Final Year Project Proposal

TU856

AI Career Advisor

Aileen Coliban
C22304291

School of Computer Science
TU Dublin – City Campus

05/10/2025

Table of Contents

Contents

Declaration

I hereby declare that the work described in this dissertation is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

Signed:

*Aileen Coliban*_____

Aileen Coliban

07/10/2025

Project Title

AI Career Advisor: AI-Driven Screening and Career Recommendation System

Summary

The AI Career Advisor is an intelligent web-based application designed to support both job seekers and recruiters by leveraging artificial intelligence and natural language processing (NLP). The system analyses job seekers CVs to extract key details such as skills, education, and professional experience, and compares them to job descriptions sourced from the job market. This process allows the system to generate insights and recommendations for suitable roles, skill development, and areas of improvement.

Recruiters will have access to a complementary portal that allows them to upload job postings or candidate CVs. The AI will automatically extract and summarise key data for each applicant, producing visual dashboards that display skill profiles, match scores, and candidate comparisons. The overall objective is to enhance transparency and efficiency in the recruitment process while supporting job seekers with actionable, data-driven feedback making recruitment easier and more ethical. The final deliverable will include a functional prototype featuring an AI-powered backend, accessible and responsive user interfaces, and a recruiter dashboard for visual analysis.

Background (and References)

For my background research, I investigated how existing AI tools like Eightfold AI and HiringBranch use AI in recruitment. I found Eightfold's approach to automated skill extraction particularly interesting, while HiringBranch's use of interactive dashboards inspired how I plan to visualise recruiter data to make decision making seamless based on candidate performance and skill fit.

In addition to industry research, several previous TU Dublin projects involving AI recommendation systems, chatbots, and dashboards provided valuable insights into data handling, user interaction, and system evaluation, which gave me inspiration for the overall structure and user flow.
I wanted to understand how AI could be used not just to help recruiters but also help job seekers identify where they stand in the job market since it can be ambiguous and stressful for the most part.

During my internship with SAP's Centre of Expertise, I worked on an AI CV Scanner that enabled recruiters to upload multiple CVs and query data interactively (e.g., "Which candidate has the most relevant experience for this role?"). This experience highlighted the potential of AI to support recruitment while also revealing challenges in bias, accuracy, and scalability. The AI Career Advisor builds on these findings by creating a system that assists both recruiters and job seekers, promoting fair and informed decision-making.

Proposed Approach

These are the main areas to my approach to this project:

1. Research
   The initial stage will involve analysing existing AI recruitment tools, academic papers, and past student projects to identify best practices in NLP-driven screening, vector similarity matching, and dashboard design.

2. Requirements Gathering
   Functional and non-functional requirements will be defined for both user groups, job seekers and recruiters. This includes features like CV upload, job posting upload, skill extraction, similarity matching, and recruiter dashboard. Informal feedback will be collected from peers who are actively applying to jobs and internship colleagues to refine the system's features.

3. Analysis and Design
   System architecture and database schemas will be developed to support both data extraction and visualisation processes. The backend will use FastAPI for API endpoints, with FAISS as the vector store and SQLite or PostgreSQL for structure data. The frontend will be simple at first for testing purposes and later expanded to include dashboards for recruiters. I will also plan out how to integrate the NLP model to extract data from CVs and job descriptions, and how those embeddings will be compared.

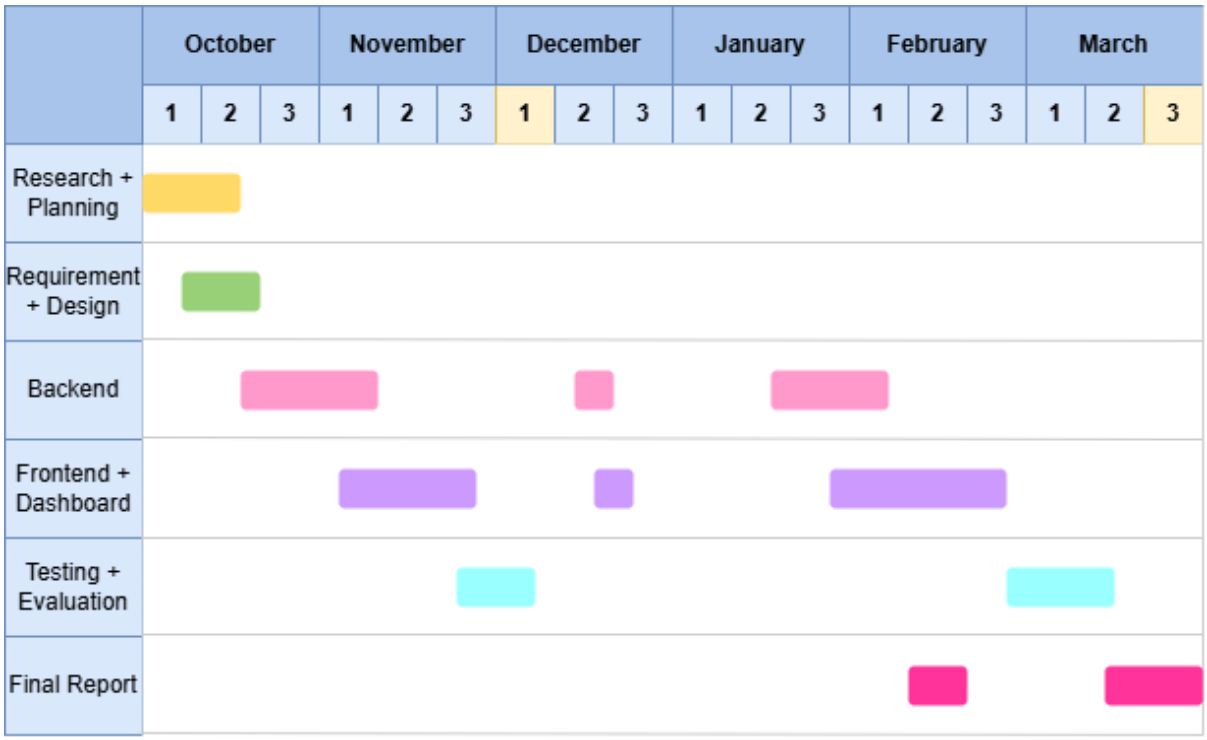
4. Implementation
   Development will begin with setting up the backend, NLP logic, a simple frontend and connecting them first. I will use a python module to read CVs and extract text, then generate embeddings using OpenAI or Hugging Face models. The FAISS vector database will store these embeddings and perform similarity searches. After that, high-level UI wireframes will also be created for both user portals, adding the dashboard and refining the design to follow Universal Design standards.

