ONTOLOGY AND LARGE LANGUAGE MODEL BASED INTELLIGENT TUTORING SYSTEM

U.H.L.P. SILVA

218547H

MSc in Artificial Intelligence

Department of Computational Mathematics
Faculty of Information Technology

University of Moratuwa
Sri Lanka

April 2024

DECLARATION

I declare that this is my own work, and this dissertation does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any other University or Institute of higher learning, and to the best of my knowledge and belief, it does not contain any material previously published or written by another person except where the acknowledgment is made in the text. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date: 2024-04-24

The above candidate has carried out research for the Masters dissertation under my supervision. I confirm that the declaration made above by the student is true and correct.

Name of Supervisor: Dr. Thushari Silva

Signature of the Supervisor: Date: 25/04/2023

DEDICATION

This research is dedicated to diligent and dedicated tutors whose passion for education has illuminated the path of countless students, including myself. Your tireless efforts, patience, and commitment to nurturing intellectual curiosity have left an indelible mark on our academic journey.

ACKNOWLEDGMENT

I extend my deepest gratitude to my supervisor, Dr. Thushari Silva, for her unwavering guidance, invaluable insights, and continuous support throughout the research process. Her expertise and encouragement have been instrumental in shaping this thesis.

I am immensely thankful to the lecturers who imparted their knowledge and expertise during my academic journey at the University of Moratuwa. Their dedication, passion, and commitment to excellence have been instrumental in shaping my understanding of the subjects covered in this thesis.

I would like to express my appreciation to my family for their unconditional love, patience, and encouragement during this journey. Their unwavering support has been my source of strength and motivation.

Finally, I acknowledge the participants of this study for their valuable contributions, without which this research would not have been possible.

Thank you all for being part of this journey.

ABSTRACT

The evolution of online education systems traces back to distance education methodologies of the 18th and 19th centuries, which laid the foundation for independent learning. The late 20th century saw the convergence of technological breakthroughs, including personal computers and the internet, leading to the development of computer-assisted instruction (CAI) and later, learning management systems (LMS). Online education systems evolved to embrace asynchronous learning, multimedia capabilities, and social learning principles, experiencing further growth during the COVID-19 pandemic.

Prominent online learning platforms like Coursera, Udemy, and Skillshare have revolutionized access to education. these platforms have limitations regarding course quality consistency, instructor engagement, and accessibility. Artificial intelligence (AI) is transforming online education through Intelligent Tutoring Systems (ITS) and Expert Systems (ES). ITSs provide personalized and adaptive learning experiences, while ES assists with decision-making and problem-solving tasks. However, existing ITSs face limitations in scalability, automation, and feedback provision.

In this research, the hypothesis posits that an ITS can be developed effectively using Ontology and LLMs. This concept is inspired by the unique capabilities of Ontology in structuring human knowledge and LLM in possessing strong Natural Language Processing (NLP) capability and general knowledge.

The proposed system generates teaching plans, resources, assessments, quizzes, grades, progress tracking, and personalized feedback. Teaching plans outline schedules and topics, while resources enhance learning. Assessments and quizzes evaluate comprehension, with grades providing insights. Progress tracking identifies strengths and weaknesses, facilitating targeted interventions. Personalized feedback guides improvement strategies.

The system is divided into four main components which are the domain knowledge model, tutoring model, student model, and user interface. These components are designed to be segregated, allowing modifications to one component without affecting the others.

the evaluation of the system is done using the focused approach and comparative analysis. Using the focused approach find out if the combination of ontology and LLM is solving the issues that can't be solved by ontology and LLM working individually in the system. Using comparative analysis find out the system is better than the selected similar systems for the experiment.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Description** **Page**

LIST OF TABLES

Table **Description** **Page**

LIST OF ABBREVIATIONS

Abbreviation	Description
AI	Artificial Intelligence
ITS	Intelligent Tutoring System
LMS	Learning Management System
NLP	Natural Language Processing
CRS	Classroom Response System
CAI	Computer-Assisted Instruction
MOOC	Massive Open Online Course
ES	Expert Systems
RDF	Resource Description Framework
OWL	Web Ontology Language
GPT-3	Generative Pre-trained Transformer 3
SQL	Structured Query Language
RDBMS	Relational Database Management System
UI	User Interface
DOM	Document Object Model

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Prolegomena

Education is a critical enabler for the development of an informed, competent, and enlightened society because it empowers people, promotes growth on both a personal and societal level, and advances knowledge across a wide range of subjects [1]. By offering tailored guidance, tailored instruction, and academic support, tutors play a critical role in education by assisting students in overcoming obstacles, realizing their full potential, and succeeding academically. However, because of several administrative responsibilities including assigning homework, grading, organizing students into groups, and giving tailored feedback, tutors are under a lot of time limitations. These laborious duties rob instructors of critical instructional time and prevent them from implementing more effective teaching strategies. In order to reduce these tedious chores and improve instructor effectiveness and efficiency, a solution must be investigated [2]. Fortunately, with the continuous advancement of technology, platforms have emerged with the potential to automate some of these administrative burdens. Online learning platforms like Course Era, Udemy, and Skillshare exemplify this progress, contributing to the overall improvement of the educational landscape [3] [4]. Furthermore, the field of Artificial Intelligence (AI) is rapidly integrating into the educational system, leading to the evolution of the Intelligent Tutoring System (ITS). Existing ITS platforms, such as Cognitive Tutor, Duolingo, and ALEKS, demonstrate the positive impact of this technology on empowering the educational system [5].

These platforms offer valuable support in enhancing educational experiences. However, it is crucial to acknowledge that current ITS solutions may have limitations. The scope of some existing systems might be restricted, and they may not achieve complete automation [6], [5]. This underscores the ongoing need for research and development efforts to explore and implement solutions that can alleviate the time constraints faced by educators, overcome the limitations of current systems, and ultimately, optimize teacher efficiency and effectiveness in the classroom. Achieving these goals will contribute not only to empowering educators but also to the advancement of the entire education system.

The background and motivation, problem description, aim, objectives, proposed solution, resources, and thesis structure are all presented in this chapter.

1.2 Background and Motivation

The educational and learning process has been happening since age. In ancient times, education was primarily transmitted through oral traditions, where elders passed on cultural values, practical skills, and religious teachings to younger generations. Furthermore, education was limited to certain social statuses [7]. The educators had the advantage of fewer students and more time to focus on individual students. Nevertheless, since there was no structured education system, knowledge passed to the student always depended on the educator, and there was no guarantee that students would get the correct amount of knowledge. Since educators were highly respected by society, there were people who were willing to work as assistants in the class.

Education is now readily accessible to everybody. As a result, Tutors have more students in their classes and need to devote more time to them. Additionally, a sizable portion of instruction is delivered online [8]. E-learning has experienced significant expansion, especially since the COVID-19 epidemic began in 2020, and restricted opportunities for in-person instruction for numerous educational institutions around the world [9]. Various Learning Management Systems (LMSs) are available to develop, manage, and distribute digital resources for face-to-face and online teaching. An LMS provides interaction between traditional teaching techniques and digital learning resources and simultaneously offers students personalized e-learning opportunities [10]. Furthermore, there are some online learning platforms such as Course Era, Udemy, and Skillshare [3]. These platforms commonly provide learners with structured learning experiences, often culminating in certificates upon completion of courses or specializations. Also offers specializations that group related courses around a specific topic or skill set and boasts a vast library of courses, encompassing a wide range of topics and skill sets as well. But the main issue of these platforms, is they are not fully automated [4].

Teaching methods and classrooms have undergone modifications and reforms in recent years due to advancements in computer hardware, software, and internet services. However, the use of AI in education has not yet caused a significant disruption. The Global AI in Education Market is expected to increase at a Compound

Annual Growth Rate (CAGR) of 47% from USD 537.3 million in 2018 to USD 3,683.5 million by 2023, according to a Research and Markets.com survey report [2]. Because of the teacher shortage in schools, Edwards, B. I. et al. experimented with using PERT (Physically – Embodied Robot Teacher) in the classroom. The Sota robot was used in classroom experiments by the writers. In the research paper, it was argued that the robot was small, legged, and portable and was capable of speech synthesis and voice recognition. It also engaged with students and administered a quiz in the classroom, to which students responded through the Classroom Response System (CRS). The robot also led a discussion with students about the quiz [11]. These authors contend that advancements in automation and AI will actually make people more human [8]. AI in education will decrease teacher workload, increase student-teacher interaction, facilitate successful learning experiences, help students identify their strengths, and foster creativity [12]. However, opposing viewpoints also exist. As an example, studies on AI see it as dangerous to shift teaching duties to machines [13].

The fast integration of AI into the educational system has resulted in the development of ITS. Examples of the platforms which are integrating AI, Cognitive Tutor, Duolingo, and ALEKS. Even with advancements these platforms also have some limitations [6]. Many systems are developed for specific subjects, lacking scalability and adaptability to diverse educational domains. Additionally, some systems are not fully automated, requiring manual intervention for content creation and administrative tasks [5]. There is a need for an approach to overcome these limitations and drawbacks of current systems as well as to automate the administrative tasks of tutors.

1.3 Problem in Brief

Existing online education systems have been identified as having limitations: they are subject-specific, lack scalability and automation, struggle with timely feedback, and burden tutors with administrative tasks. Here, the research problem is defined as creating an ITS based on ontology and LLM to overcome the above limitations.

1.4 Aim & Objectives

Aim

Develop an ontology and LLM based ITS to enhance the quality of education through intelligent, personalized, and efficient assistance.

Objectives

- A critical review of the evolution of education systems.
- In-depth study of technologies used in modern education.
- Design and develop ontology and connect ontology and LLM.
- Implement ITS based on the ontology and the LLM.
- Evaluation of the ITSs.

1.5 Proposed Solution

It has been hypothesized that the creation of an ITS and the achievement of efficient and effective solutions for education can be facilitated through the utilization of Ontology and LLMs. This idea has been inspired by the unique capabilities and advantages of Ontology and LLMs. The design of ontologies aims to convert human knowledge about the natural world and other subjects into computer-understandable structures, and new knowledge can be inferred using existing knowledge. Additionally, LLMs possess very good NLP capabilities and a good general knowledge base. By harnessing these technologies, an ITS can be developed, which is equipped with a vast repository of pedagogical knowledge and instructional strategies.

The process starts when the tutor creates the class. After that system creates the teaching plan (study plan) to follow. the system will follow the plan with the tutor and ensure the student gains the knowledge. The system will give quizzes to the student according to the teaching plan and measure the student's knowledge based on answers. Based on the student's knowledge, the system will decide to suggest study material to the student or give a more challenging quiz than the previous one. Same time system will adjust the teaching method based on the efficiency of the student.

When a student goes through this process, the system will track the student's progress and provide feedback to both the student and tutor to understand the current situation and change the teaching strategy if required.

Users of the system include students, tutors, and anyone who wants to check student progress.

Major features of the system are planning courses, smart content creation, generating assessments and quizzes, grading assessments/quizzes, tracking progress, providing feedback, adjust teaching strategies.

1.6 Resource Requirements

A computer (Processor: Intel Core i7, 16 GB DDR4, NVIDIA GeForce RTX 2060 6 GB) and tools (python, node.js, and react.js) for programming, a cloud service like AWS for hosting.

1.7 Structure of the Dissertation

The Introduction chapter covers an introduction to the area of research, its importance, major contributions/attractions, achievements, and issues. It is organized to present the background and motivation, problem in brief, aim & objectives, proposed solution, resource requirements, and structure of the dissertation.

The second chapter is Evolution and Challenges in Online Education Systems. This chapter gives a comprehensive literature review of online education systems. The literature review has been structured to cover the gestation of online education systems, and major developments in online education systems. Here we have also discussed the research gap.

The third chapter of the dissertation is Technology Adapted, and it presents our in-depth study of Ontology and LLM technology by highlighting its relevance to this research. Covered main topics are ontology modeling, large language models, integration of ontology and LLM, backend infrastructure, and user interface.

Proposed Approach is the fourth chapter, and it presents our Ontology and LLM-based approach for an ITS, it covers our hypothesis, input, output, process, features, and users.

The Design chapter presents the architecture design of our system covering top-level architecture, domain knowledge model, tutoring model, student model, and user interface.

The next chapter is Implementation, and this chapter describes the Implementation of the system with reference to the design. It covers the Overall implementation, Domain knowledge model implementation, Tutoring model implementation, Student model implementation, and User interface implementation.

The Evaluation chapter evaluates the system by covering experimental design, benchmark, and result and analysis.

The final chapter is Conclusion and Further Work. This chapter discusses the conclusion and further works by covering the Overall conclusion, Objective-wise achievements, Limitations, and Further work.

1.8 Summary

The area of our research was introduced in this chapter. The four objectives and key points of our literature review were discussed. Following that, our problem was defined briefly along with the proposed solution. Subsequently, resources were identified as necessary for the research. In Chapter 2, a comprehensive literature review on online education systems is presented.

CHAPTER 2

EVOLUTION AND CHALLENGES IN ONLINE EDUCATION SYSTEMS

2.1 Introduction

Chapter 1 presented the overall picture of this dissertation concerning the research problem and the essentials of the solution. According to that tutors have a lot of administrative work ahead of them, including making lecture modules, assigning homework, grading, monitoring progress, and giving feedback [2]. This chapter undertakes a comprehensive literature review on online education an extensive literature review on online education systems is conducted in this chapter. The landscape of education is undergoing a significant transformation, driven by the ever-growing influence of technology. Online education systems, encompassing a diverse range of platforms and methodologies, have emerged as a powerful force in this evolution. They offer a flexible and accessible alternative to traditional classroom-based learning, opening doors for individuals of all backgrounds and geographical locations to pursue knowledge and skills. This chapter delves into the fascinating story of online education systems, tracing their historical roots, charting their major developments, and critically examining the research gaps that remain to be addressed.

2.2 Gestation of online education systems

The story of online education systems stretches back further than one might imagine. The seeds of this revolution were sown long before the advent of the internet, with the emergence of distance education methodologies in the 18th and 19th centuries. Pioneering efforts included correspondence courses offered by universities, which allowed students to learn independently through mailed materials and examinations [14].

The late 20th century witnessed a convergence of technological breakthroughs that truly laid the foundation for the online education systems we know today. The development of the personal computer and the subsequent rise of the internet in the 1970s and 1980s marked a turning point. These advancements facilitated the creation of online learning platforms that transcended the limitations of previous technologies [15].

One of the earliest forms of online education in this era was computer-assisted instruction (CAI). CAI programs offered interactive exercises and drills delivered through personal computers. While these programs lacked the human element of traditional classrooms, they provided a self-paced learning experience and allowed for individualized instruction to a certain extent [16].

The widespread adoption of the Internet in the 1990s further accelerated the development of online education systems. With the advent of LMS, a centralized platform for administering online courses, dispensing educational resources, and enabling instructor-student interactions was made available. These systems offered a more structured approach to online learning, enabling instructors to create and deliver curriculum online, assess student progress, and provide feedback [9].

Massive Open Online Courses (MOOCs) had a spike in popularity in the late 1990s and early 2000s. MOOCs provide free or inexpensive access to top-notch instructional materials from esteemed colleges and organizations all around the world. Due to the democratization of education, formerly unattainable online learning possibilities became available to a global audience [17].

As online education systems matured, the focus shifted from simply replicating traditional classroom experiences online to embracing the unique affordances of the digital learning environment. Asynchronous learning emerged as a prominent feature, allowing students to access course materials and complete assignments at their own pace, a stark contrast to the fixed schedules of traditional classrooms [8].

The learning process was further enhanced by the addition of multimedia features to online learning systems. To create a more dynamic and engaging learning environment, tutors could use interactive simulations, video lectures, and collaborative learning tools [18].

The turn of the 21st century witnessed the integration of social learning principles into online education systems. Students could communicate with each other, exchange ideas, and work together on projects through discussion boards, online chat rooms, and video conferencing facilities, which helped to build a sense of community and encourage peer-to-peer learning. Furthermore, online learning has experienced tremendous growth, particularly since the COVID-19 pandemic, which necessitated remote learning for many institutions. With the flexibility and convenience of online learning environments, students can learn almost anywhere at their own pace [9].

