer.py
* Installs and sets up transformers and torch
* Implements generate\_summary(text) using HuggingFace's summarization pipeline
* Tests this module independently to ensure the summary works

**Technical Workflow**

1. User sends lecture text to /summarize route (via Postman or frontend).
2. Flask route receives and cleans the text.
3. The route calls generate\_summary(text) (from summarizer.py).
4. The model returns a short summary.
5. Flask sends the summary back to the user.

**Testing**

* Tested summarizer.py alone to ensure the model works.
* Tested /summarize API using Postman with real lecture samples.

**Deliverables by End of Day 3**

* A working API at /summarize
* A reliable summarizer.py using HuggingFace model
* Backend integration tested with Postman
* Prepared for frontend integration next