Optimising Students Learning Experience Through AI-Driven Personalised Path Generation

Edgar Park

Supervisor: *Yanchao Yu*
Internal Examiner: *Saima Rafi*

Submitted in partial fulfilment of
the requirements of Edinburgh Napier University
for the Degree of
Master of Science in Computing

School of Computing

August 18, 2025

Authorship Declaration

I, Edgar Park, confirm that this dissertation and the work presented in it are my own achievement.

Where I have consulted the published work of others this is always clearly attributed;

Where I have quoted from the work of others the source is always given. With the exception of such quotations this dissertation is entirely my own work;

I have acknowledged all main sources of help;

If my research follows on from previous work or is part of a larger collaborative research project I have made clear exactly what was done by others and what I have contributed myself;

I have read and understand the penalties associated with Academic Misconduct.

I also confirm that I have obtained informed consent from all people I have involved in the work in this dissertation following the School's ethical guidelines.

Signed:

Date:

Matriculation no:

General Data Protection Regulation Declaration

Under the General Data Protection Regulation (GDPR) (EU) 2016/679, the University cannot disclose your grade to an unauthorised person. However, other students benefit from studying dissertations that have their grades attached.

Please sign your name below one of the options below to state your preference.

The University may make this dissertation, with indicative grade, available to others.

The University may make this dissertation available to others, but the grade may not be disclosed.

The University may not make this dissertation available to others.

Abstract

This dissertation investigates the potential of AI-driven personalised learning paths to enhance the learning experience for undergraduate students within technology courses. The research addresses the main question: ”How can personalised learning paths affect undergraduate student engagement and learning effectiveness in tech courses?”. The study involves the design, development, and evaluation of an AI-driven prototype built as part of Moodle AI mock website. The system, leveraging a local large language mode, Llama3, was engineered using a two-step retrieval augmented generation (RAG) approach to generate learning paths from the module content.

The project successfully demonstrated the technical feasibility of creating a personalised learning system with a local LLM, a solution that significantly mitigates the high computational costs associated with cloud-based models, such as Gemini or ChatGPT. Initial qualitative feedback from a user survey indicated that students perceived personalised learning paths as beneficial, leading to increased engagement and a more efficient learning process.

Despite these successes, the project was subject to several limitations. The use of local LLM constrained the quality and richness of the generated content and its inability to provide a real-time external resource links. The prototype’s content retrieval was limited to text-based PDF files, restricting its ability to create a truly unique learning experience. Furthermore, the small sample size of the user evaluation means that the findings are indicative rather than statistically significant.

In conclusion, this dissertation provides a compelling proof-of-concept for the application of AI in personalised education. The developed web app offers a viable technical framework that can be expanded upon in future work by integrating more powerful models and a wider variety of content types.

Contents

List of Figures

Acknowledgements

I would like to thank my supervisor Yanchao Yu for continuous support and guidance throughout this project and also for taking the time to provide feedback during out-of-office hours.

In addition, I would like to thank all the students who participated in a user survey and who showed interest in this project.

1 Chapter 1. Introduction

1.1 Motivation

The motivation for doing this project comes from personal experience and observation of other students and their feedback. When it comes to learning new programming languages or improving existing skills, it is not always that a student wants to follow the exact structure of the university module. Nowadays, choices are more frequently implemented in modules, and students are given more freedom to choose their preferred approach towards accomplishing coursework in the end.

With this in mind, it was evident that the students complain about not being taught other programming languages and that their grades suffer. This brings to personal experience where there was not enough time to learn my chosen programming language to do the coursework. Therefore, the idea of creating a learning path generator arose, and it was motivating to know that this feature could help a lot of future students not only with learning a new programming language, but also with increasing their engagement in learning and raising their grades.

1.2 Aims & Objectives

The following is a list of aims that will be undertaken in this project:

- Research existing technologies. Find relevant systems that could potentially be used to compare to this project in the evaluation phase.
- Investigate the architecture. Find out how these existing technologies were developed and understand the fundamentals and building blocks that would be useful to use as a guide when developing an AI system in this project.
- Find relevant gaps for improvement. Find out what these existing technologies are missing or an area for improvement, and use it as a starting point.
- Build a prototype of a website with an AI learning path system integration. Build a mock website where the main project idea, an AI system, could be integrated.

- Evaluate the prototype against existing technologies. Perform a comparison of the chosen existing application and compare it with the newly built website with AI integration.

1.3 Research Question

The following research question will be answered in this dissertation:

1. How can personalised learning paths affect undergraduate student engagement and learning effectiveness in tech courses (e.g., software development)?

Furthermore, the sub-questions are:

- How can the benefits differ between theoretical and practical courses within universities' technology curriculum?
- To what extent can the personalised learning path system accurately recommend learning resources and predict student mastery in specific topics, based on course content?

By finding answers to the questions above, we can determine the effectiveness of AI systems and whether it will be useful when integrating it into university Moodle page.

1.4 Scope & Constraints

The scope of the project requires extensive research and understanding of how AI and LLMs work. A significant part of the schedule is dedicated to research, writing a literature review, and exploring existing applications. Approximately half way into the project, a development phase can commence, and all the research findings put to work. Building a website should not be a challenge; however, integrating an AI system will present a slight challenge. The time limit pressure by this time will be high and some sacrifices when building a website might need to be moved to "future work" (e.g., other non-AI features). Finally, at the last stretch an evaluation should take place with comparative analysis as well as user surveys. The hardware used for this project is a laptop and a PC. Therefore, it does not present a problem as the PC is more powerful than a laptop and can be used when performance issues occur.

Some of the constraints include hardware limitations, due to the project being developed on a laptop which is not as powerful as a PC; therefore, making it harder for (offline) LLMs to run. Another constraint is the time given to complete this project. The overall time to complete this project is two and a half month; however, due to research required prior to development, it may not be feasible to fully perfect the prototype as planned on time. Finally, programming knowledge is another constraint that could present a great challenge. With an increasing difficulty in AI implementation, it may require further reading and learning, which could result in short delays in the development schedule.

1.5 Chapter Overview

Chapter 1. Introduction is an introduction to the topic, its aims and objectives, scope, and research questions.

Chapter 2. Literature Review introduces the personalised learning system, its concept, and architecture. In addition, various techniques that can be used in achieved a desired result in personalisation. A discussion of existing applications and core technology that could be potentially used in this project.

Chapter 3. Methodology presents information about the architecture used before actual implementation. Furthermore, what tools and methods will be used. Each will have an explanation of why it was chosen and why it is convenient for this particular project. Finally, this chapter will outline the evaluation methods that will be used and why it will produce the results that will help evaluate this project.

Chapter 4. Implementation introduces the full implementation steps, with an explanation of the tools and techniques used in building this prototype of a website with AI integration.

Chapter 5. Evaluation introduces a mixture of techniques used for evaluating the prototype of a website, and the summary of actual results and user feedback.

The final Chapter 6, Conclusion, presents a summary of the main findings and outcome of this project. In addition, a discussion of results will lead to fulfillment of the main aims. Any limitations that occurred during the development of the project will be presented with potential future work and improvements.

2 Chapter 2. Literature Review

2.1 Introduction

An AI has been around for many decades (founded in 1956); however, only in the last few years we have seen a breakthrough in this field (Haenlein, 2019). From simple task automation to complex large language models to driverless cars. Nowadays, some form of AI is used almost everywhere, in business, in technology, and in daily life. Although this is a broad topic with endless possibilities for discussion, in this literature review, we are shifting the focus to AI in daily life and personalization.

The origins of modern recommender systems date back to the early 1990s when they were mainly applied experimentally to personal email and information filtering (Jannach, n.d.). Ever since then, recommender systems have been attracting more attention and evolve over time. One of the first widely recognized websites to use this feature is Amazon. Over the past two decades, they have been improving the user experience by displaying personalized recommendations based on their browsing activities (Smith, 2017).

Furthermore, with the evolution of AI, in the past few years, recommender systems evolved at much faster pace, enabling it to be modified with new algorithms and methods that enhance user experience (Carole, 2024). This literature review explores different techniques used in recommender systems and how it can be used in education and benefit students by enhancing their learning experience.

2.2 Personalized Learning System

2.2.1 Concept

AI has introduced a new and more efficient way to the learning process. It has become a vital part of the educational system that helps students to learn and achieve higher grades through adaptive learning (Yekollu, 2024). There are multiple ways of achieving a desired recommendation path as shown in the figure 1.
However, the preferred and most popular model for recommendation systems is a hybrid approach. This also applies to personalized learning systems. The hybrid approach combines multiple techniques (e.g. module content, student preferences etc.) that often lead to more accurate, diverse, and relevant results.

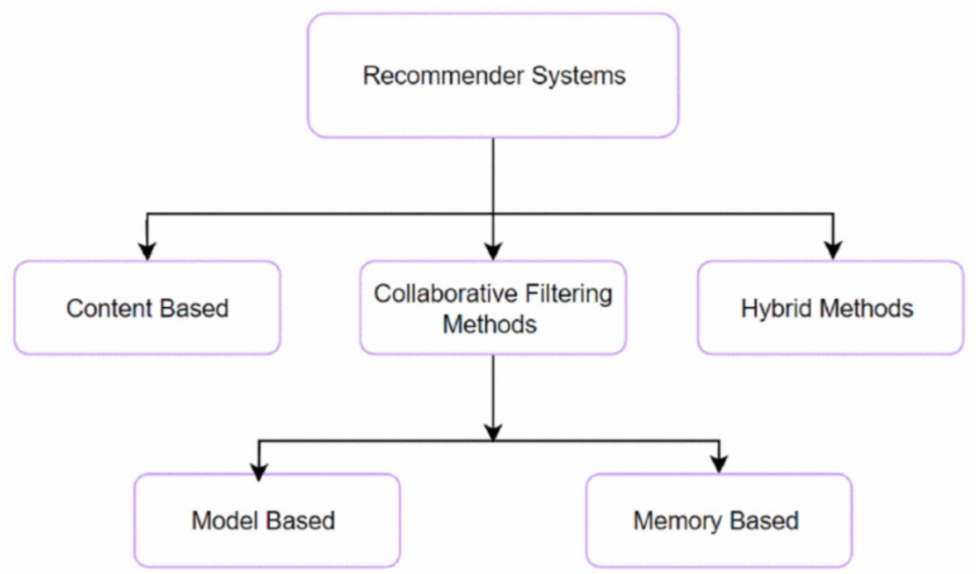


Figure 1: Recommender System Methods (Yekollu, 2024)

There are multiple existing applications that are used for different purposes when it comes to educational learning. With the power of AI/ML, AI tutors have emerged in recent years that are capable of not only generating a learning path, but can teach based on content provided, create exams, provide feedback and grade students. While other, smaller applications focuses on specific areas in education, such as generating module content.

Subsequently, a subfield of ML, deep learning has reshaped the way learning content is generated and personalized. It algorithms can simulate neural networks of human brain and therefore, extract and analyze complex datasets that can be used to tailor educational content (Naseer, 2024).

2.2.2 Architecture

Personalized learning systems aim to address the limitations of one-size-fits-all educational models by tailoring the learning experience to individual student needs, preferences, and abilities. (Stiubiener, 2007) discusses typical e-learning system architecture, and suggests that the personalized learning path should be defined and configured with navigation, adaptation, and updating rules. Therefore, suggesting that a rule-based approach should be implemented as a minimum, which would work as a safeguard to prevent the algorithm from trying to behave in unexpected ways (outside of specified boundaries).

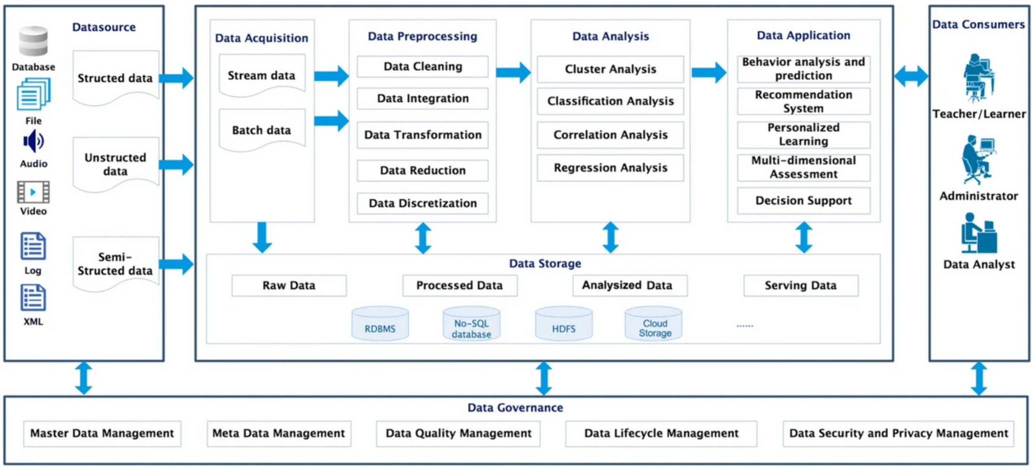


Figure 2: The Architecture of Big Data Based E-Learning Systems (Liu, 2023)

To further expand on the architecture, a personalization process occurs when multiple datasets are fed into the algorithm. Without comprehensive and relevant datasets, the PL system cannot effectively tailor learning experiences. When we talk about learning and education, it is crucial for the algorithm to be able to gather a wide range of information about individual learners and create a ”learner/student model”. Furthermore, multiple datasets with learning content are required to create a ”knowledge base”, allowing the system to understand and categorize available learning resources.

According to (Youssra, 2022) and (Brusilovsky, 2007), data can also be classified into two groups, depending on the nature and form of the information collected.

- Domain-Specific Information
- Domain-Independent Information

DSI contains records about learning activities, level of skills, and level of knowledge while DII is not related to content delivered; therefore, it facilitates the process of personalization, which includes learning styles, motivation, learning goals, background, and preferences.

(Lu, 2024) investigates it further, to integrate DSI into large language models; however, (Lu, 2024) states that it still remains a significant challenge due to the few benchmarks available to measure the effectiveness of LLMs with specific domain knowledge in the context of e-learning; therefore, making it difficult to create and implement a custom model.

So far, we have discussed an example of a hybrid approach to the architecture of the PL system that leverages the strength of each component to create a more robust and effective personalized learning experience. However, with the latest advances in AI, we can create a more dynamic and self-optimizing model, by adding ML to the existing architecture. (Wangmei, 2024) discusses genetic algorithm (optimization algorithm, evolutionary computing) and K-means clustering algorithm to optimize the collaborative filtering approach. They state that 98% of the students believe that through collaborative filtering, personalized recommendation model is more effective; therefore, improving students' learning interest, efficiency, and suggesting more accurate PL resources.

In this section, we have covered the PL architecture, from simple rule-based to more complex ML model. In the real world, there are many variations of the PL architecture. Each of them, can be built differently, depending on company or educational institution needs and requirements. We have discussed the foundational steps needed to build the personalized learning system, which can be used and enhanced in the future with AI advancements.

2.3 Techniques

Personalized learning systems heavily rely on AI and ML techniques to achieve their adaptive capabilities. The most common approach, as discussed earlier, is a *hybrid* model, which combines various techniques to overcome certain limitations and provide a more efficient adaptation.