5. Testing and Evaluation
   Testing will occur continuously, with two dedicated testing phases, one for the prototype and another for the final product. Although I will be following an Agile approach where testing happens throughout every stage of development, these two dedicated testing periods will focus specifically on analysing and improving the model and systems overall performance. I will check the accuracy of the AI's skill extraction and similarity matching using sample data. I will also be testing the recruiter dashboard to display correct information. For evaluation, I will use different metrics such as precision, recall and F1-score to assess the model, while also gathering user feedback to measure usability and accessibility.

Deliverables

The completed project will include a fully functional web application with two main portals, one for job seekers and another for recruiters. It will have an NLP model that extracts key information from CVs and job descriptions, supported by a FAISS vector database for accurate similarity matching. The recruiter portal will have a dashboard that displays candidate summaries, match scores, and insights in a clear and concise format. The front-end will be designed based on the Universal Design principles. A full final report including technical documentation and research will be done alongside a presentation outlining the project.

Project Schedule



The project is divided into development phases to ensure a functional prototype is completed by December 1st, 2025, followed by refinement and evaluation in the second semester.
The month is split up in thirds in this gantt chart, the first third of the month, middle and end of the month.

The first two thirds of October will be focused on finalising the project proposal, background research and confirming which technologies should be used which will be the research and planning. Alongside research and planning, I will be working on the requirements and design

of the system which will entail creating an architecture diagram, database schema and mock-ups for the user and recruiter portals.

With backend and frontend developments the work will be interlinked together so I can see everything in the backend is working properly. So, I will start off by making a simple frontend to test functionality of the backend, then I will make it more detailed once the project picks up steam.

The yellow highlighted thirds indicate important dates for example, the first third of December will indicate that a working prototype and the interim report will have to be completed by then.

With Testing and evaluation, there will be two periods. The first period will be testing and evaluating to make sure the prototype is working correctly and the second period will be testing and making sure the final product is up to standard.

Finally, there is time sectioned for the final report documentation and completing the presentation alongside the report.

Technical Requirements

The project will be developed using Python, FastAPI, and FAISS for the backend. The NLP model will be based on OpenAI or Hugging Face embeddings, and PyMuPDF will be used for reading PDF CVs. The frontend will be built with Streamlit, HTML, CSS and JavaScript. The database will use SQLite or PostgreSQL for scalability. All development will be done on my local machine with GitHub. For deployment, Render or Heroku will possibly be used.

Conclusion

The AI Career Advisor project combines artificial intelligence, NLP, and ethical system design to create a fairer and more efficient recruitment process. It enables job seekers to receive data-driven feedback on their career readiness while providing recruiters with an analytical dashboard to evaluate candidates objectively. By integrating AI-driven data processing with intuitive design and accessibility standards, this project demonstrates how technology can promote transparency, fairness, and inclusivity within modern recruitment systems.

