Recommendation System for Improve Primary Students Educational Performance

Yushan Viduranga Jayasekara

IT20645748

B.Sc. (Hons) Degree in Information Technology Specialized in Data Science

Department of Computer Science and Software Engineering

Sri Lanka Institute of Information Technology Sri Lanka

2024 - June

Recommendation System for Improve Primary Students Educational Performance

Yushan Viduranga Jayasekara

IT20645748

Dissertation submitted in partial fulfilment of the requirements for the Special Honours Degree of Bachelor of Science in Information Technology Specializing in Data Science.

Department of Information Technology

Sri Lanka Institute of Information Technology Sri Lanka

2024 - June

# **ABSTRACT**

The focus of e-learning systems is to deliver knowledge effectively to students, particularly in a user-friendly and flexible manner. This challenge becomes more intricate when dealing with primary students, necessitating a highly intuitive interface. This study aims to address these complexities by creating an adaptive learning system tailored to enhance the educational performance of primary students. This Component involves developing a recommendation system to improve students' educational outcomes. This innovative system administers subject-specific and general knowledge quizzes, meticulously tracking the resultant data. Leveraging this data and employing machine-learning models to gauge educational performance levels, the system channels this information through a recommendation system. This, in turn, furnishes recommendations in alignment with the student's educational proficiency. Recommendations encompass various mediums such as questionnaires, books, video lessons, and other relevant materials, each selected to elevate the student's educational aptitude. Notably, the system aims to enhance user engagement by elucidating the rationale behind each recommendation. Providing transparency in the recommendation process enables students to comprehend the basis for the suggested learning materials. The interplay of data-driven insights, machine-learning techniques, and comprehensive recommendations marks a significant advancement in fostering primary students' learning journeys. In conclusion, this research contributes to a comprehensive adaptive learning system tailored for primary students. The integrated recommendation system, guided by student performance data and machine-learning algorithms, offers tailored suggestions to bolster educational growth. By elucidating the reasoning behind each recommendation, the system promotes a deeper understanding of the learning process. This research paves the way for a more personalised and effective approach to enhancing primary students' educational performance in e-learning.

Keywords:**-** knowledge graphs, recommendation systems, explainable AI, Explainable AI, Spacy, Questionnaire, NER

# **DECLARATION**

I declare that this is my own work, and this dissertation does not incorporate without acknowledgement 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 acknowledgement is made in the text. Also, I hereby grant to the Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute my dissertation in whole or part in print, electronic or other mediums. I retain the right to use this content in whole or part in future works (such as articles or books).

|  |  |  |
| --- | --- | --- |
| Name | Student ID | Signature |
| J.A.Y.V. Jayasekara | IT20645748 |  |

Signature of the Supervisor Date : 05.04.2024

(Mr Samadhi Rathnayake)



# **ACKNOWLEDGEMENT**

I express my heartfelt thanks to Mr. Samadhi Ratnayake for his exceptional guidance and mentorship throughout this research endeavour. His deep expertise and insights have been instrumental in shaping the trajectory and outcomes of this study.

I greatly appreciate Dr. Darshan as my co-supervisor for his unstinting support and invaluable contribution. His insightful guidance and encouragement were essential in navigating the complexities inherent in this research project.

I would like to express my sincere thanks to the Honorable Research Panel Dr. Junius and Dr. Lakmini for their valuable insights and recommendations. Their constructive feedback and suggestions for improving the model along with useful updates have contributed significantly to its refinement and progress. Also, I would like to express my gratitude to our Lecturer-in-Charge Dr. Jayantha Amaraarachchi and the CDAP team for their support in ensuring the success of the project.

I am extremely grateful for the support, guidance and expertise provided by these individuals throughout this research journey. I am also thankful to all my friends who have helped to make this project a success. Their contributions have enriched the depth and quality of this work, and I am privileged to have the opportunity to collaborate with such esteemed professionals.

# TABLE OF CONTENT

[ABSTRACT 2](#_Toc1465561832)

[DECLARATION 3](#_Toc521070283)

[ACKNOWLEDGEMENT 4](#_Toc101021087)

[TABLE OF CONTENT 5](#_Toc1832989693)

[LIST OF FIGURES 6](#_Toc53980225)

[LIST OF TABLES 7](#_Toc789572815)

[LIST OF ABBREVIATIONS 8](#_Toc1972688165)

[INTRODUCTION 9](#_Toc669345390)

[1.1 General Introduction 9](#_Toc1487014976)

[1.2 Background Literature 10](#_Toc2034006360)

[1.3 Research Gap 12](#_Toc863495549)

[1.4 Research Problem 15](#_Toc2068112524)

[1.5 Research Objectives 17](#_Toc989672675)

[1.5.1 Main Objective 17](#_Toc720008994)

[1.5.2 Sub Objectives 17](#_Toc1936312514)

[2. METHODOLOGY 19](#_Toc1209526880)

[2.1 Methodology 19](#_Toc940874577)

[2.1.1 Requirement Gathering 19](#_Toc1356525062)

[2.1.2 Data Collection and Description 19](#_Toc492152644)

[2.1.3 System Designs 19](#_Toc577307180)

[2.1.3.1 Overall system diagram 20](#_Toc729071148)

[2.1.3.2 Component Architecture 20](#_Toc1088666836)

[2.1.4 Tools and Libraries 21](#_Toc818793312)

[2.1.5 Knowledge Graph Recommendation System Architecture 22](#_Toc1572835691)

[2.1.5.1 Data Collection and Augmentation 22](#_Toc1477171621)

[2.1.5.2 Build Knowledge Graph 24](#_Toc1052408945)

[2.1.5.3 Recommendation System 31](#_Toc1154199833)

[2.2 Commercialization Aspects of the Product 32](#_Toc152857968)

[2.2.1 Target Market 33](#_Toc2064829714)

[2.2.2 Revenue Generate 33](#_Toc278588840)

[2.2.3 Marketing Approach 34](#_Toc217888223)

[2.3 Testing and Implementation 36](#_Toc831096637)

[2.3.1 Functional Requirements 36](#_Toc1563897979)

[2.3.2 Non-Functional Requirements 37](#_Toc1014879562)

[2.3.3 Backend Implementation 39](#_Toc1311311496)

[2.3.4 Backend Testing 46](#_Toc2074939095)

[3. RESULTS AND DISCUSSION 49](#_Toc1950599727)

[3.1 Results 49](#_Toc423845315)

[3.1.1 Named, Entity Recognition(NER) 49](#_Toc2049047314)

[3.1.2 Knowledge Graph 50](#_Toc893526616)

[3.1.3 Recommendation system 52](#_Toc1922578386)

[3.2 Research Findings 53](#_Toc1043888003)

[3.3 Discussion 54](#_Toc1706291446)

[3.3.1 Knowledge graph and Recommendation System performance 54](#_Toc2095771807)

[3.3.2 Practical Implementations 55](#_Toc560381956)

[3.3.3 Limitations, Challengers and Future Developments 55](#_Toc1605683276)

[4. CONCLUSION 56](#_Toc476178989)

[5. REFERENCES 58](#_Toc293792889)

# LIST OF FIGURES

Figure 1: Overall System Architecture…………..…….……………………………………24

Figure 2: Component Diagram……….…………..……………………….…………………25

Figure 3: Mount Google Drive………………………………………………………………26

Figure 4: Import Libraries………………………………………………………………..….26

Figure 5: Load data using Wikipidealoder…………………….………….…………………27

Figure 6: Remove unwanted documents…………………………………………………….27

Figure 7: Tokenize sentences………………………………….…………………………….28

Figure 8:- Import Libraries for building knowledge graph………………...………………..29

Figure 9: NER Model 1……………………………………………….....…………………..30

Figure 10: NER Model 2(Save custom model) …………………………………....………..30

Figure 11: Sentence Segmentation…………………………………………….…………….31

Figure 12: Entity Extract step 1…………………………………………….……………….32

Figure 13: Entity Extract Step 2……………………………………………………………..32

Figure 14: Entity pairing……………………………………………...………...……………..33

Figure 15: Relationships Extraction…………………………………...……….…...….………33

Figure 16: Relationships list…………………………………….………………….………….34

Figure 17: Knowledge Graph construct………………………..…………………….…………34

Figure 18: Visualize Knowledge Graph……………………………………………..………….35

Figure 19: Visualize Knowledge Graph in relationship-wise…………………………………….35

Figure 20: Save and Load Knowledge Graph……………………………………...……………35

Figure 21: Pre-train graph-embedding model………………………………….……………36

Figure 22: Recommendation System Architecture……………………..………………………..37

Figure 23: Import Flask libraries…………..........…….……………………….……………….44

Figure 24: Flask app initialization…………………………………………….………………..44

Figure 25: ‘/recommendations' Route………………….……………………………….………45

Figure 26: App execute…………………………………………………....…………………..45

Figure 27: generate recommendations function……………………………...………………….46

Figure 28: add\_entity\_relationships function……………………………..…………………….46

Figure 29: Import packages for frontend development………...………....………….…………..47

Figure 30: useState & useNavigate hooks…………………………..……………………….….47

Figure 31: handleSubmit function…………………………….……….………….……………48

Figure 32: Evaluate answers……………………………………..…....……………………….48

Figure 33: Define educational levels……………………………..……………..........…………49 Figure 35: Get data to the frontend………………………………………………………….49 Figure 36: Display data………………………………….…………………….............……………50 Figure 37: Test backend using Postman……………………….………………………………..50 Figure 38: Home page…………………………………………………………………51 Figure 39: Quiz Dashboard………………………………………………………………….51 Figure 40: Quiz interface…………………………………………...……………………….52 Figure 41: Report page…………………………………………….………………………..52 Figure 42: Classification report for custom NER model…………………………...………..54 Figure 43: Knowledge graph…………………….........……………………………………..55 Figure 44: Entity availability………………………………………………………………..55 Figure 45: Generated Recommendations…………………………...……………....………….56

# LIST OF TABLES

Table 1: Research Gap....................................................................................................……16

# LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
| Abbreviation | Long Form |
| KG | Knowledge graph |
| RS | Recommendation System |
| NER | Named Entity Recogniser |
| NLP | Natural Language processing |
| KGRS | Knowledge graph recommendation system |

# INTRODUCTION

## 1.1 General Introduction

E-learning systems’ main target is to deliver user-friendly and flexible knowledge to students. When we deal with primary students, the process is more complicated and the system must be very user-friendly. Our system tries to minimise those challenges and make a valuable adaptive learning system to improve primary students' educational performance.

Students are a critical asset in providing high-quality education for any educational institution. To achieve this, it is essential for every college, school or any other institution to check the performance of students. Due to the incredible development of recent technology like social media, students are addicted to it and waste their time in vain. This is often one of the explanations for students' poor performance in academic achievement, which even results in school truancy.[5] Predicting student performance focuses learners on their performance and helps them improve their performance in the future. However, academic performance may vary from student to student as every student has a different level of performance. But we aim to bring students who have a particularly poor educational level, also improve some higher level.[1]

Some other studies have researched in the field of affective recommender systems on learning resources in virtual learning environments and e-learning to help improve content recommendations in virtual education and therefore to improve the virtual learning process to develop a new methodology to enhance the learning level of elementary students. Ultimately more personalized recommendations are needed, and that is why an effective recommender system is analysed based on the specific needs of each student, their level of expertise, their situation and how they feel during the learning process.[4]

The provided solution is a questionnaire based on the subject-wise general knowledge. Students can take quizzes very easily and give answers, Answer checking and providing educational level is fully automated so that teachers can monitor educational performance very easily. The questionnaire is a subject-wise quiz for the students and monitors the relevant results. Based on those monitored results and using a function to get educational performance levels. Then pass the educational level through a recommendation system to get the recommendation based on the relevant student’s educational level.[2] This recommendation system is developed based on knowledge graphs and knowledge graphs developed using the Spacy model, and NER to recognize entities and relationships. These recommendations explain the reason for the recommendation and perform this task using Explainable AI technology. It can analyse student performance and guide them by displaying suggestions for improvement. This recommendation may be another questionnaire, book, video lesson, or other suitable material to improve the educational level. In addition, this system tries to explain why that recommendation came and what happened inside the recommendation.

## 1.2 Background Literature

Most research work is aimed at students in universities or other higher education. Many of their research papers use students' previous performance to predict educational levels. Several research papers were conducted on how to improve the e-learning process and some others are only based only recommendation systems. Also, some of the research papers predict educational performance but do not give any recommendations on how to improve educational performance.

P Ramya, S G Balakrishnan and M Kannan propose a recommendation model for higher education students using machine learning algorithms. They use Random forests, Decision-tree and Naive Bayes Classifiers to reach maximum accuracy and their accuracy level is between 68% to 79% for the techniques they used. Throughout this paper, the prediction of scholar performance is finished utilizing the Apriori-type strategies WEKA tool[1]. A variety of methods like content-based, collaborative filter-based, and knowledge-based recommendations and observations can be used to better understand the current development and future direction of recommendation systems in e-learning[2]. The information is processed in the WEKA device itself and the algorithms needed for the model production are also available inside the tool[1].

According to the paper of Jiahong Su and Weipeng Yang artificial intelligence (AI), tools are increasingly used in the field of early childhood education (ECE) to improve learning and development among young children. Previous proof-of-concept studies have shown that AI can effectively improve teaching and learning in ECE. These publications are reviewed to evaluate, synthesize, and showcase the latest literature on AI in ECE. Many studies have shown that AI has significantly improved children's conceptions of machine learning, computer science, robotics, and other skills such as creativity, emotional control, collaborative inquiry, literacy skills, and computational thinking. This development of literacy and knowledge can be introduced as a special importance of e-learning education[4].

J. Bobadilla, F. Serradilla and A. Hernando propose Collaborative filtering adapted to recommender systems of e-learning. In the context of e-learning recommender systems, they suggest that more knowledgeable users have more weight in the recommendation calculation than less knowledgeable users. To achieve this objective, we have created several new equations in the memory-based collaborative filtering kernel, extending the existing equations to gather and process information relative to the scores obtained by each user from a variable number of level tests[2],[6].

Regardless of the method used in the CF phase, the generally followed technical objective is to minimize the prediction errors by making the accuracy of the RS as high as possible. One of the ideas emphasized in the working philosophy of RS is based on the equality between its users, not only on their ability to access the service but above all on the contribution of each of them to the recommendations that the rest can receive. A typical RS generates recommendations for each user based on the ratios provided by the users with the most similar contribution to them[6].

This questionnaire is based on the subject to check students' knowledge. Students can take quizzes very easily and give answers, Answer checking and providing educational level are fully automated so that teachers can monitor educational performance very easily. The questionnaire is a subject-wise quiz for the students and monitors the relevant results. Based on those monitored results and using a Large language model to get educational performance levels[12]. Then pass the educational level through a recommendation system to get the recommendation based on the relevant student’s educational level[2]. This recommendation system is developed based on knowledge graphs and knowledge graphs developed using the Spacy model, and NER to recognize entities and relationships.

These recommendations explain the reason for the recommendation and perform this task using Explainable AI technology. It can analyse student performance and guide them by displaying suggestions for improvement. This recommendation may be another questionnaire, book, video lesson, or other suitable material to improve the educational level. In addition, this system tries to explain why that recommendation came and what happened inside the recommendation.

This study explores the potential of Large Language Models (LLMs) in educational scenarios, specifically in concept graph recovery and question-answering. It evaluates LLMs' zero-shot performance in creating domain-specific concept graphs and introduces TutorQA, a benchmark for scientific graph reasoning and QA. The study presents CGLLM, a pipeline integrating concept graphs with LLMs for diverse questions. Results show LLMs' zero-shot concept graph recovery is competitive with supervised methods, with up to 26% F1 score enhancement in TutorQA tasks[11].

The benefits of introducing e-learning recommendation systems go beyond the achievement of learning goals. Based on the literature review, the advantages of e-learning recommendation systems can be classified in several ways, but mainly we can say that they improve the educational performance of students.[2],[8] Our proposed method is more useful in raising the level of education for primary students in the right way.

## 1.3 Research Gap

According to the information extracted from the previous research papers, none of them related to real-time educational monitoring. All of them are considered previously done test marks and only input directly into their systems. But our solution we will monitor educational levels using quizzes.

According to the available research papers[2] and resources, there are some researches developed for recommendation systems, some of them are related to e-learning systems[1],[2],[7], but none of them touches primary education enhancement using recommendation systems. Also, very few of them provide papers to enhance childhood education, but their solution is based on AI.[4] According to the extracted information we provide knowledge-based recommendations as the best method to implement solutions in our system. Some research papers talk about knowledge-based methods but they do not implement them in e-learning systems to give recommendations.[2] In addition, we will classify educational levels using ML algorithms and then provide them to the recommendation models. Some research done in the past gives recommendations for improving educational levels using machine learning algorithms only.[1] They didn't use any specific recommendation mechanism to give recommendations.

The study of Shanshan Wan and Zhendong Nilu[3] proposed a hybrid filtering (HF) recommendation approach (SI-IFL) for e-learning recommender systems. It combines the learner influence model (LIM), self-organization-based (SOB) recommendation strategy, and sequential pattern mining (SPM) to recommend learning objects to learners. LIM calculates learner influence, SOB simulates influence propagation, and SPM decides on final learning objects and navigational paths. The experimental results show that SI-IFL provides personalized and diversified recommendations, demonstrating efficiency and adaptability in e-learning scenarios.

This scoping review of Jiahong Su and Weipeng Yang[4], found that AI significantly improved children's concepts in AI, machine learning, computer science, robotics, creativity, emotion control, collaborative inquiry, literacy skills, and computational thinking. Future directions for researching AI in ECE are discussed and do not discuss real-time educational performance monitoring or recommendation systems.

All of the information extracted from the research papers and resources in recommendation systems only provides recommendations and one of them provides some justification but that paper did not use proper recommendation methods. However, the proposed solution uses knowledge-based recommendation techniques and it provides the reason behind the recommendation. In doing that process use explainable AI techniques and try to explain the reason behind the relevant recommendation and why that recommendation exactly given to improve the students' educational performance.

Archive to that process and success in this research project used explainable AI and it helps to revive as a topic of active research by the need to convey safety and trust to users in the “how” and “why” of automated decision-making in different applications.[9] Also, It helps characterize model accuracy, fairness, transparency and outcomes in AI-powered decision-making.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Research Gap | Research A  [1] | Research B  [4] | Research C  [10] | Proposed Solution |
| Provide a reason using explainable AI | No | No | No | Yes |
| Improve primary students’ educational performance | Yes | Yes | No | Yes |
| Use knowledge graph recommendation system | No | No | Yes | Yes |
| Monitor educational level using quiz | No | No | No | Yes |
| Give the most accurate recommendations | Yes | No | No | Yes |

*Table 1 - Research Gap*

This article was provided by Z. Shokrzadeh, M.-R Feizi-Derakhshi, M.-A Balafar and J. Bagherzadeh Mohasefi[10] present an architecture for AI-based recommendation systems, utilizing pre-trained knowledge graph embeddings. The architecture consists of several stages, including creating a knowledge graph, learning low-dimensional vector representations, and fine-tuning pre-trained embeddings. The method outperforms other state-of-the-art approaches in recall, precision, and F1-score, with improvements of 3.87%, 2.42%, and 6.05%, respectively.

The proposed solutions always try to provide the most accurate recommendations for the students and it explain the reason behind the recommendations as well. To get accurate recommendations, need to filter the most accurate recommendation among all the recommendations given by the knowledge graph. Computing the similarity between entities and putting the weights accordingly can filter and provide the most relevant recommendation to the users.

## 1.4 Research Problem

Technology has undeniably transformed the way we live, work, and study. Its influence on the education industry has been significant, with the introduction of e-learning technologies playing a critical part in altering traditional teaching methods. While technology has provided countless benefits, it has also highlighted certain obstacles, notably in the field of basic education. In today's fast-paced society, when both parents frequently work and have various obligations, the convenience provided by e-learning platforms is clear. However, for elementary school pupils, a lack of direct supervision and direction might present substantial challenges in their academic path. Primary school students require more hands-on help and monitoring than their older peers in higher education, who have a higher level of self-discipline and time management abilities.

The shift to e-learning platforms has unintentionally caused a schism between students, instructors, and parents, making it increasingly difficult to recognize and treat academic issues in real-time. This disparity can have far-reaching effects, since failing to recognize and address learning gaps at an early stage can exacerbate problems, ultimately impeding a child's overall development and potential.

One of the most significant issues for both parents and educators is the inability to appropriately assess a primary student's educational performance and identify areas of weakness. While conventional classroom settings allow for more direct observation and engagement, the virtual nature of e-learning platforms may mask these critical signs. Parents may detect a drop in their child's academic performance but without a thorough idea, they cannot make solutions. Moreover, there isn't any suitable platform to get students' educational performance and help to improve their educational levels on their own.

To solve these issues, there is an urgent need for a complete solution that uses technology to monitor primary kids' educational levels in real-time, identify areas of weakness, and give individualized advice to assist them overcome academic obstacles. By leveraging data analytics and machine learning algorithms, such a system might bridge the gap between kids, parents, and instructors, resulting in a more collaborative and productive learning environment.

This suggested approach calls for the creation of a recommendation system that interfaces smoothly with existing e-learning systems. Through a series of subject-specific quizzes and exams, the system would collect data on each student's performance, identifying their strengths and weaknesses across numerous academic fields. This information would then be sent into a knowledge graph recommendation engine, which would examine the student's learning habits, learning styles, and areas of difficulty.

Based on this information, the recommendation system would create individualized learning plans and tactics for each student's specific needs. But anyone doesn't know what happened inside the recommendation system, because it works like Blackbox. The system's real-time monitoring feature would allow for prompt interventions, preventing small academic troubles from growing into more serious issues. The use of such a system may go beyond e-learning, with the possibility of integration into traditional classroom environments. The potential advantages of this method are numerous. Individually, elementary pupils would benefit from a more interesting and individualized learning experience, which would instil a love of learning while also enhancing their confidence and self-esteem. Real-time monitoring and individualized advice would help parents reduce the stress and uncertainty involved with their child's educational path.

In summary, It is an adaptive learning system for enhancing primary students' educational performance and it provides questionnaires for the students' subject-wise and gives the results. Based on the results system monitors the educational level using some machine learning techniques and based on the educational level recommends how to improve students' performance using the recommendation system. using explainable AI techniques to try to provide a reason behind the recommendation.

For instructors and educational institutions, integrating such a system might simplify the process of identifying and resolving learning gaps, allowing for more effective resource allocation and targeted interventions.

## 1.5 Research Objectives

### 1.5.1 Main Objective

The main objective of this application is to develop a recommendation system to enhance primary students' educational performance in the adaptive online platform. Online learning is very famous these days and it's beneficial for t students. But when it applies to primary students some issues that comes up. The main problem is that students' educational performance goes down and there is no perfect way to improve it. As a solution to this problem, a questionnaire-based recommendation system where students, teachers or parents can see current educational performance and what are recommendations along with the reason for the recommendation, he or she must improve educational performance. To implement this, we use Machine Learning techniques and some recommendation mechanisms as well as advanced AI-based techniques like explainable AI.

### 1.5.2 Sub Objectives

**Sub Objective 1:-** How we collect students' academic performance more efficiently?

* + To do this process we make a fully automated questionnaire, which provides various types of quizzes takes the answers, and gives output as a result.

**Sub Objective 2:-** Which type of questionnaires can be used to monitor students’ performance?

* + We are going to provide subject-wise quizzes like mathematics or English, Math quizzes could have several parts like multiplication and division. Also, type of General knowledge quizzes are there because they would help to improve overall educational performance.

**Sub Objective 3:-** How to classify educational levels based on the results more accurately?

* + Get results as a percentage, pass it through the Python function and get the relevant educational level of each student.

**Sub Objective 4:-** How to provide the best and most accurate recommendations based on students' educational performance?

* + Choose a knowledge graph recommendation model to give recommendations because the end goal of the system is recommendations that need to be explained.
  + Create a knowledge graph that links educational levels to specific learning strategies, resources, and activities. Based on the predicted educational level, using the knowledge graph to retrieve recommendations tailored to that level.

**Sub Objective 5:-** How to connect RS to the explainable AI model and provide a reason.

* + For the recommendations generated, provide explanations for why certain resources or strategies are suggested based on the predicted educational level.
  + Explain the connections in the knowledge graph or rules that led to the selection of those recommendations.

# 2. METHODOLOGY

## 2.1 Methodology

### 2.1.1 Requirement Gathering

The gathering of precise requirements is the foundation of every research effort. During this stage, the research subject and background studies were carefully investigated. Extensive evaluation of established criteria offered a thorough understanding of current processes and comparable systems. The scope is methodically developed to define the project's boundaries. To learn more about this application, important stakeholders such as teachers, education professors and an IT professional were interviewed. Feedback from IT department personnel was also gathered to have a comprehensive understanding of the needed system features.

Going through these steps is critical to ensure the success of the project.

* Start with a feasibility study to see if the project is feasible.
* Refer to related research papers to identify any gaps in knowledge.
* Dive into background and literature reviews to grasp context and existing studies.
* Review collected research papers for insights from the study.
* Get user feedback and make sure their expectations align with what the system offers.
* Select components to address project requirements and complete the project scope accurately.

### 2.1.2 Data Collection and Description

Collect data using multiple surveys (interviews with school teachers), use publicly available data (on Kaggle) and access the quiz bank via API to build a quiz. Data was collected using Wikipedia sources through Wikipedia Loader and other educational guides to build a knowledge graph.

### 2.1.3 System Designs

#### 2.1.3.1 Overall system diagram

The following system architecture embodies the core of this research project. The primary objective is to develop an adaptive online learning platform to improve primary education. To improve the educational performance of primary students, the system creates a subject-specific questionnaire and gives separate questions according to subjects, analyses the answers given by the students and determines the level of education of the students according to the respective subject based on the scores obtained, thereby improving the level of education of each student provide recommendations base on their educational level, a face authentication system that improves the login process without having to remember usernames and passwords for easy login of students and makes the login process more user-friendly. To improve the personality of the students, and to address their emotional weaknesses, a unique questionnaire analyses the answers given by the students and issues unique advice to each student. In addition to this, as shown in the diagram, as a method to monitor and maintain the attention of students during the online sections, a method has been created to regain the attention of students who are less attentive by calculating the angle at which the students look away from the computer screen using the camera.

|  |
| --- |
|  |

*Figure 1: Overall System Architecture*

#### 2.1.3.2 Component Architecture

The following component architecture embodies the core of this research component. The primary objective is to develop a recommendation system to enhance primary students' educational performance. To improve the educational performance of primary students, the system creates a subject-specific questionnaire and gives separate questions according to subjects, analyses the answers given by the students and determines the level of education of the students according to the respective subject based on the scores obtained, thereby improving the level of education of each student provide recommendations base on their educational level. To provide recommendations use the knowledge graph recommendation system and use Wikipedia Loder to load educational sources then save them into to CSV file and use Named, Entity Recognizer (NER) to recognize Nodes and relationships. In addition, using custom pre-trained spacy model used for sentence segmentation and building the knowledge graph. Used explainable AI(XAI) to provide the reason behind the recommendation to help provide more explained and valuable recommendations to the students.

|  |
| --- |
|  |

*Figure 2: Component Diagram*

### 2.1.4 Tools and Libraries

* Networkx graphs:- Use to build and handle knowledge graph
* Pandas:- The Pandas library is used to load data in CSV files
* Matplotlib:- Use for visualisation knowledge graph
* NER:- Use for recognising nodes and entities.

### 2.1.5 Knowledge Graph Recommendation System Architecture

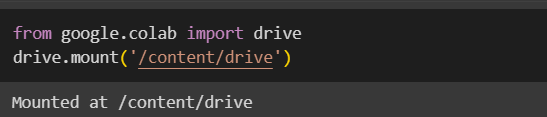
A knowledge graph recommendation system (KGRS) is based on leveraging a knowledge graph to produce more accurate suggestions. This knowledge network functions as a gigantic encyclopedia, holding information about things, users, and how they are linked. Consider an educational sources recommendation system; the knowledge graph may connect books, video lessons, quizzes, and even user preferences. When you engage with the system, it considers your previous decisions and the relationships between learning sources in the knowledge graph. Then it utilizes various algorithms to recommend comparable sources for you to follow.

This recommends instructional materials you've probably already used and helps you find hidden treasures based on the links between educational sources in the knowledge network. KGRSs are an effective tool since they may propose new products that you might have yet to discover on your own and give a more complete context for suggestions.

#### **2.1.5.1 Data Collection and Augmentation**

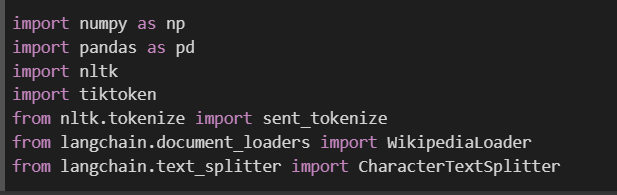
We developed a knowledge graph for generating recommendations. To build a knowledge graph collect data in Wikipedia using Wikipedia Loader and preprocess data and recognize entities and relationships.

**Step 1:-** Mount Google Drive: This step connects your Google Drive to the Colab environment, allowing you to access Wikipedia datasets stored in Drive and save the preprocessed dataset back to Drive for future use.



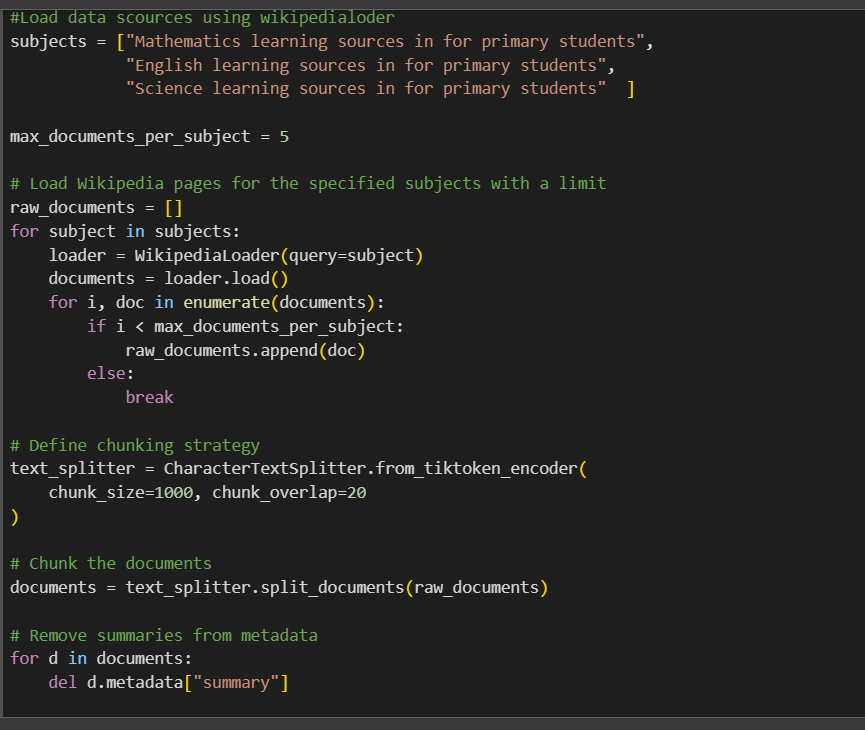
*Figure 3: Mount Google Drive*

**Step 2:-** Import necessary libraries: This step entails importing Python libraries such as pandas, scikit-learn, matplotlib, networkx, numpy, and the Google Colab drive module. These libraries are required for data pre-processing, model training, and saving/loading files from Google Drive.



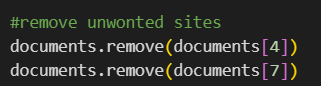
*Figure 4: Import Libraries*

**Step 3:-** Load data using Wikipidealoder: This step loads educational data sources using Wikipidealoder. To do this step we define subjects and the number of articles we want to load then define the chunking strategy and chunk document to split large texts into smaller chunks for further processing.



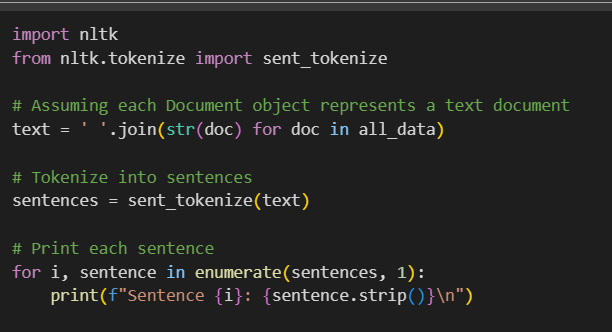
*Figure 5: Load data using Wikipidealoder*

**Step 4:-** Remove unwanted documents: In this step, we removed unwanted documents for future steps.



*Figure 6: Remove unwanted documents*

**Step 5:-** Break down the entire text into individual sentences: In this step combine documents into single “text” variables tokenize the combined sentences using the “sent\_torkanize” function and store them into “sentence” variables

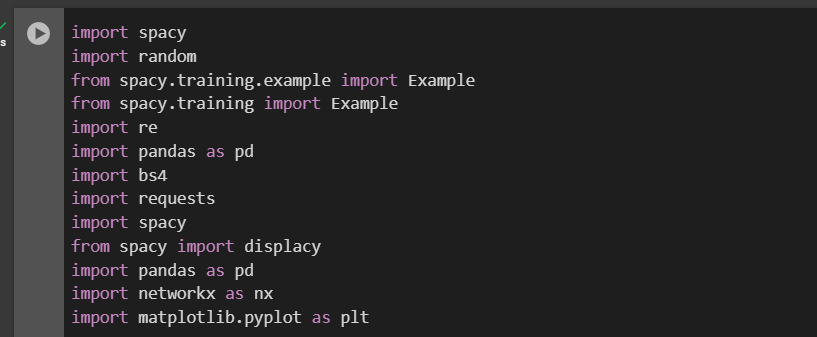
.

*Figure 7: Tokenize sentences*

#### **2.1.5.2 Build Knowledge Graph**

Using Wikipidealoader data, we identify entities and relationships using NER techniques to build a knowledge graph. Before using POS tags use a custom NER model for sentence segmentation and prepare a Wikipedia article dataset.

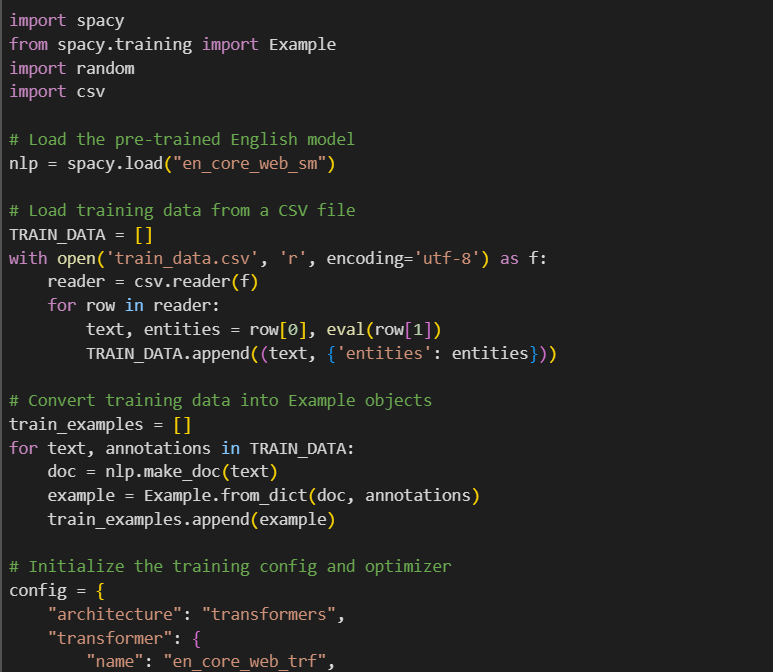
**Step 6:-** Import relevant libraries to build a knowledge graph: Import necessary libraries: This step entails importing Python libraries such as pandas, spacy, random matplotlib, networkx, request, and the Google Colab drive module. These libraries are required for data pre-processing, model training, and saving/loading files from Google Drive.



*Figure 8:- Import Libraries for building knowledge graph*

**Step 7:-** Train custom NER model using the spacy library: The spaCy library implements the Named Entity Recognition (NER) concept. The NER model is trained using a custom dataset supplied from a CSV file, and the learned model is stored on disk for future use.

The trained NER model can recognize and categorize named entities in text data, such as organizations, locations, and people. This model can be important in developing a Knowledge Graph Recommendation System for educational materials since it can extract relevant entities from text sources and create a knowledge graph that represents the relationships between these entities.



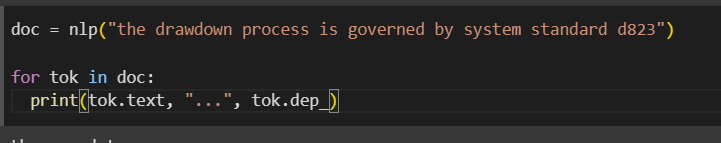
*Figure 9: NER Model 1*



*Figure 10: NER Model 2(Save custom model)*

**Step 8:-** Sentence Segmentation: This line uses the nlp object to generate a Doc object from the given text, which is most likely a natural language processing pipeline (e.g., spaCy). This loop goes through each token (tok) in the Doc object doc. It publishes the following for every token:

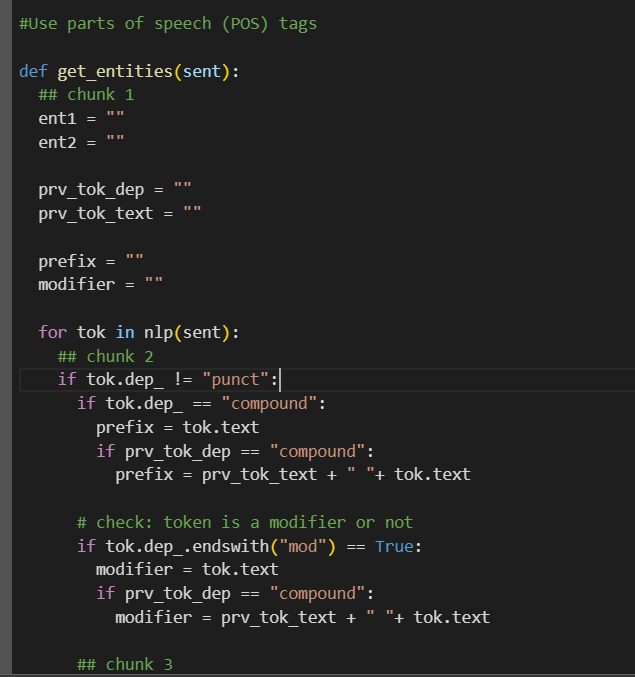
A literal string of three periods separates the token text from the dependency label. The token's dependency label describes its syntactic connection to its headword in the dependency parse.



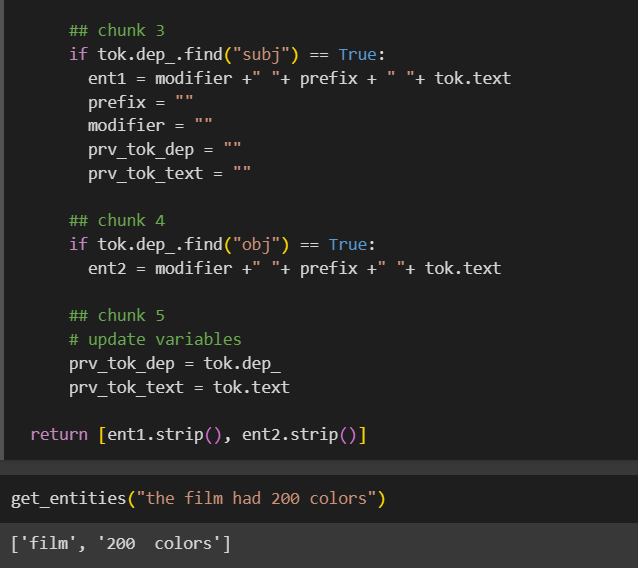
*Figure 11: Sentence Segmentation*

**Step 9.1:-** Entity extraction: The get\_entities function tries to extract two entities from a phrase by iterating over the tokens, identifying subjects and objects based on their dependency relations, and creating the entities using modifiers, prefixes, and the token's content. This function expects the presence of a nlp object, which is most likely a natural language processing pipeline (for example, spaCy or NLTK).

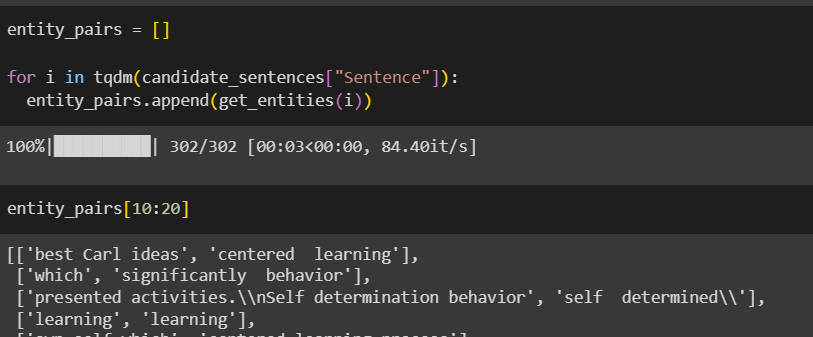
**Step 9.2:** Iterating over a list of potential sentences stored in candidate\_sentences["Sentence"] with the tqdm library, a Python progress bar library. For each phrase i in the list, the get\_entities function is invoked, and a resulting list of two entities is added to the entity\_pairs list. The line entity\_pairs[10:20] returns a slice from the entity\_pairs list that includes index 10 (inclusive) to index 20 (exclusive).



*Figure 12: Entity Extract step 1*

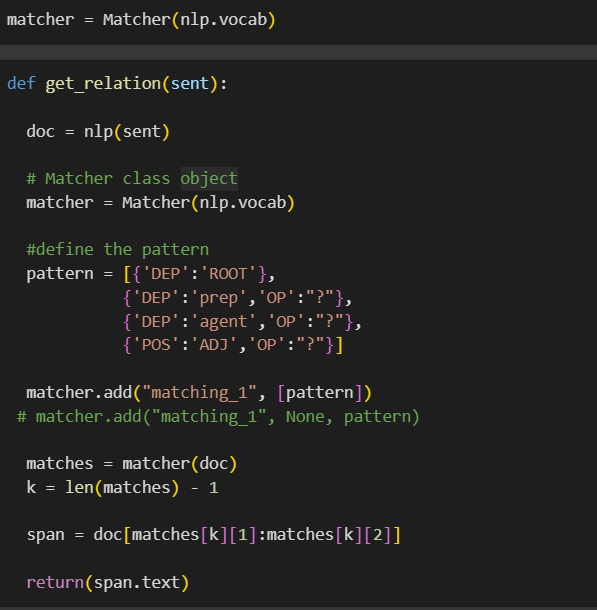


*Figure 13: Entity Extract Step 2*

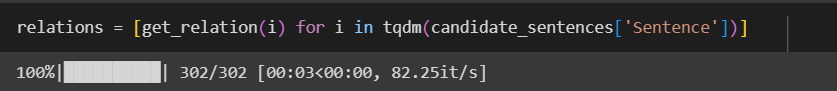


*Figure 14: Entity pairing*

**Step 10:-** Relationship Extraction: We define the method get\_relation, which accepts a phrase as input and attempts to extract a relation from it using the spaCy Matcher object. The pattern corresponds to a root token, followed by an optional preposition, an optional agent, and an optional adjective. The function returns the extracted relation, which is the text of the matched tokens.



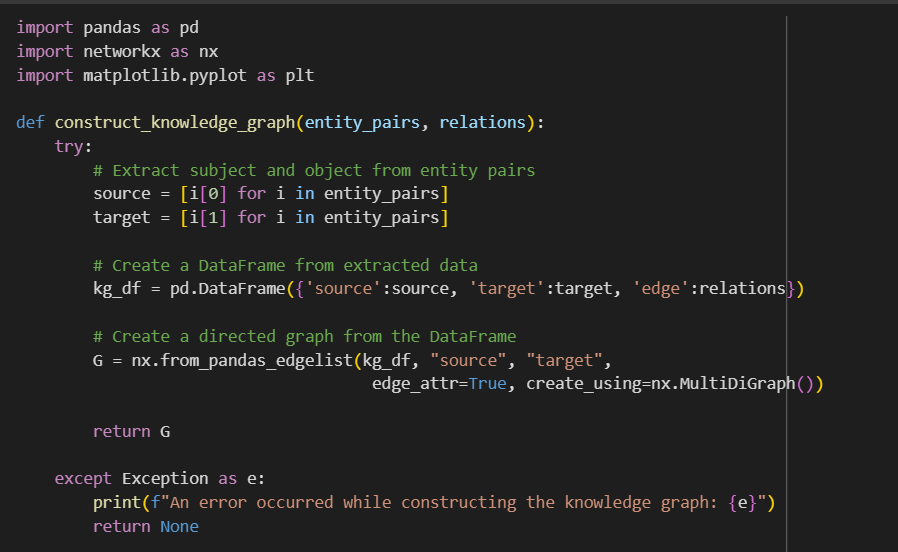
*Figure 15: Relationships Extraction*



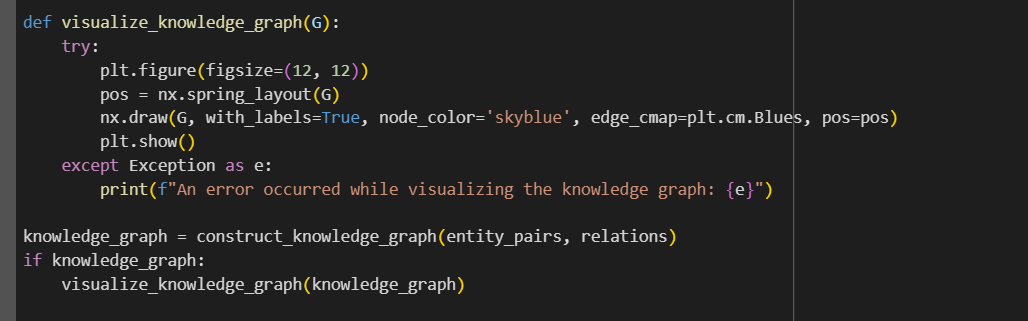
*Figure 16: Relationships list*

**Step 11:** Build a Knowledge Graph: We provide two functions: construct\_knowledge\_graph and visualize\_knowledge\_graph. The construct\_knowledge\_graph function accepts two lists as input: entity\_pairs (a collection of lists containing pairings of entities) and relations (a collection of relations retrieved from sentences). It generates a knowledge graph by first building a pandas DataFrame from the incoming data and then converting it to a NetworkX-directed graph, with nodes representing things and edges representing relations between them.

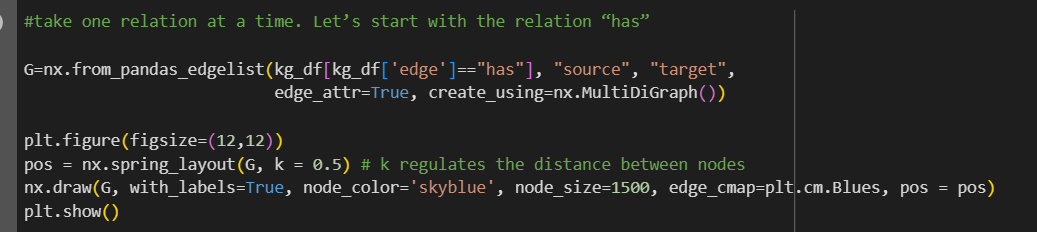
The visualize\_knowledge\_graph function accepts a NetworkX graph as input and creates a knowledge graph display with the NetworkX and Matplotlib libraries. It specifies node colours, edge colours, and layout and shows the final graph. The code combines the two methods by first creating a knowledge graph from the entity\_pairs and relations lists.



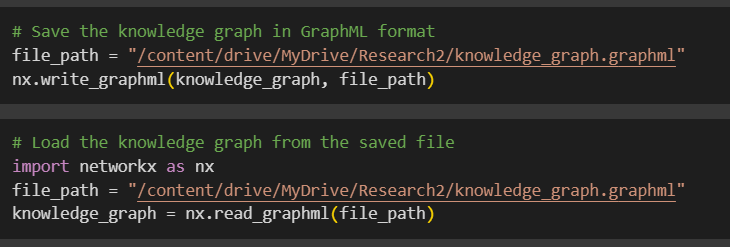
*Figure 17: Knowledge Graph construct*



*Figure 18: Visualize Knowledge Graph*



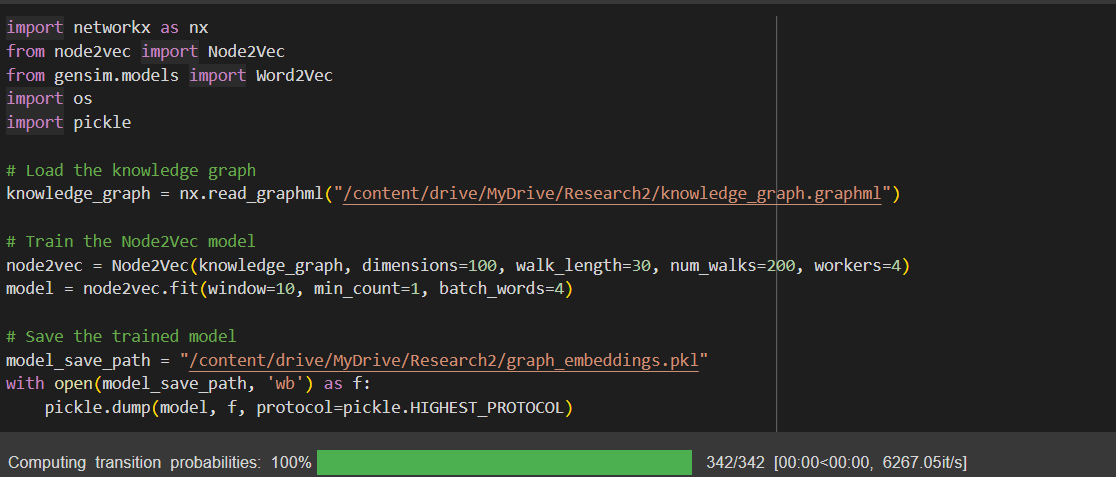
*Figure 19: Visualize Knowledge Graph in relationship-wise*



*Figure 20: Save and Load Knowledge Graph*

#### **2.1.5.3 Recommendation System**

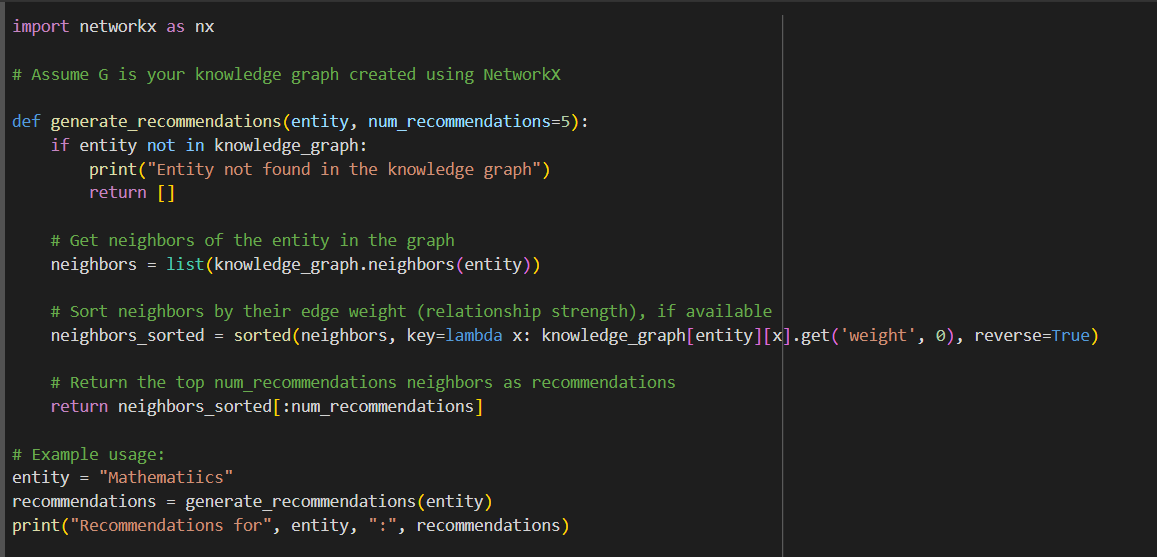
**Step 12:** Pre-train graph-embedding model: To import a pre-trained graph-embedding model, use the Gensim library to load graph\_embeddings.model. The model employs dense vector representations to encapsulate the semantic and structural links among things inside the knowledge network.



*Figure 21:* *Pre-train graph-embedding model*

**Step 13:** Recommendation System architecture: It is assumed by the code that NetworkX has already been used to generate a knowledge graph with the name knowledge\_graph. It outputs the suggestions after using the entity "Mathematics" to invoke the generate\_recommendations method.

This solution uses a knowledge graph represented as a NetworkX graph object to generate suggestions. It sorts the provided entity's immediate neighbours alone, considering edge weights (if available). More advanced methods, such as machine learning models, graph embeddings, or context-aware recommendations, can be applied in backend implementation processes to provide more precise and tailored suggestions to the user.



*Figure 22: Recommendation System Architecture*

# **2.2 Commercialization Aspects of the Product**

### **2.2.1 Target Market**

We concentrate on the fundamental educational unit—the interaction between elementary school instructors, parents, and students—as part of our goal to develop young minds. We understand how important each is in forming a child's foundation in school and general well-being. Our system identifies potential educational gaps that youngsters may encounter, therefore meeting the special demands of this target population. We provide focused direction and assistance to teachers as well as parents, with the ultimate goal of increasing kids' academic performance and giving them the tools they require to succeed.

We want to integrate our technology into the educational environment by reaching out beyond the home. To include our materials in their curricula, we actively cooperate with educational establishments and schools. This guarantees that more instructors and students may easily use our resources, resulting in a more complete and productive learning environment for everyone.

### **2.2.2 Revenue Generate**

We have a flexible revenue plan that targets various user categories.

* **Subscriptions for Individual Users:** We provide tiers of subscription options tailored to the requirements of educators, parents, and even kids (under parental supervision). Features like detailed success reports, individualized learning routes, and access to a larger library of instructional materials might be made available with these programs.
* **Promotion for Educational Institutions:** We are aware of the particular difficulties that affect educational establishments. In addition to premium choices that enable features like gamified learning experiences for students, customizable reporting for instructors, and bulk student accounts for simpler management, we provide freemium models with basic functions. Schools can also take advantage of the platform's promotional options to highlight their initiatives or accomplishments.
* **Premium Features:** To improve the learning process, we provide several premium features. This might include advanced analytics for parents and instructors, curated quiz banks with skill-focused examinations, and access to a personalized recommendation engine that makes learning material suggestions based on individual strengths and limitations.
* **Strategic Partnerships:** We are currently investigating joint ventures with content producers and educational platforms. Together, we can provide value-added services like well-being and emotional intelligence modules, carefully chosen content collections that complement certain curricula, and access to business leaders for seminars and consultations. This enables us to offer a more comprehensive educational program that takes into account students' academic and personal development.

### **2.2.3 Marketing Approach**

**1. User validation and focused beta testing.**

* Educators for a Focus Group: Assemble a broad group of educators from all primary school levels. Organize seminars to present the system and get input on how well it works, how well the content relates to the curriculum, and how it could affect students' learning.
* Pilot Project in Collaboration with Educational Institutions: To introduce the system into classrooms, collaborate with a few chosen schools. Track student progress and use statistics to confirm that it is successful in filling in learning gaps. Review instructor and student input to improve the material collection and recommendation system.

**2. Partnerships for Content and the Freemium Model.**

* Offer a freemium approach in which educators can utilize restricted recommendations or pre-made tests as basic features. For a membership price, provide premium services including individualized learning pathways, comprehensive analytics, and access to a larger knowledge graph.
* Content Partnerships: Work together with creators of curricula and suppliers of educational content. To accommodate different learning styles and grow the knowledge graph, include their excellent course materials in the system.

**3. Awareness of Multiple Channels and Teacher Involvement**

* Targeted Online Marketing: Launch focused online advertising campaigns on social media sites that parents and educators frequently visit. Emphasize the system's advantages, such as enhanced student involvement and tailored learning.
* Webinars and Education Conferences: To highlight the features of the system, conduct webinars and take part in education conferences. To spark curiosity and motivate educators to investigate its possibilities, provide free trials and materials.
* Media Coverage by Industry: Create press releases and media kits emphasizing your system's innovative features and benefits. To get favourable press, get in touch with websites and newspapers that are devoted to education.

**4. Creating a Network of Teachers and Success Stories**

* Educator Community Forum: Establish a specific online discussion board where educators may exchange best practices, work together to solve problems, and exchange ideas.
* Teacher Testimonials: Gather endorsements and success stories from instructors who have witnessed successful results in their learning environments. Use these narratives as highlights on your website and promotional materials.
* Research and Case Studies: Work with partner schools to conduct research and provide case studies that show how well the system works to enhance student learning.

**5. Market expansion and strategic alliances**

* Integrations of Educational Technology: Collaborate with providers of educational technology to link your system to well-known learning management systems (LMS). This will improve usability and streamline the way that current school processes are used.
* Global Growth: Investigate and investigate global markets for possible growth. Take into account localization efforts to modify the system's user interface and content for various linguistic and educational environments.
* Partnerships with Academic Institutions: Advocate for the acceptance of your system as a useful instrument for enhancing primary school results by interacting with legislators and educational authorities.

# **2.3 Testing and Implementation**

### **2.3.1 Functional Requirements**

* **Questionnaire:**
  + The system should be able to provide quizzes for the relevant subject students choose and save the responses.
  + The system should be able to evaluate the answers students are provided.
  + The system should be able to define the educational level of the students based on the results they get in quizzes.
* **Knowledge Graph:**
  + The system should have a Knowledge Graph that can add new nodes when a new student does the quiz and connect particular educational sources to that student.
* **Recommendation system:**
  + The system should provide the best recommendation to the student using the knowledge graph.
* **Result Dashboard:**
  + The system should be able to display the student’s marks when he/she obtains the quiz.
  + The system should be able to display the educational performance of the students' based on the marks they get on the quiz.
  + The system should be able to display relevant recommendations and other relevant information.

### **2.3.2 Non-Functional Requirements**

**Performance:**

* The system should be able to define real-time educational levels to provide recommendations.
* The knowledge graph and recommendation system must provide more accurate recommendations to the student without much delay.

**Scalability:**

* The knowledge graph should be able to dynamically change and increase nodes when the new student does the quiz.

**Availability:**

* The system should be able to be available with minimum downtime and available most of the time for every user.

**Accuracy:**

* The system should be able to provide more accurate recommendations and define educational performance levels correctly.

**Maintainability:**

* The system should be able to fix bugs and handle errors easily.

**Usability:**

* The system should be able to provide clear instructions to the users.
* The user interfaces must be interactive, easy to navigate and user-friendly to the students.

**Reliability:**

* The system needs to function consistently and not experience frequent crashes.

**Security:**

* The system should implement authentication and authorization mechanisms to protect sensitive information.
* The system should be able to encrypt student marks, and educational levels when it is stored in databases.

**Fault Tolerance:**

* The system must be able to recover from malfunctions and manage mistakes or unexpected inputs smoothly.

**Extension:**

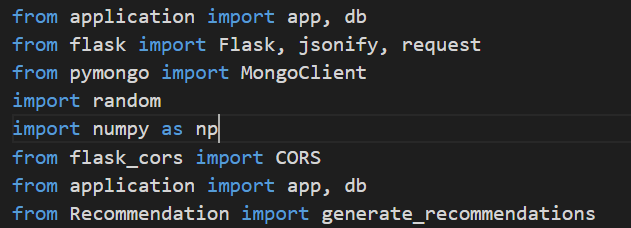
* The system's architecture needs to be modular and adaptable, enabling the future incorporation of new functionalities or system integration.

### **2.3.3 Backend Implementation**

In this section, we discuss how the backend and front end of this component were implemented.

**Import Relevant models and libraries**

* Flask, jsonify, request: Import necessary modules from Flask for building the web application and handling HTTP requests.
* CORS: Import CORS-related modules to enable Cross-Origin Resource Sharing (CORS) for handling requests from different origins.
* MongoClient: Import the necessary module to connect the Python flask app to the MongoDB database.
* Generate\_recommendations: Import generates recommendations function for generating recommendations based on students' educational performances.



*Figure 23: Import Flask libraries*

**Flask App Initialization:**

* Create a Flask app name called “app”.
* Initialize CORS with the app and path used for running front end.



*Figure 24: Flask app initialization*

**“/recommendations” Route:**

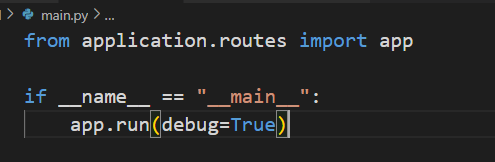
* Define a route call /recommendations that accepts POST requests.
* Define a function call handle\_recommendation\_request() to handle the POST request to /recommendation.
* Inside handle\_recommendation\_request(), extract JSON data from the request using request.get\_json().
* Initialize variables for assigning particular students' data.
* Check students’ educational level through Knowledge Graph.



*Figure 25: ‘/recommendations’ Route*

**App Execute:**

* The if \_\_name\_\_ == '\_\_main\_\_': block ensures that the Flask app is only run when the script is executed directly (not imported).
* Run the Flask app in debug mode (app.run(debug=True)), which enables debugging features and auto-reloads the server on code changes.



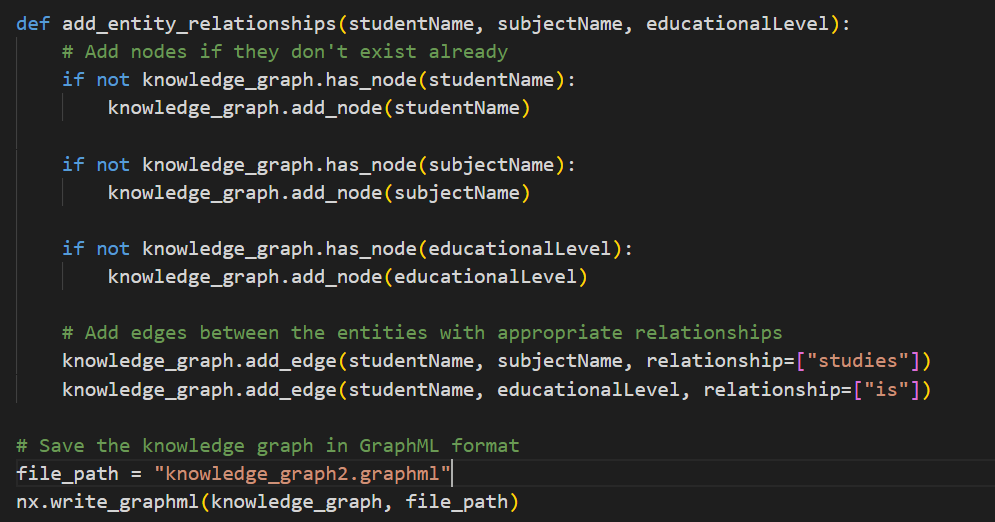
*Figure 26: App execute*

**“Generate recommendations” Function**

* This function is used to generate recommendations for the students based on their educational levels.
* Use knowledge graph and graph embedding model it generate recommendations.
* Considering cosine similarity along with entity embedding and neighbor embedding try to generate more accurate and personalised recommendations. 

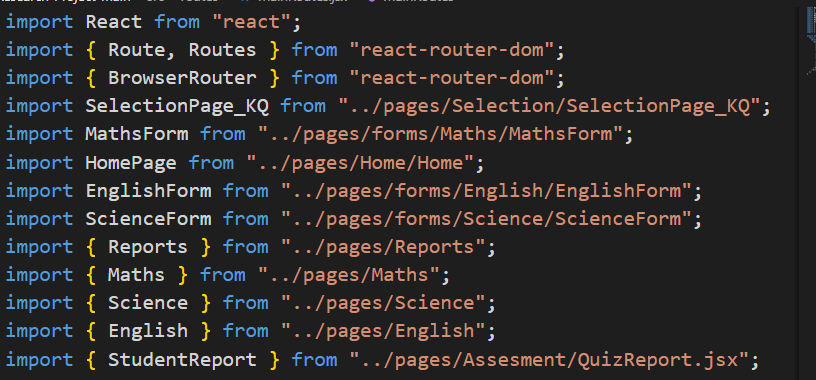
*Figure 27: generate recommendations function*

**‘Add\_entity\_relationships’ function:**

* This function is used to add new nodes and relationships to the knowledge graph.
* Before adding new nodes check that the node is already available in the knowledge graph. If it is not available new node and relationship will be added to the knowledge graph.
* After saving the updated knowledge graph for use generate recommendations. 

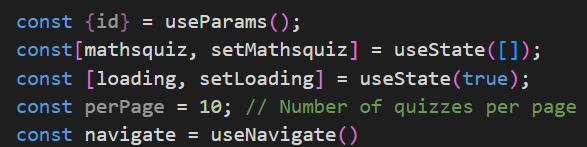
*Figure 28: add\_entity\_relationships function*

**Frontend development:**



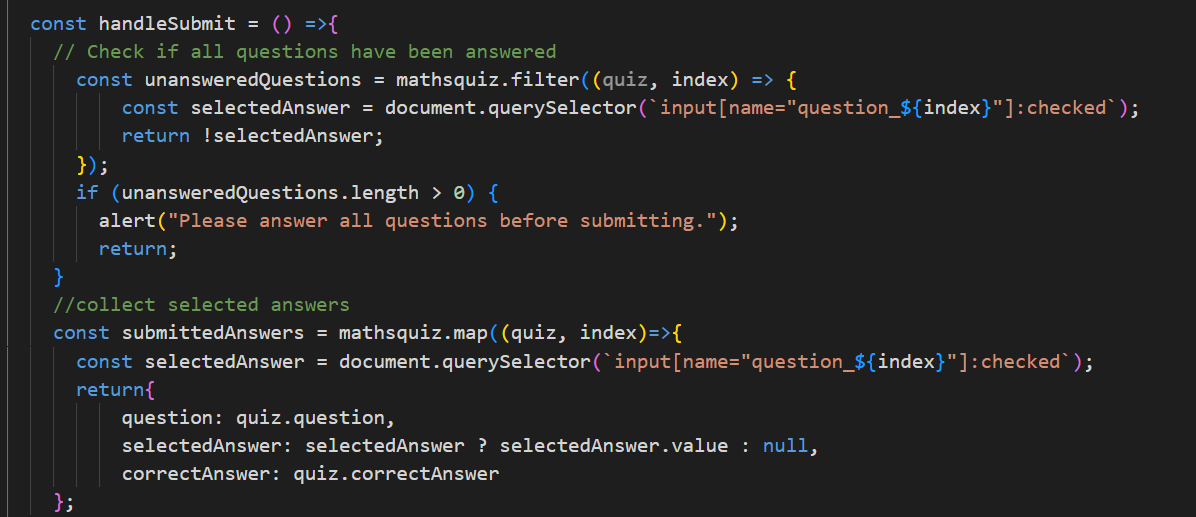
*Figure 29: Import packages for frontend development*

**useState and useNavigate Hooks:**  This code snippet uses multiple useState hooks to manage the state within the MathsQuiz component. Each useState hook initializes a piece of state and provides a function to update that state. In addition useNavigate hook use for pass score of the quiz and subject name to the Quizreport component.



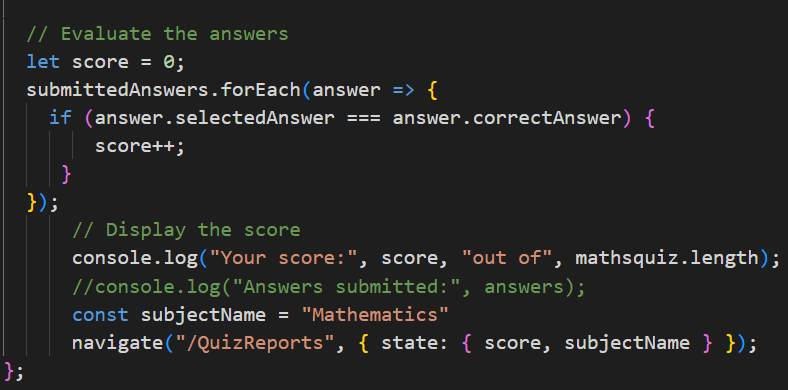
*Figure 30: useState & useNavigate hooks*

**“handleSubmit” Function:** this function is triggered when submitting the question form. When the student clicks the “Submit” button, the system checks if the student answers all the questions. Otherwise, it shows an alert to the student for asking to answer all the questions before submitting.

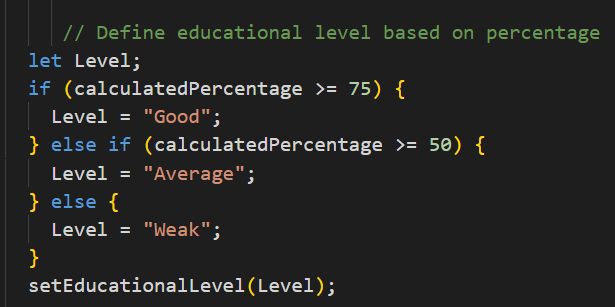


*Figure 31: handleSubmit function*

In addition, it collects the answers students were given and evaluates them. Then define how many correct answers is there to pass it result page.



*Figure 32: Evaluate answers*

Based on the marks students are taking, define educational levels to give recommendations.

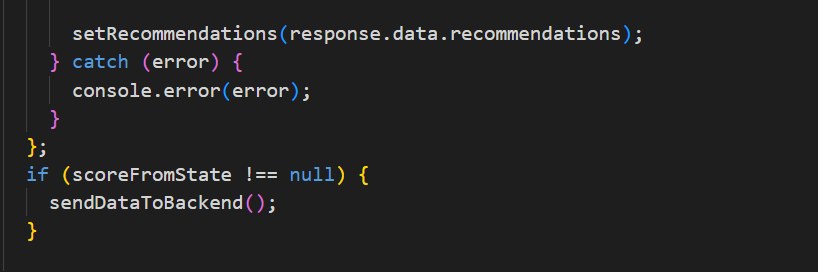
*Figure 33: Define educational levels*

Sent student data to the backend to get recommendations.



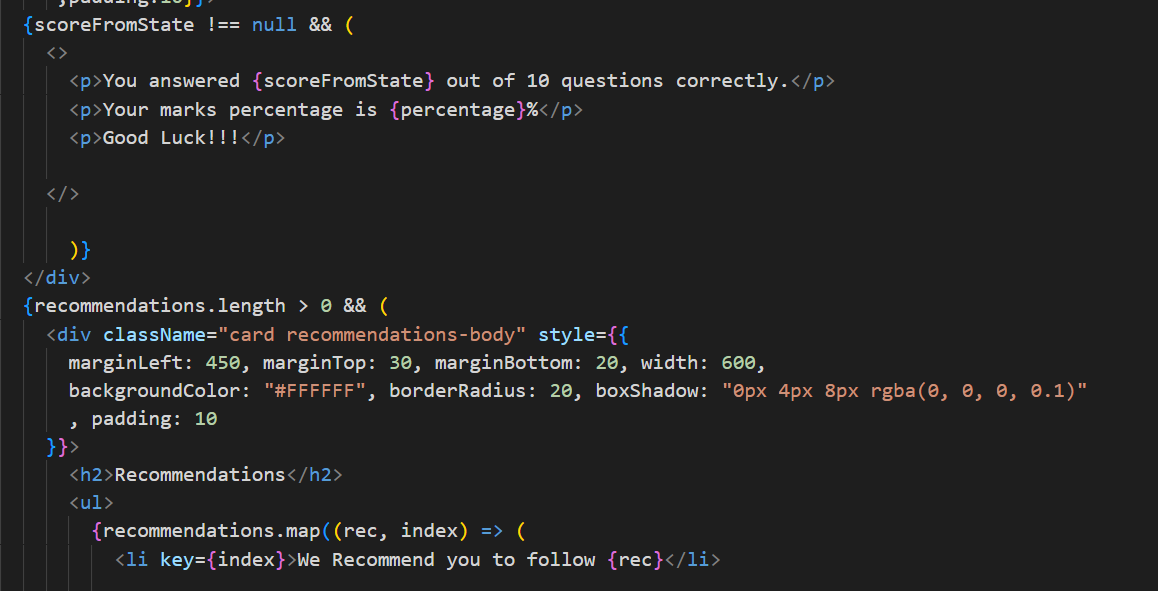
*Figure 34: Send data to the backend*

Get recommendations to the front end.



*Figure 35: Get data to the frontend*

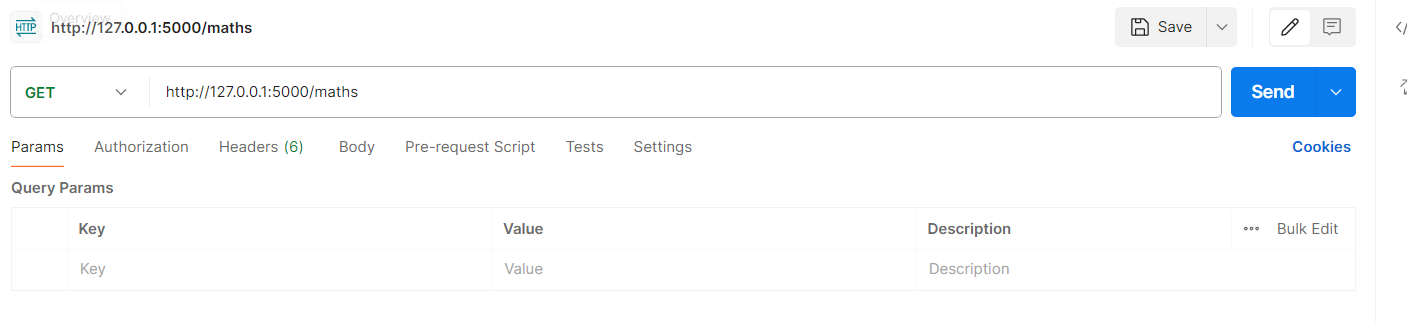
Display Student reports including marks they took on the quiz and recommendations for improving educational performances.



*Figure 36: Display data.*

### **2.3.4 Backend Testing**

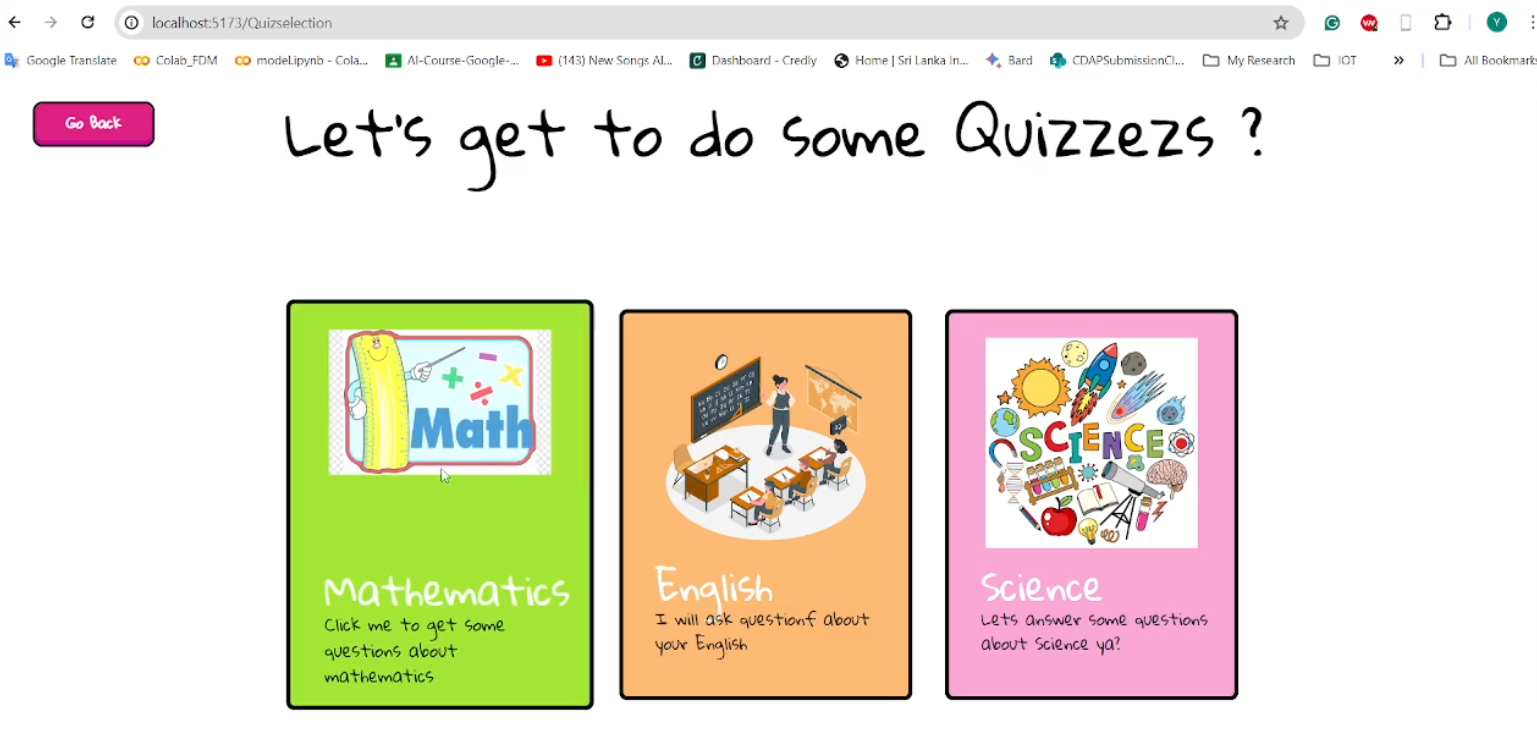
For backend testing, use the Postman tool to send **GET** requests to get data in the MongoDB database to verify APIs working correctly.



*Figure 37: Test backend using Postman*



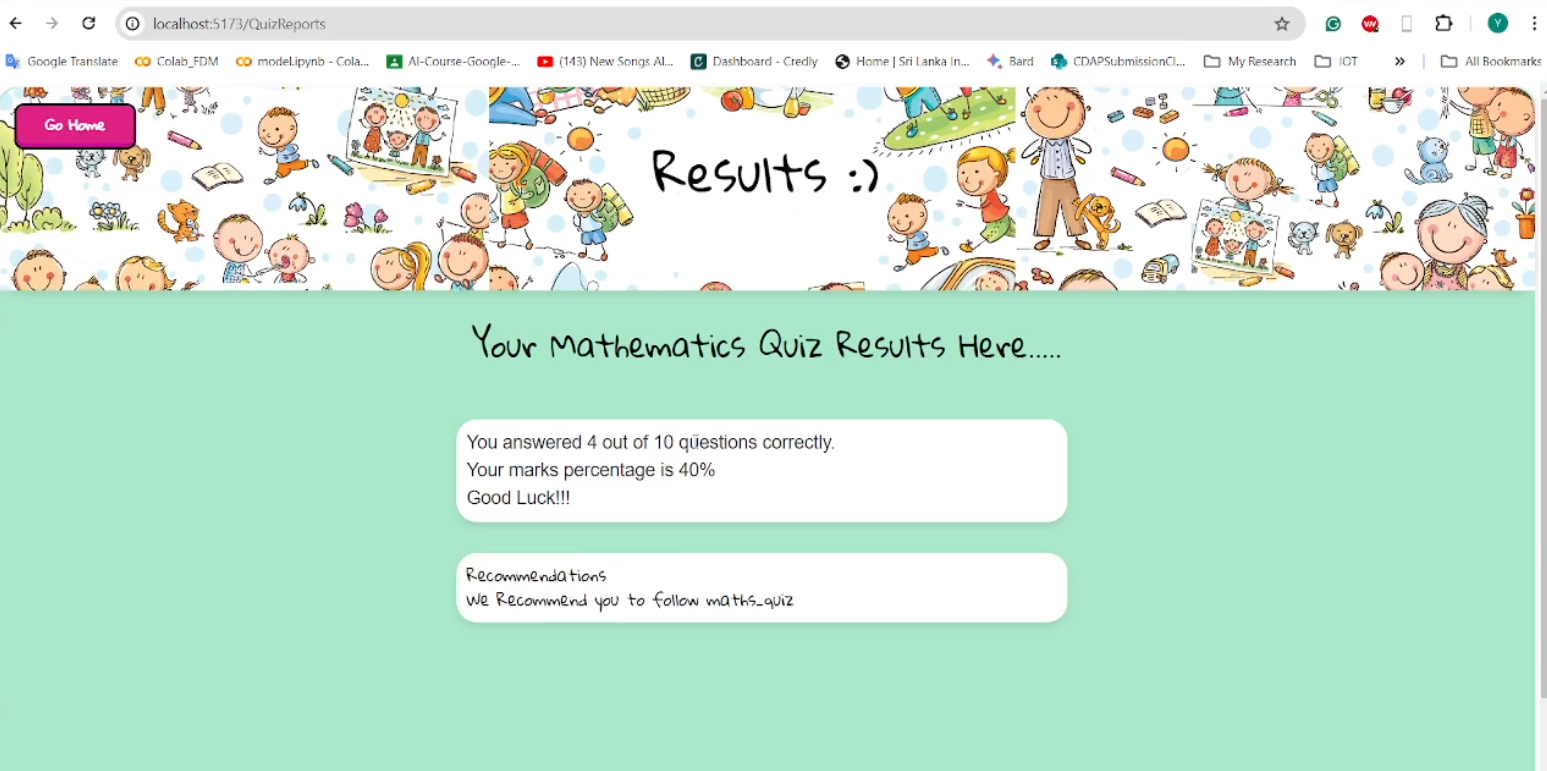
*Figure 38: Home page*



*Figure 39: Quiz Dashboard*



*Figure 40: Quiz interface*



*Figure 41: Report page*

# **3. RESULTS AND DISCUSSION**

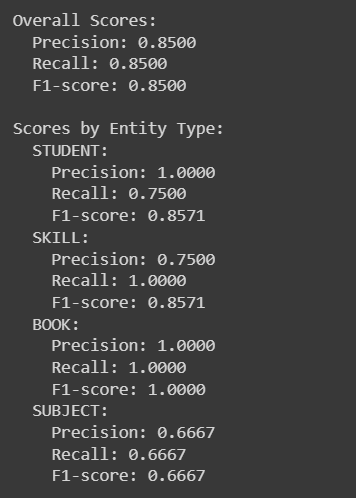
## **3.1 Results**

This section analyses the results and findings obtained from our experiments with the questionnaire, knowledge graph development, recommendation system and XAI results. The experiments aimed to assess the impact of data augmentation, preprocess, NER recognition, and performance of the custom spacy model. Additionally, we discuss how the recommendation system works and the efficiency of the recommendation.

### **3.1.1 Named, Entity Recognition(NER)**

Our work focuses on identifying and extracting relationships between things in instructional texts utilizing Named Entity Recognition (NER) and spaCy. We have trained a pre-trained English model to detect items like people (PERSON), skills (SKILL), books (BOOK), and academic subjects (SUBJECT) by fine-tuning it using domain-specific data. Annotated examples demonstrate the promising performance of our bespoke NER model. For example, in the line "Kamal is good for subtraction," it properly recognized "Kamal" as a PERSON and "subtraction" as a SKILL; in the sentence "Basic\_maths is the average level mathematics book," it correctly identified "mathematics" as a SUBJECT and "Basic\_maths" as a BOOK. In addition to entity identification, we have created functions that extract significant associations. In the first example, the function "is good for" shows Kamal's proficiency with subtraction.

Rich context and insightful information are provided by this method for educational applications, allowing for resource suggestions and tailored learning. With a small training set, our present model is still in its infancy. Sentences including author names or sentences of various complexity levels may be too difficult for it to handle. Our goal is to improve our model's ability to handle intricate educational claims and create a comprehensive educational knowledge graph as we grow our dataset. This study improves the comprehension of instructional materials and their context, laying a strong foundation for future educational AI applications.

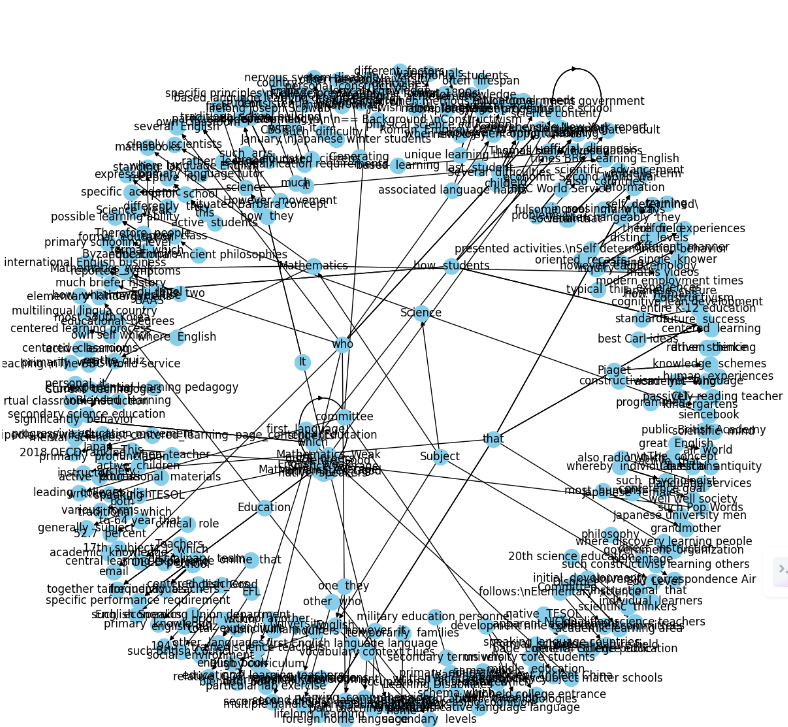


*Figure 42: Classification report for custom NER model*

### **3.1.2 Knowledge Graph**

Using extracted entities and relationships I construct the knowledge graph. The structured information included in instructional texts is captured by the knowledge graph that is created using the entity pairs and relationships that are supplied. Academic disciplines, people, abilities, books, and other things are represented by nodes, and the interactions between these entities are shown by directed edges.