This historical trajectory of online education systems reveals a continuous process of innovation and adaptation driven by technological advancements.

2.3 Major developments in online education systems

The landscape of education has undergone significant transformations in recent decades, driven by factors such as increased accessibility, the rise of online learning, and the integration of technology. This section will delve deeper into these key developments, exploring their impact on educators, students, and the overall learning experience.

“Course Era” is one of the most popular online learning platforms. Coursera provides learners with a structured learning experience, often culminating in certificates upon completion of courses or specializations. Coursera goes beyond individual courses by offering specializations that group related courses around a specific topic or skill set. These specializations provide a structured learning path, guiding learners through a comprehensive program of study [19]. Upon successful completion of a Coursera specialization, learners can earn certificates that are recognized by industry leaders. Many Coursera courses integrate project-based learning activities, allowing learners to apply their knowledge and skills to real-world scenarios. This practical approach fosters deeper understanding and hones the ability to solve problems effectively. While Coursera offers advantages it has some limitations as well. platform has potential issues with course quality consistency across different providers. Also, the platform is not fully automated. Nevertheless, Coursera has had a significant impact on online education by providing accessible and high-quality learning opportunities to a global audience [3].

Another prominent online learning platform, "Udemy", operates as an online marketplace where instructors can create and sell their own online courses. Udemy boasts a vast library of courses, encompassing a wide range of topics and skill sets. This diversity allows learners to find courses that align with their specific interests and career aspirations [4]. The majority of Udemy courses utilize video lectures as the primary mode of instruction. These lectures are often created and delivered by subject matter experts, providing learners with valuable insights and practical guidance. However, this platform has some limitations as well. Since instructors independently create Udemy courses, the quality and consistency of content can vary. It's crucial for learners to carefully evaluate instructor credentials and course reviews before enrolling. Also, while some Udemy instructors offer interaction through Q&A forums or discussion boards, the level of engagement can vary. Learners seeking a more interactive learning experience may need to look elsewhere. Furthermore, Udemy primarily focuses on individual learning through video lectures and downloadable resources. Collaboration features like group projects or real-time discussions are often limited or absent. Real-time feedback is also absent in this approach [20].

Skillshare is another online learning platform. The platform is a subscription-based platform featuring creative and practical skill-building courses. The platform caters to individuals seeking to develop or enhance their skills in various creative fields, ranging from animation and design to photography and music production. Skillshare emphasizes project-based learning and fosters a community-driven environment to encourage interaction and collaboration among learners. Many courses utilize a project-based learning approach, where learners work on creative projects throughout the course to solidify their understanding and showcase their skills. Skillshare has some drawbacks as well. It requires a monthly or yearly subscription fee to access its full course library. This can be a barrier for learners with limited budgets. While Skillshare offers a free trial period, the majority of its courses are only accessible with a paid subscription. The project-based learning approach may not be ideal for learners seeking a strong theoretical foundation in a particular creative skill [21].

<u>AI for online education systems</u>

With advancement of the technology, AI is rapidly transforming the landscape of online learning platforms, making the learning experience more personalized, engaging, and effective [22]. The integration of AI has given rise to two prominent educational technology categories such as ITS and Expert Systems (ES).

ITSs represent the primary application of AI in education. ITSs are designed to provide personalized and adaptive learning experiences to students. By leveraging AI algorithms, these systems can analyze individual student performance, identify areas of strength and weakness, and tailor instructional content accordingly. Through interactive exercises, feedback mechanisms, and dynamic adjustments, ITSs aim to optimize the learning process, fostering greater comprehension and retention of educational material [23].

In the realm of ITS, various systems have been developed to enhance learning experiences. One such system is the "Cognitive Tutor". The system driven by AI is used to deliver individualized mathematics training [24]. "Duolingo" is an additional AI-powered language learning application that employs this technology to customize lessons according to each user's learning trajectory [6]. Another adaptive learning system is called "ALEKS," which evaluates students' knowledge, modifies the difficulty of the questions based on that information, and creates individual learning paths for each student. However, the system's primary areas of concentration are mathematics and after-school activities [5].

Several shortcomings are revealed by a thorough analysis of the current ITS. A lot of systems are made for specific subjects and aren't flexible enough to scale to different educational fields. Furthermore, some systems need personal intervention for administrative and content development duties because they are not entirely automated. Additionally, current systems might find it difficult to give students timely, individualized feedback, which would impede their ability to learn.

ES, on the other hand, are highly developed software programmers that mimic human experts' decision-making processes in particular fields. These systems handle complicated issues and offer recommendations or counsel appropriate for experts by utilizing rule-based reasoning, knowledge bases, and inference engines. While ES has diverse applications across various fields, including medicine, finance, and engineering, its role within education is typically more focused on assisting educators and students with specialized knowledge and problem-solving tasks [25].

Despite the significant contributions of ES to the broader landscape of AI, ITSs hold particular significance within the realm of education. Their ability to adapt instruction to the unique needs and learning styles of individual students has made them invaluable tools for enhancing educational outcomes and facilitating more personalized learning experiences. As technology continues to evolve, the ongoing development and refinement of ITSs promises to further revolutionize the educational landscape, empowering learners and educators alike with innovative AI-driven solutions [26].

2.4 Research gap and conclusion

A comprehensive review of existing online education systems reveals several key limitations. Notably, many systems are designed for specific subjects, hindering their scalability and adaptability to various educational domains. This lack of flexibility restricts their broader application and potential impact. Additionally, some systems are not fully automated, requiring significant human intervention for content creation and administrative tasks. This dependence on manual work not only increases the workload for educators but also limits the overall efficiency and effectiveness of these platforms.

Furthermore, a prevalent limitation identified in the research is the struggle of existing systems to provide timely and personalized feedback to students. This lack of individualized support can hinder student learning progress and reduce the effectiveness of the online learning experience. As identified in the literature, tutors spend a significant amount of time on administrative tasks such as creating lectures, assigning work, grading, tracking progress, and providing feedback.

Considering the time-consuming tasks of tutors and the limitations of existing applications, it becomes clear that there is a pressing need for an approach that can overcome the shortcomings of existing online education systems. This approach not only automates time-consuming administrative tasks but also offers a more intelligent and personalized learning experience. This strategy will deliver tutors useful information on student progress, learning patterns, and areas of difficulty by utilizing AI-driven insights and analytics. Equipped with this information, tutors can adjust their methods of instruction, interventions, and methods of providing feedback in order to maximize the learning results of their students.

Furthermore, this approach incorporates adaptive learning capabilities. By analyzing student learning styles, preferences, and proficiency levels, the system can personalize learning pathways and content. This ensures that students receive targeted instruction and support, leading to a more engaging and effective learning experience with improved outcomes.

2.5 Summary

A comprehensive literature review on online education systems is presented in this chapter. The gestation of online education systems and Major developments in online education systems are discussed and identified the research gap as a limitation of existing online education systems. The technological framework and adaptations employed in the development will described in the next chapter.

CHAPTER 3

TECHNOLOGY ADAPTED

3.1 Introduction

Chapter 2 covered the literature review on online education systems by covering the Gestation of online education systems, Major developments in online education systems, and finally the Research gap and conclusion. This chapter presents the technological framework and adaptations employed in the development. This outlines the key technologies integrated into the system and discusses their roles in facilitating effective ITS by covering ontology modeling, integration of ontology and LLM, backend infrastructure, and user interface.

3.2 Ontology Modeling

Ontology, in a general sense, refers to the philosophical study of existence, being, or reality. It seeks to systematically categorize and analyze the fundamental concepts and relationships that underlie our understanding of the world. In the realm of information science and AI, ontology takes on a more specific meaning as a formal representation of knowledge within a particular domain. Here, ontology serves as a structured framework for organizing and representing concepts, entities, and their interrelationships in a coherent and meaningful way. By defining a common vocabulary and specifying the properties and relationships between different elements, ontology enables shared understanding and facilitates interoperability among different systems and applications. Ontologies play a crucial role in various fields, including information retrieval, knowledge management, and semantic web technologies, by providing a formal basis for reasoning, inference, and knowledge representation.

The adaptation of ontology in ITS serves as a foundational element to enhance the effectiveness and intelligence of the tutoring process. Ontology provides a structured framework for organizing domain-specific knowledge, defining concepts, and specifying the relationships between them. By incorporating ontology into an ITS, we can achieve several key objectives. Firstly, ontology enables a deeper understanding of the subject matter by formalizing and structuring domain knowledge, thereby facilitating more accurate and contextually relevant tutoring interactions. Secondly, ontology supports personalized instruction by allowing the system to tailor its responses and instructional strategies to the individual learner's profile and needs. Thirdly, ontology-based reasoning enhances the system's ability to adapt and evolve over time, as it can incorporate new knowledge and adjust its instructional approach accordingly. Overall, the integration of ontology in ITS not only enriches the tutoring experience but also enables more intelligent and adaptive support for both tutors and students.

Ontology modeling plays a fundamental role in structuring knowledge and defining the relationships between concepts within the tutoring domain. The development of an ontology involves the creation of a formal representation of knowledge, typically organized in a hierarchical structure with defined properties and relationships. In the context of the ITS, ontology modeling serves several purposes:

- **Conceptual Framework**: The ontology establishes a conceptual framework that encapsulates the domain knowledge relevant to the tutoring tasks. This framework defines the key concepts, their properties, and the relationships between them.
- **Domain Understanding**: By formalizing domain knowledge into an ontology, the tutoring system gains a deeper understanding of the subject matter. This understanding enables more effective reasoning and decision-making processes during tutoring interactions.
- **Personalization**: Ontologies facilitate personalized tutoring experiences by enabling the system to adapt its instructional strategies based on the individual learner's profile and preferences. Personalization is achieved through the semantic interpretation of learner data and the application of inference rules defined within the ontology.

The development process outlines the steps taken to develop the ontology, including domain analysis, concept identification, and ontology construction. When discussing the methodologies used to ensure the ontology's completeness, coherence, and scalability. When detailing the implementation of the ontology within the tutoring system it includes Protégé as ontology editor, Pellet as reasoning engine, RDF and OWL as ontology languages, and Apache Jena Fuseki as SPARQL endpoint.

3.3 Large Language Models - LLMs

LLMs represent a transformative breakthrough in NLP, revolutionizing how machines understand and generate human language. These models, typically built using deep learning architectures such as transformers, are trained on vast amounts of text data, enabling them to capture complex linguistic patterns and semantic relationships. LLMs exhibit remarkable capabilities in tasks such as language generation, text completion, sentiment analysis, and machine translation. By leveraging pre-training techniques followed by fine-tuning on specific tasks or domains, LLMs demonstrate remarkable versatility and adaptability across a wide range of applications. However, their immense size and computational requirements pose challenges in terms of training and deployment. Ethical considerations also arise concerning potential biases in training data and the responsible use of language generation capabilities. Nonetheless, LLMs continue to drive innovation in natural language understanding and generation, promising profound implications for communication, content creation, and human-computer interaction.

The integration of LLMs into ITS represents a pivotal advancement in enhancing the tutoring experience. LLMs excel in natural language understanding and generation, enabling more seamless and intuitive communication between the system and users. By leveraging LLMs, the system can comprehend and respond to learner queries, provide explanations, offer feedback, and generate educational content in a manner that closely mimics human-like interaction. Moreover, LLMs facilitate personalized instruction by analyzing learner responses and tailoring feedback and instructional materials to individual needs and preferences. Their adaptability and versatility allow the system to dynamically adjust its instructional strategies based on real-time learner interactions, promoting deeper engagement and more effective learning outcomes. Overall, the integration of LLMs empowers the system to deliver more personalized, adaptive, and interactive tutoring experiences, significantly enhancing its efficacy and utility in supporting students' educational journey.

By selecting the OpenAI LLM model GPT-3 as the LLM model the system can employed for various purposes:

- **Natural Language Understanding**: LLMs excel at understanding and generating natural language text, enabling seamless communication between the system and the users (tutors and students). They can comprehend and respond to queries, provide explanations, and offer feedback in natural language.

- **Content Generation**: LLMs can generate educational content, including explanations, examples, and practice questions, based on the input provided by tutors and students. This capability enhances the richness and diversity of instructional materials available within the system.
- **Adaptive Feedback**: LLMs can analyze learner responses and provide adaptive feedback tailored to the individual's strengths, weaknesses, and learning objectives. By leveraging contextual understanding and semantic coherence, the system can deliver feedback that is relevant and actionable.

3.3 Integration of Ontology and LLM

The integration of ontology and LLM technologies forms the backbone of the System's intelligence and adaptability. By combining the structured knowledge representation offered by ontologies with the NLP capabilities of LLMs, the system achieves a synergistic effect:

- **Semantic Reasoning**: The ontology provides a semantic framework for interpreting and contextualizing the input/output generated by the LLM. This semantic grounding enables more robust reasoning and inference capabilities, enhancing the system's ability to understand and respond appropriately to tutoring interactions.
- **Personalized Instruction**: The integration of ontological reasoning with LLM-based natural language understanding enables the system to deliver personalized instruction tailored to the unique needs and preferences of each learner. The system can dynamically adjust the instructional content and delivery based on the learner's profile, learning style, and progress.
- **Continuous Improvement**: Through ongoing interaction with tutors and students, the system can continuously refine its ontology and LLM components, incorporating new knowledge, refining inference rules, and improving language understanding capabilities. This iterative process of improvement ensures that the system remains up-to-date and adaptive to evolving educational needs.

3.4 Backend Infrastructure

The Backend Infrastructure of the system leverages Node.js and MySQL technologies to support its operations effectively. Node.js serves as the runtime environment for executing server-side JavaScript code, offering scalability and high performance for handling concurrent requests. With its event-driven architecture and non-blocking I/O operations, Node.js enables the system to efficiently manage user interactions, data processing, and communication with external services. Furthermore, Node.js's extensive ecosystem of modules and libraries provides developers with a rich set of tools for building robust backend functionalities, such as handling authentication, managing sessions, and integrating with other systems.

Complementing Node.js, MySQL is employed as the relational database management system (RDBMS) to store and manage structured data within the system. MySQL offers reliability, performance, and scalability for handling large volumes of data generated by user interactions, content management, and system operations. Through its support for SQL (Structured Query Language), MySQL enables developers to define complex queries, manage transactions, and ensure data integrity within the system. Additionally, MySQL's features such as indexing, caching, and replication contribute to optimizing database performance and ensuring smooth operation even during peak usage periods. By leveraging Node.js and MySQL technologies in its Backend Infrastructure, the system achieves robustness, scalability, and efficiency in supporting intelligent tutoring for tutors and students.

3.4 User Interface

The User Interface (UI) of the System is powered by React.js, a popular JavaScript library for building interactive and dynamic user interfaces. React.js excels in creating reusable UI components, enabling developers to efficiently manage and maintain complex UI structures within the system. With its component-based architecture, React.js promotes modularity and scalability, allowing for the creation of highly responsive and intuitive interfaces. This facilitates seamless interactions between users and the system, enhancing the overall user experience.

Moreover, React.js's virtual DOM (Document Object Model) abstraction optimizes UI rendering performance by selectively updating only the components that have changed, minimizing unnecessary re-renders and improving responsiveness. This feature is particularly advantageous in the context of the system, where real-time communication and updates are essential for effective tutoring assistance. Additionally, React.js's ecosystem provides a wealth of tools, libraries, and community support, enabling developers to implement advanced UI features, state management solutions, and integration with other technologies seamlessly. Overall, React.js serves as a robust foundation for the User Interface of the system, empowering users with a modern, intuitive, and efficient platform for delivering personalized learning.

3.7 Summary

The technological framework and adaptations employed in the development of the system were presented in this chapter. The main technologies presented as Ontology with OWL and RDF, LLM as OpenAI GPT-3, backend infrastructure with Node.js and MySQL, and user interface with React.js. The Novel approach to the system will be described in the next chapter.