2.3.1 Knowledge Tracing

Knowledge Tracing (KT) is a data mining technique that is used to track and predict students' evolving comprehension during the learning process (Shaka, 2023). KT technique offers valuable information student progress by analyzing their performance over time, as shown in Figure 3. This is a fundamental ML technique with a range of algorithms, such as traditional

Bayesian Knowledge Tracing (BKT) to more advanced algorithms, such as Deep Knowledge Tracing (DKT), see Figure 4. The latter is a deep learning

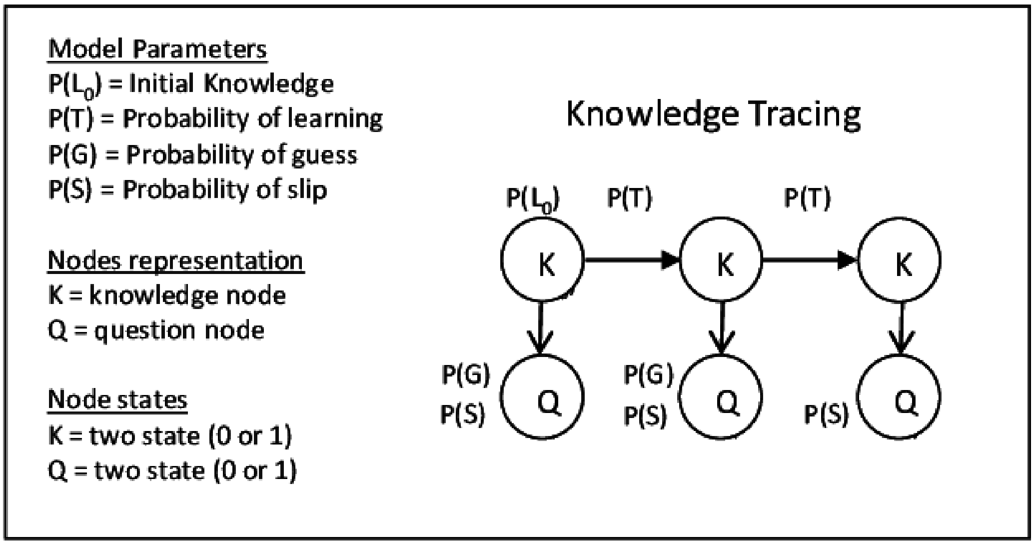


Figure 3: Standard Knowledge Tracing Model (Pardos, n.d.)

model that uses recurrent neural networks (Shaka, 2023).

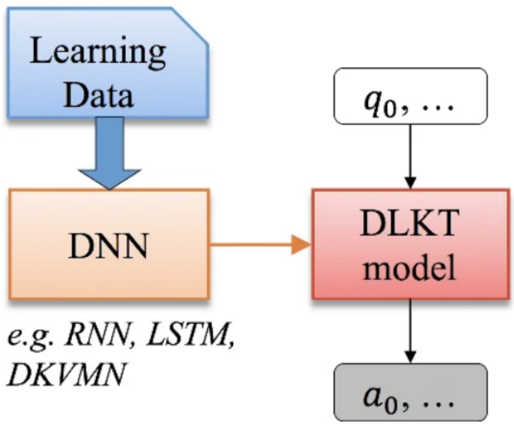


Figure 4: Deep Learning based KT framework (Shaka, 2023)

2.3.2 Content Based Filtering

Content-Based Filtering (CBF) provides recommendations to users based on the similarity of items that have been recommended to previous users to produce appropriate recommendations where items can be represented by applying feature extraction to represent them as vectors (Julianti, 2022). To further expand on this, the algorithm extracts relevant features or attributes from an "item", and therefore converts it to a numerical format as mentioned previously, vectors. Thereafter, a "user profile" is created based on the items with which a user interacted previously. In education, a user is a student that interacts with specific "items" when searching for something (e.g. practical content that was helpful in overcoming the problem). By accessing this content by many students, the algorithm would find similarities and treat this as a "helpful" solution that can be suggested to another student with the same problem.

However, we argue that clicking on the content provided by the school or university for a module might not always be beneficial. At times, the algorithm would suggest content that is clicked by many, but it does not solve the problem; therefore, it does quite the opposite, it can introduce even more problems or confusion. (B.Thorat, 2015) discusses this issue and agrees that the CBF system is anticipated to increase diversity, because they help to discover new products; however, some algorithms may accidentally do the opposite. In order to have useful and valuable suggestions, we suggest adjusting CBF to take into consideration time spent by the user on a specific content, scroll depth, interactions with the content. By implementing new features, the CBF would dive deeper into engagement metrics and separate valuable content from the rest.

2.3.3 Collaborative Filtering

Although this technique is more related to the personal recommender, it can also be used for personalized learning recommendations as well. This approach is similar to content-based filtering; the only key difference is that the algorithm is trying to identify patterns in the learning behaviors of several (similar) students. The fundamental idea is that if users x and y rate two items similarly or have similar behaviors, they will act on other items similarly (Chatti, 2013). For instance, if a group of students indicated that they struggled with something specific and that there was a particular exercise that was highly effective for them to overcome the problem, the system might recommend the same exercise to new students with similar struggles.

Because CF has no special requirements, it is easier for the algorithm to find users with similar interests, and therefore; this is a perfect approach for many commercial applications (such as Amazon, Netflix, Spotify, YouTube, and more) (zhihua, 2020). These applications have very large datasets and,

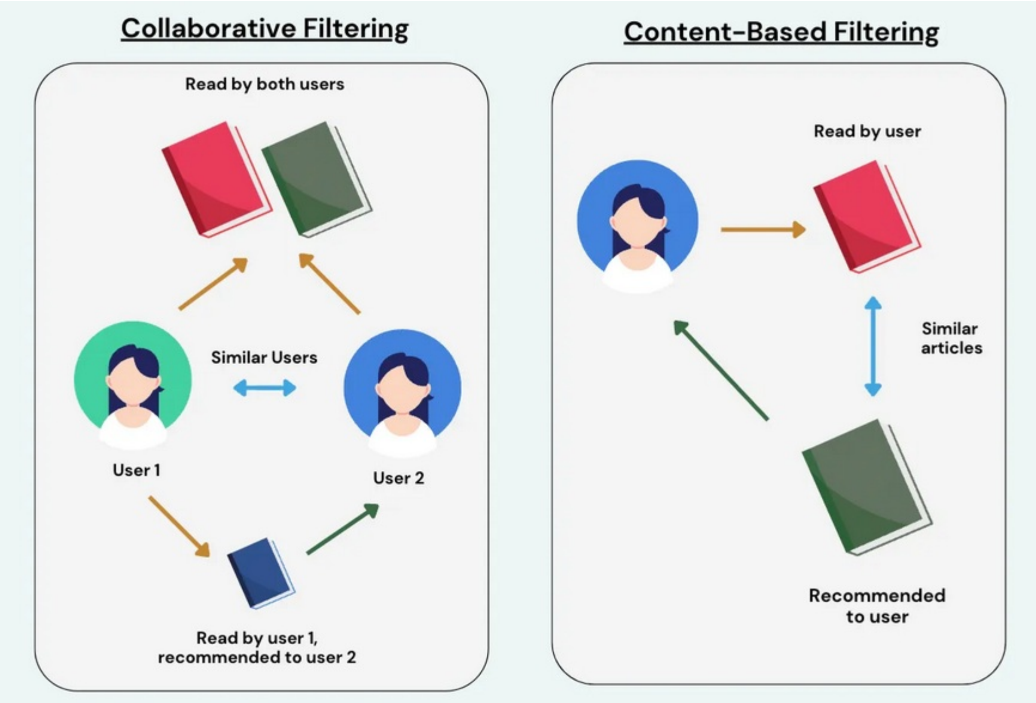


Figure 5: Filtering Models (Subramanian, n.d.)

therefore, it can be hard to extract information and find similarities by using traditional CF models. Moreover, (zhihua, 2020) explores various approaches to CF and suggests that the machine learning approach is the most common these days, especially with recent advances in the artificial intelligence sector.

Finally, this approach may not be suitable for educational purposes, mainly due to the special requirements that are needed to suggest the right content to students. Therefore, we suggest considering a CBF approach when creating a personalized learning system in educational institutions. The comparison of both CBF and CF can be seen in Figure 5.

2.3.4 Deep Learning

A subfield of machine learning plays a crucial role in creating more sophisticated personalization. With the use of neural networks (Figure 7), complex educational datasets can be analyzed and more accurate pathways can be developed for student learning progression. Models such as deep neural networks (DNNs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs) can automatically extract features from the data by learning from students' historical data, such as learning behaviors, grades, and preferences, and perform more complex recommendation calculations (Song, 2024).

Moreover, Natural Learning Processing (NLP) models are often built with

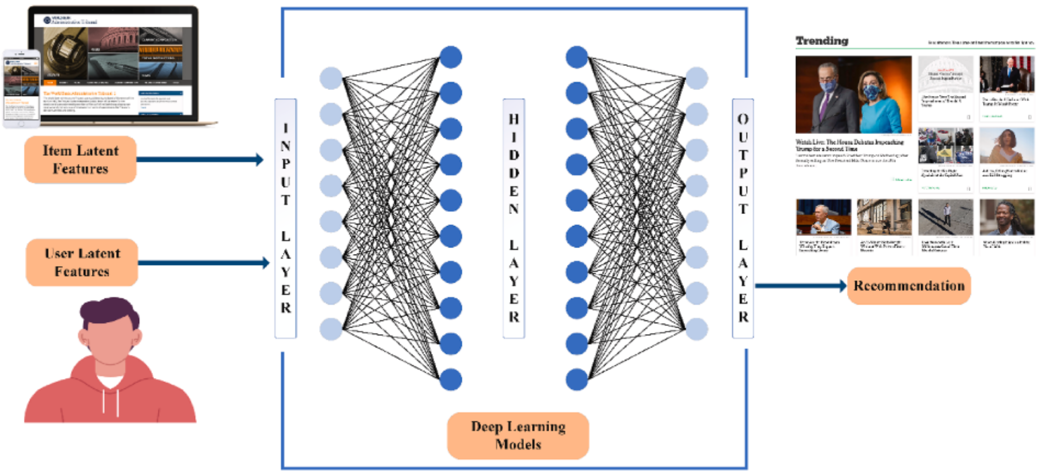


Figure 6: DL based solution to news RS (Talha, 2023)

deep learning architecture; therefore, enabling the AI tutor to understand student queries and provide a textual answer. Furthermore, for more tailored learning generation, GenAI can be used as part of Personalized Learning System to generate a variety of new content. (Feuerriegel, 2024). With DL and GenAI, personalized learning system can be taken to the next level and suggest even more accurate and relevant results. Although this approach is more complex than CBF with ML, we strongly suggest exploring it, which can make a PL system more advanced, especially with the use of neural networks (NN - see Figure 6).

2.3.5 Rule Based System

Nowadays, many new systems utilize the power of self-evolving AI; however, in educational institution, there are certain rules that AI must adhere to (e.g., prerequisites). This is where rule-based implementation is useful. This approach ensures that AI-generated learning paths do not skip any curriculum requirements and are according to standards. Essentially, the rule-based model implements safeguards that the AI algorithm must follow.

(Raj, 2019) discusses the implementation of a rule-based expert system that recommends content to the learner, where the learner is modeled using a

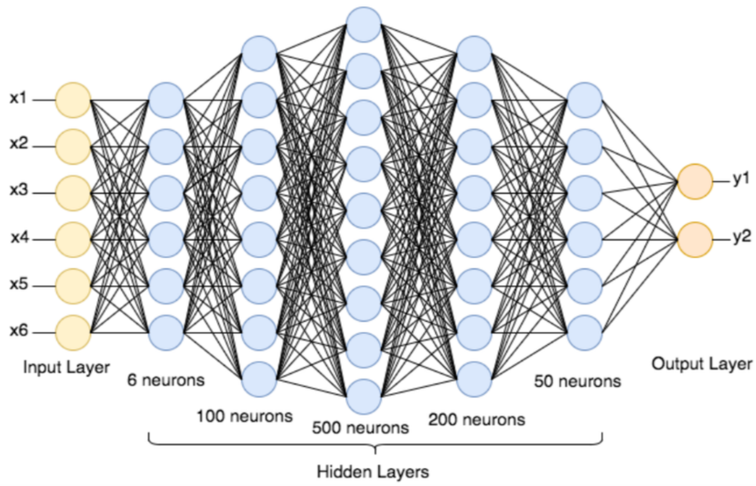


Figure 7: Deep Neural Network Architecture (Bahi, 2018)

probabilistic learning style model. Initially, the system analyzes different learning styles and a model is created based on probabilistic nature. Thereafter, a rule-set is defined using this knowledge, which maps a new learning style for the learner. Finally, another rule-based system ranks the learning objects that corresponds to the learning style created previously. Based on the structure of this proposed structure, we can determine that the rule-based system is effective when the learner content and other material is known and analyzed (probabilities assigned and the learner model created) prior to any adaptation attempts. Another approach taken by (Pelánek, 2024) suggests a less optimized framework, which ensures that all important aspects are taken into account, as seen in Figure 8. The latter framework is based on IF-ELSE rule-set, which can be easily interpretable. (Pelánek, 2024) discusses large scale learning environments, and that this simple framework could be effective in recommending learning content in larger environments. Each recommendation is a consequence of a specific interpretable rule.

Although there is no evidence of ML being used in both frameworks discussed in this section, it is clear to say that the rule-based approach can be beneficial for small and larger learning environments. These rule-based systems are highly customisable and can fit within any educational institution. However; we argue that rule-based systems can be challenging and with serious limitations. Data sets that should be analyzed to create the learners profile are needed. This can introduce a challenge if this is done without machine learning.
Furthermore, while a rule-based system is customisable, based on the latter

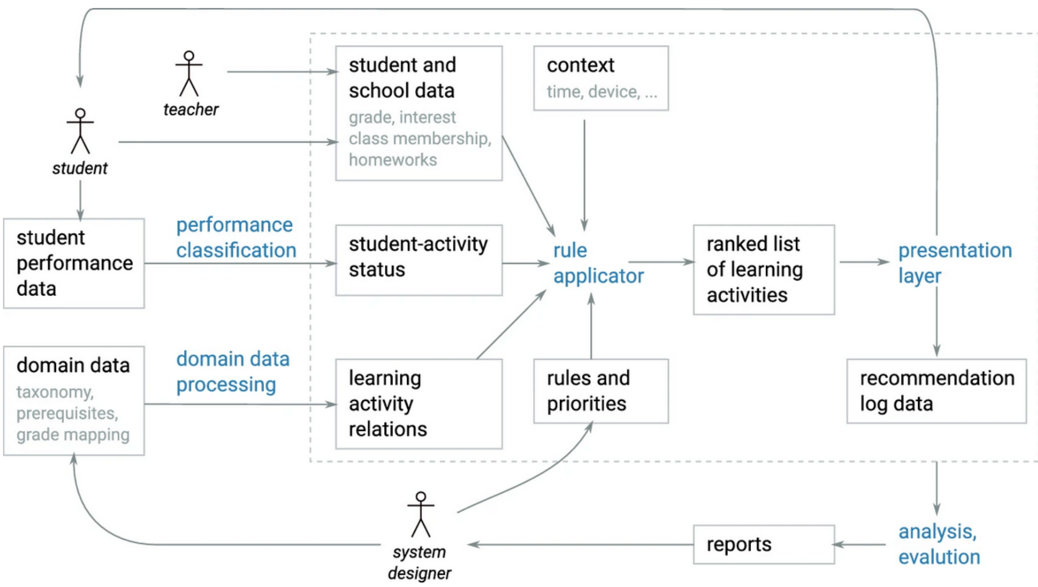


Figure 8: Proposed Rule-Based Framework by (Pelánek, 2024)

framework discussed, we can determine that a manual approach is required to build the system or add new rules to it. Finally, certain limitations, such as not expanding the rule set over time, can introduce limited recommendation options and reduce the adaptability of the system. In conclusion, based on the frameworks (Pelánek, 2024) (Raj, 2019) discussed in this section, we can determine that a rule-based approach without ML can only be effective in smaller environments, rather than larger.

2.4 Existing Applications

2.4.1 Adaptemy

Adaptemy was developed by *Adaptemy Limited*, an Irish company founded in 2013 by Conor O'Sullivan (CEO) and Ciarán Cannon (CTO).

Their goal is to provide AI-powered adaptive learning solutions that transform educational content into personalized learning experiences. Furthermore, they aim to make learning more efficient and effective by tailoring it to individual student needs.

Adaptemy's primary domain is adaptive learning technology within the education sector. It is a platform that provides adaptive learning solutions to higher education institutions (see Figure 9). It offers an AI-powered adaptive learning engine that aims to transform traditional educational content to personalized learning experiences. Adaptemy's AI Engine is developed based on existing research in the areas of Intelligent Tutoring Systems and Adaptive E-Learning. It uses a curriculum model, a content model, and a learner model (I. Ghergulescu, 2021).

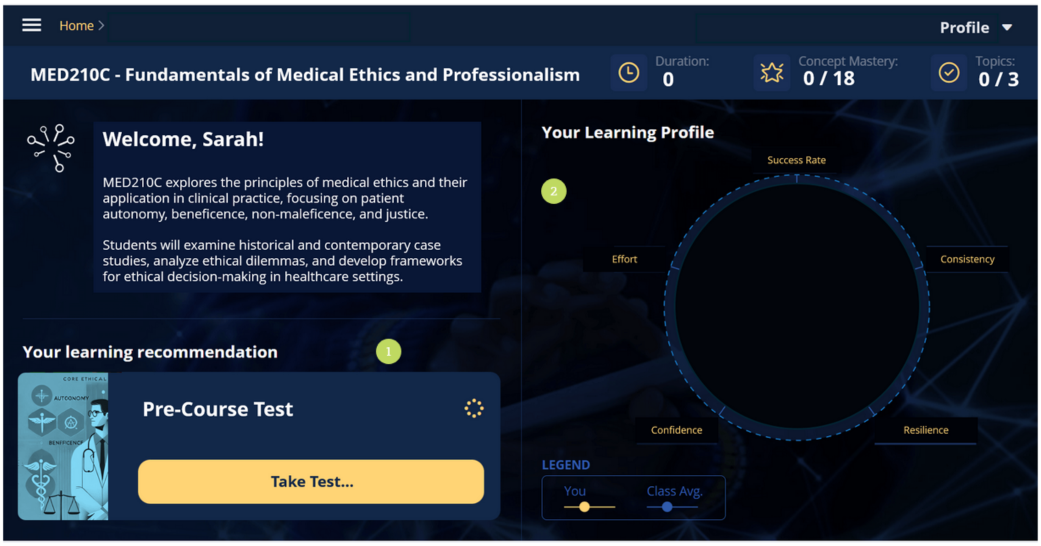


Figure 9: Student Dashboard. (Source: *www.adaptemy.com*)

The Curriculum Model (see Figure 10) contains relationships and concepts of the three models mentioned previously. It has multiple layers of adaptation and personalisation as well as misconception detection. Adaptemy engine treats the curriculum model as a "knowledge domain" which consists of two networks:

- Hierarchy Network. Structures curriculum map in a way that is easier to navigate.
- Prerequisite Network. It defines the relationships between the "knowledge" items.

Both models contain information about the relationships. This rich data indicate the strength of the link and allow for misconception detections, a complex learner model, information propagation, and for multiple layers of personalization and adaptation (D. I. Ghergulescu, n.d.).

To improve the propagation settings, Adaptemy's engine needs to update

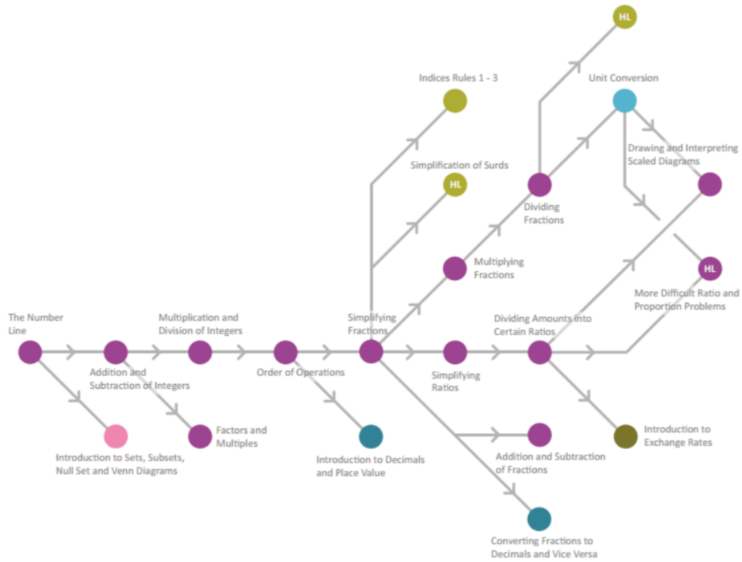


Figure 10: Curriculum Model. (Source: *www.adaptemy.com*)

its curriculum model (see Figure 11). Through Bayesian Networks, the engine improves the existing prerequisite network based on learning data. In the first stage, it performs a data-driven network learning. Subsequently, it also performs a shape-based network optimization. The optimization is based on the learning dataset that improves the accuracy of the curriculum model. Furthermore, data-driven network algorithm uses a hybrid approach for network structure and parameter learning, resulting in an improved prerequisite network structure.

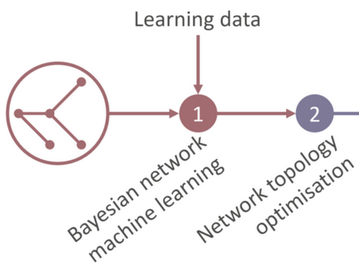


Figure 11: Curriculum Update. (Source: *www.adaptemy.com*)

Although Adaptemy's learning platform is a great application for teaching and learning, it also has some downsides. For instance, in order to provide truly personalized learning paths, there has to be a vast amount of diverse

content available. Without it, the curriculum model will not produce the best results when it comes to personalization.
Furthermore, the content created must be of good quality and very accurate; otherwise, curriculum mapping will not be as effective, which can result in the adaptive system not performing optimally.

2.4.2 MoodleLMS

MoodleLMS was developed by Martin Dougiamas, with the first version of Moodle released in 2002.

It is a globally recognized and widely adopted open source learning management system. Moodle provides a flexible, secure and robust platform for creating personalized online learning environments (see Figure 12). Many universities across the globe use Moodle. The platform is highly customisable, and can be personalized according to educational institution needs.

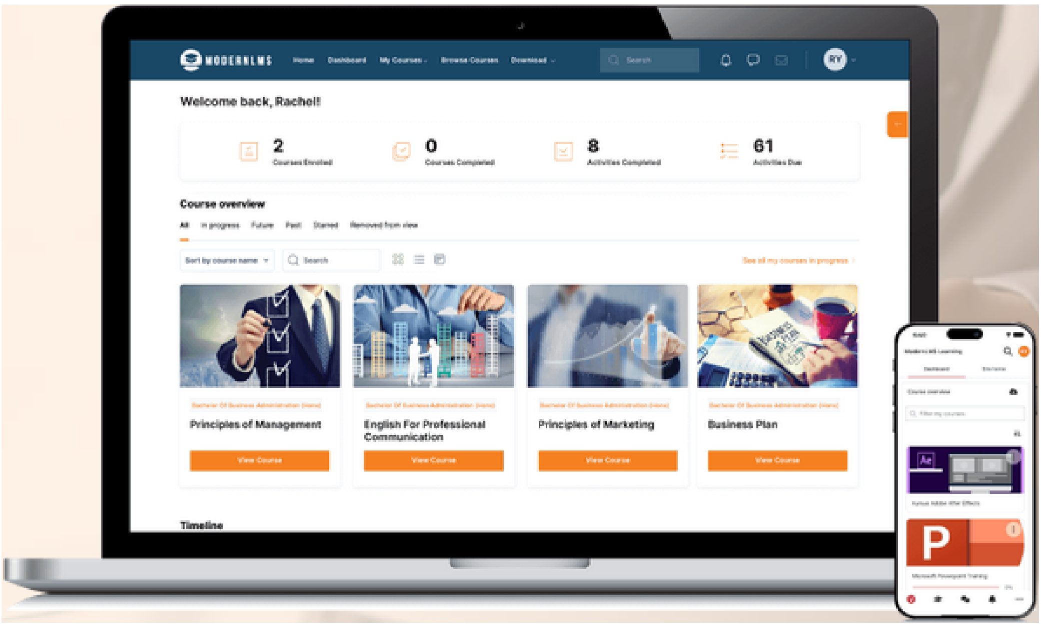


Figure 12: MoodleLMS. (Source: *www.moodle.com*)

Key functionalities of MoodleLMS include:

- Course Management. Educators can create and structure online courses, upload content and various resources, and also add interactive learning activities (e.g. assignments, quizzes, surveys).

- User Management. Admin function has full control of the website. Can switch between editing and non-editing teacher. Enrollement management.
- Assessments and Grading. Automated grading, can also choose manual grading.
- Logs and Reports. Standout feature, Moodle supports many third-party reporting plugins that help to produce various reports and logs, such as student performance, engagement, completion.
- Highly Customisable. Because it is an open source application, it can be customized in any way that suits the user.

While MoodleLMS is advanced and great application, it refrains from using AI and adopts a human-centered approach. It uses a three-layered architecture (web server, database, and Moodle data Folder), and it has over 1900 ready-to-use plugins available. According to their website (www.moodle.com), Moodle is built on Apache, MySQL, PHP stack, and has a comprehensive API for developers.

2.5 Core Technology

2.5.1 Large Language Models (LLM)

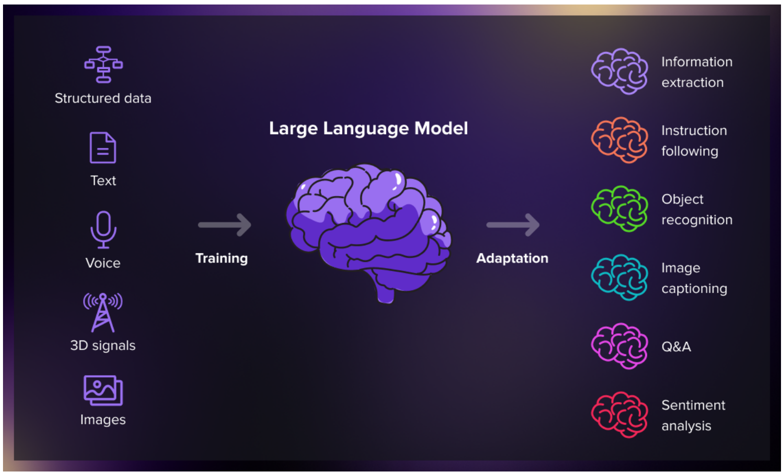


Figure 13: Large Language Model (LLM). (*Image Credit to RandomResearchAI*)

Large Language Models (Figure 13) are profoundly transforming personalised learning path generation by enabling highly dynamic, interactive, and contextually rich educational experiences. As (Ng, 2024) mentions, this personalized approach not only helps with a deeper understanding and retention of knowledge, but also increases student engagement and motivation, which are essential for successful outcomes in education. Some of the more capable LLMs that are used in various domains, including education, are GPT-4 and Llama 2-70B.

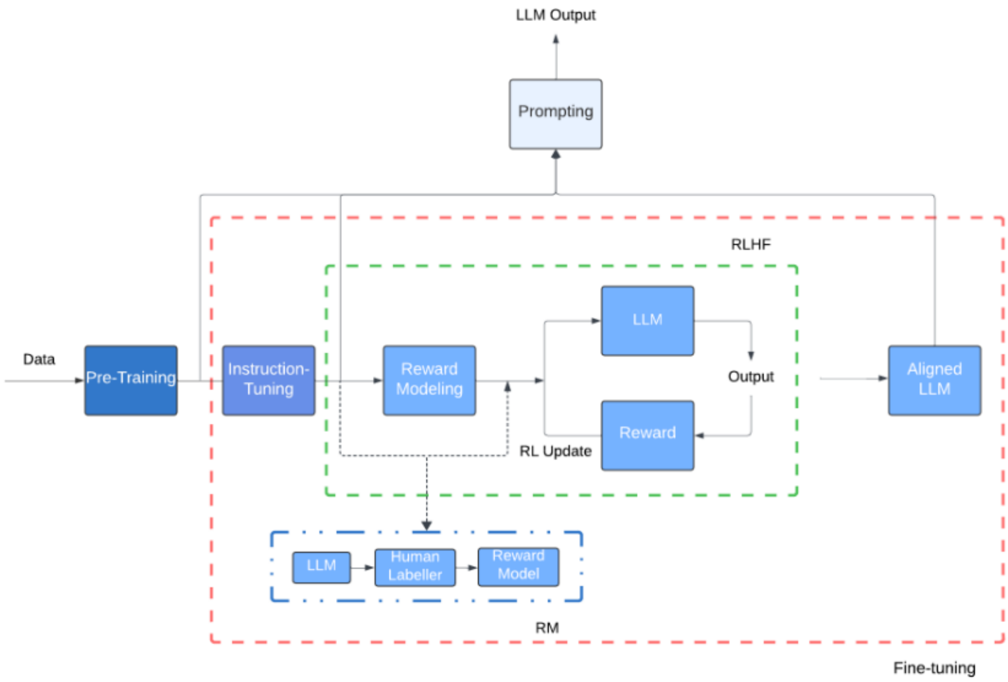


Figure 14: Various Stages of LLMs from pre-training to prompting (Naveed, 2024).

Although it is a powerful tool that a student can utilize, it comes with certain limitations and challenges. (Razafinirina, 2024) discusses these challenges by exploring foundational functionalities and argues that LLMs require extensive computational resources; therefore, this issue is particularly evident in educational contexts where high-performance computing might be restricted. Various LLM models (see Figure 14) often have millions or even billions of parameters that can overwhelm traditional computing systems.

To address these issues, (Razafinirina, 2024) suggests fine-tuning models for

education. One approach is curriculum-based fine-tuning, which integrates a learning progression into the fine-tuning procedure; therefore, exposing LLM to assignments that are increasingly intricate. This approach aims to enhance the LLMs ability to learn and understand diverse assignments incrementally, making it a more viable model to incorporate within the educational institutions.
Another approach eliminates the need for pre-labeled data and introduces instruction-based fine-tuning LLM model. Using large datasets (with labeled instances) can be time-intensive and come at a high cost. However, feeding in the precise instructions will remove the need of pre-labeled data and therefore, improve user experience and interpretability within the educational institutions.

With the right skills and techniques, LLM models can be customized according to educational needs and therefore, make them a valuable tool that students can use to learn or accomplish specific tasks. We have explored the issues educational institutions face when using LLMs and investigated instruction-based and curriculum-based fine-tuning approaches, which can be useful for both educators and students in educational institutions.

2.5.2 Information Retrieval

Information Retrieval (IR) is the process of obtaining information from a collection of resources, although it does not explicitly return the exact information, but anything relevant to the request by the user. The best example of IR are web search engines such as Google or Yahoo. Retrieving unstructured documents in response to a natural language query is the core task in information retrieval (Zamani, 2018).
Information Retrieval system can store, represent, access, and organize the information items. There are several types of IR as seen in Figure 15.

- *Adversarial IR.* The main focus is on managing and mitigating data that has been maliciously manipulated. For instance, this occurs when a user intentionally tries to spam their content into the search engine so that their ranking would be higher than it should be (Farooq, n.d.).

- *Automatic Summarization.* This is a critical task in natural language processing (NLP). It aims to make longer text into shorter versions while preserving the most important information.

- *Multi-Document Summarization.* It is similar to latter type; however, this focuses on information overload with multiple text documents.

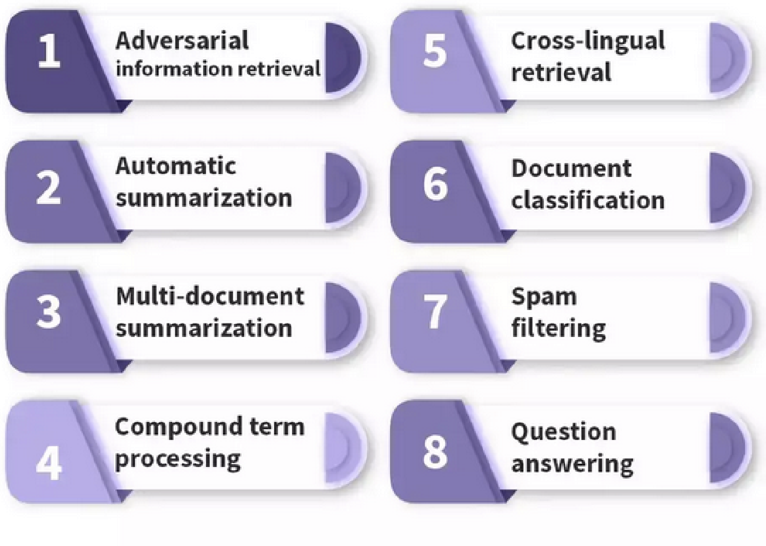


Figure 15: Types of Information Retrieval. (Source: *www.engati.com*)

The aim of MDS is to extract and summarize the most important information from multiple document, eliminate redundancy and maintain document consistency.

- *Compound Term Processing.* This technique aims to handle muti-word terms as a single unit, rather than multiple separate words. It identifies, analyzes, and interprets multiple words into one.
- *Cross-Lingual Retrieval.* This approach addresses the challenges of retrieving information from a source when a user writes a query in a different language (Yu, 2021).
- *Document Classification.* A core task of IR and NLP. It assigns labels or predefined categories to documents based on their content. This approach is excellent for unstructured text data, which can be organized and made available for more efficient information management, retrieval, and analysis.
- *Spam Filtering.* A simple automated tool that identifies and filters out unwanted spam from legitimate messages. It is a specific technique that aims to separate relevant and irrelevant information.
- *Question Answering.* This is an advanced IR type that focuses on providing precise answers to natural language questions. According

to (Abbasiantaeb, 2021), the early days of QA were rule-based and therefore limited to fewer answers. However, QA was revolutionized by Deep Learning (DL), which enabled the AI to learn and adapt using neural networks, resulting in endless answers and conversations with the natural language user.

Information Retrieval approach is evolving on a yearly basis, from rule-based to neural networks and LLMs. Today, in education it is very easy and simple for a student or educator to ask and retrieve specific information.
Furthermore, IR is a broad field that has many different approaches and techniques. We will highlight some of the components and features that IR possesses as seen in Figure 16.

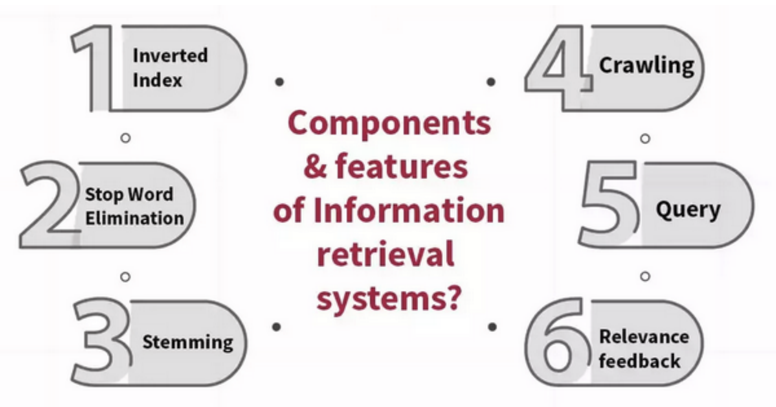


Figure 16: IR components. (Source: *www.engati.com*)

- *Inverted Index.* Primary data structure of most IR systems. It is a data structure for every word and document.
- *Stop Word Elimination.* These are the words that are not useful for searching. Such as letter "a" in "a word", the letter would be eliminated to make the search faster and easier. However, this feature is not always beneficial, as some words require to have a specific letter or word with it, and if the search engine eliminates it, this could make the search produce inaccurate results.
- *Stemming.* The feature extracts the base form of words by removing the ending of a word.

- *Crawling.* The aim of crawling is to gather relevant web pages as efficiently and as quickly as possible, whilst linking them together.
- *Query.* A simple query such as user input that provides instructions to the search engine.
- *Relevance Feedback.* Gathers user feedback based on the initial return of results from a query and determines whether the results were relevant or not.

In this section, we have described each component and feature of the information retrieval approach. Although this is a broad subject, multiple techniques can be customized and adapted to the design of a learning path system for educational institutions.

2.5.3 Neural Ranker

Neural Ranker (NR) is a technique within the field of Information Retrieval (IR). This modern approach ranks search results based on relevancy and the order in which they should appear. Furthermore, unlike previously hand-crafted rule-sets and manual approaches, NR now utilises the power of deep learning (DL) and neural networks (NN) (Zamani, 2018).

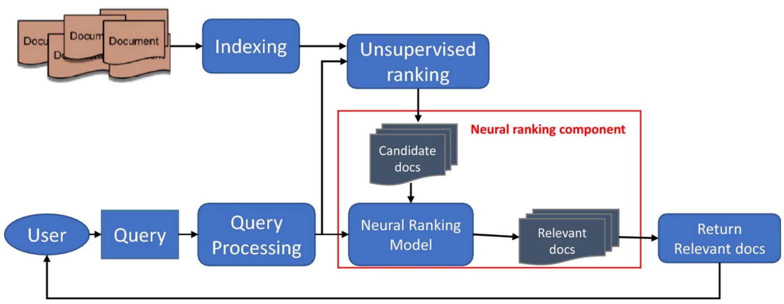


Figure 17: Neural Ranking Model for Document Retrieval (Trabelsi, 2021)

Many neural ranking models (see Figure 17) have complex architectures, therefore computing the query-document relevance score using the neural ranking model for every document in the initial large collection of documents leads to a significant increase in the latency for obtaining a ranked list of documents from the user's side (Trabelsi, 2021).

A document retrieval architecture in Figure 17, uses two types of rankings. Initially, it uses unsupervised ranking which takes in users query as well as initial set of indexed documents. This stage of ranking gathers all the possibly relevant documents (there will be a mixture of relevant and irrelevant documents at this stage). Furthermore, the unsupervised ranker then passes on documents to the next stage, "re-ranking" which, is essentially a neural ranking model where neural networks take part in separating irrelevant document from the ones that user requested. Subsequently, all relevant documents are returned and displayed to the user.

In education, neural rankers are highly relevant and are increasingly being used, especially in personalized learning. Neural networks can help a student create and personalize their learning path while maintaining consistency and displaying relevant content through the ranking stages. Moreover, with a pool of available learning resources, neural ranker can optimize and recommend sequential learning paths. Neural Ranker can be used with a variety of recommender system techniques, such as collaborative filtering (as discussed in the previous section) or Recurrent Neural Networks (RNN).
In conclusion, neural rankers in education and personalized learning go beyond the simple recommendation system, they aim to find the optimal sequence to maximize learning outcomes for each learner.

2.5.4 Neural Network

Neural Network (NN) is a type of machine learning, which is inspired by the function of a human brain. It consists of multiple nodes (input, hidden, output) and the connections between the nodes that are adjusted during learning/training process (see Figure 18).

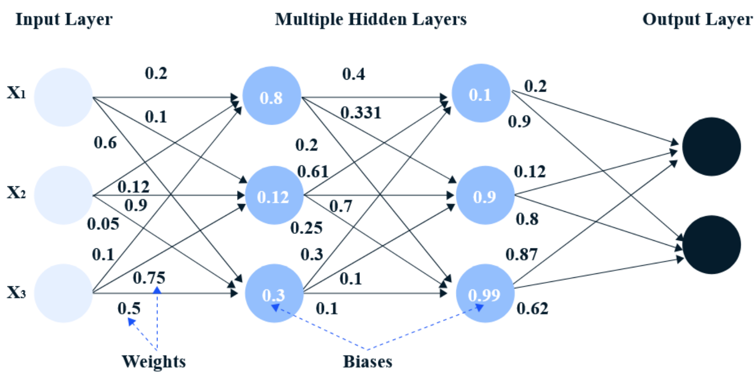


Figure 18: Neural Network (Takyar, 2023)

AI and NN (Figure 18) have been with us for decades; however, it has only recently evolved and started to gain recognition. In the education sector, this is still in an experimental stage; thus far, it is slowly growing. Neural networks are not easily implemented and can be a challenge when it comes to understanding the integration of NN in the education sector. Some of the challenges that educational institutions face are:

- Many educators lack knowledge about AI and how to use it effectively. There are plenty of educators who are ”old school” and therefore, they do not include AI practice in their daily work routine.

- Data privacy and ethical concerns. Bigger educational institutions have more diverse students; therefore, it is a challenge to adjust AI algorithms correctly and avoid any bias or other situation where a student would be put at a disadvantage.

- Resources required to run powerful AI systems. Some AI with neural network require more expensive and bigger hardware. This poses a challenge for an educational institution due to the lack of funding or affordability.

Although these challenges are temporary and will be solved over time, on a brighter side, there are multiple benefits that can be achieved with the use of AI and NN. One of the benefits is training a neural network to understand student learning behavior, keep track of their progress, and suggest learning paths.
One of the recommended approaches to learning path generation is Reinforcement Learning (RL). (Sakar, 2021) proposes this approach with the Generative Adversarial Network (GAN), which is used for rapid adaptation during realistic simulations of the students performance.

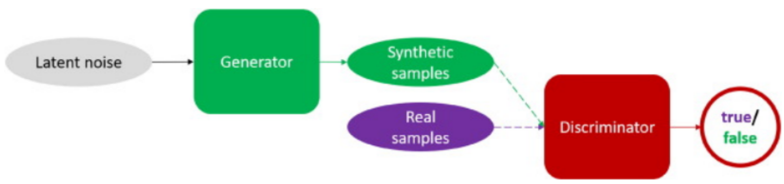


Figure 19: GAN architecture (Sakar, 2021)

In the Figure 19, two neural networks (generator and discriminator) are trained. The discriminator is trained first with both real (label ”True”) and fake (label ”False”) samples of data. It helps the discriminator to distinguish fake from real data. Subsequently, a discriminator is trained again, but on the fake samples from the generator. In this step, the weights and biases are updated after the gradient of the error is back-propagated back to the generator (Khayatian, 2021). Finally, these two steps are repeated sequentially. The training continues until a balance is reached. The generator becomes adept at creating highly realistic data, and the discriminator can no longer reliably tell the difference between real and fake samples.

We have discussed a single approach that can be used in a learning path generation system. By using GANs, we can generate a lot more learning paths whilst dynamically adapting to student progress or struggles. In addition, with reinforcement learning (RL) integration, the agent can receive rewards, which can lead to better learning outcomes and engagement. Finally, this approach can produce a highly personalized, adaptive, and engaging student learning experiences.

2.5.5 Educational Data Mining

Educational Data Mining has become an effective tool for exploring hidden relationships in educational data and predicting students' academic achievements (Yağcı, 2022). With the use of machine learning, data mining has become a great way to discover new potentials and patterns that can improve the learning path of students. Furthermore, this approach is beneficial not only for students but also for the whole educational institution. Any university or college is able to data mine specific information about their students and, therefore, improve student experience on campus, for instance.

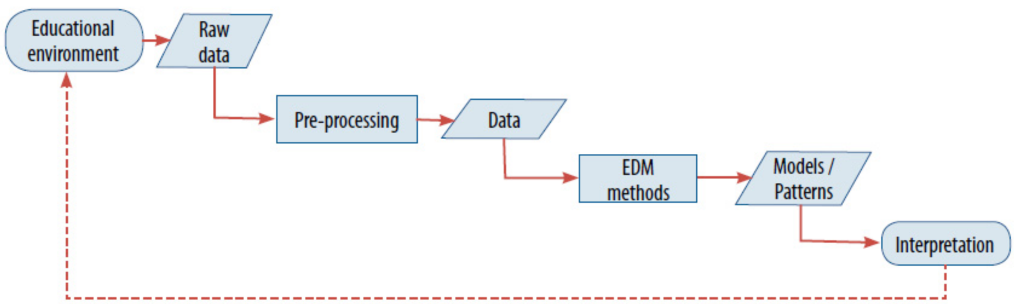


Figure 20: Overview of EDM methods (Liñán, 2015)

In Figure 20 (Liñán, 2015) discusses how the EDM methods are applied.

The author states that the EDM model can be grouped into a few different categories:

1. Student modeling, where student data can be used alongside EDM methods in order to design a better customized learning path based on differences between students.

2. Pedagogical support, where efficient educational support could be identified.

3. Scientific research, where certain applications may help to build and therefore test educational theories and formulate new hypothesis.

Based on previous research, it is evident that most Learning Management Systems (LMSs) use their own tools to automatically data mine and gather information whilst generating statistical reports of course development; however, the tools they use are often basic. For instance, Moodle (we discussed about this LMS in the existing work section) can generate several types of reports, such as, logs for students and activities, recent activity, data on activity completion, and time taken.

(Yağcı, 2022) proposes machine learning approach to identify key factors affecting certain areas of education. This includes how to improve the academic performance of an institution and what is affecting student grades. In fact, AI, machine learning, and neural networks can significantly improve data mining techniques and become a more robust tool for identifying specific issues or relationships within the educational institution.

Neural Networks (NN - Figure 21) is a technique that most new applications nowadays use. It makes the application more efficient, faster, and automated. (Okewu, 2021) argues that despite the capability of Artificial Neural Network (ANN) based EDM for efficiently classifying student learning behavior and predicting their performance, the concept of ANN in higher education has certain gaps and challenges that can prevent ANN from performing efficiently.
Based on recent studies, we agree that certain gaps in this field can disrupt EDM performance and produce less accurate results. However, with the speed of evolving AI in this decade, it is possible that specific gaps and challenges will be overcome, with AI and machine learning finally being able to effortlessly mine data and produce very accurate statistics on students and educational institutions.

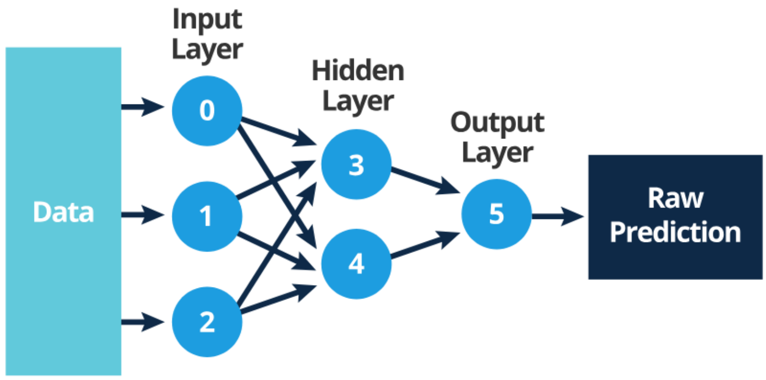


Figure 21: Neural Network in Data Mining. (Source: *www.smartsheet.com/neural-network-applications*)

2.6 Conclusion

In this chapter, we have discussed a variety of techniques for personalisation in both educational and shopping systems. From basic knowledge tracing in education to more sophisticated deep learning techniques in shopping systems such as Amazon. Content-based filtering and collaborative filtering are both widely used techniques (more in shopping systems than in educational systems), although it is highly recommended to use this technique in combination with other techniques such as neural networks to make the system more sophisticated, but this requires a bigger AI implementation rather than a small feature.

Furthermore, we have investigated existing applications such as Adaptemy, and MoodleLMS. Both are Learning Management Systems (LMS), meaning that most of the website is managed by AI, such as generating course content and being an AI tutor. These applications are larger scale than this project and therefore a less sophisticated application was investigated, Moodle. In particular Edinburgh Napier University Moodle page. Moodle is part of MoodleLMS, but without the AI tutors (it is custom tailored to university's needs). This application is a perfect website that can be improved and later compared to this project.

Finally, we have discussed the core technology behind the personalised learning systems and what could potentially be used within educational institutions. The main focus was on large language models (LLMs), as it is very useful, since it can generate content and have billions of parameters (when used online, Gemini/ChatGPT/Deepseek). Moreover, we have explored other technologies, such as educational data mining, which is an emerging discipline. The main purpose for it is to mine and analyse data collected as the teaching goes on. Although this technique is not part of learning path generation, it can certainly help with gathering data and providing statistics in order to find out if learning path generator feature has improved student performance and their grades.

3 Chapter 3. Methodology

3.1 Introduction

This chapter outlines the methodological approach to developing the Moodle AI website prototype. It discusses various methods and approaches used in developing the architecture, in particular, the client side, the server side, and the database layers. Furthermore, this chapter explains the reason for choosing these specific techniques and why they are beneficial to the Moodle AI website.

This chapter is divided into three main sections. The first section details the development architecture, explaining the rationale behind the selection of a client-side model and a three-tier architectural pattern. The second section provides a comprehensive overview of the selected tools and models, justifying the choice of each component in the technology stack and how it contributed to the project's objectives. The final section discusses the evaluation methods and explains why these methods are beneficial for evaluating the Moodle AI prototype.

3.2 Development Architecture

Moodle AI uses a three-tier architecture as the foundational model for the system. This model divides the application into three distinctive layers: the presentation layer, the logic layer, and the data layer.

1. **The presentation layer (client-side)**. This layer is responsible for the user interface and all interactive elements that a user can see and interact with. The main role of this layer is to capture user input and render data, delegating all data processing and business logic to the logic layer. This decoupling ensures that changes to the user interface can be made with minimal impact on the underlying system.

2. **The logic layer (server-side)**. This is a backend layer, which consists of business logic of the the website. It was implemented as a RESTful API using the Flask framework. The server is stateless, which means that it does not retain any client-specific session data between requests. Its responsibilities include user authentication, data retrieval from the database, file management, and control of AI to generate learning paths.

3. **The data layer (database)**. This layer is responsible for persistent storage and management of all application data. A MySQL relational database was chosen to accomplish these tasks. It stores user information, course and module structures, paths to uploaded files (PDF), and personalised learning paths after it is generated for a student.

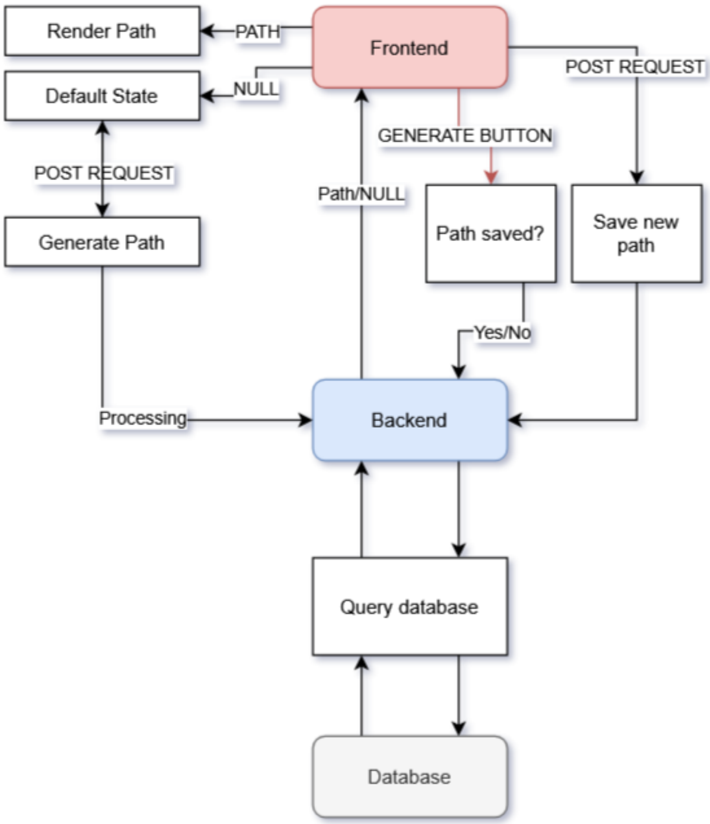


Figure 22: Data flow from frontend to backend to database

This architectural choice provides significant advantages. It allows for independent development and deployment on the client and server, facilitates scalability by enabling each layer to be scaled separately, and enhances security by creating a barrier between the UI and the database. Figure 22 provides a simple visual information of the data flow between the frontend, the backend, and the database.

3.3 Selected Tools/Models

3.3.1 Frontend: React.js

React.js was chosen as the JavaScript library for building the user interface. Its component-based architecture was highly beneficial, allowing for creation of reusable UI elements (such as widgets and accordions). This modularity streamlined the development process and improved code maintainability. Furthermore, React's use of virtual DOM ensures efficient updates and a highly responsive user experience, which is critical for a modern web application. The *react-router-dom* library was used to implement client-side routing, allowing seamless navigation.

3.3.2 Backend: Flask

A micro-web framework for Python was selected for the backend API development. Its lightweight nature provides flexibility required to build a custom application such as Moodle AI website. Python was selected as the main programming language for development due to its extensive ecosystem of libraries, in particular for data processing and AI. In the Moodle AI project, *mysql.connector* library provided a straightforward interface for database interactions, while *PyMuPDF* was essential for the text extraction required by the AI feature.

3.3.3 Database: MySQL

MySQL was chosen as the relational database management system due to its reliability, performance, and wide usage in the industry. To avoid manually creating a database, we have used XAMPP software package, which comes with an integrated web server (Apache) and also MySQL database. XAMPP is suitable for local development environments, to build and test web applications.

The structured MySQL environment was ideal for managing relationships between courses, modules, users, and files (see Figure 23). Furthermore, the *learning-path* column in the users table was designed to store the JSON data, providing a flexible way to save the structured output from the learning path generator without requiring a more complex database schema.

Figure 23: XAMPP MySQL environment

3.3.4 AI Model: Ollama (Llama3)

A key requirement in this project was to select a suitable LLM model that could be integrated into the website. For development purposes, it was more suitable to install a local LLM with limited parameters so that AI performance did not suffer. Ollama was selected as a tool to manage and server local LLMs. It provides a simple and efficient way to run models locally and exposes an OpenAI compatible API endpoints, which significantly simplifies the integration process.

The Llama3 model was chosen as the primary LLM for this project. It offers a strong balance of performance and resource requirements, making it suitable for running on local development hardware (a PC is more suitable than a laptop). Its proficiency in following complex instructions and generating structured JSON output was critical to making our learning path generator feature effective. Whilst online LLM models such as ChatGPT or Gemini offer much higher output quality, it comes at a certain cost, where you eventually have to spend real money to buy tokens in order to continue using it. This is why a local version of LLM was used; while it has limited parameters, it is also cost-free, which ensured unlimited testing during the prompt engineering phase.

In this project, we have utilised the power of OpenAI Python library to connect to local Ollama server instead of the official OpenAI servers. When we generate a new path, a new request is made to the *generate-path* endpoint. With this call, Ollama looks for a model parameter, which is set to Llama3; therefore, this informs Ollama to use it to process the requests. One of the critical instructions is for Llama3 model to ensure that its output is a JSON object, which the backend can then easily parse and it back to frontend.

3.4 Evaluation Methods

To assess the success and effectiveness of the Moodle AI website prototype, we have adopted two types of evaluation. This approach combines quantitative and qualitative data collection to provide a comprehensive understanding of the systems performance, usability, and overall impact. The evaluation consists of two key components: a comparative analysis and a user survey.

3.4.1 User Survey

A user survey was created to gather feedback from the primary target audience: MSc students. The email with the survey was sent to 100+ students with expectancy of around 20 actually doing it. The survey consists of a mixture of multiple choice and checkbox style questions. The primary objectives of the survey were to:

- **Gather usefulness feedback**: Determine how valuable students find the AI-powered features, in particular, the ability to generate personalised and programming language-specific study plans.
- **Asses study patterns**: Collect feedback on the existing Moodle website and how students use it (e.g., purpose of using it, how many times per week, quality of content).

This survey was structured into sections covering current study habits, first impressions of the tool, and overall impact on learning experience.

3.4.2 Comparative Analysis

In addition to the user survey, a comparative analysis was conducted to benchmark the Moodle AI prototype against the existing ENU Moodle website. This analysis focused on the value that AI integration can bring to existing Moodle page. The key criteria for comparison included:

- **Personalisation**: The ability of the system to adapt content to individual student needs. This was a primary focus, comparing the static content delivery of Moodle with the dynamic, user-driven generation of study plans in the Moodle AI website prototype.
- **Student Engagement**: Evaluating the potential of AI features to increase student interaction with learning materials.
- Functionality: Comparison of the feature, highlighting the unique capabilities offered by the AI-powered tool.

This comparative analysis helps to contextualise the projects achievements and provides a clear measure of the innovation and improvement offered by the Moodle AI system over the existing solution.

3.5 Conclusion

In conclusion, this chapter has detailed the systematic methodology employed in the construction and evaluation of the Moodle AI website prototype. The selection of an iterative development model, combined with a robust three-layer architecture, provided a solid foundation for the project. The chosen technologies: React, Flask, MySQL, and a locally run Ollama LLM - were carefully selected to meet the specific demands of the system, enabling the development of a feature-rich and responsive application.

The evaluation strategy, which combines user feedback through surveys with an objective comparative analysis, was designed to provide a holistic assessment of the project success. This comprehensive approach ensures that the findings, which will be presented and discussed in the following chapter, are well-rounded and credible.

4 Chapter 4. Implementation

4.1 Introduction

In this chapter, we discuss the implementation for both the website and the learning path generator feature. For the frontend, we have used React and JavaScript to create a nice looking UI and utilise various libraries that made the website more interactive. Furthermore, for responsiveness, we used a combination of bootstrap and custom made stylesheets. In the backend, we used Python/Flask approach, with the MySQL database to store various user information.

4.2 Core System Implementation

4.2.1 System Architecture

The website is built on a client-side architecture, which separates the UI from the business logic and data management.

1. **React**: The client-side application is responsible for all user interactions. It handles user login, displays courses and modules, and allows users to interact with specific features (colour note widget). It also communicates with the backend via asynchronous API calls.

2. **Flask**: The server-side application acts as the central hub. It exposes a series of RESTful API endpoints to handle requests from the frontend. Its key responsibilities include authenticating users, retrieving courses and module data, managing files (upload/delete), and handling database operations.

4.2.2 Database Management

A connection to the MySQL database is established for each incoming API request. The *mysql.connector* library is used to execute SQL queries. The database schema is designed to handle relationships between users, courses, modules, and user-specific content such as notes and uploaded files, as seen in Figure 24.

4.2.3 API Endpoints

The backend exposes several key endpoints to support the frontend functionality (see Figure 25).

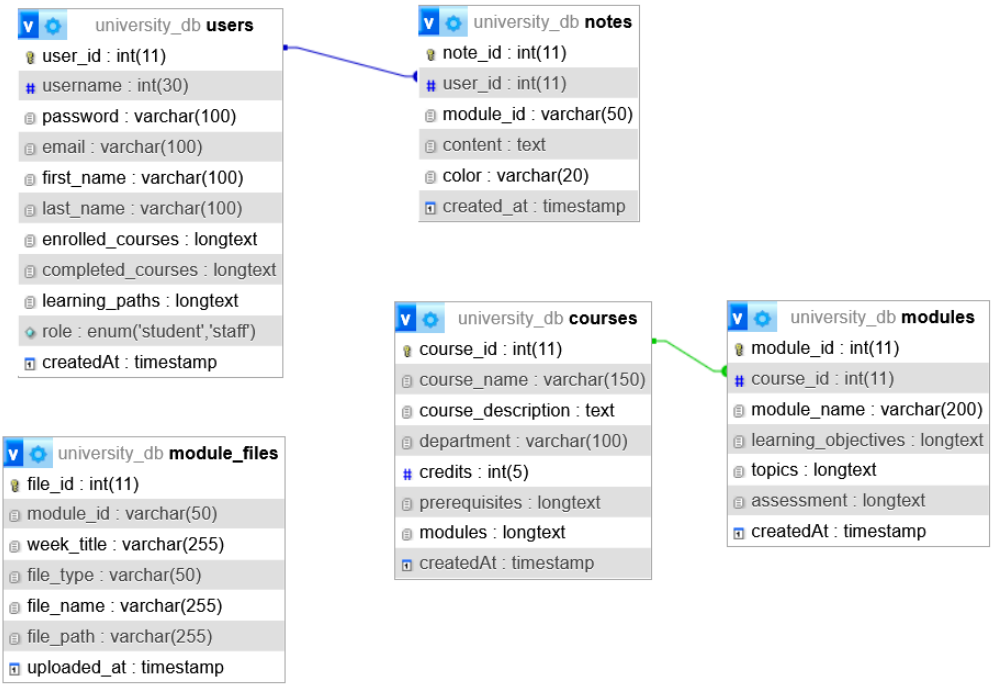


Figure 24: Database Structure

- */api/login*: Authenticates user credentials against the *users* table in the database.
- */api/courses* and */api/modules*: Both retrieve and deliver course and module information based on the logged-in user's role and enrollments.
- */api/notes*: A note taking feature, with create, read, and delete functionality.
- */api/upload* and */api/files*: Manages the uploading and retrieval of module related files, such as lecture PDFs (which are crucial for the learning path generator).

4.2.4 Frontend Component Structure and Routing

The website is structured into a series of modular components. The main *App.jsx* component servers as the entry point and defines the applications routes. It uses a *ProtectedRoute* (see Figure 26) component to protect routes that require authentication, ensuring that only logged-in users can access

```python
@app.route('/api/login', methods=['POST'])
def login():...

@app.route('/api/courses', methods=['GET'])
def get_courses():...

@app.route('/api/modules', methods=['GET'])
def get_modules():...

@app.route('/api/notes', methods=['GET'])
def get_notes():...

@app.route('/api/notes', methods=['POST'])
def add_note():...

@app.route('/api/notes/<int:note_id>', methods=['DELETE'])
def delete_note(note_id):...

@app.route('/api/upload', methods=['POST'])
def upload_file():...

@app.route('/api/files', methods=['GET'])
def get_files():...

@app.route('/uploads/<filename>')
def uploaded_file(filename):...

@app.route('/api/files/<int:file_id>', methods=['DELETE'])
def delete_file(file_id):...
```

Figure 25: API endpoints

course content. This component checks the users authentication state and redirects to thee login page if they are not authenticated.

```jsx
<Route path="/ai/ethics" element={<ProtectedRoute><Ethics /></ProtectedRoute>} />
```

Figure 26: Protected Route

4.2.5 Layout and Styling

The visual layout of the website was built using bootstrap CSS framework. The module page is built on a responsive three-column grid: the main content area for weekly topics, and two sidebars for visual information and interactive widgets. Furthermore, using bootstrap was essential to make the website adaptive to different screen sizes.

In addition to this, a custom CSS stylesheet (Figure 27) was used to adjust the sizes, borders, colour, and placement of the elements within the bootstrap grid. This custom CSS was used to style specific elements, such as the weekly content accordions and the sidebar widgets, ensuring a consistent layout that aligns with the project goals.

```
.week-accordion-header {
    background-color: #306083;
    color: white;
    padding: 0.75rem 1.25rem;
    font-weight: 600;
    cursor: pointer;
    display: flex;
    align-items: center;
    gap: 0.75rem;
    border-radius: 0.25rem;
    border-bottom: solid 2.5px black;
}
.week-accordion-content {
    padding: 1.25rem;
    background-color: #fff;
    border: 1px solid #dee2e6;
    border-top: none;
    border-radius: 0 0 0.25rem 0.25rem;
}
```

Figure 27: Custom Weekly Accordion CSS

4.2.6 State Management and User Authentication

Users authentication status is managed through Reacts Context API in the dedicated *AuthContext.jsx* file. This provides a clean and efficient way to share users data across all components.

1. The *Login.jsx* component calls the login function from the *AuthContext.jsx*.

2. This function saves users data (*userId*, *role*, and *firstName*) to the browsers *localStorage*.

3. When the application is first loaded or refreshed, the AuthContext checks localStorage for the saved data. If it exists, the user is automatically logged back in.

4.3 Learning Path Generator AI

This section provides a detailed technical approach to the implementation of the AI-powered Learning Path Generator, a key feature of the Moodle AI website prototype. The objective is to create a system capable of generating personalised, content-aware study plans for students based on existing module materials. The implementation follows a client-server architecture, utilising a combination of modern web technologies and a locally-run large language model (llm) to ensure a cost-effective and responsive solution.

4.3.1 The Stack

- **Frontend** (Client): React.js for building a dynamic and interactive user interface.
- **Backend** (Server): Flask, a lightweight Python web framework, for handling API requests and business logic.
- **Database**: MySQL, for storing user data and module file information.
- **AI Model**: Llama3, run locally via Ollama, to provide the core generative capabilities without incurring API costs during development. Furthermore, it would be ideal to use API KEY for ChatGPT or Gemini; however, a user needs to have an active subscription and tokens available.

The system is designed around a three-tier architecture, ensuring a clear separation between the user interface, main logic, and data storage.

4.3.2 System Architecture

1. **Presentation** (React - Frontend): The user interacts with a widget within the module page. This component is responsible for capturing user inputs (chosen difficulty and programming language), sending requests to the backend, and rendering the generated learning path.
2. **Logic** (Flask - Backend): The Flask server acts as the intermediary. It exposes an API endpoint that receives requests from the frontend. Its primary responsibilities include retrieving module data from the database, processing PDF files, interacting with the AI model, and returning the structured learning path. It also has a save/load capability that saves personalised path for each user separately in the database.

3. **Data** (MySQL - Database): The database stores essential information, including user credentials, paths to the PDF files associated with each module, and JSON data of each student's generated learning paths in a dedicated *learning-paths* column.

The end-to-end data flow for a generation requests is as follows:

1. A student selects their desired difficulty level and a programming language in the React widget and clicks the "generate" button.
2. The React client sends a POST request containing the *moduleId*, *userId*, and user selections to the */api/generate-path* endpoint on the Flask server.
3. The Flask server queries the MySQL database to find the file paths for all PDFs associated with the given *moduleId*.
4. The server reads the content of these PDFs from the local file system.
5. This content is then processed and sent to the locally-run Ollama LLM.
6. The LLM returns a structured JSON object representing the learning path.
7. The Flask server forwards this JSON response to the React client.
8. The React client parses the JSON and dynamically renders the learning path, and then sends a second request to a */api/save-path* endpoint to store the generated path in the user's record in the database.

4.3.3 User Interface and State Management

The widget's UI consists of two dropdown menus and a button, managed using the *useState* hook to track the user's selected *difficulty* and *programming language* as seen in Figure 28. An additional state variables (*isLoading*, *error*, *pathData*) are used to manage the component's lifecycle during the asynchronous API call, allowing the UI to display loading indicators, error messages, or the final result.

4.3.4 User-Specific Persistence

For more personalised experience, generated learning paths are stored in the database. This ensures that a path can be loaded to the website using any browser on any device.

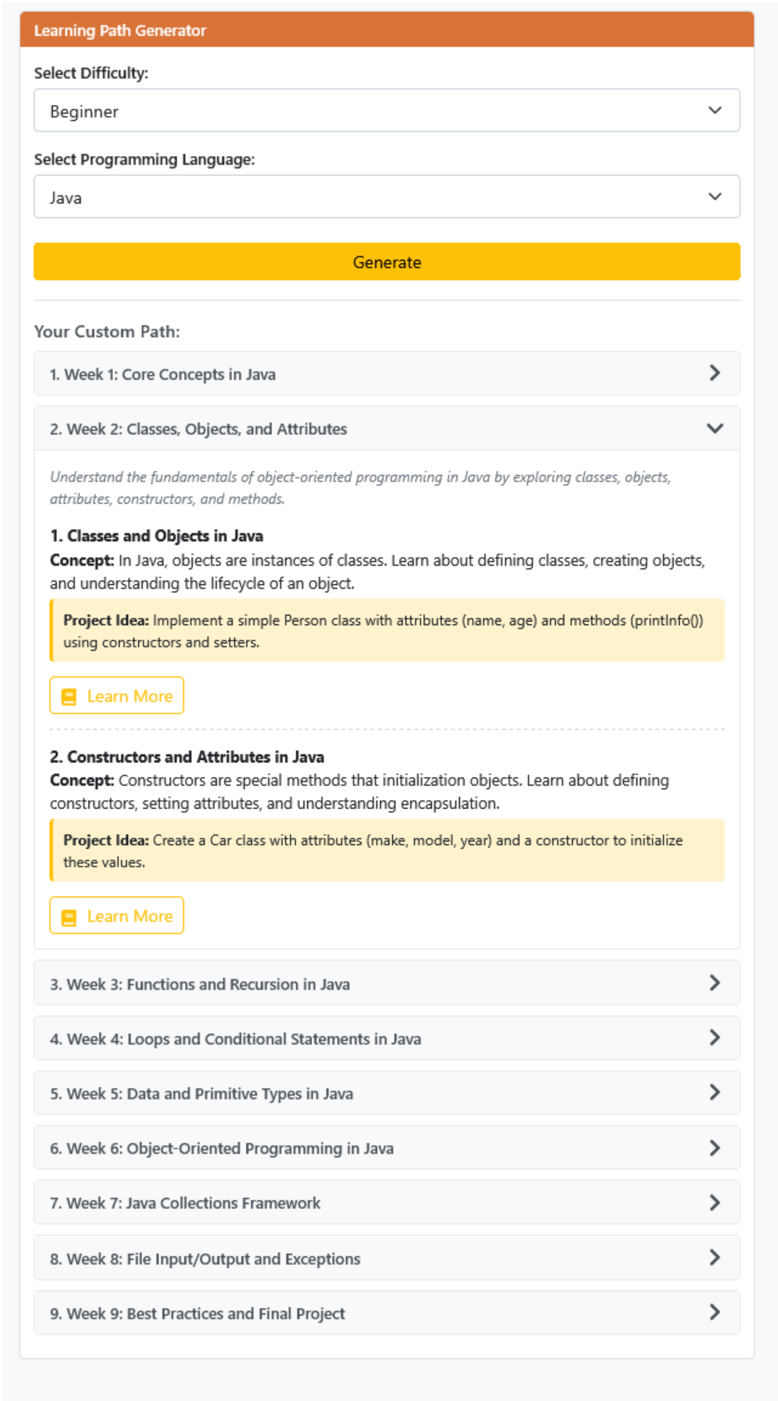


Figure 28: Learning Path Generator

When a new path is generated, the React component makes a POST request to a */api/save-path* endpoint on the Flask server. The request includes the *userId*, *moduleId*, and JSON data of the learning path. The backend then updates the *learning-paths* column for that specific user in the database. Conversely, a *useEffect* hook in the learning path widget runs when the component first mounts or when the logged-in user changes. This will call a */api/load-path* endpoint, which retrieves and displays the user's previously saved learning path for that specific module.

4.3.5 Content Retrieval and Processing

Upon receiving a request at the */api/generate-path* endpoint, the server first interacts with the MySQL database to fetch the file paths of all PDFs for the specified module. A key robustness feature was implemented here: the code iterates through the retrieved paths and uses Python's *os.path.exists()* to verify that each file is physically present on the server, as can be seen in Figure 29. This prevents the application from crashing if a file is missing, allowing it to skip the file and use only the available content. The text of all PDFs is then extracted using the *PyMuPDF* library and concatenated into a single string.

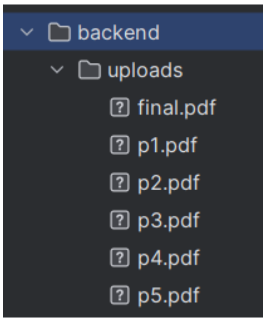


Figure 29: PDF files for LPG

4.3.6 AI Integration and Prompt Engineering

A significant challenge with running a local LLM is the limited processing power compared to cloud-based APIs, especially when dealing with large amounts of text. In this case, computer hardware matters, and excessively long generation times occurred with frequent timeouts when sending all module PDFs to the LLM.

To address this issue, we have introduced a more sophisticated two-step Retrieval-Augmented Generation (RAG).

1. **Summarise**. A complete and concatenated text from all PDFs is sent to *Ollama llama3* model with a simple, direct prompt: "summarise the key topics from the following text". This is a relatively fast operation for the model.

2. **Generate**. The summary from the previous step is then used as the context for the main (system, see Figure 30) prompt. The second (user, see Figure 31) prompt is far more detailed and contains the critical instructions for the AI.

It is engineered to:

- Enforce JSON output. Instructing the model that its output must be a single, valid JSON object, enabling easy parsing on the frontend.

- Translate concept. The model should use the provided summary to identify language-agnostic concepts (e.g., control flow, OOP, version control) and generate all new content in the student's chosen programming language (*target-language*).

- Include external resources. A model should display a *resource-link* for each topic. Because the local model cannot access the internet, a placeholder (implemented in the frontend) is shown instead of a real resource link, which should take a student to a website with learning content relevant to the specific weeks topic.

By using this approach, we have reduced the cognitive load on the local model, bringing the average generation time down from over 10 minutes (tested on a laptop, it is much faster with a PC) to under 60 seconds.

```
# Prompt for Ollama API
system_prompt = f"""
You are an expert academic advisor and curriculum designer who can translate educational concepts between programming languages.
Your task is to generate a structured, practical learning path for a student.
The output MUST be a single, valid JSON object and nothing else.
"""
```

Figure 30: Main System Prompt

```
user_prompt = f"""
The student's desired parameters are:
- Difficulty Level: '{difficulty}'
- Target Programming Language: '{target_language}'

CRITICAL INSTRUCTION: The following is a SUMMARY of the core concepts for a university module.
Your primary job is to take these summarized topics and create a detailed learning path.

All explanations, project ideas, and code examples in your output MUST be in the student's target language: '{target_language}'.

--- SOURCE SUMMARY (for concept extraction only) ---
{module_summary}
--- END OF SOURCE SUMMARY ---

The JSON object you return must have a root key 'learningPath' which is an array of 'modules'.
Each module object in the array must contain:
1. "module_number": An integer (e.g., 1).
2. "title": A string (e.g., "Week 1: Core Concepts in {target_language}").
    This should correspond to the weekly topics from the source summary.
3. "description": A brief string explaining the module's goals in the context of the target language.
4. "topics": An array of topic objects, breaking down the module's key concepts.

Each topic object in the 'topics' array must contain:
1. "topic_number": An integer (e.g., 1).
2. "title": A string (e.g., "Variables and Data Types in {target_language}").
3. "concept": A string (2-3 sentences) explaining the core concept in simple terms, specifically for '{target_language}'.
4. "project": A string describing a small, practical exercise or a thought-provoking question that requires
    writing code in '{target_language}'.
5. "resource_link": A valid, high-quality URL to a relevant resource for the topic
    (e.g., MDN for JavaScript, python.org for Python).
"""
```

Figure 31: User Prompt

4.4 Conclusion

This chapter has provided a comprehensive technical overview of the development of the Moodle AI website prototype. The implementation was successfully executed by following a structured approach: building a core website with necessary functionality and integrating the innovative AI-powered learning path generator.

The core system was built upon a robust three-layer architecture, utilising React.js to create a dynamic user interface, and a Flask backend to manage server side logic and data persistence with a MySQL database. Some of the key features such as the component-based user interface, persistent user sessions were managed through a custom *AuthContext*, and database-driven content delivery was successfully implemented, resulting in a functional and user-friendly application.

The development of the AI learning path generator consisted of a locally run LLM to create a personalised learning experience. The implementation was a two step summarisation and generation process which proved to be a critical solution to overcome the performance limitations of a local model, enabling the system to handle larger volumes of text from module documents efficiently. Furthermore, the successful integration of database persistence for the generated paths ensured that each student experience is both unique and consistent across sessions.

In conclusion, the implementation phase has resulted in a fully functional website prototype that successfully integrates a custom built AI personalisation feature.

5 Chapter 5. Evaluation

In this project, we have used two techniques to gather feedback and compare this project to another existing application, as described in previous chapters. By conducting a user survey in Figure 42, we were able to find some crucial details about the Moodle and learning path generator. The responses to the survey are shown in Figure 43. The survey focused on the existing Moodle page and the learning path generator that was built in this project. Users were asked to answer a series of questions related to the current use of Moodle website as well as a series of questions about the learning path generator and its usefulness.

The first part produced valuable feedback with most of the students saying that the current Moodle page needs to be improved further. They stated that the overall structure of the Moodle page sometimes is good, and sometimes is bad, depending on a lecturer and their uploaded content. Another few students mentioned that while the structure is descent enough to do the coursework, they do not further learn or improve their skills. Moreover, most of the students agree that each module of the course should have the option to do the coursework in their preferred programming language. Based on these questions and responses, it is clear that students are not very happy with the way the module content is structured, which discourages them from engaging with the module material/content more. Thus, leading to possibly lower grades than expected.

In part two of the survey, we have introduced a learning path generator and asked students if they would see this feature to be useful when integrated into the Moodle website. All students said they would be very likely to use this feature and that this would definitely boost student engagement in learning and raise their grades a little. Furthermore, it is important that students can personalise their Moodle page throughout their study career; All the students who did the survey agreed that personalisation is important and that a learning path generator would definitely make it more personalised and interesting to use.

It is clear to see that a learning path generator is a very useful feature that students will like and use. The prototype shown to selected number of students gained attraction and interest. However, there are some limitations and possible drawbacks when it comes to adapting this feature to all university courses. The prototype specifically focused on a technology field (software development course). Whilst this works on any course which includes programming languages, it may require some refinement for other courses that focuses more on the theoretical part such as business. With this in mind, the prototype should be expanded, thus it could adapt to specific courses and structure of the content.

In conclusion, the prototype of a learning path generator attracted the interest of various students, who said that this feature would be very useful and increase student learning engagement. Furthermore, after conducting a survey, we were able to understand how students use Moodle and how often. Thus, the LPG feature can be refined and upgraded to a better version. While this played as a starting point, further user testing and surveying is required to understand more about what students want and how the learning path generator could be used by all students in all university courses.

6 Chapter 6. Conclusion

6.1 Fulfillment of Aims

The aims stated in Chapter 1 have been fulfilled. We have researched existing technologies, finding two learning management systems that works as an AI tutor which produces content, grades, and handles system management automatically. This, therefore was a sophisticated LMS and beyond the scope of our proposed project. We have looked further into simpler versions, finding ENU Moodle page, which presented a perfect application to compare against our built prototype.

After exploring the LMS applications, we have investigated their architecture, and how they were built. While some information was not publicly available, MoodleLMS was an open source app which we downloaded and examined the insights. As stated before, the LMS architecture is complex and we were only interested in learning path generation part. This particular part was made of a rule-based, collaborative or content-based filtering approach. Thus, it acted as a starting point for building our prototype.

Another aim was to find an existing gap that could be improved. Therefore, we chose to investigate the ENU Moodle page and find an area for improvement. Since Moodle does not have a learning path generator, it served as a perfect application that could be further enhanced with AI technology.

We have used ENU Moodle website to build a similar looking Moodle AI prototype website. The website has login feature, courses, modules, and the PDF files with mock data which were uploaded to the module weekly schedule and used in generating new learning paths. Subsequently, we have developed a learning path generator that was integrated into the Moodle AI website. It was built using local LLM for development reasons and was successfully tested on a higher end PC.

Finally, a survey was sent to multiple students and feedback was collected and then summarised. The information was used to evaluate our prototype and to understand if students find it useful, and also to determine what further work is required to make the learning path generator a better version for all students. We have then compared our prototype to the existing ENU Moodle, and came to conclusion that our Moodle AI prototype, specifically learning path generator, should be integrated into the ENU Moodle website.

6.2 Limitations

While the Moodle AI website prototype successfully demonstrates the core concept of an AI-powered learning path generator, is it important to acknowledge the limitations inherent in its design and implementation. These limitations provide valuable context for the evaluation results and offer clear directions for future work.

6.2.1 Local LLM Performance and Hardware Dependency

The decision to use a locally run LLM via Ollama was crucial for cost-effective development and testing. However, this approach introduces a significant performance bottleneck. The speed of the AI-powered features, in particular the PDF file summarisation and generation of learning paths, is entirely dependent on the user's local hardware.

When running LLM on a laptop, the CPU usage is almost 100 percent; therefore, generation time can be lengthy, especially when processing larger documents. The implementation in this project has a 60-second timeout, which was necessary to implement in order to prevent the application from loading indefinitely. In real world production environment, relying on client side or non-GPU server hardware would likely result in a poor user experience.

6.2.2 Inability of Local LLM to Generate Valid External Links

A key limitation discovered during development and testing was the local models inability to provide valid, real-world hyperlinks for the "Learn More" feature. As the Ollama model run entirely offline, it has no access to the live internet to verify or search for URLs.

When prompted to provide a *resource-link*, the model would create an imaginary hyperlink to the file which does not exist (e.g., */cpp-intro.html*). This is a fundamental constraint of using offline models for tasks that require real-world, up-to-date information. Although a placeholder was implemented to handle this limitation, the model still generated imaginary links every few generations.

6.2.3 Dependency on Text Based PDF Content

The systems ability to understand module content is entirely reliant on its capacity to extract raw text from PDF files using the *PyMuPDF* library.

This approach works well for documents that are text based; however, it would fail completely with other common educational formats, such as presentations, video/audio, or images.

The quality and accuracy of the AI generated learning paths are therefore directly tied to the quality and format of the uploaded source materials. The system currently lacks the sophisticated optical character recognition or multi-modal capabilities required to process a wider variety of educational content.

6.2.4 Scope of Evaluation

The evaluation of the prototype was conducted with a small amount of university students. While this provided essential qualitative feedback and initial usability insights, the sample size is not large enough to be considered statistically significant. The findings are therefore indicative of the potential of the system, but cannot be generalised to the entire student population. A more extensive study with a larger and more diverse user group would be required to draw more definite conclusions about the impact of features on learning outcomes.

6.3 Future Work

The following is a list of proposed improvements for the Moodle AI website:

- Replace local Llama3 LLM model with an online model such as ChatGPT or Gemini. This will greatly improve the overall performance of the learning path generator. In addition, local model has limited amounts of parameters and data; therefore, it cannot generate the best possible answer and provide best links to learning material. By using an online model, the learning path generator would become more accurate and reliable with the output.
- Currently, the LPG is going through weekly content and specifically looking for PDF files so it could read it and summarise. The learning path generator should be able to understand more diverse amounts of files and data, such as code files or a presentation. With an upgrade like this, not only the LPG would be able to summarise content from multiple file formats, but would also be easier for moderators to upload teaching content without thinking that they need specifically PDF file for it.

- Introducing validation so that the learning path generator would verify if each link to external website is safe and of a good quality would greatly add to quality of the output and make it a more reliable and accurate tool.

By taking these steps, the learning path generator can be greatly improved and become a powerful tool in the future of education.

6.4 Critical Evaluation of The Project

This section provides a critical evaluation of the ”Optimising Students Learning Experience Through AI-Driven Personalised Path Generation” project. It combines the findings from the literature review, methodology, and implementation to assess the overall success, limitations, and contribution to the field of education.

The core objective in this project was to investigate the effectiveness of personalised learning paths in enhancing undergraduate student engagement and learning outcomes in technology courses. The main research questions is: ”How can personalised learning paths affect undergraduate student engagement and learning effectiveness in tech courses?”. The development of the Moodle AI website prototype successfully demonstrated the technical feasibility of this concept. The system, built on React.js, Flask, and MySQL, effectively integrated a local LLM (Llama3), to generate personalised learning paths from module content.

The system's two step retrieval augmented generation (RAG) approach proved to be a critical innovation for overcoming the computational limitations of running a local large language model (Llama3). By first summarising the content from PDF files and then using that summary to the main generation prompt, the system reduced the average generation time from over 5 minutes to under 60 seconds (although the computer still must be higher end). This approach addresses a significant challenge identified in the literature by (Razafinirina, 2024), who highlights the extensive computational resources required for LLMs in educational contexts. This practical solution demonstrates a viable pathway for educational institutions to leverage powerful AI models on more limited hardware.

Furthermore, the website prototype's ability to enforce structured JSON output from the LLM and the subsequent parsing on the frontend showcase a successful application of prompt engineering to ensure reliable AI behaviour.

Moreover, the user survey provided initial evidence of the system's potential, with students finding the concept of a personalised learning path beneficial for increasing their engagement and efficiency, as well as grades and learning.

While the project was successful in achieving its aims, there are some limitations. The reliance on a local LLM, while its a practical solution for cost and testing, it also introduced constraints on the quality and accuracy of the output. The limited parameter set of Llama3 meant that the generated content was not as accurate or as rich as could have been if using a cloud based model such as Gemini or ChatGPT. Crucially, the local model's inability to access the internet meant that the external links did not appear, and so we had to introduce placeholders to replace it. Thus, it also affected the learning path generator by limiting its full potential.

The scope of content retrieval was another significant limitation. The system was designed to process only text-based PDF files. As noted in future works section, this leaves the system unable to handle a wide variety of educational content, such as images and presentations. This limitation directly ties to the challenges of Domain-Specific information and Domain-Independent information discussed in the literature review. Without the ability to process a broader range of content, the system's learner model is incomplete; therefore, limiting its ability to create a truly adaptive learning experience.

Finally, the user survey's small sample size means the findings are indicative rather than statistically significant. While the qualitative feedback was positive, a larger, more diverse user study would be necessary to draw definite conclusions about the LPG system's impact on a broader student population. This gap in the evaluation process means that the sub-questions of the main research question, particularly those regarding the differences between theoretical and practical courses and the prediction of student mastery, could not be fully addressed.

In conclusion, this project successfully validated the core hypothesis that AI-driven personalised learning paths can be a powerful tool for improving student learning. The developed prototype stands as a proof of concept, demonstrating a technical approach to integrating an LLM within an educational framework. While the project achieved its primary objectives, a critical evaluation reveals key limitations related to the LLM's capabilities and the system's concept processing scope. Addressing these limitations in future work will be essential to transforming this prototype into a comprehensive and robust educational tool capable of fully realising the potential of an AI in personalised learning.

Appendices

MSc dissertation check list

Student Name: Edgar Park	Matric: 40310835

Milestones	Date of completion	Target deadline
Proposal	10/06/25	Week 3
Initial report	14/07/25	Week 7
Full draft of the dissertation	20/08/25	2 weeks before final deadline

Learning outcome	The markers will assess	Pages [1]	Hours spent
Learning outcome 1 Conduct a literature search using an appropriate range of information sources and produce a critical review of the findings.	* Range of materials; list of references * The literature review/exposition/background information chapter	P74-p76 P12-p35	200 hours
Learning outcome 2 Demonstrate professional competence by sound project management and (a) by applying appropriate theoretical and practical computing concepts and techniques to a non-trivial problem, or (b) by undertaking an approved project of equivalent standard.	* Evidence of project management (Gantt chart, diary, etc.) * Depending on the topic: chapters on design, implementation, methods, experiments, results, etc.	P73 P37-p52	200 hours
Learning outcome 3 Show a capacity for self-appraisal by analysing the strengths and weakness of the project outcomes with reference to the initial objectives, and to the work of others.	* Chapter on evaluation (assessing your outcomes against the project aims and objectives) * Discussion of your project's output compared to the work of others.	P54-p55 P54-p59	100 hours
Learning outcome 4 Provide evidence of the meeting learning outcomes 1-3 in the form of a dissertation which complies with the requirements of the School of Computing both in style and content.	* Is the dissertation well-written (academic writing style, grammatical), spell-checked, free of typos, neatly formatted. * Does the dissertation contain all relevant chapters, appendices, title and contents pages, etc. * Style and content of the dissertation.		80 hours
Learning outcome 5 Defend the work orally at a viva voce examination.	* Performance * Confirm authorship		1 hour

Have you previously uploaded your dissertation to Turnitin?	<u>Yes</u>/No
Has your supervisor seen a full draft of the dissertation before submission?	<u>Yes</u>/No
Has your supervisor said that you are ready to submit the dissertation?	<u>Yes</u>/No

Figure 32: Checklist

EDINBURGH NAPIER UNIVERSITY SCHOOL OF COMPUTING

MSc INITIAL REPORT

1. Student details

Last (family) name	Park
First name	Edgar
Napier matriculation number	40310835

2. Project details

Project title	Optimising Student Learning Experience Through AI-Driven Personalised Path Generation
Summary of project (300 words) Include a note of its aims, the main research questions, the methods to be deployed, deliverables, and means of evaluating the project work as a whole.	*With an increasing usage and interest in artificial intelligence, it is evident that AI is making its way to be a part of everyday life. Currently, the education sector is not utilizing AI as much and it is still in the experimental stage.* *This project aims to investigate further into AI use in education. In particular, its use in generating specific learning paths for students. Moreover, this project seeks to answer the following research questions:* o *How can personalized learning paths affect undergraduate student engagement and learning effectiveness in tech courses (e.g. software development)?* ▪ *How can the benefits differ between theoretical and practical courses within universities technology curriculum?* ▪ *To what extent can personalized learning path system accurately recommend learning resources and predict student mastery in specific topics, based on course content.* *In addition, the aims of the project are as follows:* o *Research existing technologies.* o *Investigate the architecture.* o *Find relevant gaps for improvement.*

Figure 33: IPO page 1

- *Build a prototype of a website with AI learning path system integrated.*
- *Evaluate the prototype against the existing technologies.*

Based on research, as described in literature review, this project will be focusing on enhancing undergraduate student-learning experience; therefore, implementing an AI based software application, which will be integrated into a website prototype.

The implementation will be mainly Python and Machine Learning based. A prototype of a website will be built (similar to Moodle) where the AI system will be integrated at a later stage. The project will have a few datasets created from scratch; however, the project does not require large datasets and therefore, MySQL will be primarily used for databases.

The AI implementation is the most tricky part of the project. Based on the literature review, we can determine that a hybrid approach is best for this, and therefore, the implementation will use neural networks and collaborative/content-based filtering approach.

The success of this project will be measured in the evaluation stage at the end of the dissertation. The website with AI integration, its effectiveness and robustness will be compared to the existing Moodle (ENU) website, also MoodleLMS, and finally user testing will provide valuable feedback that will also be used in answering research questions of this project.

3. Literature review

See attached **PDF** file for literature review.

4. Annotated contents list for the dissertation

Please show the proposed structure for the final version of the dissertation. This should give a brief indication of what each section will contain.

1. Introduction

   Motivation

   Motivation for choosing this project.

Figure 34: IPO page 2

Introduction to the Topic
Information about Aims & Objectives, Research Question, and Scope & Constraints.

Introduction to the Layout of This Document
Brief description of each chapter in the Dissertation.

2. Literature Review
Research, findings, and discussion of the current knowledge relevant to the project. Addressing research questions and gaps in the knowledge.

3. Methodology
An in-depth discussion and description of various tools/models/algorithms & architecture that is used in developing the proposed software/web applications.

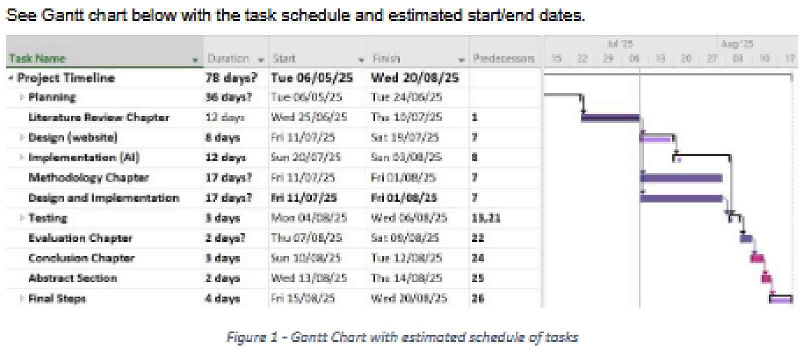
4. Design & Implementation
An actual roadmap to how the proposed software/web application is built and implemented. In-depth guide to algorithms and techniques used in the development phase.

5. Evaluation
Evaluation of approach & techniques used for gathering results. Subsequently, a presentation and summary of results. The software/web application will be compared to the existing platform. Moreover, user testing will be conducted to gather live feedback and results.

6. Conclusion
Discussion of Results
An in-depth discussion of the results gathered from testing & evaluation; therefore, drawing conclusions based on this discussion.

Fulfilment of Aims
A discussion of proposed aims in this dissertation and how they were accomplished.

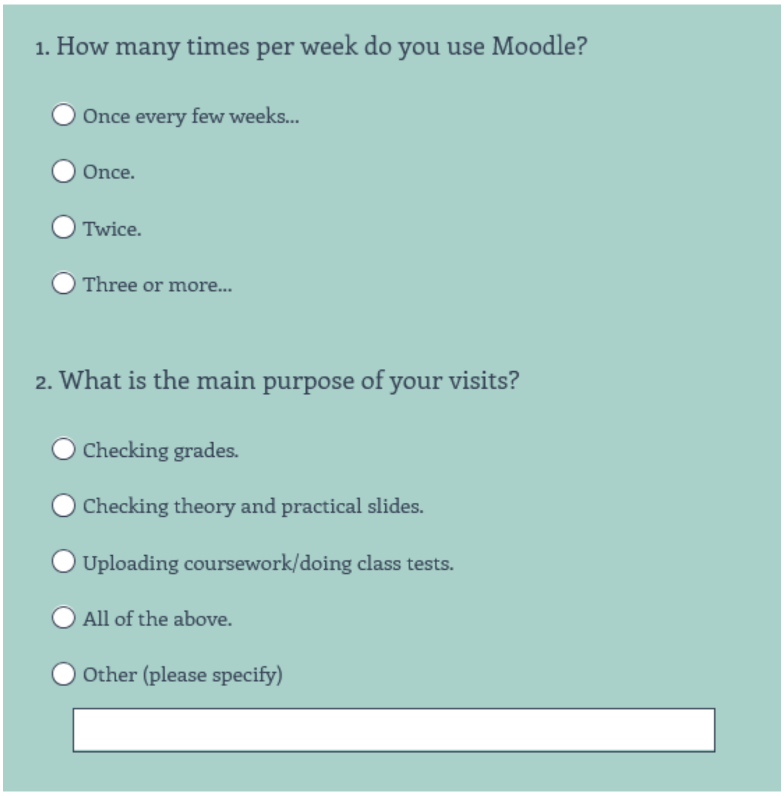
Figure 35: IPO page 3


See Gantt chart below with the task schedule and estimated start/end dates.

Task Name	Duration	Start	Finish	Predecessors
Project Timeline	78 days?	Tue 06/05/25	Wed 20/08/25	
Planning	36 days?	Tue 06/05/25	Tue 24/06/25	
Literature Review Chapter	12 days	Wed 25/06/25	Thu 10/07/25	1
Design (website)	8 days	Fri 11/07/25	Sat 19/07/25	7
Implementation (AI)	12 days	Sun 20/07/25	Sun 03/08/25	8
Methodology Chapter	17 days?	Fri 11/07/25	Fri 01/08/25	7
Design and Implementation	17 days?	Fri 11/07/25	Fri 01/08/25	7
Testing	3 days	Mon 04/08/25	Wed 06/08/25	13,21
Evaluation Chapter	2 days?	Thu 07/08/25	Sat 09/08/25	22
Conclusion Chapter	3 days	Sun 10/08/25	Tue 12/08/25	24
Abstract Section	2 days	Wed 13/08/25	Thu 14/08/25	25
Final Steps	4 days	Fri 15/08/25	Wed 20/08/25	26

Figure 1 - Gantt Chart with estimated schedule of tasks

Figure 36: IPO page 4

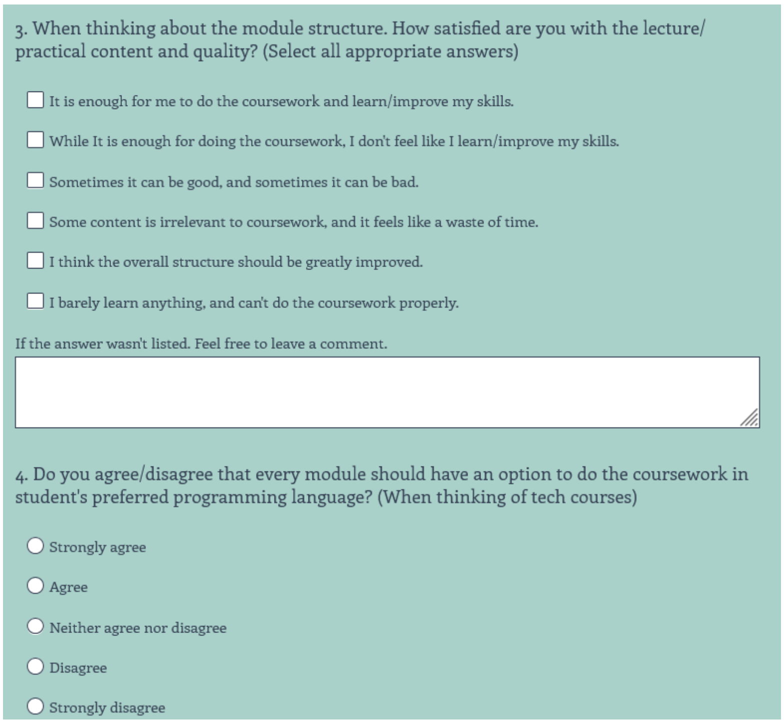

1. How many times per week do you use Moodle?

- Once every few weeks...
- Once.
- Twice.
- Three or more...

