



CourseCompanion

Student: Hermes Bonilla

Mentor: Elaria Ghobrial, *Product Owner*

Instructor/Faculty: Masoud Sadjadi, *Florida International University*

Our Solution

With no previous work on this project, this is the first semester that this software solution was developed. Our tech stack includes Node.js, ChatGPT API, CSS, HTML, and JS.

- Current contributions include the connection between the server and ChatGPT's API, the design and implementation of the student UI view, and the integration of the class response prompts based on the corresponding syllabus.
- Potential future contributions may include adding a view that allows the instructor to upload any documents in addition to the syllabus. An important feature that should also be implemented is linking a student's Canvas login to student view of CourseCompanion.

Problem Statement

In response to the recurring issue of students repeatedly asking professors the same questions about course details, we identified a pattern: students often seek reassurance from direct answers rather than consulting the provided syllabus.

To address this pattern and alleviate the burden on professors, we developed CourseCompanion—an AI-powered chatbot accessible via our website. CourseCompanion leverages the course syllabus to provide students 24/7 immediate support and accurate responses to their questions.

Personal Contributions

Throughout this semester, my focus was handling the AI response generation side of our project. Some of my contributions are as follows:

Parsing User Input into Relevant Categories:

- Parsed user input received from website into categories that provides syllabus context to the chat generation model.

Regex for Syllabus Category Relevance:

- Scanned for off-topic questions using Regex to then signal to the user that their question cannot be answered since they are not relevant to the class.

Client-Side Integration with OpenAI API:

- Created POST request to OpenAI API's chat completions endpoint with relevant payload parameters (model type, prompt, user input, etc.)
- Display the AI's response by creating an HTML element and appending it to the chat box if response is successful.

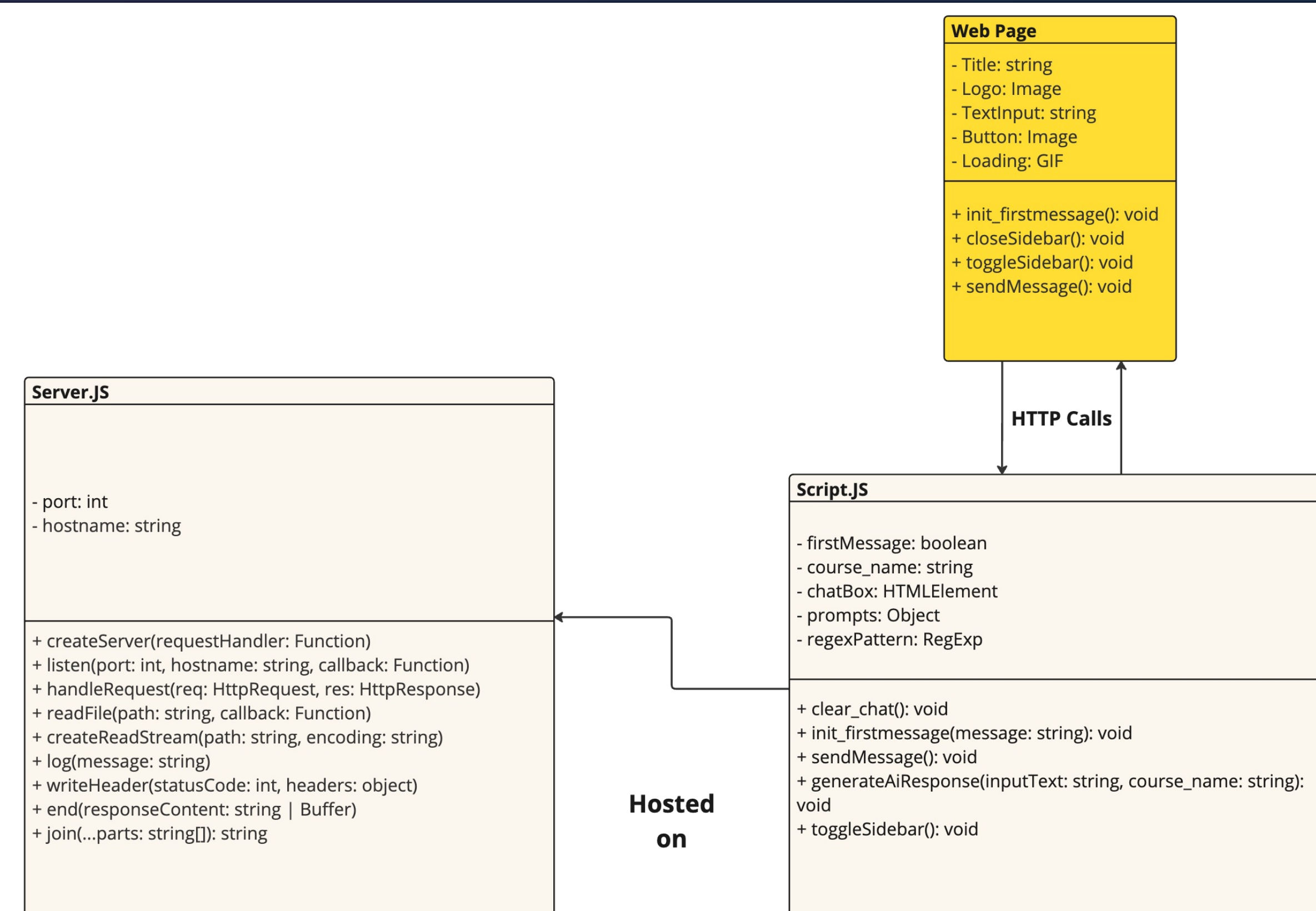
UI Class Button functions:

- Created functions that UI buttons used to clear previous chat history when users changed classes and initialize first messages.

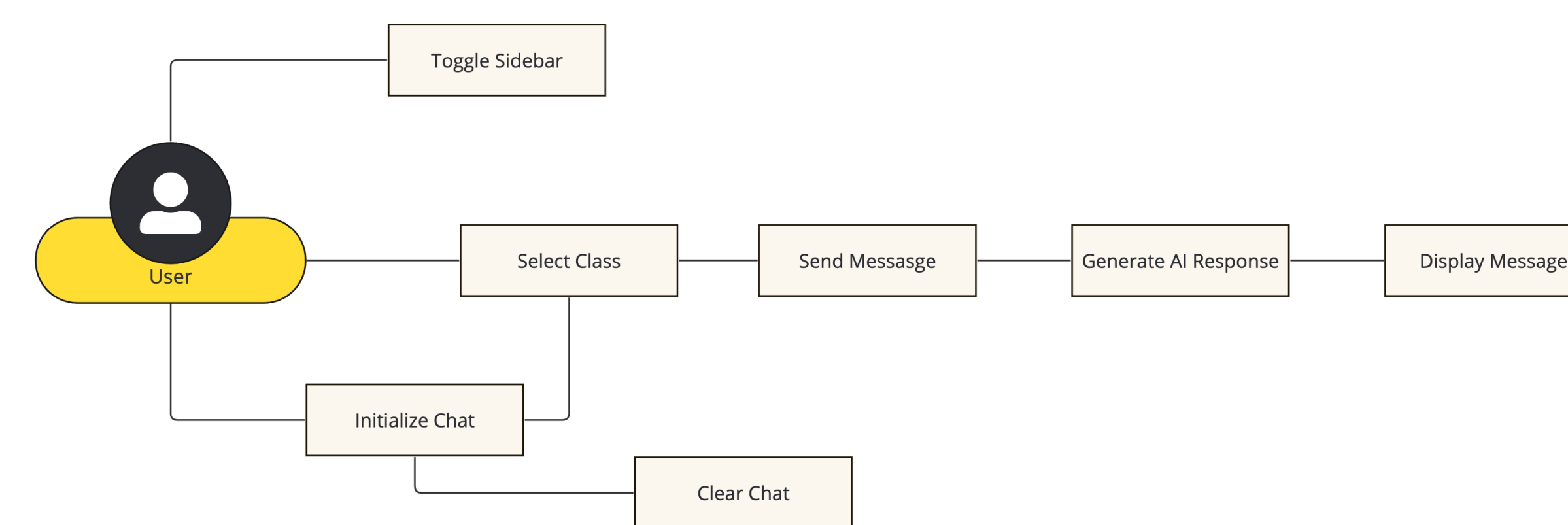
Error Handling:

- Implemented error handling in chat generation responses and user input to catch errors rather than break our code.

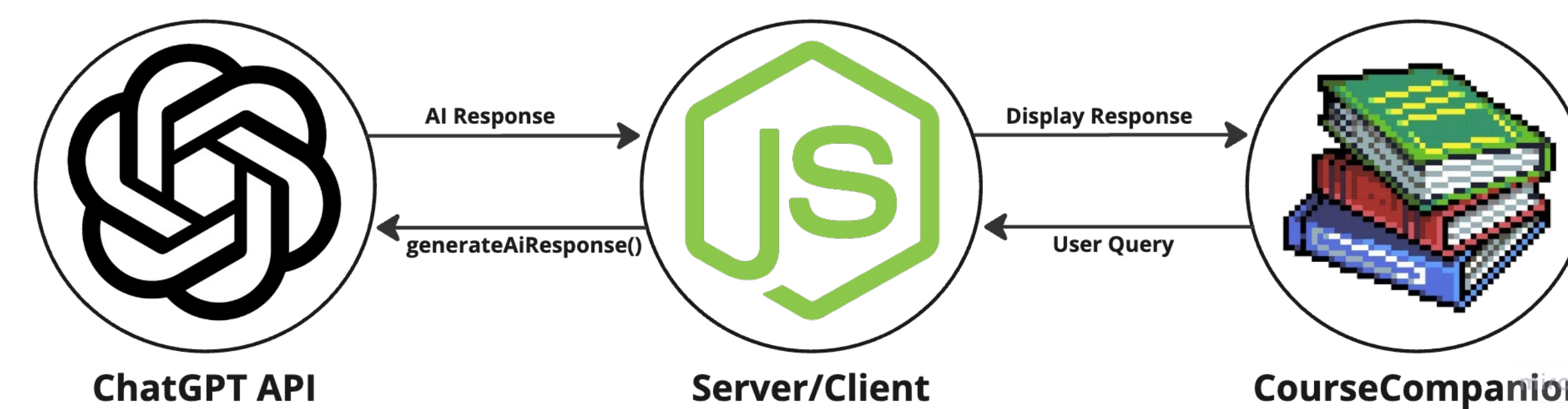
Object Design



Use Case Diagram



Architecture Diagram



Technologies



Summary

As a team, we focused on developing a user-friendly webpage for students to interact with an AI chatbot that answers questions about a course syllabus. We were able to establish a connection between ChatGPT's API and our HTTP server. With this set up, we designed and integrated a webpage that allows the student to interface with the server. To ensure that our goals were met, our team tested our solution with current FIU students and took note of their feedback. Moving forward, subsequent teams are encouraged to prioritize the development of a teacher interface incorporating upload functionality, as well as the integration of student Canvas accounts into our CourseCompanion platform.

References