References

A Guide to Artificial Intelligence Resume Screening. (n.d.). Retrieved from HiringBranch: https://www.hiringbranch.com/blog/artificial-intelligence-resume-screening

Agentic AI Talent Platform for Smarter Hiring. (n.d.). Retrieved from SeekOut: https://www.seekout.com/

AI Career Coach. (n.d.). Retrieved from Sapia: https://sapia.ai/phai/

AI talent acquisition & recruiting platform . (n.d.). Retrieved from Eightfold: https://eightfold.ai/

COACH: AI-powered Career Coach . (n.d.). Retrieved from COACH: https://www.aicareercoach.org/

Gomes, P. (n.d.). *Generative AI Project: CV Analyzer*. Retrieved from Medium: ttps://patotricks15.medium.com/generative-ai-project-python-langchain-cv-analyzer-c56f07c7f1f0

Sonawale, Y. (n.d.). *How to Build an LLM-Based Resume Analyzer*. Retrieved from mercity: https://www.mercity.ai/blog-post/build-an-llm-based-resume-analyzer?utm_source=chatgpt.com

Appendix A: First Project Review

Title: NLPurchase – eCommerce Chatbot

Student: Stephanie Finn

Description (brief):
This project was based on creating an AI chatbot that could help users browse and buy products directly through Facebook Messenger. It used Wit.ai to process natural language so it could understand user queries and respond like a real assistant. It has a backend built in Node.js with a MongoDB database that handled all the products, conversations, and recommendations. The bot could suggest products, keep track of preferences and guide users through the shopping process in chat form.

What is complex in this project:
The complexity was connecting multiple systems together such as the NLP model, database and Facebook's APIs, requiring managing real time data and keeping responses fast and accurate. Natural language can be unpredictable alongside getting the chatbot to properly understand human input and respond correctly. Designing intent recognition and entity extraction tool a lot of fine tuning, making it the main complexity of this project.

What technical architecture was used:
The technical architecture consisted of Node.js for the backend, MongoDB for storing data, Wit.ai handled all the NLP implementation, and the Facebook Messenger APIs so users could chat to it directly.

Explain key strengths and weaknesses of this project, as you see it.
One of the key strengths of this project was the use of NLP to create a natural conversational environment.
However, the chatbot depended on predefined intents, which means it wasn't as flexible or adaptive as newer NLP models. It was also limited to Facebook Messenger.
Analysing this specific projects strengths and weaknesses, it is relevant to my project because it shows how NLP can be used to interpret user input and give meaningful, human like response. I liked how it handled user interaction and guided people through making decisions.

Appendix B: Second Project Review

Title: Auto-Efficiency – Machine Learning for Predictive Analysis and Recommender System

Student: Glory Pierce Eguare

Description (brief):
This project was based on building a machine learning model that predicts a car's fuel efficiency and recommends similar vehicles based on user preferences. The recommender system compared car specs like horsepower, engine size, and weight to find the most similar vehicles for the user. It combined predictive analytics with recommendation logic and included a dashboard where users could view the model's predications and recommendations.

What is complex in this project:
The complexity is within the training and optimising the prediction model and then connecting it to a user facing dashboard that updates dynamically. Managing the data between the ML model, backend API, and frontend dashboard required good system design and consistent testing.

What technical architecture was used:
It had a Flask backend, a TensorFlow/Keras model for prediction, and a React/Ionic frontend for the dashboard. The project followed an Agile development process.

Explain key strengths and weaknesses of this project, as you see it.
The key strength was how it combined machine learning with an interactive dashboard to make the data easy to understand. The connection between the predictive model, backend and frontend worked well. It was also well structured and modular, which made it easy to follow and potentially scale for bigger demands. One weakness was the the dataset might have limited how well the model could generalise its predictions.
This project is close to what I want to do with the AI Career Advisor. I liked how it linked an AI model to a live dashboard and turned predictions into something users could easily interpret. My system will do something similar, extracting and matching data using AI and visualising it clearly for recruiters and job seekers.

Appendix C: Prompts Used with ChatGPT

Prompt 1:
I am student doing Computer Science. I am in my final year starting to do my final year project and I need some ideas. Here are the modules I have done throughout the 4 years of my degree:
Year 1: Algorithm and Design and Problem Operating Systems 1 Computer Architecture and Technology Microprocessor Systems Data Exploration Mathematics 1 Communications for Computer Tech Web Development Information Technology Fundamentals Program Design Programming in C
Year 2: Algorithms and Data Structures Human Computer Interaction Data Communications Legal and Professional Issues Mathematics 2 Web Development 2 Software Engineering 1 Software Engineering 2 Databases 1 Object Oriented Programming in Java and Python Operating Systems 2
Year 3: Mobile Software Development Intro to Artificial Intelligence Cloud Computing Client Serving Programming Databases 2 Software Engineering 3 Year 4: Advanced Security 1 Advanced Databases Machine Learning Forensics
I have also completed 2 internships, one in optum as a data scientist where I made an application that uses predictive analytics and machine learning to create a loan calculator and the second one I was at SAP CoE as a business processes consultant, where I completed a couple of projects stemming from and AI CV scanner which allows recruiters to input pdfs of applicants and query them through an AI and compare CVs, to a Gamify Networking application that allows interns to network freely. Based on the information I have given to you above what ideas would be good for a final year project.

Prompt 2:
What are some references and tools I can research for my AI Career Advisor project?

Prompt 3:
What would be a suitable tech stack?

Prompt 4:
For the background I focused on looking at how Eightfold does their CV screening for recruiters, I also looked at hiringbranch for examples of how dashboards are used to visualise the data during screening so decision making can be quick, I am looking at previous final year projects from the years before what type of projects should I focus on for reference?

Prompt 5:
What would my project schedule look like so I can have a working prototype by December 1st and a complete project by march?

Prompt 6:
What ways can I test an AI model?

Appendix D: