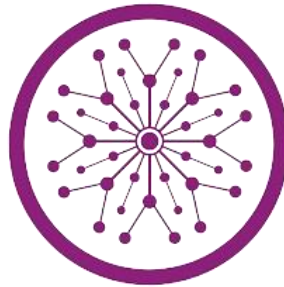


Programming for AI

Lab

Task 12



The Superior University

Faculty of Computer Science & Information

Technology

2023-2027

Submitted To:

Sir Rasikh Ali

Submitted By:

Qaswa Chaudhary

Roll No:

SU92-BSDSM-F23-002

Section:

4A

Department:

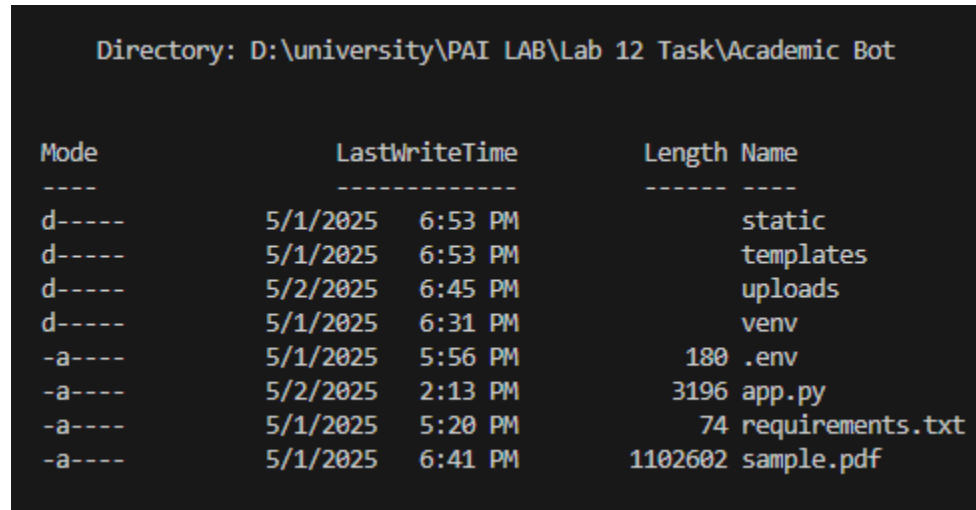
Software Engineering

Project: Academic Research Assistant Bot

Overview:

This is a Flask web application that allows users to upload academic PDFs and ask research-related questions. It uses **Sentence Transformers** for semantic chunking & embedding, **FAISS** for vector search, and **OpenAI's API** for intelligent answering based on relevant PDF content.

Project Structure:



```
Directory: D:\university\PAI LAB\Lab 12 Task\Academic Bot
```

Mode	LastWriteTime		Length	Name
d----	5/1/2025	6:53 PM		static
d----	5/1/2025	6:53 PM		templates
d----	5/2/2025	6:45 PM		uploads
d----	5/1/2025	6:31 PM		venv
-a----	5/1/2025	5:56 PM	180	.env
-a----	5/2/2025	2:13 PM	3196	app.py
-a----	5/1/2025	5:20 PM	74	requirements.txt
-a----	5/1/2025	6:41 PM	1102602	sample.pdf

Environment Variables (.env)

Variable	Description
----------	-------------

MODEL_NAME	OpenAI model name (e.g., gpt-4)
------------	---------------------------------

API_KEY	Your OpenAI API key
---------	---------------------

BASE_URL	OpenAI API base URL
----------	---------------------

Functional Components

1. PDF Text Extraction (extract_chunks)

- Opens a PDF using PyMuPDF (fitz).
- Extracts paragraphs by splitting page text at double newlines (`\n\n`).

2. Embedding Creation (create_embeddings)

- Uses SentenceTransformer (all-MiniLM-L6-v2) to generate vector representations of chunks.

- Creates a FAISS index for fast similarity search.

3. Chunk Search (search_chunks)

- Encodes the user's question into a vector.
- Searches the top 3 most similar chunks using FAISS.
- Returns relevant content.

4. Chat API (chat_with_bot)

- Constructs a prompt with the selected context.
- Sends it to the OpenAI API to generate a detailed answer.

Routes

@app.route('/', methods=['GET', 'POST'])

- **GET:** Renders the form to upload a PDF and input a question.
- **POST:**
 - If upload_new is submitted, resets session.
 - If a PDF is uploaded, extracts text and builds index.
 - If a question is submitted, returns a response from the bot.

Front-End (templates/index.html)

- Provides a user-friendly form to upload a PDF and enter questions.
- Displays uploaded filename.
- Shows the answer from the bot in a styled box.

Styling (static/style.css)

- Dark theme with teal highlights.
- Responsive layout with padding and shadows.
- Styled input, button, and answer display.