

# AI Career Coach Chatbot Project Report

**Project Overview:** The AI Career Coach Chatbot project aims to develop a personalized, AI-powered chatbot to provide career guidance specifically for Computer Science students. The chatbot leverages advanced natural language processing (NLP) techniques to offer personalized advice, career development tips, and interview preparation assistance. This project is designed to fill the gap in personalized mentorship resources for students entering the tech industry.

**Development Process:** The development process began with creating a project repository and setting up the initial structure using Flask/Django. After establishing the basic framework, the integration of OpenAI's GPT-4o through the ChatGPT API was performed, ensuring the chatbot could generate meaningful and context-aware responses.

Defining user personas was a critical step to tailor the content and responses provided by the chatbot. By identifying the needs and preferences of the target audience, the chatbot could be designed to address the specific concerns and requirements of Computer Science students.

**Content Development:** All relevant content was stored in PDF format and organized into a single folder. To facilitate easy retrieval and efficient data processing, a Python script named `extract_texts.py` was developed. This script converted the PDFs into a single CSV file, making the data manageable and ready for embedding.

**Technical Implementation:** The core functionality of the chatbot is powered by the `app.py` file. The environment setup involves loading environment variables and setting the OpenAI API key using the `dotenv` library. Extracted texts from the PDFs are loaded and split into manageable chunks for embedding.

LangChain components were then initialized. OpenAIEmbeddings were created and stored in a FAISS vector store, enabling efficient search through the text data for relevant information. The vector store's retriever was used to fetch documents relevant to user queries, ensuring contextually accurate responses.

**Transition to Streamlit:** Initially, the project utilized a Flask application to serve the chatbot. However, for better front-end integration and user experience, the project transitioned to Streamlit. This transition simplified the development process and made the `run.py` and Flask application unnecessary.

**Challenges:** One significant challenge was the initial limitation of parsing only the first 3500 characters of text. This was resolved by implementing LangChain, allowing the chatbot to parse the full document efficiently. Additionally, several deprecation warnings related to the OpenAIEmbeddings class and BaseRetriever.get\_relevant\_documents method were encountered. While fixing these issues caused compatibility problems with the OpenAI version used, the current implementation continued to function as intended.

**Future Plans:** The future direction of the project involves developing a tool that allows anyone to create their own custom chatbot. Additionally, a humanized AI model with a human face will

be created for more engaging interactions. A study will be conducted to compare user feedback between the chatbot and the humanized model. The findings will be documented in a formal research paper and presented at a tech conference.

**Conclusion:** The AI Career Coach Chatbot project successfully developed a personalized career guidance tool for Computer Science students. Despite facing several challenges, the project achieved its goals and laid the groundwork for future enhancements and broader applications. The next steps include expanding the project's capabilities and sharing the findings with the broader tech community.