2. What is the main purpose of your visits?

- Checking grades.
- Checking theory and practical slides.
- Uploading coursework/doing class tests.
- All of the above.
- Other (please specify)

Figure 37: Survey Questions


3. When thinking about the module structure. How satisfied are you with the lecture/ practical content and quality? (Select all appropriate answers)

- [ ] It is enough for me to do the coursework and learn/improve my skills.
- [ ] While It is enough for doing the coursework, I don't feel like I learn/improve my skills.
- [ ] Sometimes it can be good, and sometimes it can be bad.
- [ ] Some content is irrelevant to coursework, and it feels like a waste of time.
- [ ] I think the overall structure should be greatly improved.
- [ ] I barely learn anything, and can't do the coursework properly.

If the answer wasn't listed. Feel free to leave a comment.

4. Do you agree/disagree that every module should have an option to do the coursework in student's preferred programming language? (When thinking of tech courses)

- Strongly agree
- Agree
- Neither agree nor disagree
- Disagree
- Strongly disagree

Figure 38: Survey Questions

5. **Scenario**: A university has introduced an AI assistant to its Moodle page. The AI assistant is implemented as a widget on a side of the page. The main purpose is for the AI assistant to generate a new learning path for a student based on their preferred programming language. The AI assistant takes into consideration lecturers provided lecture and practical slides. It then generates a weekly path similar to module content, but in your preferred programming language. It also provides relevant links to various websites where you can learn more about "current week" topic.

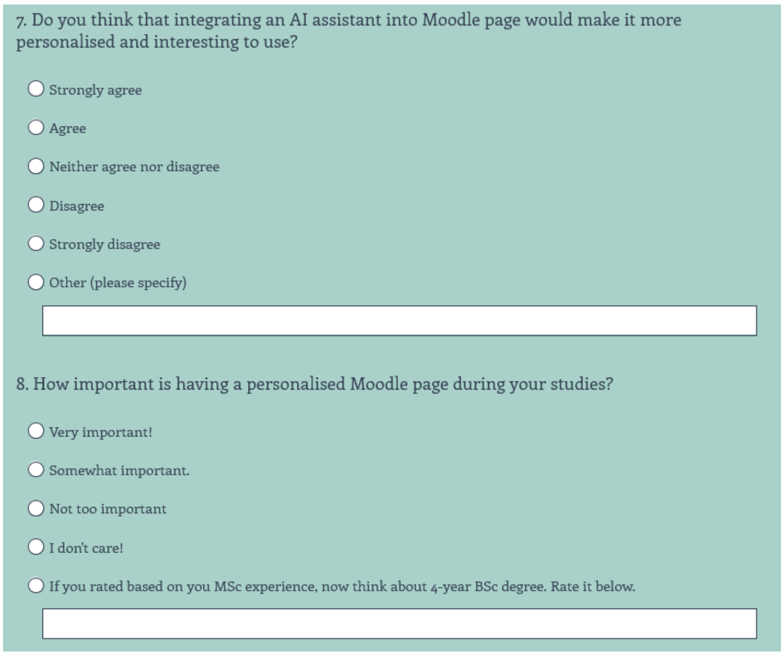
Question: How likely are you to use such a feature?

- ○ Very likely
- ○ Likely
- ○ Neither likely nor unlikely
- ○ Unlikely
- ○ Very unlikely

6. Do you think that this AI assistant would beneficial for students, increase their learning, and raise their grades? (Select all that apply)

- ☐ Absolutely! An AI assistant would help lots of students.
- ☐ Yes! Programming in your favourite language would boost student engagement in learning.
- ☐ The concept is great, but further testing and surveying is needed to see if it works.
- ☐ Im not sure. Lecturers might not like it.
- ☐ Im happy with the content provided by the lecturer.
- ☐ I might use such a feature, but rarely.

Figure 39: Survey Questions



7. Do you think that integrating an AI assistant into Moodle page would make it more personalised and interesting to use?

- Strongly agree
- Agree
- Neither agree nor disagree
- Disagree
- Strongly disagree
- Other (please specify)

8. How important is having a personalised Moodle page during your studies?

- Very important!
- Somewhat important.
- Not too important
- I don't care!
- If you rated based on you MSc experience, now think about 4-year BSc degree. Rate it below.

Figure 40: Survey Questions

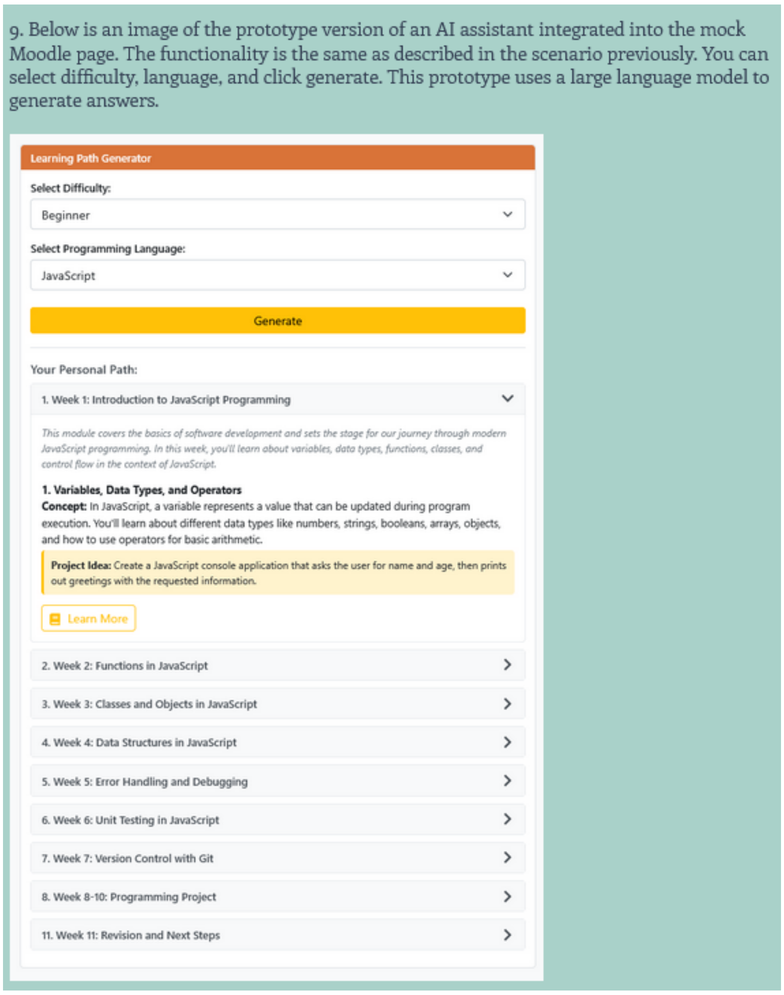


Figure 41: Survey Questions

How likely would you be using such a prototype?

- Very likely
- Likely
- Neither likely nor unlikely
- Unlikely
- Very unlikely

Other (please specify)

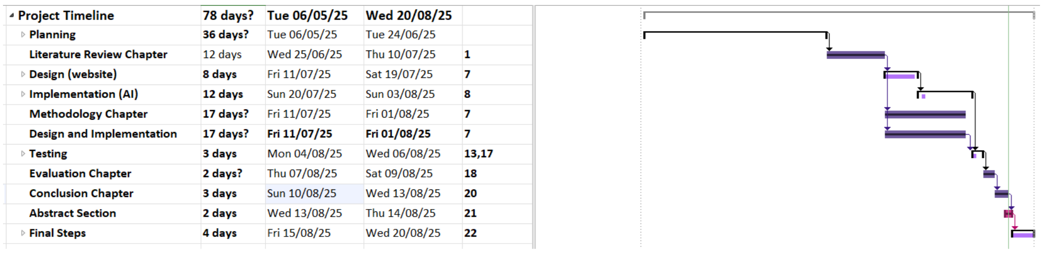
10. When it comes to large language models (LLMs), do you think it could generate an accurate learning path each time?

- Yes! I believe LLMs are improving and very accurate.
- Depends on the LLM. Some are good, some are not!
- Inaccuracies will happen regardless.
- Maybe accurate first time and then not so accurate.
- Other (please specify)

Figure 42: Survey Questions

Question	Answers
Q1	A-1, B-0, C-1, D-3
Q2	A-1,B-1,C-1,D-3
Q3	A-1, B-3, C-3, D-2, E-3
Q4	A-1, B-2, C-2, D-0, E-0
Q5	A-4, B-2, C-0, D-0
Q6	A-3, B-3, C-1, D-0, E-0, F-0, G-0
Q7	A-1, B-4, C-0, D-0, E-0, F-0
Q8	A-2, B-4, C-0, D-0, E-0
Q9	A-3, B-2, C-1, D-0, E-0
Q10	A-3, B-2, C-2, D-0, E-0

Figure 43: Survey Answers



Project Timeline	78 days?	Tue 06/05/25	Wed 20/08/25	
Planning	36 days?	Tue 06/05/25	Tue 24/06/25	
Literature Review Chapter	12 days	Wed 25/06/25	Thu 10/07/25	1
Design (website)	8 days	Fri 11/07/25	Sat 19/07/25	7
Implementation (AI)	12 days	Sun 20/07/25	Sun 03/08/25	8
Methodology Chapter	17 days?	Fri 11/07/25	Fri 01/08/25	7
Design and Implementation	17 days?	Fri 11/07/25	Fri 01/08/25	7
Testing	3 days	Mon 04/08/25	Wed 06/08/25	13,17
Evaluation Chapter	2 days?	Thu 07/08/25	Sat 09/08/25	18
Conclusion Chapter	3 days	Sun 10/08/25	Wed 13/08/25	20
Abstract Section	2 days	Wed 13/08/25	Thu 14/08/25	21
Final Steps	4 days	Fri 15/08/25	Wed 20/08/25	22

Figure 44: Gantt Chart

References

Abbasiantaeb, S., Zahra; Momtazi. (2021). Text-based question answering from information retrieval and deep neural network perspectives: A survey. , *6*.

Bahi, M., Meriem; Batouche. (2018). Deep learning for ligand-based virtual screening in drug discovery. , 1-5.

Brusilovsky, A., Peter; Kobsa. (2007). User models for adaptive hypermedia and adaptive educational systems. *Lecture Notes in Computer Science*, 3-53.

B.Thorat, R. B. S., Poonam; M. Goudar. (2015). Survey on collaborative filtering, content-based filtering and hybrid recommendation system. *International Journal of Computer Applications*, *110*(4), 31-36.

Carole, H. C., Kouayep Sonia; Kim. (2024). Enhanced experiences: Benefits of ai-powered recommendation systems. , 216-220.

Chatti, S. T. H. S. U., Mohamed Amine; Dakova. (2013). Tag-based collaborative filtering recommendation in personal learning environments. *IEEE Transactions on Learning Technologies*, *6*(4), 337-349.

Farooq, S. (n.d.). A survey on adversarial information retrieval on the web.

Feuerriegel, J. J. C. Z. P., Stefan; Hartmann. (2024). Generative ai. *Bus Inf Syst Eng*, *66*(1), 111-126.

Ghergulescu, D. I. (n.d.). How do we update the curriculum model?

Ghergulescu, I. (2021). Extending ai-enabled adaptive learning with subskills modelling: Integrating adaptemy ai engine with algebrakit engine.

Haenlein, A., Michael; Kaplan. (2019). A brief history of artificial intelligence: On the past, present, and future of artificial intelligence. *AI Magazine*, *61*(4), 5-14.

Jannach, P. R. F. Z. M., Dietmar; Pu. (n.d.). Recommender systems: Past, present, future. *AI Magazine*(42), 3-6.

Julianti, Y. Y. B. B. W., Ramaddan; Heryadi. (2022). Recommendation system model for personalized learning in higher education using content-based filtering method. *IEEE Xplore*, 1-6.

Khayatian, Z. B. A., Fazel; Nagy. (2021). Using generative adversarial networks to evaluate robustness of reinforcement learning agents against uncertainties. *Energy and Buildings*, *251*, 111-334.

Liu, D., Mengchi; Yu. (2023). Towards intelligent e-learning systems. *Education and Information Technologies*, *28*(7), 7845-7876.

Liñán, PBI A. J., Laura Calvet; Pérez. (2015). Educational data mining and learning analytics: differences, similarities, and time evolution. *RUSC.*

Universities and Knowledge Society Journal, *12*(3), 98-112.

Lu, C.-C. T. H.-Y., Ruei-Shan; Lin. (2024). Empowering large language models to leverage domain-specific knowledge in e-learning. *Applied Sciences*, *14*(12), 5264.

Naseer, M. N. T. M. A.-A., Fawad; Khan. (2024). Integrating deep learning techniques for personalized learning pathways in higher education. *Heliyon*, *10*.

Naveed, A. U. Q. S. S.-M. A. S., Humza; Khan. (2024). A comprehensive overview of large language models.

Ng, Y., Chee; Fung. (2024). Educational personalized learning path planning with large language models.

Okewu, P. M. S. M. R. D. R., Emmanuel; Adewole. (2021). Artificial neural networks for educational data mining in higher education: A systematic literature review. *Applied Artificial Intelligence*, *35*(13), 983-1021.

Pardos, N., Zachary; Heffernan. (n.d.). Introducing item difficulty to the knowledge tracing model. , 243-254.

Pelánek, T. J. P., Radek; Effenberger. (2024). Personalized recommendations for learning activities in online environments: A modular rule-based approach. *User Modeling and User-Adapted Interaction*, *34*(4), 1399-1430.

Raj, R., Nisha S; V G. (2019). A rule-based approach for adaptive content recommendation in a personalized learning environment: An experimental analysis. *2019 IEEE Tenth International Conference on Technology for Education*, 138-141.

Razafinirina, W. G. M. T., Mahefa Abel; Dimbisoa. (2024). Pedagogical alignment of large language models (llm) for personalized learning: A survey, trends and challenges. *Journal of Intelligent Learning Systems and Applications*, *16*(4), 448-480.

Sakar, S. (2021). Personalized learning path generation in e-learning systems using reinforcement learning and generative adversarial networks. , 92-99.

Shaka, D. B. K. N., Martha; Carraro. (2023). Personalised programming education with knowledge tracing. , 47.

Smith, G., Brent; Linden. (2017). Two decades of recommender systems at amazon.com. *IEEE Internet Computing*, *21*(3), 12-18.

Song, X. (2024). Research on a personalised education recommendation system based on deep learning: From learning data to customised learning paths. *IEEE*, 147-152.

Stiubiener, W. V. R. M. C., Itana; Ruggiero. (2007). An approach to personalisation in e-learning. *IEEE Xplore*, 189-193.

Subramanian, M., Arun; Arora. (n.d.). Building your own recommender system.

Takyar, A. (2023). Neural networks: Architecture, applications, case studies, development and implementation.

Talha, H. U. I. S. A. M. I. T. F. M., Mian Muhammad; Khan. (2023). Deep learning in news recommender systems: A comprehensive survey, challenges and future trends. *Neurocomputing*, *562*, 126-881.

Trabelsi, Z. D. B. H. J., Mohamed; Chen. (2021). Neural ranking mmodels for document retrieval. *Information Retrieval*, *24*(6), 400-444.

Wangmei, S. Y. P. K. T. C. W., Chen; Zepeng. (2024). Applying machine learning algorithm to optimize personalized education recommendation system. *Journal of Theory and Practice of Engineering Science*, *4*(1), 102-108.

Yağcı, M. (2022). Educational data mining: Prediction of students' academic performance using machine learning algorithms. *Smart Learning Environments*, *9*(1), 1-19.

Yekollu, T. S. B. S., Roop Kumar; Bhimraj Ghuge. (2024). Ai-driven personalized learning paths: Enhancing education through adaptive systems. , 507-517.

Youssra, J. H. T. N. B., Bellarhmouch; Adil. (2022). A proposed architectural learner model for a personalized learning environment. *Education and Information Technologies*.

Yu, H. L. P., Puxuan; Fei. (2021). Cross-lingual language model pretraining for retrieval. *Proceedings of the Web Conference*, 1029-1039.

Zamani, M. C. W. B., Hamed; Dehghani. (2018). From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing. , 497-506.

zhihua, X. F. X. X. C. Y. C. W. Z., Cui; Xianghua. (2020). Personalized recommendation system based on collaborative filtering for iot scenarions. *IEEE Transactions on Services Computing*, *13*(4), 685-695.