CHAPTER 4

PROPOSED APPROACH

4.1 Introduction

Chapter 3 justified the suitability of the Ontology and LLM as the technology adapted for developing an ITS. This chapter presents our Ontology and LLM-based approach to building an ITS, named OLMxITS, an acronym for Ontology and LLM-based ITS. This chapter presents the OLMxITS approach by covering our hypothesis, input, output, process, features, and users. Among others, the process specifically identifies the task within OLMxITS which provides the insight into design of the solution.

4.2 Hypothesis

It has been hypothesized that an ITS can be created and efficiency and effective solutions for education can be achieved through the utilization of Ontology and LLM. This idea has been inspired by the unique capabilities and advantages of Ontology and LLMs. Ontologies are designed to convert human knowledge about the natural world and other subjects into computer-understandable structures, and new knowledge can be inferred using existing knowledge. Additionally, LLMs possess very good NLP capability and a good general knowledge base. By harnessing these technologies, ITS can be developed, which possesses a vast repository of pedagogical knowledge and instructional strategies.

4.3 Input

The system collects essential user details for account creation and identification purposes. These details include the user's email address, first name, last name, user role (whether they are a student or a tutor), and password. With these inputs, the system can distinguish between different users and their roles within the platform, ensuring personalized access and appropriate permissions. Additionally, the user login process requires users to provide their email address and password, serving as authentication credentials to access their accounts securely.

Furthermore, when creating classes, the system gathers specific details to tailor the educational experience. This includes curriculum, subject, grade, and component specifications, alongside the tutor's preferred number of hours for the class. These inputs help customize the learning journey for students and facilitate effective teaching strategies. Moreover, assessments and quizzes within the system allow users to select answers, enabling them to demonstrate their understanding of the material. Figure 1 shows the summarized view of the above inputs.

4.4 Output

The system generates various outputs to support the educational process comprehensively. Firstly, it formulates a teaching plan, or study plan, based on class details and desired teaching hours. This plan outlines the schedule and allocation of time for each topic, providing a structured framework for both tutors and students to follow. Additionally, the system provides teaching resources as PDFs tailored to each topic within the curriculum. These resources enhance understanding and engagement, enriching the learning experience.

Furthermore, the system offers assessments and quizzes to evaluate student comprehension. Upon completion, it provides grades for these assessments and quizzes, offering insights into student performance and proficiency levels. Leveraging this data, the system tracks progress over time, identifying strong and weak areas in each topic. With this information, it suggests teaching strategies personalized to address individual learning needs effectively. Moreover, the system delivers personalized feedback to students, offering guidance for improvement and reinforcement of concepts. These outputs collectively empower tutors to adapt their teaching methods and facilitate targeted interventions, fostering continuous academic growth and achievement. Figure 1 shows the summarized view of the above outputs.

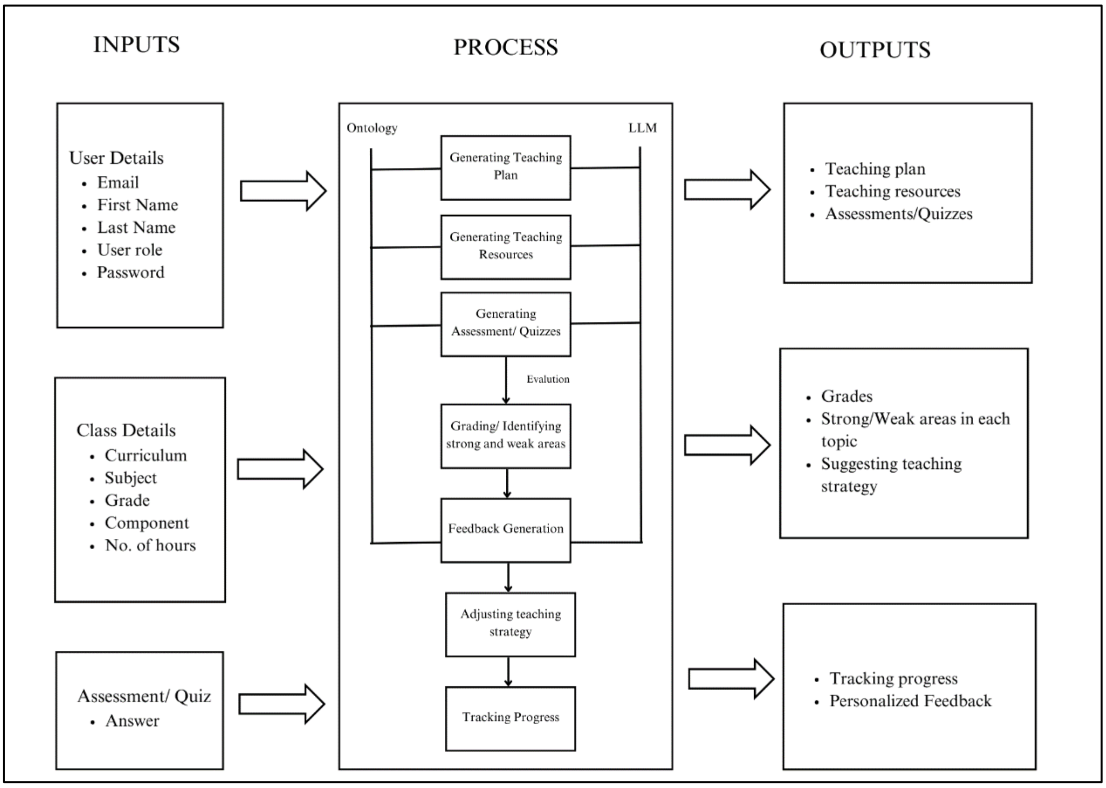

Figure 1: Inputs and Outputs

4.5 Process

In OLMxITS, the process is initiated by student account creation. Accounts for students can be created by every tutor in the system. When a student account is created, the student's email address, first name, last name, and password for student login need to be provided. After the account is created, the password can be shared with the student by the tutor. The system allows students to log in using their email addresses and the passwords shared with them.

The subsequent step involves the creation of the class. To initiate the class creation process, one of the available students and the curriculum, subject, grade, and component for the class need to be selected by the tutor. Additionally, the tutor is required to input their preferred total number of hours for the class. Upon the creation of the class by the tutor, the teaching plan for the class is generated by the system. During the generation of the teaching plan, the preferred total number of hours is utilized by the system to determine the amount of time allocated for studying each topic.

Following the creation of the teaching plan, the system proceeds to ascertain the initial activity as per the teaching plan. This activity may consist of a diagnostic assessment intended to evaluate the student's comprehension of the subject area, or it may entail delving into the first topic designated for the class. In the event that a diagnostic assessment is the initial activity, the system automatically generates the assessment, encompassing questions spanning all areas within the subject domain, with the objective of gauging the student's existing knowledge. Conversely, if the first topic is the designated activity, the system generates corresponding content in PDF format. This content is made accessible to students for viewing or downloading at their discretion.

Following the establishment of the plan, the subsequent phase entails the implementation of said plan by the student and tutor. Should the initial activity involve a diagnostic assessment, students are tasked with its completion. Given the evaluative nature of the assessment, it is accompanied by a predetermined time frame within which students must endeavor to address all posed questions. Upon either the student's completion of all questions or the expiration of the allotted time, the assessment is automatically submitted to the system for analysis. Subsequently, the system proceeds to assess the submitted answers and assign grades, thereby discerning the student's proficiency level in the subject matter. Based on this assessment, the system generates tailored content for the first topic, considering the student's demonstrated level of knowledge, and subsequently proposes the most suitable teaching strategy for the initial class to the tutor.

Following the teaching of the first topic to the student utilizing the generated content, the tutor proceeds with the instructional process. Once the student has assimilated the topic, they may advance to the subsequent activity, which is a Formative assessment of the preceding topic, referred to within the system as a quiz. Notably, this quiz is devoid of any time constraints, affording students the opportunity for multiple attempts and practice sessions. During successive attempts, the system ensures that identical questions are not replicated, and adjusts the difficulty level of questions based on the student's performance in previous attempts, potentially presenting questions of lesser, greater, or equivalent difficulty levels.

Diverging from traditional assessments, the quiz format provides immediate feedback to students upon submission of their answers, indicating the correctness of their responses and furnishing explanations delineating the rationale behind each correct or incorrect answer. In instances of incorrect responses, elucidation is provided on how students can arrive at the correct answers. Through perusal of these explanations, students can augment their comprehension and subsequently furnish correct responses in subsequent attempts, should analogous questions recur.

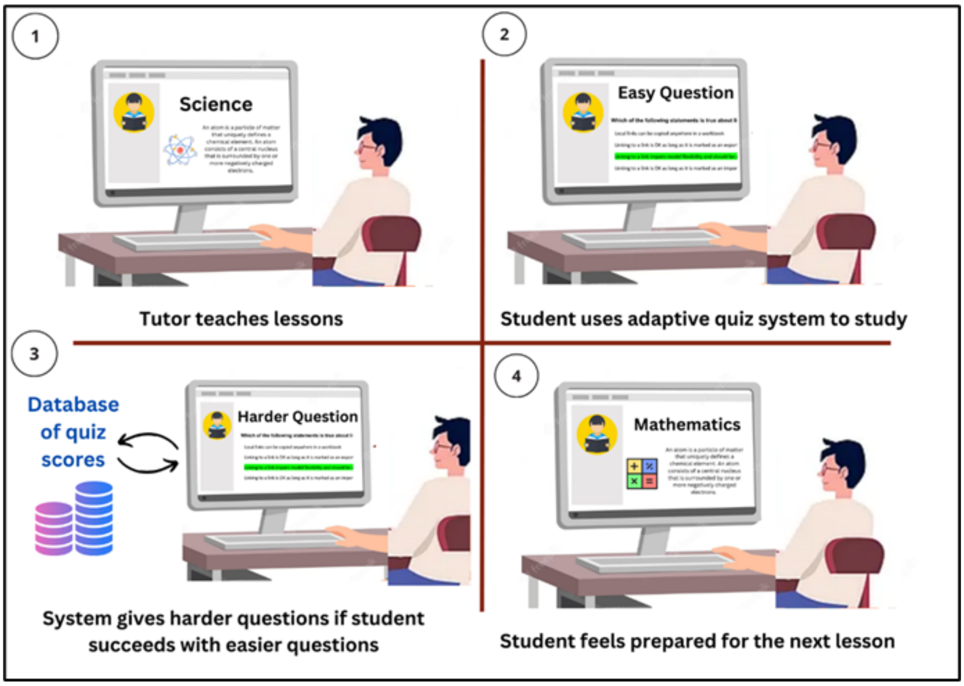

Figure 2: Example scenario.

Upon completion of the quiz, wherein students provide answers to all posed questions, the system undertakes the grading process, discerning the student's areas of strength and weakness. Leveraging this data, the system formulates personalized feedback for the student, thereby facilitating targeted improvement strategies. Moreover, contingent upon the student's proficiency level, the system evaluates the necessity for adjustments to the current teaching strategy. Should modifications be deemed requisite, the system proffers suggestions to the tutor accordingly.

Throughout multiple attempts, questions are tailored to fortify weak areas and incorporate more challenging queries in areas of demonstrated proficiency. Through iterative engagement, students can enhance their overall grades and receive pertinent feedback conducive to academic advancement.

Upon the completion of several topics and quizzes, students progress to a summative assessment, which is designed to evaluate their overall progress and readiness for examinations. Assessments can only be attempted once, thus it is recommended that students engage in related quizzes multiple times to enhance their knowledge before undertaking the assessment.

These assessments are subject to time limits for completion, compelling students to endeavor to respond to all questions within the designated timeframe. Upon completion of all questions or expiration of the time limit, the assessment is graded by the system. Similar to quizzes, the system discerns the student's strengths and weaknesses, subsequently generating personalized feedback. If any adjustments in teaching strategy are deemed necessary, recommendations will be communicated to the tutor. Finally, the overall grade and feedback for the class are updated according to student progress. While students complete topics and undertake quizzes and assessments, the system monitors their progress and presents it to both the tutor and the student in the form of a graph.

Upon the completion of all topics, quizzes, and assessments, the class is concluded. At this juncture, if students achieve a good overall grade, it serves as an indicator of their knowledge level in the subject and their readiness for the upcoming exam.

4.6 Features

The major features of OLMxITS encompass several key functionalities.

These include:

- Generation of resources tailored to the class, facilitating comprehensive learning experiences.
- Development of assessments, including diagnostic assessments aimed at gauging initial understanding and summative assessments focused on overall progress and exam readiness.
- Creation of quizzes, serving as formative assessments to enable ongoing evaluation and learning reinforcement.
- Grading of assessments and quizzes, providing objective evaluation of student performance.

- Identification of students' areas of strength and weakness, enabling targeted instructional interventions.
- Generation of personalized feedback, offering tailored guidance for individual student improvement.
- Suggesting teaching strategies based on student performance data, and providing recommendations for optimizing instructional effectiveness.
- Tracking of students' progress over time, allowing for continuous monitoring and adjustment of instructional strategies.

4.7 Users

Users of the OLMxITS include students, tutors, and anyone who wants to check student progress.

4.8 Summary

The approach to OLMxITS was presented in this chapter. The input has been defined, and, more importantly, nine major tasks have been identified namely creating teaching plans, generating resources, creating assessments and quizzes, grading assessments and quizzes, identifying strong and weak areas, generating personalized feedback, tracking progress, and suggesting teaching strategies within OLMxITS. The design of OLMxITS will be described in the next chapter with reference to the process identified here.

CHAPTER 5

DESIGN

5.1 Introduction

Chapter 4 explained the approach in OLMxITS. Where identified nine major tasks namely creating teaching plans, generating resources, creating assessments and quizzes, grading assessments and quizzes, identifying strong and weak areas, generating personalized feedback, tracking progress, and suggesting teaching strategies. This chapter presents the architecture design of OLMxITS covering top-level architecture, domain knowledge model, tutoring model, student model, and user interface.

5.2 Top-level architecture

The system development plan involves dividing it into four main components, as depicted in Figure 4. These components are designed to be segregated, thereby permitting modifications to be applied to one component without necessitating alterations to the others.

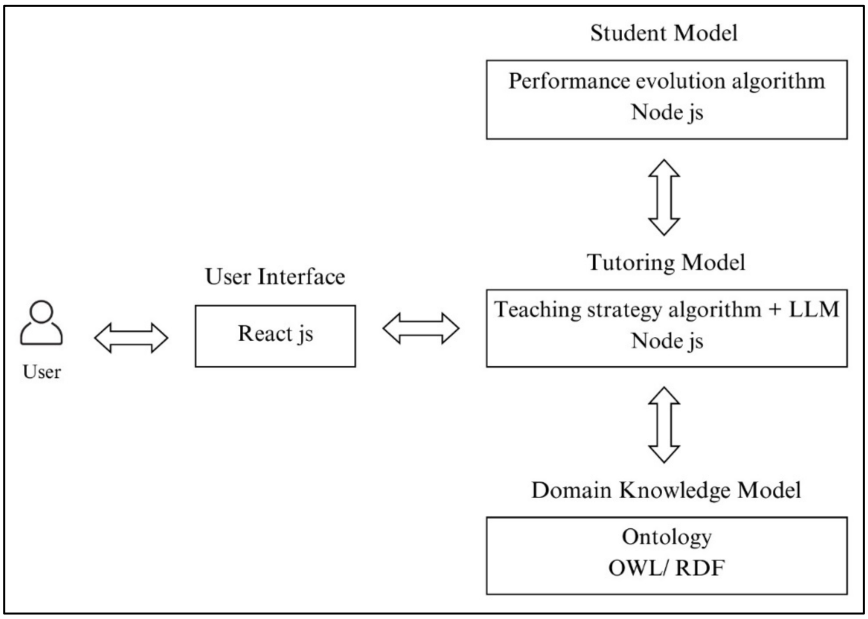

Figure 3: Summarized workflow

Two strategies can be employed by this model:

- Black Box Expert Model
- Glass Box Expert Model

The Black Box Expert Model is a strategy where the results are kept hidden from both students and other parts of the system. It's like a closed box where you can't see what's happening inside. This approach doesn't show how the expert makes decisions or solves problems. The focus is on providing the final results without revealing the process behind them.

