

Enhanced Personalized Learning Experiences by Leveraging Knowledge Graphs and Prompt Engineering

MSc Research Project
Data Analytics

Malav Naik
Student ID: x23271779

School of Computing
National College of Ireland

Supervisor: Jorge Basilio

**National College of Ireland
Project Submission Sheet
School of Computing**



Student Name:	Malav Naik
Student ID:	x23271779
Programme:	Data Analytics
Year:	2024
Module:	MSc Research Project
Supervisor:	Jorge Basilio
Submission Due Date:	9/15/2025
Project Title:	Enhanced Personalized Learning Experiences by Leveraging Knowledge Graphs and Prompt Engineering
Word Count:	XXX
Page Count:	2

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	15th September 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission , to each project (including multiple copies).	<input type="checkbox"/>
You must ensure that you retain a HARD COPY of the project , both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhanced Personalized Learning Experiences by Leveraging Knowledge Graphs and Prompt Engineering

Malav Naik
x23271779

1 Introduction

The setup handbook includes libraries, software and hardware settings and critical code blocks relating to every phase of implementation.

2 System Requirement

2.1 Hardware Requirement

Server specifications to host AI/ML models, API layer and databases:

- Multiple-core CPU (at least 4 cores, modern intel or amd or equivalent)

- Adequate RAM (at least 16 GB; 32 GB- or more, depending on large-scale or production hosting)

- GPU (NVIDIA CUDA-supporting GPU recommended over LLM inference/fine-tuning, in case it is done on-premises)

- SSD (256GB at the very minimum; more in case of requiring to store large knowledge graphs or conversation log)

- Constant internet connection to host cloud based and API integrations

- Development machine (local frontend / backend coding/testing):

 - Contemporary PC/laptop having not less than 8GB RAM and 4-core processor

 - Development could be made without GPU, but one would need it to do local LLM experiments

2.2 Software Requirement

Operating System:

- Linux (Ubuntu/Debian/CentOS) for backend and cloud/server; Windows/Mac for local development

The use of critepython programming language has been applied. Python is an interpreted general purpose, high, and dynamic programming language ManeVanderPlas (2016). The Application development is supported through Object-Oriented technique of programming. It provides numerous advanced data structures in abundance and is simple to learn and exercise. Python is sweet and and easy to learn language, though it

is an appealing language used in Application Development. influential and robust. Géron (2022)

Coding has been done using Google Colab tool. Jupyter is open source and free Jupyter The Jupyter Project created notebook. Jupyter notebook is comparable to an interactive lab notebook in that it stores data, comments and code that allow editing that data. You can also run the notebook code and afterwards store the results. Brownlee (2018)

Backend and API:

Python 3.8+ (Flask server for REST APIs, AI/ML logic) SQLite (datastorage)

Node.js Express.js (specifically mentioned for API Gateway pattern)

Frontend:

React.js (with Redux and Tailwind CSS for state management and responsive UI)

2.3 Library and Packages

Python Libraries:

Flask (REST APIs WebSocket support)

TensorFlow/PyTorch (for ML/AI layers, as reference frameworks)

neo4j (for knowledge graph integration)

Redis-py (Redis operations)

OpenAI API client (connecting to GPT-3/3.5 or other LLMs)

scikit-learn, numpy, pandas (data analysis, entity resolution, feature engineering)

transformers / huggingface (LLM fine-tuning, if local)

JavaScript/Frontend Libraries:

React.js, Redux (state management)

Tailwind CSS (styling)

Socket.io or equivalent WebSocket library (real-time chat)

Axios/fetch (API requests)

References

Brownlee, J. (2018). How to setup a python environment for machine learning and deep learning. Accessed: 10 August 2025.

URL: <https://machinelearningmastery.com/setup-python-environment-machine-learning-deep-learning-anaconda/>

Géron, A. (2022). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 3rd edn, O'Reilly Media, Inc.

VanderPlas, J. (2016). *Python Data Science Handbook: Essential Tools for Working with Data*, O'Reilly Media, Inc.