Undergraduate Final Project: ITMD: 447 Web Systems Integration

Objective:
Develop a Flask-based Python web application to upload PDFs and generate a summary using OpenAI's GPT-3.5 Turbo model. This project will emphasize secure, testable, and efficient code.

Use this project as an opportunity to produce something you can talk about and show to potential employers during interviews and on your resume.

Instructions:

Download the Starter Template:
- Navigate to the Blackboard course portal.
- Locate the Assignments section and go to the Final Project.
- Download the provided pdf_summarizer_template.zip containing starter code.

Set Up the Development Environment:
- Extract the downloaded zip file.
- Navigate to the project directory in your terminal.
- Run pip install -r requirements.txt.

OpenAI Registration and API Key Setup:
- Register for an OpenAI account using your IIT hawk email. This will grant you $5 free usage provided by OpenAI for this project.
- Get your API key from the OpenAI Dashboard.
- Securely store your API key using an environment variable or a .env file. Never hardcode the API key.

Developing the Application:
- Your application should allow users to upload PDF files and return summarized content.
- Use the provided template code as a foundation. Integrate the required logic for file handling, text extraction, and OpenAI API interaction.
- Handle file validation, exceptions, and potential errors.

Implementing Unit Tests:
- **PDF Upload Test:** Verify that a PDF file is uploaded correctly, and that non-PDF files are rejected.
- **Summary Generation Test:** Ensure the OpenAI API call is made and a summary is returned after a valid PDF upload.
- **Exception Handling Test:** Check the application's response to possible exceptions like empty files or an unavailable OpenAI API.

Secure Coding Practices:

- **File Validation:** Ensure only valid PDFs are accepted. Reject files with wrong extensions or corrupted content.
- **API Interaction Safety:** Ensure sensitive data, like the API key, is not exposed. Handle timeouts and errors gracefully.
- **Data Presentation:** Escape any data being rendered to prevent Cross-Site Scripting (XSS) attacks. Avoid displaying sensitive error messages to the user.

Integrating Stylish UI Elements:

Sample solution Screenshots are presented on the next page for inspiration. Your solution does not need to look exactly like mine, but should incorporate similar elements to stylize and make the web pages user friendly and presented clearly.

For the **Upload Page (upload.html)**:

- Use the provided template code
- Apply CSS styles
- The upload button should animate slightly on hover to give feedback to the user.

For the **Summary Page (summary.html)**:

- The summary should be presented in a centered box.
- Implement the JavaScript function copyToClipboard() to allow users to copy the summary to their clipboard.
- The 'Copy to Clipboard' and 'Upload another PDF' buttons should be side-by-side, and both should have hover animations.

Submission:

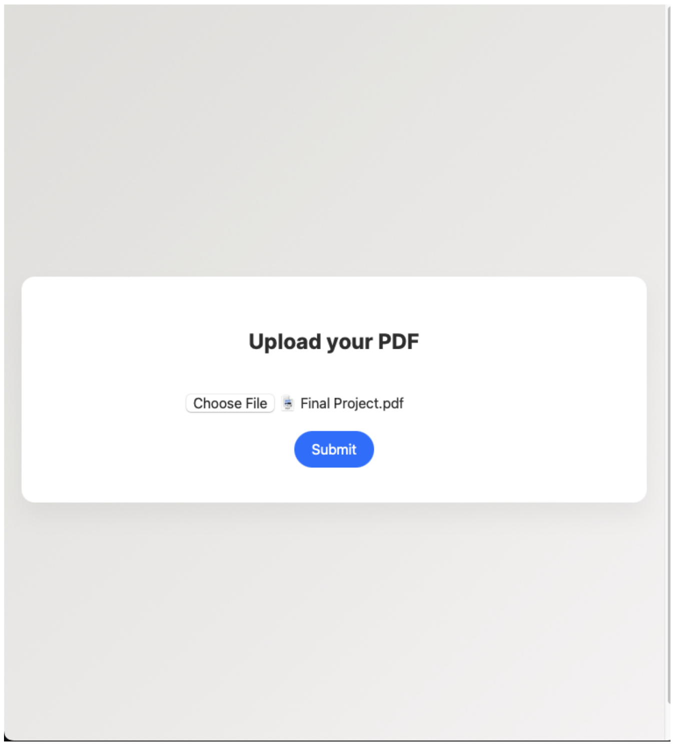
- After thoroughly testing all functionalities, compress your project folder (excluding the virtual environment) into a zip.
- Submit the zip file on Blackboard before the deadline.

Grading Criteria:

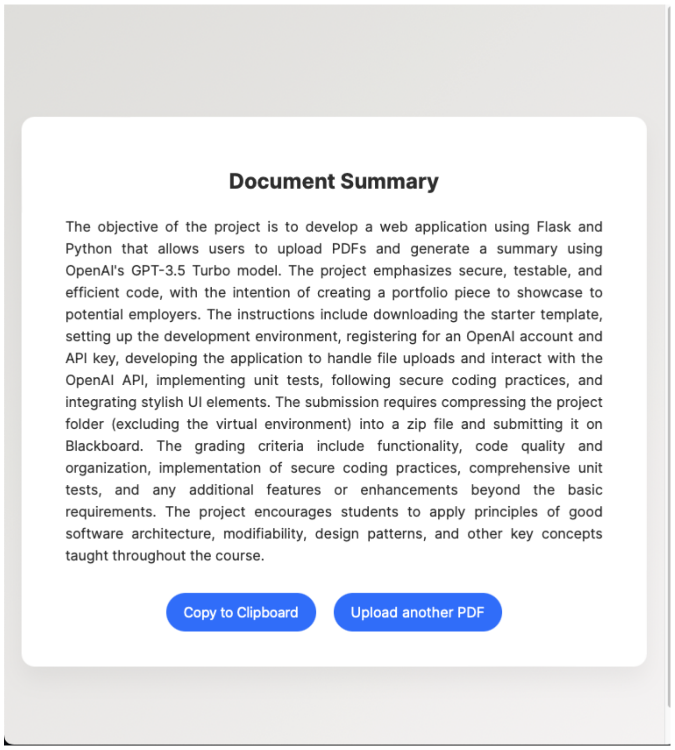
- Functionality and correctness of the application.
- Quality and organization of code.
- Proper implementation of secure coding practices.
- Comprehensive unit tests ensuring reliability.
- Additional features or enhancements implemented beyond the basic requirements.

Note:
Revisit course modules and apply principles of good software architecture, modifiability, design patterns, and other key concepts taught throughout the duration. This project is a platform for you to showcase all that you've learned. Best of luck!

upload.html



summary.html



Document Summary

The objective of the project is to develop a web application using Flask and Python that allows users to upload PDFs and generate a summary using OpenAI's GPT-3.5 Turbo model. The project emphasizes secure, testable, and efficient code, with the intention of creating a portfolio piece to showcase to potential employers. The instructions include downloading the starter template, setting up the development environment, registering for an OpenAI account and API key, developing the application to handle file uploads and interact with the OpenAI API, implementing unit tests, following secure coding practices, and integrating stylish UI elements. The submission requires compressing the project folder (excluding the virtual environment) into a zip file and submitting it on Blackboard. The grading criteria include functionality, code quality and organization, implementation of secure coding practices, comprehensive unit tests, and any additional features or enhancements beyond the basic requirements. The project encourages students to apply principles of good software architecture, modifiability, design patterns, and other key concepts taught throughout the course.