The Glass Box Expert Model is a strategy where the results are visible to students. It's like having a transparent box where you can see what's happening inside. This approach aims to make the expert's thinking process clear to students. By showing the results and the steps taken to achieve them, students can better understand the reasoning behind the decisions made by the expert. It promotes transparency and helps students learn not only the final answers but also the thought process behind them.

The Glass Box Expert Model was chosen because it allows students to see the results. This way, the results can be seen, allowing students to understand how decisions are made and problems are thought through by the expert. Transparency and clarity are provided, which can be really helpful for students in learning and understanding the subject better.

5.3 Domain knowledge model

The model can be referred to by various names including Expert Model, Domain Expert, and Knowledge Model, which embodies expertise within a specific domain. It serves as the knowledge base of a system, encapsulating experts' understanding of a particular area. It also can be termed as Subject Model or Idealized Model, since it represents a refined and optimized version of domain knowledge, aiming to achieve specific targets. Additionally, it functions as a Correction Model, ensuring accuracy through error correction mechanisms. Ultimately, it acts as a repository of expertise, facilitating informed decision-making and understanding within the system's operational domain.

Within OLMxITS, this model serves as the backbone where the Ontology resides, containing a comprehensive array of subject-specific knowledge. It encompasses a vast array of educational components, including syllabi, topics, subtopics, minor topics, model questions, practice questions, quizzes, assessments, and insights into students' progress. Figure 5 shows all the main classes and subclasses inside the ontology.

Within this model, students rely on it as a guiding resource, leveraging its wealth of expertise to navigate learning experiences effectively. It facilitates OLMxITS in evaluating students’ actions and choices within programs, discerning their accuracy, and offering diagnostic insights into any errors. Through this process, the Domain Model empowers OLMxITSs to provide targeted feedback and support, aiding students in their educational journey.

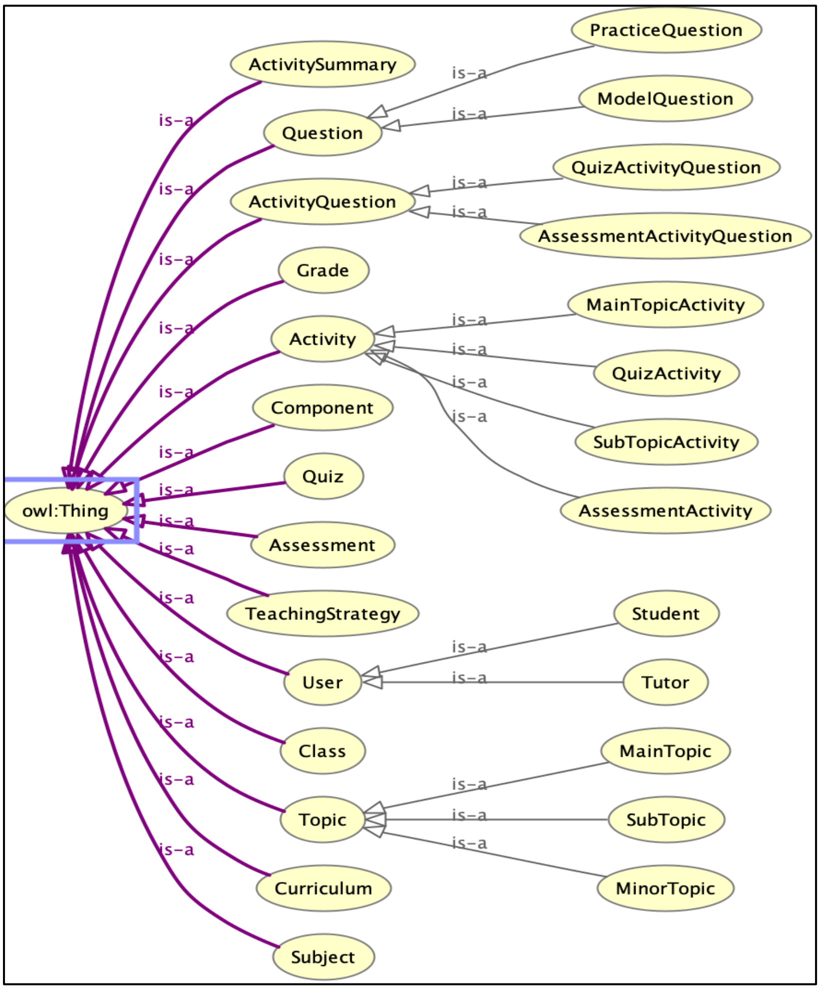

Figure 4: Main classes and subclasses in Ontology.

The Ontology within this model provides a structured framework for organizing and accessing this wealth of knowledge. Syllabi outlines the scope and sequence of learning, while topics and subtopics break down the curriculum into manageable units. Model questions offer examples of expected learning outcomes, guiding students in their understanding. Practice questions and quizzes provide opportunities for reinforcement and self-assessment, fostering active engagement and mastery of concepts. Assessments track students' progress and performance, offering valuable feedback to both students and tutors.

By centralizing these resources within OLMxITS, tutors and students alike benefit from a cohesive platform that supports personalized learning experiences. Students can access a diverse range of materials tailored to their needs, while instructors can monitor progress and adapt instruction accordingly. Ultimately, this model serves as a dynamic hub for knowledge dissemination and skill development within the realm of online education.

5.4 Tutoring model

This model can be referred to by various names including Instruction Model, Pedagogic Model, Tutoring Strategy Model, Teacher Model, Tutorial Knowledge Model, and Instructor Model. It serves as the backbone for implementing teaching strategies within OLMxITS. This model is responsible for encoding and applying various instructional methodologies and pedagogical approaches within the OLMxITS.

By leveraging this model, OLMxITSs can effectively deliver instruction by employing three main teaching strategies which are Mastery learning, Discovery learning, and Guided practice (Figure 6). These three strategies are accompanied by two types of assessments which are Formative assessments and Summative assessments (Figure 7).

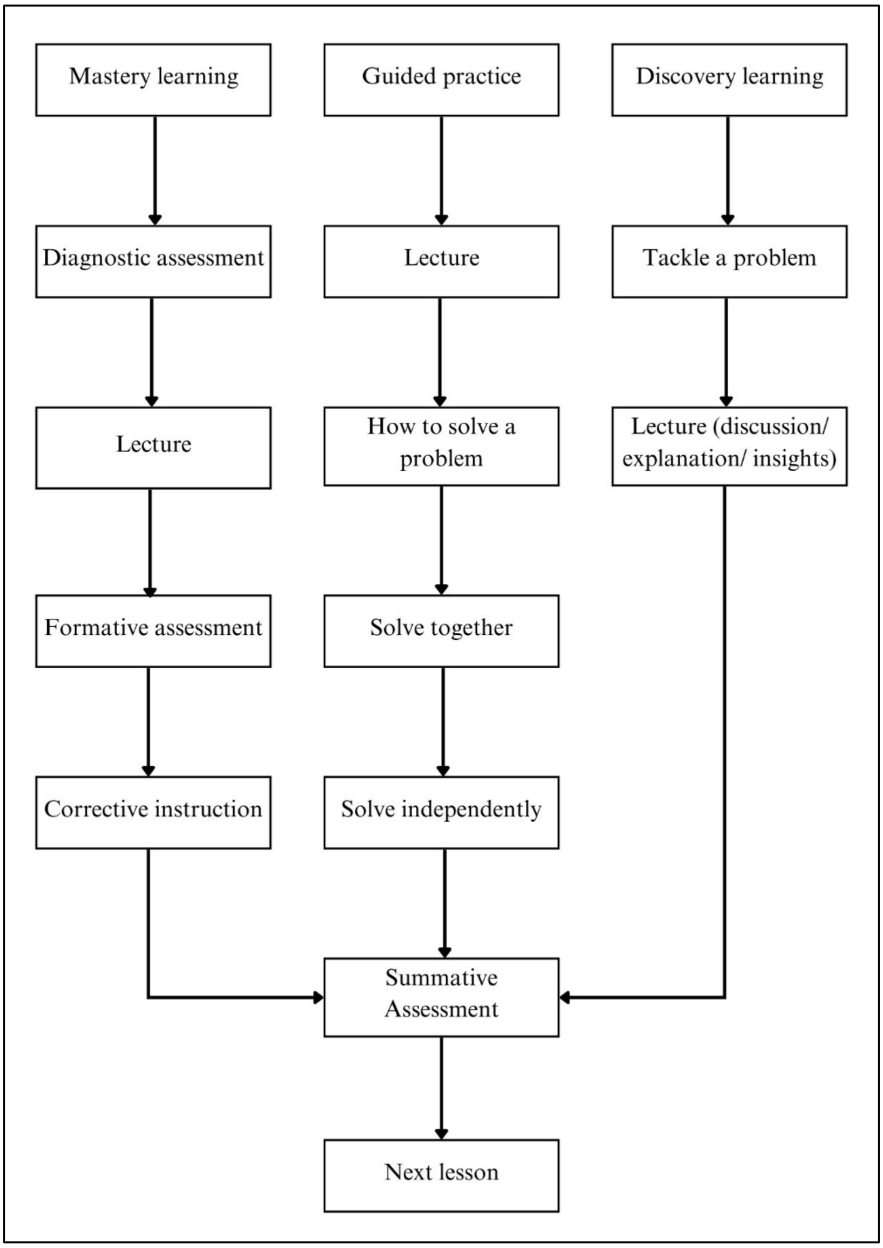

Figure 5: Teaching strategies.

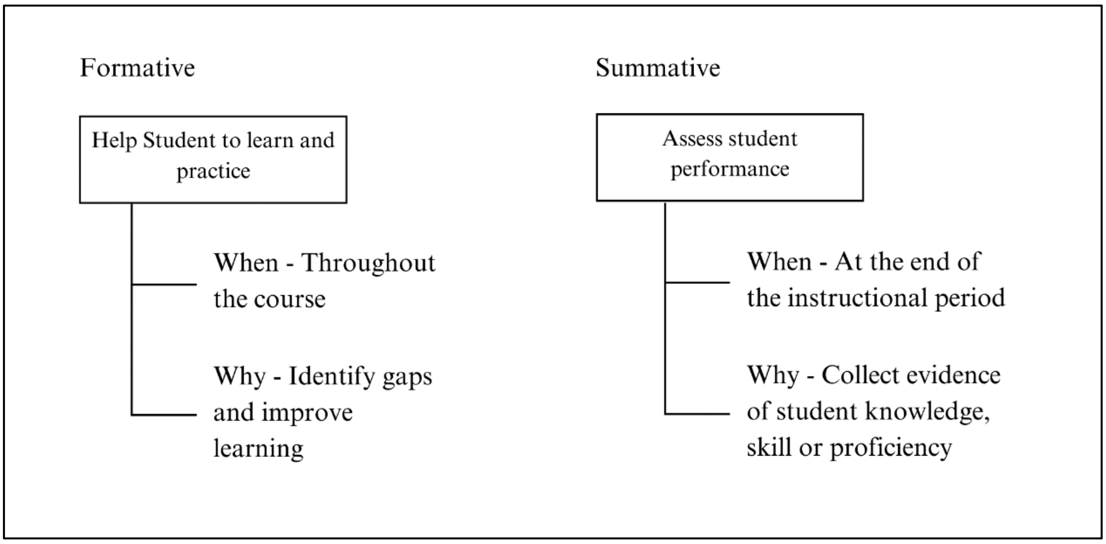

Figure 6: Assessment types.

This model also contains the connection to the LLM. Mainly pass data to this model and here it gets improved with the help of LLMs NLP capability. LLMs are statistical systems. When probability distribution $P(w_1...w_L)$ of words $w_1...w_{k-1}$, the approximated probability of the word w_k, i.e. $P(w_k |w_1w_2...w_{k-1})$. Using entropy can check the probability approximates [27].

$$L = -\frac{1}{N}\sum_{i=1}^{N-n} \log P\left(w_{i+n} \mid w_i w_{i+1} \ldots w_{i+n-1}\right)$$

e^{-L} is referred to as perplexity. When model size (N), computational resources (C), and dataset size (D) power law relationship between these and the model perplexity [27].

$$L(N) = \left(\frac{N_c}{N}\right)^{\alpha_N}, \quad \alpha_N \sim 0.076, N_c \sim 8.8 \times 10^{13}$$
$$L(D) = \left(\frac{D_c}{D}\right)^{\alpha_D}, \quad \alpha_D \sim 0.095, D_c \sim 5.4 \times 10^{13}$$
$$L(C) = \left(\frac{C_c}{C}\right)^{\alpha_C}, \quad \alpha_C \sim 0.050, C_c \sim 3.1 \times 10^{8}$$

The common workflow is shown in Figure 8. This same workflow is used to generate resources, assessments, and feedback with relevant pieces of information. Within OLMxITS, this model shoulders the responsibility of crafting comprehensive teaching plans tailored to students' needs. It generates a wealth of resources for classes, ranging from instructional materials to supplementary resources, ensuring a rich and varied learning experience. Additionally, the model generates quizzes and assessments to gauge students' comprehension and progress accurately.

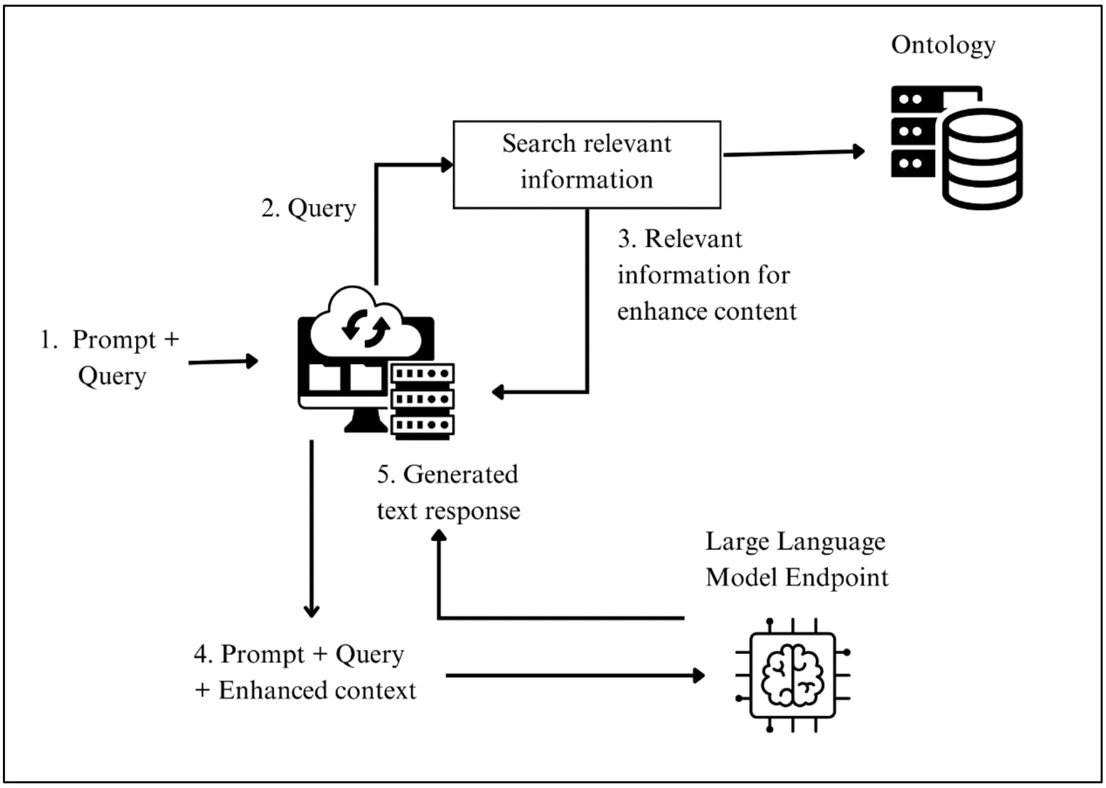

Figure 7: Common workflow for combination of LLM and Ontology

Moreover, the model fosters a responsive and personalized learning environment, where teaching strategies are tailored to meet the diverse needs of students. Whether through differentiated instruction, adaptive learning pathways, or targeted interventions, the model ensures that each student receives the support and guidance necessary to achieve their learning objectives. Following is the pseudocode for the process.

```
// Class Creation Process
CreateClass(selectedStudent, curriculum, subject, grade, component,
totalHours):
    tutorInputTotalHours = InputTotalHours()
    teachingPlan = GenerateTeachingPlan(totalHours)

// Teaching Plan Generation
GenerateTeachingPlan(totalHours):
```

```
    // Utilize preferred total number of hours to determine time allocated
for each topic
    // Generate teaching plan based on time allocation for each topic
    return teachingPlan

// Initial Activity
InitiateActivity(teachingPlan):
    activity = teachingPlan.getFirstActivity()
    if activity == DiagnosticAssessment:
        GenerateDiagnosticAssessment()
    else if activity == FirstTopic:
        GenerateContentPDF()

// Diagnostic Assessment Generation
GenerateDiagnosticAssessment():
    // Generate questions spanning all areas within the subject domain
    // Submit assessment to system for analysis

// Content Generation
GenerateContentPDF():
    // Generate content for first topic in PDF format

// Implementation of Teaching Plan
ImplementTeachingPlan(teachingPlan):
    // Based on teaching plan, proceed with instructional process
    // Evaluate student's proficiency level
    // Generate tailored content for first topic
    // Propose suitable teaching strategy to tutor

// Formative Assessment (Quiz)
FormativeAssessment():
    // Provide immediate feedback upon submission of answers
    // Tailor questions based on student performance
    // Generate personalized feedback for student

// Summative Assessment
```

```
SummativeAssessment():
    // Evaluate overall progress and readiness for examinations
    // Generate personalized feedback
    // Communicate adjustments in teaching strategy if necessary

// Update Class Grade and Feedback
UpdateGradeAndFeedback():
    // Update overall grade and feedback according to student progress

// Monitor Student Progress
MonitorProgress():
    // Present progress to tutor and student in graph format

// Conclusion of Class
ConcludeClass():
    // Assess overall grade to indicate knowledge level and readiness for
exam

// Main Process
Main():
    selectedStudent = SelectStudent()
    CreateClass(selectedStudent, curriculum, subject, grade, component,
totalHours)
    InitiateActivity(teachingPlan)
    ImplementTeachingPlan(teachingPlan)
    FormativeAssessment()
    SummativeAssessment()
    UpdateGradeAndFeedback()
    MonitorProgress()
    ConcludeClass()
```

Ultimately, this model represents a dynamic interplay between students' experiences and expert knowledge, guiding the instructional process and promoting meaningful learning outcomes. Through its adaptive nature and focus on individualized support, the model empowers tutors to facilitate engaging and effective learning experiences for all students.

5.5 Student model

This model can be also referred to as the Learner Model and the Student Diagnosis Model. Its main duty is evaluating student performance compared to other students and identifying strong and weak areas based on students' current knowledge. To do performance evaluation OLMxITS uses an algorithm (Figure 9). First, it collects all students' progress data from the ontology. Then it normalizes the scores in each assessment to ensure that they are on the same scale. After applying a clustering algorithm such as K-Means to group students based on their normalized scores across assessments use the Euclidean distance equation to calculate the distance between each student's normalized scores and the centroids of the clusters. Finally, assign each student to the cluster with the nearest centroid based on the calculated distances.

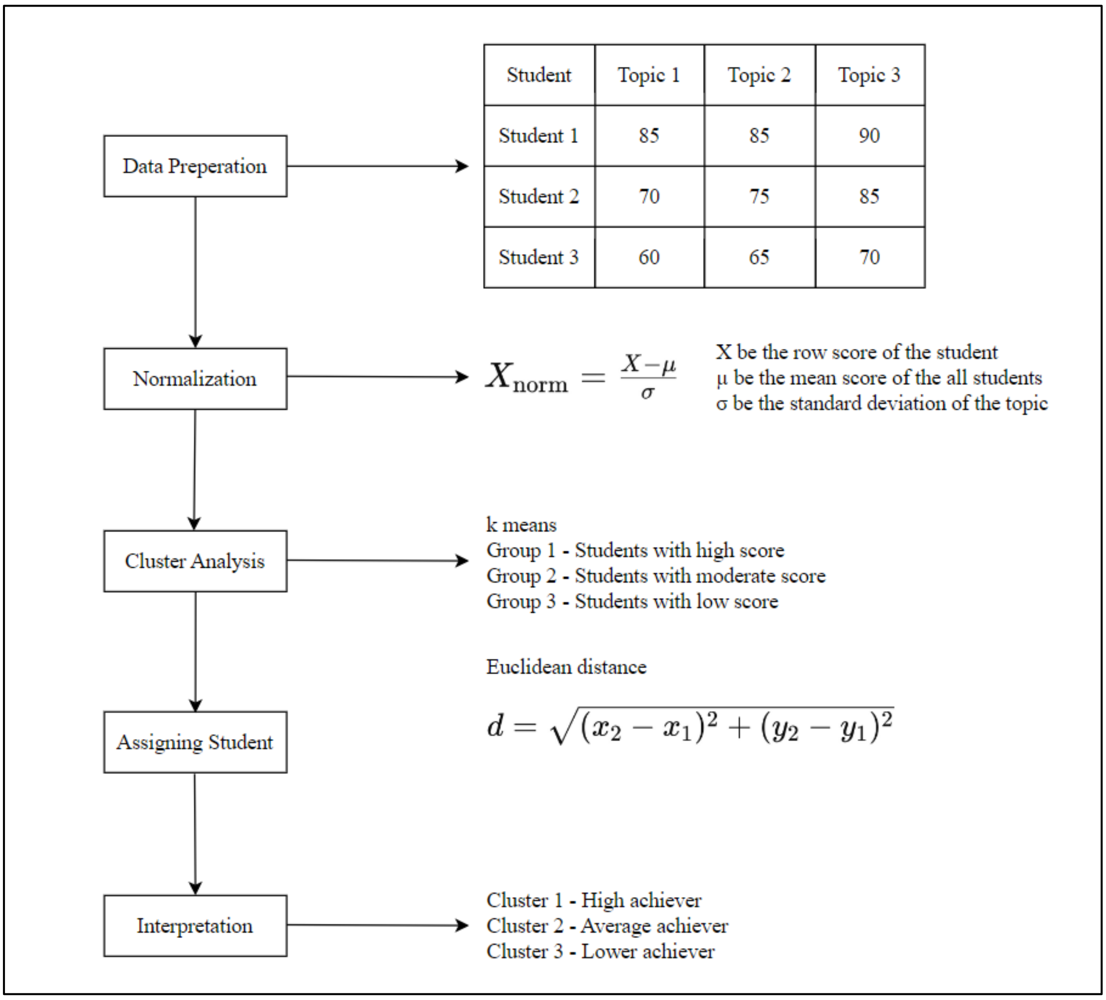

Figure 8: Performance evaluation algorithm.

Moreover, the Student Model plays a crucial role in identifying areas where students may require additional assistance or enrichment. By diagnosing learning gaps and misconceptions, it guides tutors in implementing effective strategies to address these challenges and foster academic growth. Ultimately, this model serves as an invaluable tool for promoting personalized learning experiences and maximizing student success within educational contexts.

5.6 User Interface

The Model User Interface serves as the crucial link between users and the system, facilitating interaction with the learning content and providing feedback. It encompasses both functional and aesthetic elements, aiming to create an intuitive and engaging user experience.

Functionally, the interface enables users to navigate through the system, access learning materials, and engage with interactive components such as quizzes and assessments. It provides mechanisms for users to answer questions, save their progress, and receive guidance, warnings, or corrections as needed. This ensures that users can effectively interact with the learning content and receive support in real time.

Aesthetically, the interface is designed with the user in mind, prioritizing simplicity, clarity, and ease of use. It employs intuitive navigation patterns, clear visual cues, and responsive design elements to enhance usability and accessibility. The interface should be designed to be user-friendly, allowing students to quickly and effortlessly access the information they need without encountering unnecessary barriers or confusion.

Moreover, the interface should be adaptable to different devices and screen sizes, ensuring a seamless experience across desktops, laptops, tablets, and smartphones. This flexibility enables users to engage with the learning content anytime, anywhere, enhancing the accessibility and convenience of the learning experience.

Ultimately, the Model User Interface plays a crucial role in shaping the overall user experience within the system, empowering students to effectively interact with the learning content and achieve their educational goals. By prioritizing usability, accessibility, and user-centric design principles, the interface enhances engagement, fosters learning, and maximizes the impact of the educational experience.

5.7 Summary

This chapter presented the design of OLMxITS. The top-level architecture was defined, and four main models, namely, the domain knowledge model, tutoring model, student model, and user interface, were identified within our OLMxITS architecture. The implementation of OLMxITS with reference to the design identified here will be described in the next chapter.

CHAPTER 6

IMPLEMENTATION

6.1 Introduction

Chapter 5 explained architectural design in OLMxITS. By covering top-level architecture and four main components of it namely domain knowledge model, tutoring model, student model, and user interface. This chapter presents the implementation of OLMxITS covering overall implementation, domain knowledge model, tutoring model implementation, student model implementation, and user interface implementation.

6.2 Overall implementation

For initial development, only one subject and curriculum, Cambridge IGCSE Mathematics – 0580, were selected. The ontology was modeled as the domain knowledge model, and knowledge about Cambridge IGCSE Mathematics was embedded as the first step of implementation. Subsequently, the tutoring model and student model were implemented as two separate services within one code base. The tutoring model was connected to LLM. Finally, a separate react.js project was developed for the user interface, which was then connected to the other models.

6.3 Domain knowledge model implementation

The main component of the domain knowledge model is the ontology. Initially, all the classes and subclasses outlined in the design chapter were created using the Protégé editor. Following this, properties not available in OWL and RDF were established as object properties and datatype properties. Subsequently, axioms, subclass definitions, and equivalent class rules were formulated for the previously established classes and subclasses utilizing the defined properties.

Subsequently, individuals corresponding to classes were generated based on the Cambridge IGCSE Mathematics syllabus. Individuals for topics, subtopics, and minor topics were initially created, along with their associated property values (Figure 9).

Figure 9: Classes, data properties, object properties in ontology.

Upon completion of the preceding tasks, the ontology underwent reasoning using the Pallet reasoner. During this phase, Pallet was employed to perform reasoning operations on OWL and RDFs. Additionally, for more sophisticated reasoning capabilities, the Semantic Web Rule Language (SWRL) was utilized. Bellow SWRL rules are for calculating the total duration for a component using individual subtopic durations.

```
// Add Chain Duration For First SubTopic

MainTopic(?t) ^ hasFirstSubTopicForMainTopic(?t, ?st) ^
durationForTopic(?st, ?d) -> durationForSubTopicFromMainTopicStart(?st, ?d)

// Add Chain Duration For SubTopic

durationForSubTopicFromMainTopicStart(?t, ?d) ^ hasNextTopic(?t, ?nt) ^
durationForTopic(?nt, ?ntd) ^ swrlb:add(?ndnt, ?d, ?ntd) ->
durationForSubTopicFromMainTopicStart(?nt, ?ndnt)

// Add Duration For Topic

durationForSubTopicFromMainTopicStart(?st, ?d) ^ MainTopic(?t) ^
hasLastSubTopicForMainTopic(?t, ?st) -> durationForTopic(?t, ?d)

// Add Chain Duration For FirstTopic

Component(?c) ^ hasFirstMainTopicForComponent(?c, ?t) ^ durationForTopic(?t,
?d) -> durationForMainTopicFromComponentStart(?t, ?d)

// Add Chain Duration For Topic

swrlb:add(?ndnt, ?d, ?ntd) ^ durationForMainTopicFromComponentStart(?t, ?d)
^ hasNextTopic(?t, ?nt) ^ durationForTopic(?nt, ?ntd) ->
durationForMainTopicFromComponentStart(?nt, ?ndnt)

// Add Duration For Component

durationForMainTopicFromComponentStart(?t, ?d) ^
hasLastMainTopicForComponent(?c, ?t) ^ Component(?c) ->
durationForComponent(?c, ?d)
```

Subsequently, the ontology was saved as a turtle file and imported into an Apache Jena Fuseki server. Furthermore, a SPARQL endpoint was established to facilitate connections with other models as required. Figure 10 shows the graph of the implemented ontology.

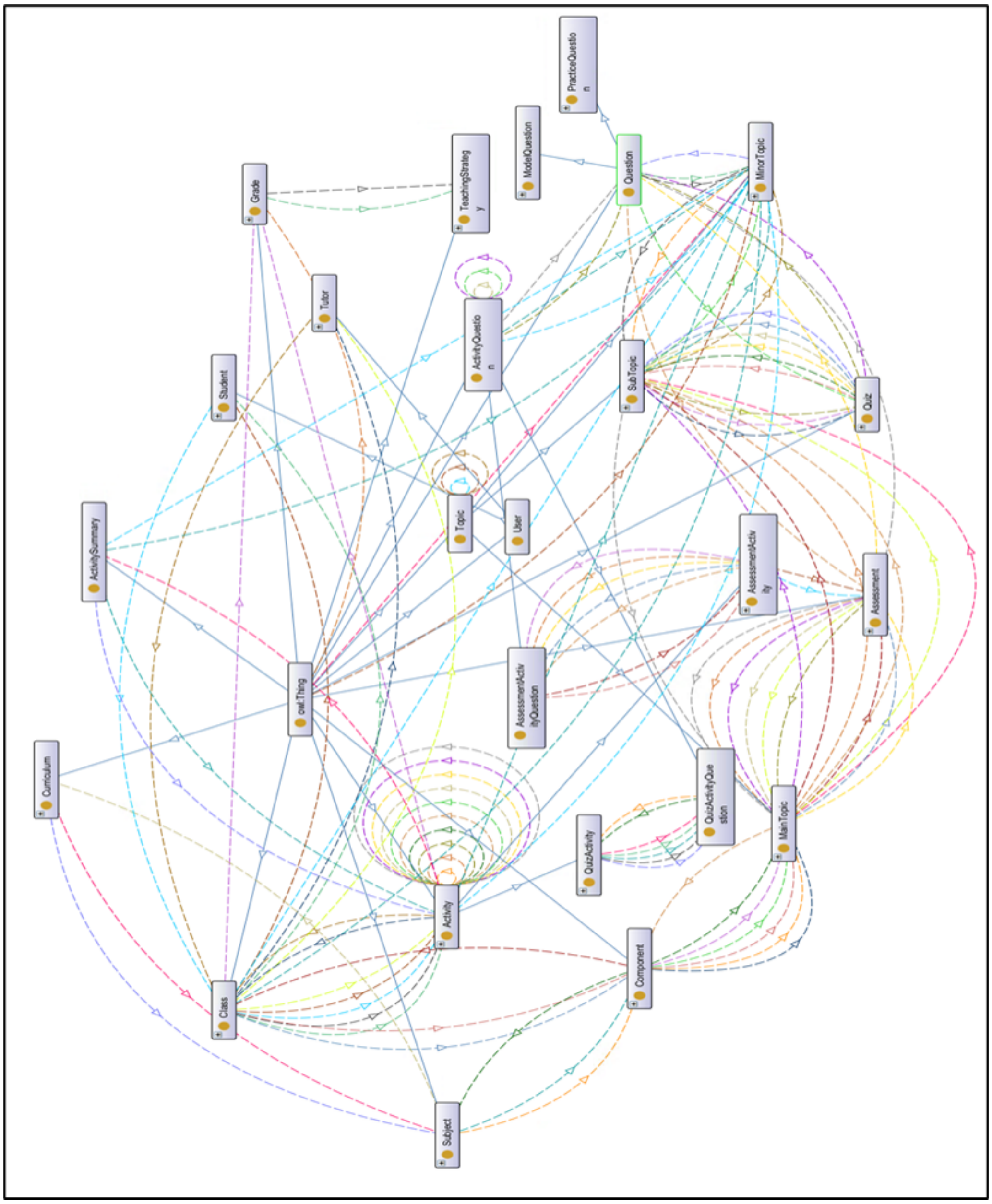

Figure 10: Ontology graph.

6.4 Tutoring model implementation

Initially, teaching strategies were implemented using Node.js in accordance with the design chapter. Subsequently, the OpenAI GPT-03 model was integrated as LLM and a SPARQL endpoint was established for querying the ontology. During the implementation of the teaching plan, the tutoring model first retrieved syllabus details from the ontology by querying the SPARQL endpoint. Following this, a plan was devised based on the selected teaching strategy (Figure 10)

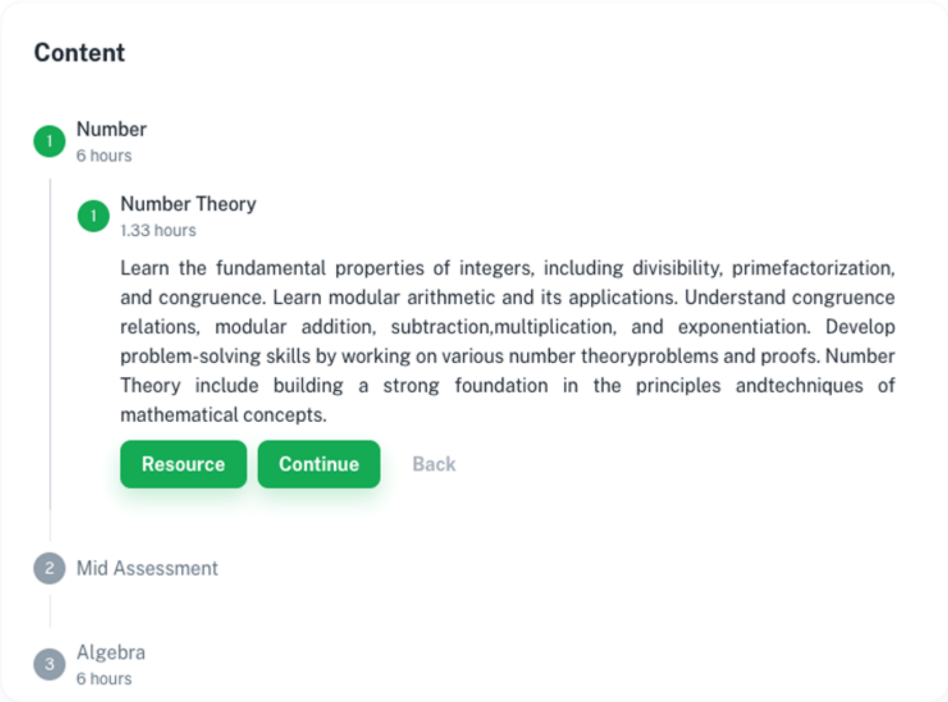

Figure 11: User Interface - Teaching plan.

Following that, for resource generation, querying the ontology for essential details was initially implemented. Subsequently, that information was passed to the GPT-03, and the generated content was obtained. Finally, it was stored as PDFs for convenient utilization.

For assessment generation, the initial step involved retrieving template questions from the ontology. Subsequently, they were passed to the GPT-03, and similar questions were obtained. Following the submission of the student's answer, the generation of explanations was implemented similarly by passing the student's answer to the GPT-03 along with ontology data and obtaining the explanation. The process of generating feedback for students was also executed in the same manner. The figure 11 depicts the feedback generated for a class.

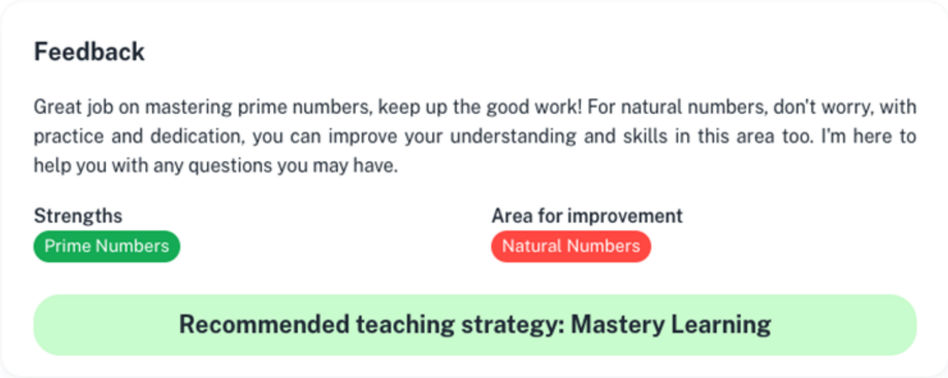

Figure 12: User Interface - Personalized feedback.

6.5 Student model implementation

The performance evaluation algorithm was implemented using Node.js in accordance with the design chapter. In the initial stages of implementation, student performance details were obtained from the ontology and passed to the implemented algorithm for evaluation. Once the evaluation process concluded, the student's grade and their strong and weak areas were identified. This analysis is illustrated in the accompanying Figure 12.

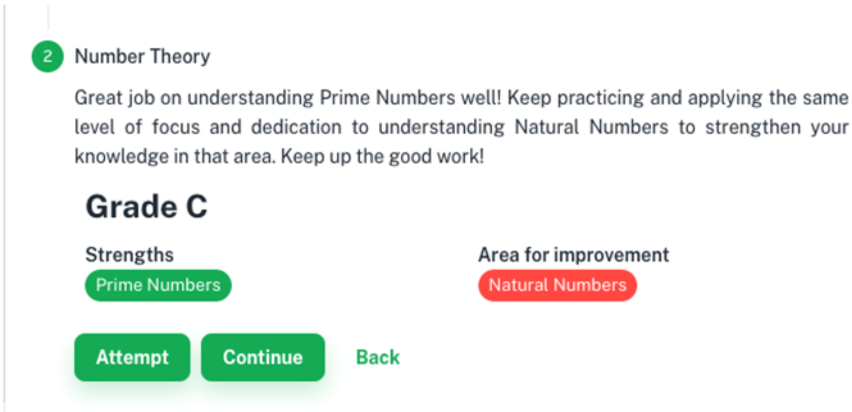

Figure 13: User Interface - Performance evaluation.

Progress tracking was implemented in such a manner that performance details were sent back to the ontology for recording and subsequent querying to display to the student. The process of displaying progress to the student was depicted in the accompanying figure 13.

Figure 14: User Interface - Progress tracker.

6.6 User interface implementation

The user interface was implemented using React.js. React.js components were created to facilitate navigation, access to learning materials, and interaction with quizzes and assessments. React.js state and props were utilized to enable users to answer questions, save progress, and receive real-time feedback. Event handling in React.js was employed to trigger guidance, warnings, or corrections based on user actions, ensuring effective interaction with the learning content (Figure 14).

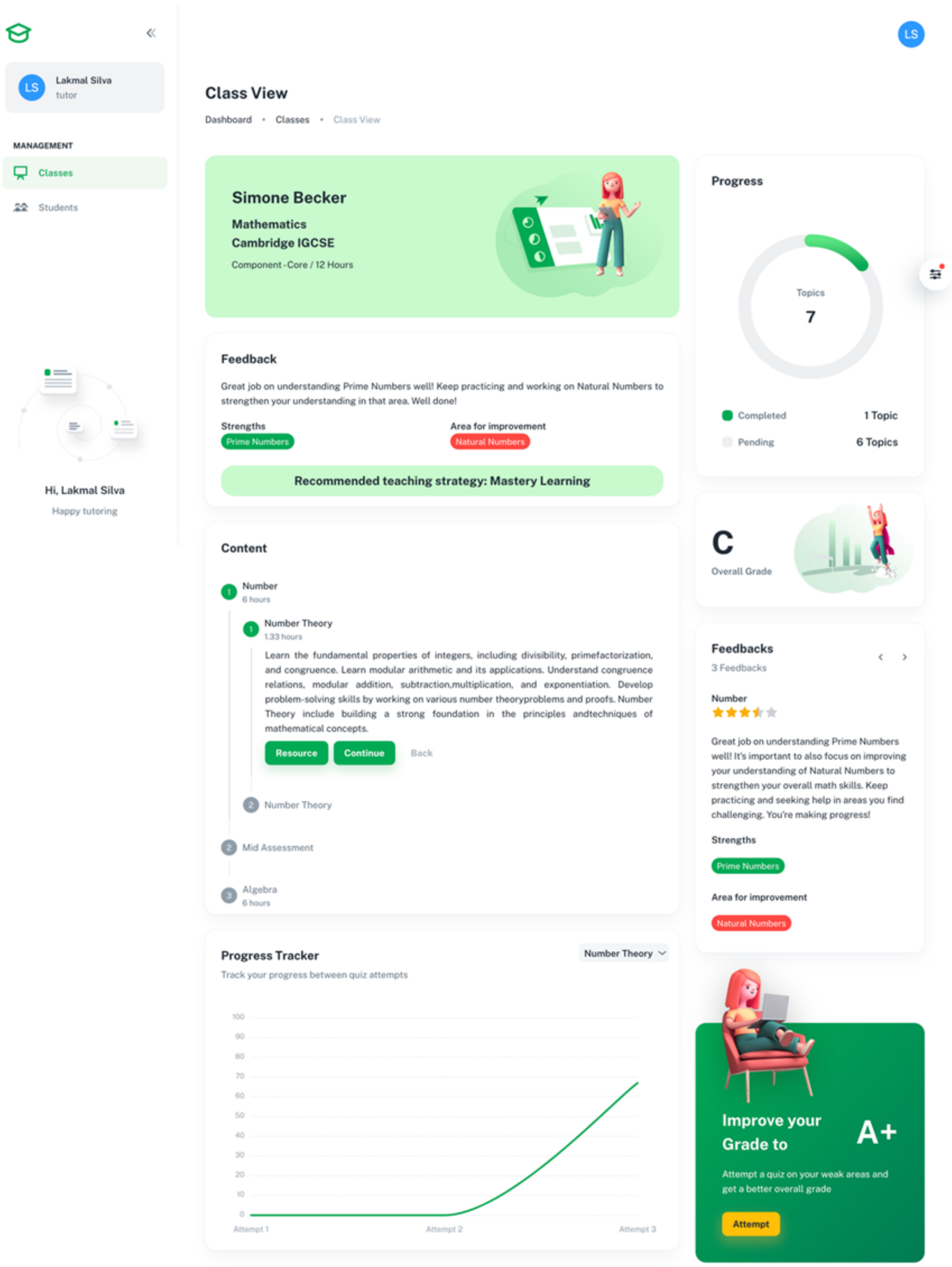

Figure 15: User Interface - Desktop view.

React.js's component-based architecture was leveraged to design UI components with simplicity, clarity, and ease of use in mind. React.js's conditional rendering capabilities were utilized to dynamically adjust the UI based on user interactions or device characteristics, prioritizing user-friendliness. React.js's responsive design features, such as media queries, were used to ensure that the interface adapts seamlessly to various devices and screen sizes (Figure 15).

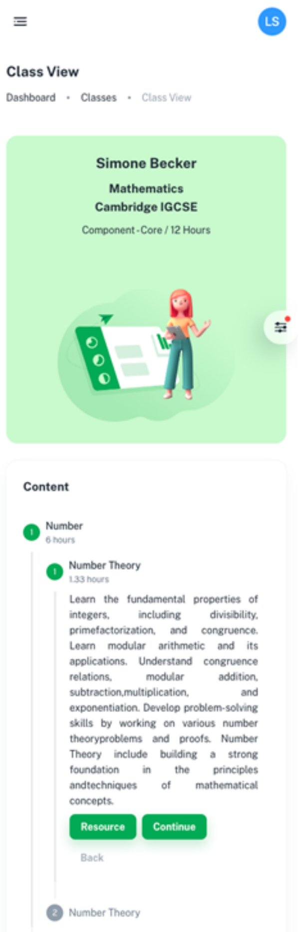

Figure 16: User Interface - Mobile view.

Figure 17 and Figure 18 show the UI of the class creation form and how explanations of answers for questions are shown in the UI.

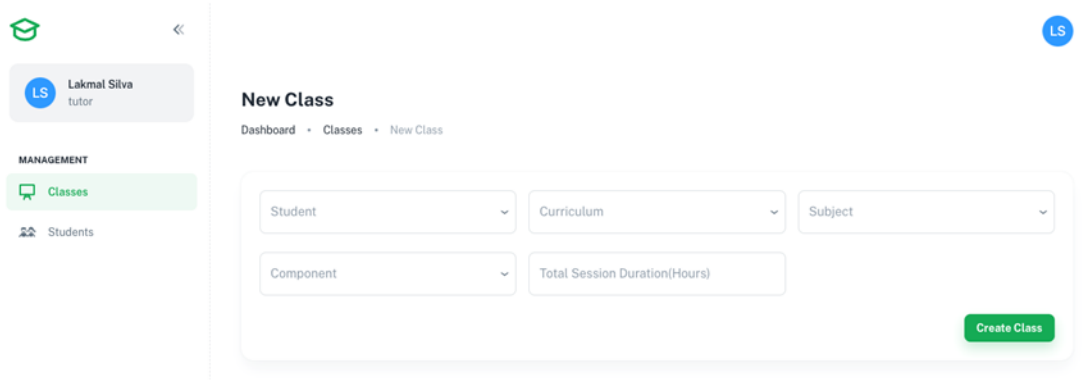

Figure 17: Class create form.

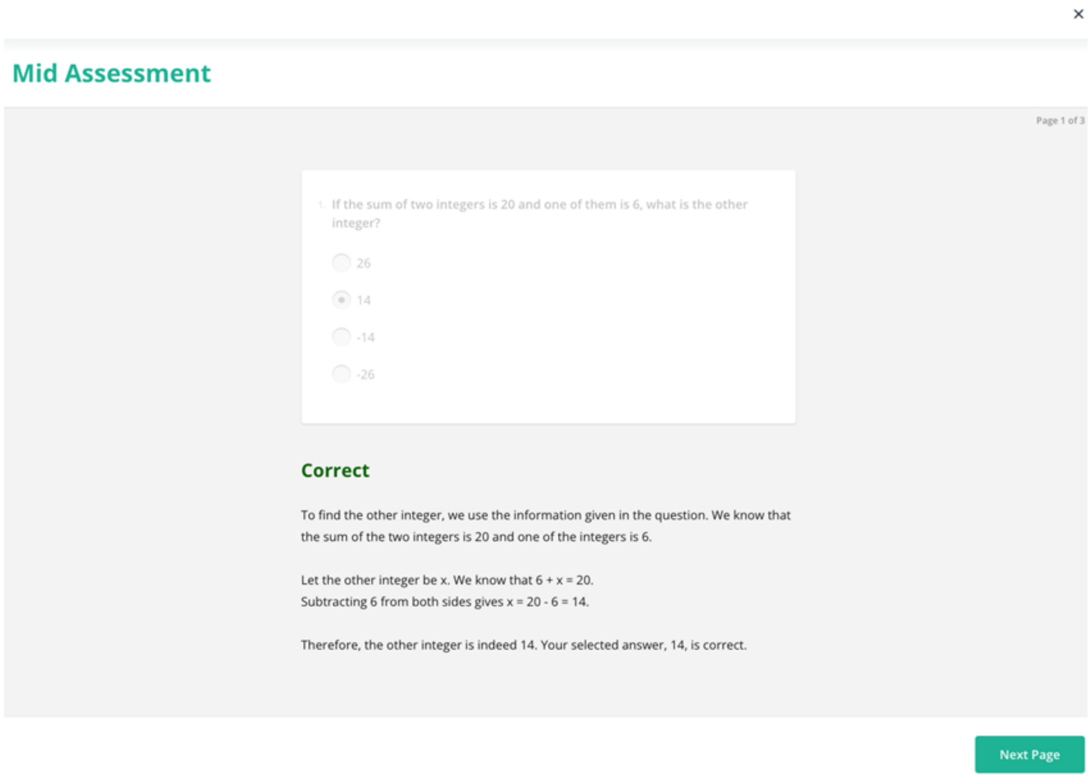

Figure 18: Explaining answers for questions.

6.7 Summary

This chapter presented the implementation of OLMxITS. The overall implementation, and four main models, namely, the domain knowledge model implementation, tutoring model implementation, student model implementation, and user interface implementation, were discussed. The evaluation of OLMxITS will be described in the next chapter.

CHAPTER 7

EVALUATION

7.1 Introduction

Chapter 6 explained the implementation of OLMxITS. By covering overall implementation, domain knowledge model implementation, tutoring model implementation, student model implementation, and user interface implementation. This chapter presents the evaluation of OLMxITS covering experimental design, benchmark, and results and analysis.

7.2 Experimental design

Experiments are meticulously designed to comprehensively assess the effectiveness and efficiency of OLMxITS features. In the initial phase of experimentation, a focused approach is taken, targeting three core features: resource generation, assessment generation, and feedback generation.

In order to provide a thorough evaluation, teaching strategies, and student grades are deliberately assigned manually to all student accounts involved in these experiments. This meticulous approach ensures that each experiment accounts for the diverse educational contexts and learner profiles.

To capture the full spectrum of possible interactions, all conceivable combinations between teaching strategies and students' grades are systematically tested. This exhaustive methodology aims to explore how varying instructional approaches and student performance levels influence the efficacy of the OLMxITS features.

Moreover, to ascertain the robustness and reliability of the findings, each test is iterated multiple times, exceeding a threshold of 10 repetitions. This rigorous repetition helps to mitigate the impact of random variability and ensures that the observed results are consistent and dependable across multiple iterations. By emphasizing consistency and reliability, the validity of the experimental outcomes is upheld, affirming the credibility and trustworthiness of the study's findings.

For resource generation testing, a meticulous approach is adopted whereby resources are generated repeatedly for the same topic and subtopics, each time under varied combinations of teaching strategies and student grades. This method ensures a comprehensive evaluation of the system's ability to adapt and tailor resources to suit diverse instructional contexts and learner profiles. By systematically varying the teaching strategies and student grades, the experiment aims to elucidate how different instructional approaches and student performance levels influence the quality and relevance of the generated resources.

Similarly, the assessment generation process undergoes rigorous testing, wherein assessments are repeatedly generated following the generation of feedback for each assessment. This sequential approach ensures that the assessments accurately reflect the feedback provided, maintaining coherence and alignment between assessment and instructional feedback. By subjecting the assessment generation process to iterative testing, the experiment aims to assess the system's proficiency in generating assessments that effectively evaluate learner understanding and progress. This comprehensive evaluation of both resource and assessment generation processes is essential for gauging the system's overall effectiveness in supporting teaching and learning activities.

In further sets of experiments, the OLMxITS undergoes comparative analysis with similar systems to assess its unique contributions and advantages. A total of six systems are carefully chosen for comparison, comprising a diverse array of online learning platforms and ITSs.

Among the online learning platforms selected for comparison are Coursera, Udemy, and Skillshare, renowned for their extensive course offerings and user-friendly interfaces. These platforms serve as benchmarks for assessing the OLMxITS's capabilities in delivering educational content and facilitating self-paced learning experiences on a large scale.

In addition, three ITSs are included in the comparison: Cognitive Tutor, Duolingo, and ALEKS. Each of these systems specializes in providing personalized and adaptive learning experiences tailored to individual learner needs. By juxtaposing the OLMxITS against these established ITSs, the experiment aims to highlight its innovative features and effectiveness in leveraging ontology and LLMs for educational purposes.

Through comprehensive comparative analysis with a diverse range of educational platforms and tutoring systems, the experiment seeks to elucidate the distinctive advantages and potential applications of the OLMxITS in enhancing teaching and learning outcomes across various domains and disciplines. This comparative approach provides valuable insights into the relative strengths and weaknesses of each system, guiding future developments and advancements in intelligent tutoring technologies.

7.3 Benchmark

In the initial series of experiments, three distinct benchmarks are meticulously chosen to gauge the performance and functionality of the OLMxITS. Each benchmark represents a specific configuration of the system, enabling a focused evaluation of its capabilities.

The first benchmark involves testing with only the ontology enabled, aiming to assess the system's performance solely based on ontological knowledge representation. This benchmark serves as a baseline for evaluating the effectiveness of ontology-driven content generation and organization within the OLMxITS.

The second benchmark focuses on testing with only the LLM enabled, isolating the impact of NLP capabilities on the system's performance. By examining the system's ability to generate content and responses solely based on language model processing, this benchmark elucidates the contribution of LLM to the overall functionality of the OLMxITS.

The third benchmark encompasses testing with both the ontology and LLM enabled, representing the full integration of both knowledge representation and NLP within the system. This benchmark allows for an assessment of the synergistic effects resulting from the combined utilization of ontology and LLM, highlighting the enhanced capabilities and adaptive features of the OLMxITS.

Against each benchmark, a thorough evaluation is conducted across four key subareas. These are the coverage of necessary educational areas in generated content, the dynamic generation of content according to varying teaching strategies and student grades, the quality of NLP, and the absence of unnecessary or irrelevant content in the generated content.

In the second set of experiments, a comparative analysis is conducted to evaluate the OLMxITS against selected similar systems. The comparison is centered around key features essential for effective teaching and learning experiences, including course planning, content generation, providing assessment, personalized feedback, progress tracking, and teaching strategy adjustment.

Through this comprehensive comparative analysis, the second set of experiments aims to elucidate the unique strengths and advantages of OLMxITS in supporting effective teaching and learning practices, providing valuable insights for educators, researchers, and developers in the field of ITSs.

7.4 Results and analysis

For the initial set of experiments (focused evaluation)

Resource Generation:

- **Covered Necessary Areas**: This criterion assesses whether the generated resources address all essential educational topics and subtopics. Results indicate that the combination of ontology and LLM (Ontology + LLM) performs well in covering necessary areas, while ontology alone (Ontology) also achieves this but to a lesser extent.
- **Generated Dynamically**: Refers to the ability of the system to dynamically generate resources based on teaching strategy and student grade. Both configurations with LLM enabled (LLM and Ontology + LLM) excel in dynamic generation.
- **Delivered Quality NLP**: This criterion evaluates the quality of Natural Language Processing in the generated resources. Both configurations with LLM enabled (LLM and Ontology + LLM) deliver high-quality NLP outputs.
- **Absence of Unnecessary or Irrelevant Content**: Determines whether the generated resources contain unnecessary or irrelevant content. The combination of ontology and LLM (Ontology + LLM) performs well in eliminating unnecessary content.

Table 1: Focused evaluation - Resource generation

	Ontology	LLM	Ontology + LLM
Covered necessary areas	Yes	No	Yes
Generated dynamically	No	Yes	Yes
Delivered quality NLP	No	Yes	Yes
Absence of unnecessary or irrelevant content	Yes	No	Yes

Assessment Generation:

- **Covered Necessary Areas**: Evaluate whether the generated assessments adequately cover essential educational topics. Similar to resource generation, the combination of ontology and LLM (Ontology + LLM) excels in covering necessary areas.
- **Generated Dynamically**: Both configurations with LLM enabled (LLM and Ontology + LLM) demonstrate dynamic generation capabilities.
- **Delivered Quality NLP**: LLM-enabled configurations (LLM and Ontology + LLM) deliver assessments with high-quality NLP.
- **Absence of Unnecessary or Irrelevant Content**: The combination of ontology and LLM (Ontology + LLM) effectively eliminates unnecessary or irrelevant content from generated assessments.

Table 2: Focused evaluation - Assessment generation

	Ontology	LLM	Ontology + LLM
Covered necessary areas	Yes	No	Yes
Generated dynamically	No	Yes	Yes

Delivered quality NLP	No	Yes	Yes
Absence of unnecessary or irrelevant content	Yes	No	Yes

Feedback Generation:

- **Covered Necessary Areas**: Determines whether the generated feedback addresses essential educational topics. Once again, the combination of ontology and LLM (Ontology + LLM) excels in covering necessary areas.
- **Generated Dynamically**: Configurations with LLM enabled (LLM and Ontology + LLM) demonstrate dynamic feedback generation.
- **Delivered Quality NLP**: Both configurations with LLM enabled (LLM and Ontology + LLM) deliver feedback with high-quality NLP.
- **Absence of Unnecessary or Irrelevant Content**: The combination of ontology and LLM (Ontology + LLM) effectively eliminates unnecessary or irrelevant content from generated feedback.

Table 3: Focused evaluation- Feedback generation

	Ontology	LLM	Ontology + LLM
Covered necessary areas	Yes	No	Yes
Generated dynamically	No	Yes	Yes
Delivered quality NLP	No	Yes	Yes
Absence of unnecessary or irrelevant content	Yes	No	Yes

Overall, the evaluation indicates that the combination of ontology and LLM yields the most favorable results across all aspects of resource, assessment, and feedback generation, followed by ontology alone. LLM alone exhibits limitations in coverage, dynamic generation, NLP quality, and the elimination of unnecessary content, highlighting the importance of integrating ontology-based knowledge representation with NLP capabilities for effective ITS functionality.

For the second set of experiments (comparative analysis)

Course Planning:

- OLMxITS offers dynamic course planning, allowing for adaptable and personalized learning pathways based on learner needs and progress.
- Coursera, Udemy, Skillshare, and ALEKS provide static course planning, offering predefined courses without dynamic adjustments.
- Cognitive Tutor does not offer course planning functionality.

Content Generation:

- OLMxITS provides dynamic content generation, tailoring instructional materials based on learner characteristics and progress.
- Coursera, Udemy, Skillshare, and ALEKS offer static content generation, providing predefined course content without adaptability.
- Duolingo offers dynamic content generation for language learning.

Providing Assessment:

- OLMxITS offers dynamic assessment provision, adjusting assessments based on learner progress and performance.
- Coursera, Udemy, Skillshare, and ALEKS provide static assessment provision, offering predefined assessments without adaptability.
- Duolingo offers dynamic assessment provision for language learning.

Personalized Feedback:

- OLMxITS provides dynamic personalized feedback tailored to individual learner needs and performance.
- Coursera, Udemy, Skillshare, and Cognitive Tutor do not offer personalized feedback.
- Duolingo and ALEKS offer dynamic personalized feedback for language learning and math respectively.

Progress Tracking:

- All platforms, including OLMxITS, Coursera, Udemy, Skillshare, Cognitive Tutor, Duolingo, and ALEKS, offer progress tracking functionality to monitor learner advancement and performance.

Teaching Strategy Adjustment:

- OLMxITS allows for dynamic adjustment of teaching strategies based on learner progress and feedback.
- Coursera, Udemy, Skillshare, and Duolingo do not offer teaching strategy adjustments.
- Cognitive Tutor and ALEKS provide teaching strategy adjustment capabilities.

Table 4: Comparative analysis with similar systems

	OLMxITS	Coursera	Udemy	Skillshare	Cognitive Tutor	Duolingo	ALEKS
Course planning	Available – dynamic	Available - static	Available - static	Available - static	Not available	Available - static	Available - static
Content generation	Available - dynamic	Available - static	Available - static	Available - static	Not available	Available - dynamic	Available - static
Providing assessment	Available - dynamic	Available - static	Available - static	Available - static	Not available	Available - dynamic	Available - dynamic
Personalized feedback	Available - dynamic	Not available	Not available	Not available	Available	Available - dynamic	Available - dynamic
Progress tracking	Available	Available	Available	Available	Available	Available	Available
Teaching strategy adjustment	Available	Not available	Not available	Not available	Available	Not available	Available

Overall, OLMxITS stands out for its dynamic course planning, content generation, assessment provision, personalized feedback, and teaching strategy adjustment features, offering a comprehensive and adaptive learning experience. While other platforms excel in certain areas, OLMxITS combines these features to provide a versatile and effective ITS.

7.5 Summary

This chapter presented the evaluation of OLMxITS. Experimental design, benchmark, and results and analysis were discussed by doing two sets of experiments with a focused approach and comparative analysis and find out combination of ontology and LLM is solving the issues that can't be solved ontology and LLM working individually in the system. Further, find out OLMxITS is better than the selected similar systems for the experiment. The conclusion and further work will be described in the next chapter.

CHAPTER 8

CONCLUSION AND FURTHER WORK

8.1 Introduction

Chapter 7 explained the evaluation of OLMxITS. Where discussed focused evaluation on resource generation, assessment generation, feedback generation, and comparative analysis of six similar systems. This chapter presents the conclusion and further work of OLMxITS covering the overall conclusion, objective-wise achievements, limitations, and further work.

8.2 Overall conclusion

In conclusion, this study has contributed to the advancement of ITSs by integrating ontology and LLMs to enhance educational practices. Through a critical review of education system evolution and an in-depth study of modern education technologies, the groundwork was laid for designing and developing ontology and connecting ontology to LLM. Subsequently, ITSs were implemented based on these technologies, demonstrating promising results in resource generation, assessment provision, and personalized feedback delivery. The evaluation of these systems underscored their effectiveness in improving tutoring experiences and supporting personalized learning.

8.3 Objective-wise achievements

A Critical Review of the Evolution of Education Systems

This objective was fulfilled through a thorough examination of the historical development and transformation of education systems. The review provided insights into the evolution of pedagogical theories, instructional methods, and technological advancements shaping modern educational practices. By analyzing the historical context, trends, and challenges, this objective laid the groundwork for understanding the need for innovative approaches to education.

An In-depth Study of Technologies Used in Modern Education

The study delved into contemporary technologies employed in educational settings, ranging from LMS to AI and NLP. By exploring the potential applications, benefits, and limitations of these technologies, this objective facilitated a comprehensive understanding of the technological landscape in education. This knowledge served as a basis for identifying opportunities to leverage technology to enhance teaching and learning experiences.

Design and Develop Ontology and Connecting Ontology and LLM

This objective encompassed the design and development of ontology while also establishing the connection between ontology and LLMs. The process began with conceptualizing and formalizing domain knowledge relevant to educational content and pedagogical practices, resulting in the creation of a comprehensive ontology. Additionally, efforts were directed toward integrating ontology with LLM to facilitate seamless knowledge transfer and utilization within ITS. This integration enabled LLM to leverage ontological structures for generating contextually relevant and coherent educational content, enhancing the adaptability and effectiveness of ITS features. Through meticulous design and development, the synergy between ontology and LLM was harnessed to enrich the capabilities of ITSs and support personalized learning experiences.

Implement Intelligent Tutoring Systems Based on Ontology and Large Language Model

The implementation objective entailed integrating ontology and LLM into the architecture of ITS. This involved the incorporation of ontology-based knowledge representation for organizing educational content and LLM-based NLP for generating instructional materials and providing personalized feedback. By implementing ITS based on ontology and LLM, this objective demonstrated the feasibility and effectiveness of leveraging these technologies to enhance tutoring experiences.

Evaluation of the Intelligent Tutoring Systems

Finally, the objective of evaluating the ITS was realized through systematic experimentation and analysis. The evaluation process assessed the performance, and effectiveness with ITS features, including resource generation, assessment provision, and feedback delivery. By conducting rigorous evaluations, this objective provided empirical evidence of the capabilities of ontology and LLM-based ITSs.

8.4 Limitation

Despite the notable achievements attained throughout the research process, various limitations have been encountered, presenting opportunities for further refinement and exploration. One significant challenge arose from constraints in data availability for ontology development, hindering the depth and breadth of knowledge representation. This limitation underscores the importance of expanding data sources and employing diverse data collection methods to enrich ontology content. Furthermore, while the ontology design has laid a strong foundation, further refinement is necessary to enhance knowledge representation accuracy, ensuring alignment with educational objectives and domain-specific nuances. Another crucial aspect involves the integration of ontology and LLMs with existing educational infrastructures, a complex endeavor that demands seamless interoperability and compatibility with diverse systems and platforms. Additionally, the intricacies of user interface design pose challenges in optimizing usability and accessibility, necessitating iterative refinement and user-centered approaches. Moreover, the generalizability of findings may be constrained by the study's scope and context, emphasizing the need for cautious interpretation and consideration of contextual factors when extrapolating results to diverse educational settings. Addressing these limitations through ongoing research and collaborative efforts will be essential for realizing the full potential of ontology and LLM-based ITSs and fostering innovation in education.

8.5 Further work

In light of the identified limitations, several avenues for further work present themselves, offering opportunities to refine and expand upon the research findings to advance the field of ontology and LLM-based ITSs. Firstly, efforts to address data availability constraints for ontology development should be prioritized, encompassing initiatives to curate and augment datasets relevant to diverse educational domains. This involves leveraging data mining techniques, collaborative data-sharing platforms, and domain-specific partnerships to enrich the ontology with comprehensive and high-quality educational content. Moreover, further refinement of ontology design is essential to enhance knowledge representation accuracy and coherence, necessitating ongoing validation and validation processes informed by domain experts and educational practitioners.

Additionally, seamless integration of ontology and LLM-based ITSs with existing educational infrastructures remains a crucial area for further exploration. This requires interdisciplinary collaboration between researchers, educators, and technologists to develop standardized protocols, interoperability frameworks, and integration tools facilitating the seamless assimilation of ontology and LLM-based ITSs into diverse educational ecosystems. Moreover, the refinement of user interface design presents an opportunity to optimize user experiences and engagement with ITSs. This involves conducting user-centered design studies, usability testing, and iterative interface refinement to ensure intuitive navigation, accessibility, and user satisfaction across diverse user demographics and contexts.

Furthermore, longitudinal studies and real-world deployments are warranted to evaluate the long-term impact and effectiveness of ontology and LLM-based ITSs in diverse educational settings. Longitudinal studies can provide insights into the sustainability, scalability, and efficacy of ITSs over extended periods, shedding light on their transformative potential in fostering personalized and adaptive learning experiences. Additionally, real-world deployments enable researchers to gather authentic user feedback, iterate on system functionalities, and address emerging challenges in situ, facilitating continuous improvement and innovation in ontology and LLM-based ITSs.

Lastly, exploring novel applications of ontology and LLM-based ITSs beyond traditional educational contexts offers avenues for interdisciplinary collaboration and innovation. This includes leveraging ITSs for workforce training, professional development, lifelong learning, and skill acquisition in diverse domains such as healthcare, business, and industry. By extending the scope of ontology and LLM-based ITSs to new domains and applications, researchers can unlock new opportunities for enhancing human learning and performance across diverse contexts and populations.

8.6 Summary

In summary, this study represents a significant step towards leveraging ontology and LLMs to advance ITS and transform education. The achievements made in critical review, technological study, design, implementation, and evaluation underscore the potential of ontology and LLM-based ITSs in fostering personalized and adaptive learning experiences. While limitations exist, they serve as opportunities for further refinement and innovation. Moving forward, continued research and development efforts hold the promise of realizing the full potential of ontology and LLM in shaping the future of education.

REFERENCES

[1] L. Stošić, “THE IMPORTANCE OF EDUCATIONAL TECHNOLOGY IN TEACHING,” *IJCRSEE*, vol. 3, no. 1, pp. 111–114, Jun. 2015, doi: 10.23947/2334-8496-2015-3-1-111-114.

[2] U. Sharma, P. Tomar, H. Bhardwaj, and A. Sakalle, “Artificial Intelligence and Its Implications in Education,” 2021, pp. 222–235. doi: 10.4018/978-1-7998-4763-2.ch014.

[3] T. Oktavia, H. Prabowo, Meyliana, and S. H. Supangkat, “The Comparison of MOOC (Massive Open Online Course) Platforms of edX and Coursera (Study Case: Student of Programming Courses),” in *2018 International Conference on Information Management and Technology (ICIMTech)*, Jakarta: IEEE, Sep. 2018, pp. 339–344. doi: 10.1109/ICIMTech.2018.8528178.

[4] L. Li, Z. Swiecki, D. Gašević, and G. Chen, “Popularity Prediction in MOOCs: A Case Study on Udemy,” in *Artificial Intelligence in Education*, M. M. Rodrigo, N. Matsuda, A. I. Cristea, and V. Dimitrova, Eds., Cham: Springer International Publishing, 2022, pp. 607–613. doi: 10.1007/978-3-031-11644-5_56.

[5] S. D. Craig *et al.*, “The impact of a technology-based mathematics after-school program using ALEKS on student’s knowledge and behaviors,” *Computers & Education*, vol. 68, pp. 495–504, Oct. 2013, doi: 10.1016/j.compedu.2013.06.010.

[6] M. Nushi, M. H. Eqbali, D. Boulevard, and E. S. Tehran, “DUOLINGO: A MOBILE APPLICATION TO ASSIST SECOND LANGUAGE LEARNING,” *Teaching English with Technology*, vol. 17, no. 1, pp. 89–98.

[7] A. Briggs, “The Study of the History of Education,” *History of Education*, vol. 1, no. 1, pp. 5–22, Jan. 1972, doi: 10.1080/0046760720010102.

[8] A. Göçen and F. Aydemir, “Artificial Intelligence in Education and Schools,” *Research on Education and Media*, vol. 12, pp. 13–21, Dec. 2020, doi: 10.2478/rem-2020-0003.

[9] S. B. Dias, S. J. Hadjileontiadou, J. Diniz, and L. J. Hadjileontiadis, “DeepLMS: a deep learning predictive model for supporting online learning in the Covid-19 era,” *Sci Rep*, vol. 10, no. 1, Art. no. 1, Nov. 2020, doi: 10.1038/s41598-020-76740-9.

[10] S. A. Aljawarneh, “Reviewing and exploring innovative ubiquitous learning tools in higher education,” *J Comput High Educ*, vol. 32, no. 1, pp. 57–73, Apr. 2020, doi: 10.1007/s12528-019-09207-0.

[11] B. I. Edwards and A. D. Cheok, “Why not Robot Teachers: Artificial Intelligence for Addressing Teacher Shortage.” Preprints, Dec. 04, 2017. doi: 10.20944/preprints201712.0022.v1.

[12] H. Haseski, “What Do Turkish Pre-Service Teachers Think About Artificial Intelligence?,” *International Journal of Computer Science Education in Schools*, vol. 3, Sep. 2019, doi: 10.21585/ijcses.v3i2.55.

[13] N. Humble and P. Mozelius, "Artificial Intelligence in Education - a Promise, a Threat or a Hype?," presented at the European Conference on the Impact of Artificial Intelligence and Robotics 2019 (ECIAIR 2019), Oxford, UK, Academic Conferences and Publishing International Limited, 2019, pp. 149–156. Accessed: Jun. 03, 2023. [Online]. Available: https://urn.kb.se/resolve?urn=urn:nbn:se:miun:diva-37620

[14] L. Harasim, "Shift happens: online education as a new paradigm in learning," *The Internet and Higher Education*, vol. 3, no. 1, pp. 41–61, Jan. 2000, doi: 10.1016/S1096-7516(00)00032-4.

[15] L. Harasim, "Shift happens: online education as a new paradigm in learning," *The Internet and Higher Education*, vol. 3, no. 1–2, pp. 41–61, Jan. 2000, doi: 10.1016/S1096-7516(00)00032-4.

[16] D. M. Early *et al.*, "Teachers' Education, Classroom Quality, and Young Children's Academic Skills: Results From Seven Studies of Preschool Programs," *Child Development*, vol. 78, no. 2, pp. 558–580, 2007, doi: 10.1111/j.1467-8624.2007.01014.x.

[17] C. M. Stracke, S. Downes, G. Conole, D. Burgos, and F. Nascimbeni, "Are MOOCs Open Educational Resources? A Literature Review on History, Definitions and Typologies of OER and MOOCs," *Open Praxis*, vol. 11, no. 4, pp. 331–341, 2019.

[18] R. Allen, "The Web: interactive and multimedia education," *Computer Networks and ISDN Systems*, vol. 30, no. 16, pp. 1717–1727, Sep. 1998, doi: 10.1016/S0169-7552(98)00200-1.

[19] M. Goldwasser, C. Mankoff, K. Manturuk, L. Schmid, and K. Whitfield, "Who is a Student: Completion in Coursera Courses at Duke University," *Current Issues in Emerging eLearning*, vol. 3, no. 1, Apr. 2016, [Online]. Available: https://scholarworks.umb.edu/ciee/vol3/iss1/8

[20] B. Panth and R. Maclean, Eds., *Anticipating and Preparing for Emerging Skills and Jobs: Key Issues, Concerns, and Prospects*, vol. 55. in Education in the Asia-Pacific Region: Issues, Concerns and Prospects, vol. 55. Singapore: Springer Singapore, 2020. doi: 10.1007/978-981-15-7018-6.

[21] M. Bankovska, "Analysis of E-learning Platforms: Comparison Between Udemy and Skillshare," *PECR*, vol. 80, Dec. 2023, doi: 10.7546/PECR.80.23.05.

[22] E. Kasneci *et al.*, "ChatGPT for good? On opportunities and challenges of large language models for education," *Learning and Individual Differences*, vol. 103, p. 102274, Apr. 2023, doi: 10.1016/j.lindif.2023.102274.

[23] H. S. Nwana, "Intelligent tutoring systems: an overview," *Artif Intell Rev*, vol. 4, no. 4, pp. 251–277, Dec. 1990, doi: 10.1007/BF00168958.

[24] S. Ritter, J. R. Anderson, K. R. Koedinger, and A. Corbett, "Cognitive Tutor: Applied research in mathematics education," *Psychonomic Bulletin & Review*, vol. 14, no. 2, pp. 249–255, Apr. 2007, doi: 10.3758/BF03194060.

[25] U. M. Abdelsalam, "A Proposal Model of developing Intelligent Tutoring Systems based on Mastery Learning," pp. 106–118, Jan. 2014.

[26] A. Fahmy and J. Lagowski, “The use of systemic approach in teaching and learning chemistry for the 21st century,” *Pure and Applied Chemistry - PURE APPL CHEM*, vol. 71, pp. 859–863, Jan. 1999, doi: 10.1351/pac199971050859.

[27] S. Das, “Evaluating the Capabilities of Large Language Models for Spatial and Situational Understanding,” Department of Engineering University of Cambridge, 2023.

APPENDIX A

```
4 usages   Lakmal Silva *
private async getActivity(activity: string, getAttempt = true) {
  const query = `
    SELECT *
    WHERE {
      :${activity} a :Activity.
      :${activity} :activityStatus ?status.
      OPTIONAL { :${activity} :hasFirstSubActivity ?firstSubActivity. }
      OPTIONAL { :${activity} :hasFirstQuizActivityQuestionForQuizActivity ?firstQuizActivityQuestion. }
      OPTIONAL { :${activity} :hasFirstAssessmentActivityQuestionForAssessmentActivity ?firstAssessmentActivityQuestion. }
      OPTIONAL { :${activity} :hasParentActivity ?parentActivity. }
      OPTIONAL { :${activity} :hasNextActivity ?nextActivity. }
      OPTIONAL { :${activity} :hasPreviousActivity ?previousActivity. }
      OPTIONAL { :${activity} :hasSubTopicForActivity ?subTopic. }
      OPTIONAL { :${activity} :hasQuizForActivity ?quiz. }
      OPTIONAL { :${activity} :hasMainTopicForActivity ?mainTopic. }
      OPTIONAL { :${activity} :hasAssessmentForActivity ?assessment. }
      OPTIONAL { :${activity} :durationForActivity ?duration. }
      OPTIONAL { :${activity} :content ?content. }
      OPTIONAL { :${activity} :feedback ?feedback. }
      OPTIONAL { :${activity} :hasActivitySummaryForActivity ?activitySummary. }
      OPTIONAL {
        :${activity} :hasStrongMinorTopic ?strongMinorTopic.
        ?strongMinorTopic rdfs:label ?strongMinorTopicLabel.
      }
      OPTIONAL {
        :${activity} :hasWeakMinorTopic ?weakMinorTopic.
        ?weakMinorTopic rdfs:label ?weakMinorTopicLabel.
      }
      OPTIONAL { :${activity} :feedback ?feedback. }
      OPTIONAL { :${activity} :marksPercentage ?marksPercentage. }
    }
  `;
  const result = await this.sparql.select(this.sparql.queryBuilder(query));
```

Figure 19: Get user activity function.

```
1 usage   Lakmal Silva
private async getAssessmentAvg(quiz: string) {
  const query = `
    SELECT (AVG(?marks) AS ?avgMarks)
    WHERE {
      :${quiz} :hasAssessmentActivityQuestionForAssessmentActivity ?quizActivity.
      ?quizActivity :marks ?marks.
    }
  `;
  const result = await this.sparql.select(this.sparql.queryBuilder(query));

  if (result && result.length > 0) {
    return {
      instant: quiz,
      avgMarks: result[0].avgMarks ? parseFloat(result[0].avgMarks.value) : undefined,
    };
  } else {
    return undefined;
  }
}
```

Figure 20: Get assessment average function.

```
2 usages  Lakmal Silva
private async getClassAvg(instant: string) {
  const query = `
    SELECT (AVG(?marks) AS ?avgMarks)
    WHERE {
    {
        ?quiz a :MainTopicActivity.
        ?quiz :hasClassForActivity :${instant}.
        ?quiz :marksPercentage ?marks.
    }
    UNION
    {
        ?quiz a :AssessmentActivity.
        ?quiz :hasClassForActivity :${instant}.
        ?quiz :marksPercentage ?marks.
    }
    }
  `;
  const result = await this.sparql.select(this.sparql.queryBuilder(query));

  if (result && result.length > 0) {
    return {
      instant: instant,
      avgMarks: result[0].avgMarks ? parseFloat(result[0].avgMarks.value) : undefined,
    };
  } else {
    return undefined;
  }
}
```

Figure 21: Get average marks for the class.

```
5+ usages  Lakmal Silva
private async generateFeedback(data) {
  let content = `generate short feedback for a student `;
  if (data.strongMinorTopic && data.strongMinorTopic.length > 0) {
    content += `his strong areas are ${data.strongMinorTopic} `;
  }
  if (data.weakMinorTopic && data.weakMinorTopic.length > 0) {
    content += `his weak areas are ${data.weakMinorTopic}`;
  }
  const gptRes = await this.gpt.ask( messages: [
    { role: 'system', content: 'You are a maths tutor, you give short feedbacks to the student.' },
    {
      role: 'user',
      content: `${JSON.stringify(content)}`,
    },
  ]);

  console.log(gptRes.message.content);

  return `${gptRes.message.content
    .replace( searchValue: /\r?\n|\r/g,  replaceValue: '')
    .replace( searchValue: "''''",  replaceValue: "'")
    .replace( searchValue: '"',  replaceValue: "'")
    .replace( searchValue: /\"/g,  replaceValue: "'")}`;
}
```

Figure 22: Generate feedback.

```
while (minorTopic) {
  const minorTopicObj = await this.getMinorTopicFromSubTopic(minorTopic);
  content += `<h2>${minorTopicObj.label}</h2>`;

  const paragraph = await this.gpt.ask( messages: [
    { role: 'system', content: 'You are a maths tutor, you generate content for students' },
    {
      role: 'user',
      content: `expand ${minorTopicObj.content} to a student.${
        processedTopics.length > 0 ? ` student already know ${processedTopics}.` : ''
      }. give only the expanded content.`,
    },
  ]);

  processedTopics.push(minorTopicObj.label);

  console.log(paragraph.message.content);

  content += `<p>${paragraph.message.content}</p>`;

  minorTopic = minorTopicObj.nextTopic;
}
```

Figure 23: Generate resources for a class.

```
for await (const question of questions) {
  const gptRes = await this.gpt.askJson( messages: [
    {
      role: 'system',
      content: 'You are a maths tutor, you generate similar question to given question in JSON without justifying the content.',
    },
    {
      role: 'user',
      content: `${question.content}`,
    },
  ]);

  console.log(gptRes.message.content);

  question.content = gptRes.message.content.replace( searchValue: /\r?\n|\r/g,  replaceValue: '');

  activityQuestion = await this.createAssessmentActivityQuestion(instant, question, activityQuestion);
  if (isFirstActivityQuestion) {
    isFirstActivityQuestion = false;
    await this.updateAssessmentActivityWithActivityQuestions(instant, activityQuestion);
  }
}
```

Figure 24: Generate questions.