

Project Overview:

About: This project is a web application that integrates an AI chatbot capable of processing and responding to user inputs based on uploaded documents. The core idea is to create a tool where users can not only chat with the AI but also upload assignments, articles, or notes and then prompt the AI for clarifications from those documents. The application will be powered by a React front-end and a Django back-end. The AI model itself will be sourced from Hugging Face. Users will have the ability to type custom prompts into a text box, and the AI will respond with detailed answers based on the content of the documents they've uploaded. This feature not only supports general prompts but also shows the capabilities of AI as an asset.

Why: This project was chosen because I have an interest in AI and seeing how influential artificial intelligence will be in the future, I want to start learning about it. Additionally, while I have worked with React and Django individually, I want to try to build a full stack application that can take event calls.

Accomplishing: Create an intuitive platform for users to chat with AI, upload documents, and receive responses based on the document content.

Frontend (React) Goals:

- Simple, user-friendly design.
- Textbox for entering prompts.
- Upload button for document submission.
- Help button for usage instructions.

Backend (Django) Goals:

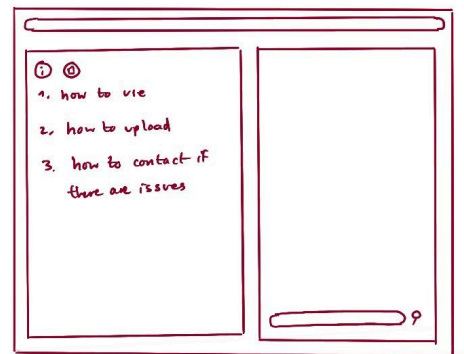
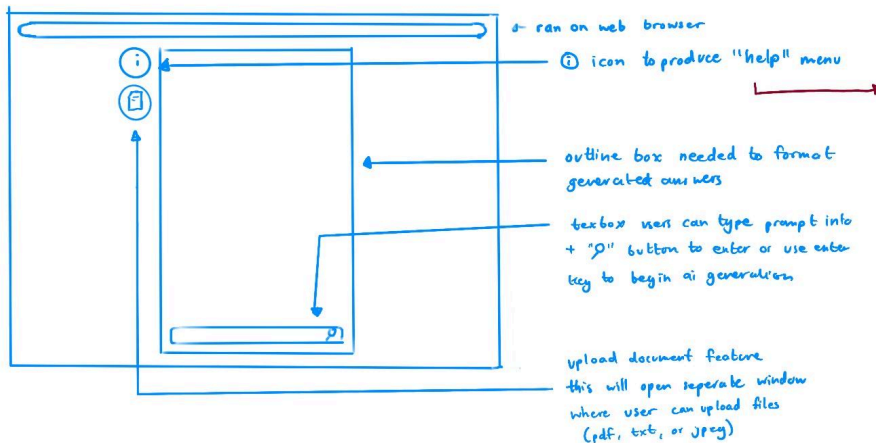
- Handle document uploads and storage securely.
- Process user inputs and pass relevant data to the AI model.
- Generate and return context-aware AI responses.

AI Model (Hugging Face LLM):

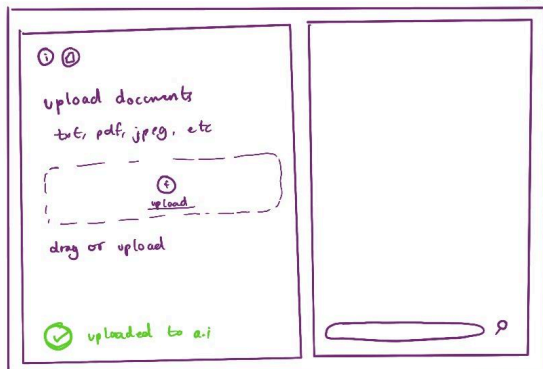
- Analyze uploaded documents to provide tailored, detailed answers.
- Understand and synthesize complex document topics.
- Deliver both direct responses and in-depth explanations when needed.

Real-time Interaction:

Application Design Mockup:



* chat box moves to the right. user can still use a.i. and prompt and upload feature. However, there is a window explaining how to use, how to upload documents, and how to contact for errors

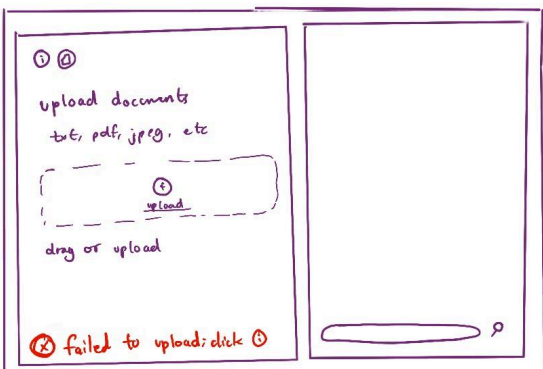


- chatbox is still moved to the right
- gives description of what files to upload and not accepted types
- when successfully uploaded, Ⓢ will appear at the bottom meaning user can write prompts

clicking the upload Ⓢ will open this window even from Ⓢ menu. Can move to help Ⓢ window in the upload Ⓢ menu. Clicking on the icons twice goes back to original (chat in the center)

final notes for development

- keep color scheme light with pastels
- avoid harsh borders, include drop shadows to create windows
- if can be included, hovering over buttons make them larger / highlighting
- sliding animation of chatbox left to right
- upload box made out of dotted lines



- if upload fails 'Ⓢ' will appear as well as a link to the help Ⓢ page

Project Plan:

It is uploaded as a PDF and an Excel File (Gantt Chart)

GitHub Repository:

<https://github.com/amanda-ucc/grade12-culminating>