Day 3 - AI Class Assistant

Text Summarization Module (2-Person Task Split)

Project Folder Structure (Backend - Flask)

/AI-Class-Assistant-Backend

app.py Main Flask app (Person A edits this)

summarizer.py Summarization logic (Person B edits this)

uploads/ Uploaded lecture files

requirements.txt

Person A - Work in app.py

- Purpose: Handle API routes and requests

1. Import libraries and generate_summary()

```
from flask import Flask, request, jsonify
from flask_cors import CORS
from summarizer import generate_summary  # Person B's function
```

2. Setup Flask app

```
app = Flask(__name__)
CORS(app)
```

3. Add the summarize route

```
@app.route('/summarize', methods=['POST'])
def summarize_text():
    data = request.get_json()
    text = data.get('text', '')

    if not text:
        return jsonify({'error': 'No text provided'}), 400

    if len(text) > 1000:
```

```
        text = text[:1000]

    summary = generate_summary(text)
    return jsonify({'summary': summary})
```

Person B - Work in summarizer.py

- Purpose: Handle the summarization logic using NLP model

1. Install required libraries

```
pip install transformers torch
```

2. Write summarizer function in summarizer.py:

```
from transformers import pipeline

summarizer = pipeline("summarization")

def generate_summary(text):
    result = summarizer(text, max_length=150, min_length=30, do_sample=False)
    return result[0]['summary_text']
```

How to Test It (Both Together)

1. Run Flask:

```
python app.py
```

2. Use Postman or curl to test:

- URL: http://127.0.0.1:5000/summarize
- Method: POST
- Body (JSON):

```
{
  "text": "Paste a long lecture or article text here..."
}
```

Response:


```
{
  "summary": "This is the summarized version of the input text."
}
```