The graph would display nodes for "Kamal," "subtraction," "Basic\_maths," and "mathematics," with edges representing relationships like "is good for" and "is the average level science book" based on sentences like "Kamal is good for subtraction" and "Basic\_maths is the average level mathematics book," for example. The educational domain is represented visually by this graph, which facilitates understanding of the relationships between various entities—such as students and their talents or books and their subjects. Through the identification of important patterns and insights, the visualization supports applications such as resource recommendations and personalized learning.



*Figure 43: Knowledge graph*



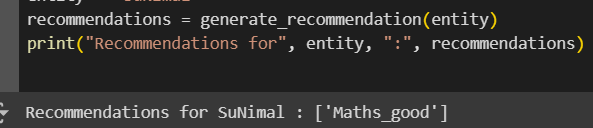
*Figure 44: Entity availability*

### **3.1.3 Recommendation system**

The offered code builds a recommendation system using a knowledge graph and a pre-trained graph embedding model. A GraphML file is used to load the knowledge graph, which contains relationships between educational elements like books, skills, and subjects. Next, these items are mapped to high-dimensional vectors using a pre-trained graph embedding model, which is loaded from a pickle file to capture semantic similarities and associations.

A random user profile vector is made to mimic a person's preferences to produce suggestions. The function returns the embedding of an item in the knowledge graph together with the embeddings of its neighbours. Cosine similarity scores are calculated by comparing the embeddings of the item and its neighbours with the user profile. The highest suggestions are chosen based on the summation of these ratings, which balances the power of the entity and its neighbours.

In addition, the system has a feature that determines whether an entity is an educational resource by looking at node properties and making sure it falls into one of the designated educational categories—books, films, courses, or quizzes, for example. Using the knowledge graph and embedding model, this recommendation system efficiently proposes instructional materials based on a user's profile. For instance, if the input entity is "Basic\_maths," the algorithm might suggest books on advanced mathematics, courses like algebra, and practical abilities like problem-solving as associated educational resources. By matching users' interests with appropriate resources, this strategy improves the educational experience by enabling tailored and contextually relevant recommendations.



*Figure 45: Generated Recommendations*

## **3.2 Research Findings**

The primary objective of this research was to develop an adaptive online learning platform that improves the educational performance of primary school students. A comprehensive approach included a questionnaire on the subjects, based on the marks students obtained on the quiz defining the educational level of the student. Using a custom NER model to extract entities and relationships to construct the knowledge graph. Used a knowledge graph and graph embedding model, to develop a recommendation system to generate recommendations based on the student's educational performance to improve their current educational level.

Our research project created a knowledge graph and recommendation system to apply advanced natural language processing (NLP) techniques to educational materials. Using domain-specific annotations, we first fine-tuned a pre-trained spaCy model ("en\_core\_web\_sm") to identify important educational elements, such as people (PERSON), skills (SKILL), books (BOOK), and academic subjects (SUBJECT). In statements such as "Kamal is good at subtraction," for instance, the model correctly classified "subtraction" as a SKILL and "Kamal" as a PERSON. Likewise, in "Basic\_maths is the average level mathematics book," "mathematics" was classified as a SUBJECT and "Basic\_maths" as a BOOK. The algorithm was able to learn and correctly tag these things thanks to annotated training data.

We created custom routines that use dependency parsing to extract relationships between entities in addition to entity recognition. The functions successfully detected significant associations, such as "is good for" in "Kamal is good for subtraction." By giving the recognized items context, this connection extraction significantly increased the depth of the knowledge acquired. With NetworkX, we created a directed knowledge graph with the entities and relationships that were extracted. This graph provided a structural and visual representation of the educational domain by representing items as nodes and their relationships as edges. This graph's representation made it simpler to comprehend how various things are associated by illuminating how intertwined educational resources are.

Next, we used a pre-trained graph embedding model to create a recommendation system. To capture the semantic commonalities between things, this model projected them to high-dimensional vectors. We generated random user profile vectors and calculated cosine

similarity scores between these profiles and the entity embeddings to mimic user preferences. Using this method, we were able to provide customized recommendations. For example, if the input entity was "Basic\_maths," the algorithm might recommend books on advanced mathematics or other relevant areas as related educational resources. We put in place a screening procedure that confirmed the entities' educational nature to guarantee the relevancy of the suggestions. To achieve this, node properties like type and category were checked to ensure that only educational materials including books, films, courses, and quizzes were included. The effectiveness of this filtering system was essential to preserving the recommendations' relevance and calibre.

## **3.3 Discussion**

The preceding section's results provide insightful information on how well the Knowledge Graph (KG) and Recommendation System (RS) function as well as how to make more accurate recommendations. The purpose of this discussion part is to clarify the wider importance of this research and to go deeper into these results.

### **3.3.1 Knowledge graph and Recommendation System performance**

Strong results have been obtained in detecting educational items and connections inside elementary school texts using the proprietary Named Entity Recognition (NER) model. It outperformed generic models in properly identifying people, abilities, books, and academic disciplines. The basis for building an accurate knowledge graph that reflects the educational domain is laid by the model's accuracy in entity and relationship extraction. The accuracy of entity recognition and connection extraction was enhanced by the combination of NER, knowledge graph, and graph embedding model, which surpassed conventional recommendation algorithms. This innovative method supports tailored teaching and has the potential for further use in the field of education.

### **3.3.2 Practical Implementations**

The adaptive online learning platform that has been built holds great practical implications for improving primary school students' educational experiences. The platform creates a complete knowledge graph, detects important educational entities and relationships, and provides individualized learning suggestions by employing powerful natural language processing (NLP) methods and machine learning models. By adjusting instructional materials to meet the requirements of each student, this adaptive method promotes a more stimulating and productive learning environment. Teachers may use more focused and efficient teaching tactics by using the system's individualized suggestions, which can help them understand the individual learning profiles and academic achievement of each of their pupils. The involvement, motivation, and academic performance of students can all be enhanced by this individualized help.

The approaches and technology created in this study have wider uses outside of elementary schools. The recommendation system and knowledge graph are adaptable to many subjects and educational levels, making them a useful tool for teachers. Moreover, additional fields including online courses, professional training, and educational content delivery systems might benefit from the use of customized learning and entity-relationship extraction techniques.

### **3.3.3 Limitations, Challengers and Future Developments**

The availability of quality and diverse educational data for primary students might be limited, which could impact the performance of machine learning models and the breadth of recommendations. In addition, Educational levels and performance might not be solely determined by quiz scores. Other factors like individual learning styles, external influences, and socio-economic background could also play a significant role. Also, Highly interpretable models might sacrifice predictive accuracy.

The main challenge is Choosing and implementing appropriate explainability techniques that can effectively communicate the reasoning behind predictions and recommendations. Also designing a recommendation system that can provide personalized recommendations for each student's unique learning needs and preferences.

Further personalization of knowledge graph-based recommendations is viable, potentially aided by AI chatbots for customized student queries. Reinforcement learning presents another avenue for an increasingly tailored system.

# **4. CONCLUSION**

The creation of an adaptive online learning platform to improve primary school children's academic performance was the main goal of this research project. This was accomplished by using a multimodal strategy that included tests to ascertain students' educational status and questionnaires to assess students' competency levels. Our study used a customized Named Entity Recognition (NER) model that was adjusted with annotations appropriate to the domain to precisely identify and extract things including people (PERSON), books (BOOK), skills (SKILL), and academic subjects (SUBJECT). In statements such as "Kamal is good at subtraction," for example, the model classified "subtraction" as a SKILL and "Kamal" as a PERSON. According to the statement, "Basic\_maths is the average level mathematics book," "mathematics" is a SUBJECT and "Basic\_maths" is a BOOK.

In addition to identifying entities, we created unique methods that utilized spaCy's dependency parsing capabilities to derive significant connections among entities. These connections, like "is good for" in "Kamal is good for subtraction," were essential in giving the information gathered dimension and nuance. We used NetworkX to create a directed knowledge graph using the entities and relationships that we had retrieved. Nodes in the network stood for the entities, and edges for their relationships. The interconnectedness of educational resources was demonstrated by this graph, which offered a visual and structural picture of the educational realm.

We created a recommendation system with a pre-trained graph embedding model to customize the learning process. To capture the semantic commonalities between things, this model projected them to high-dimensional vectors. We generated individualized suggestions by calculating cosine similarity scores between these profiles and the entity embeddings by incorporating user profile vectors, which simulated user preferences. For instance, if the input entity was "Basic\_maths," the algorithm might recommend books on advanced mathematics or other relevant areas as related educational resources.To make sure that suggestions are relevant, we put in place a filtering system that checks node properties like type and category to confirm that the entities are educational. Making sure that only relevant educational materials were recommended was crucial to preserving the suggestions' quality and applicability.

In conclusion, our study effectively illustrated how to parse and comprehend instructional texts using natural language processing (NLP) and machine learning approaches to create a knowledge tree and a customized recommendation system. This adaptive learning platform shows how many entities are connected and gives a comprehensive perspective of the educational area in addition to personalized recommendations. Our research provides a sophisticated and contextualized comprehension of educational information, laying a solid foundation for future educational AI applications. In the end, our study improves adaptive curriculum tools and intelligent tutoring systems greatly, which benefits primary school children's academic performance.

# 5. REFERENCES

[1] P Ramya and S G Balakrishnan and M Kannan, *Recommendation system to improve students performance using machine learning,* Tamil Nadu, India: Department of Computer Science and Engineering, Mahendra Engineering College, 2020.

[2] Qian Zhang, Jie Lu, Guangquan Zhang, Recommender Systems in E-learning, Sydney: University of Technology, 2020.

[3] Shanshan Wan and Zhendong Nilu, *A Hybrid E-Learning Recommendation Approach Based on Learners’ Influence Propagation,* Beijing: [University of Civil Engineering and](https://www.researchgate.net/institution/Beijing_University_of_Civil_Engineering_and_Architecture) Architecture, 2019.

[4] Jiahong Su and Weipeng Yang, *Artificial intelligence in early childhood education: A scoping review*, China: Faculty of Education, The University of Hong Kong, 2021.

[5] Matthew Etherington, *E-Learning pedagogy in the Primary School Classroom: the McDonaldization of Education,* Vol.3. Australia: Australian Journal of Teacher Education, 2008.

[6] Fergyanto Efendy Gunawan, *A Learning Recommendation to Improve Electronic Textbook Learning Experience,* Vol.13. Japan: ICIC EXPRESS LETTERS, 2019.

[7] J. Bobadilla, F. Serradilla, A. Hernando, MovieLens, *Collaborative filtering adapted to recommender systems of e-learning,* Vol.22. Madrid Spain: Universidad Politecnica de Madrid, 2009.

[8] Salman Fraihat and Qusai Shambour, *A Framework of Semantic recommendation system for E-learning*, Jordan: Software Engineering Department, Faculty of Information Technology, Al-Ahliyya Amman University, PO Box 19328, Amman, 2014.

[9] [Roberto Confalonieri](https://wires.onlinelibrary.wiley.com/authored-by/Confalonieri/Roberto), [Ludovik Coba](https://wires.onlinelibrary.wiley.com/authored-by/Coba/Ludovik), [Benedikt Wagner](https://wires.onlinelibrary.wiley.com/authored-by/Wagner/Benedikt), [Tarek R. Besold](https://wires.onlinelibrary.wiley.com/authored-by/Besold/Tarek+R.), *A historical perspective of explainable Artificial Intelligence*, Vol.11. [WIREs Data Mining and Knowledge Discovery](https://wires.onlinelibrary.wiley.com/journal/19424795), 2020.

[10] Z. Shokrzadeh, M.-R. Feizi-Derakhshi, M.-A. Balafar, and J. Bagherzadeh Mohasefi, “Knowledge graph-based recommendation system enhanced by neural collaborative filtering and knowledge graph embedding,” *Ain Shams Engineering Journal*, p. 102263, Apr. 2023, doi: https://doi.org/10.1016/j.asej.2023.102263.

[11] R. Yang, Boming Yang, Sixun Ouyang, Tianwei She, Aosong Feng, Yuang Jiang, Freddy Lecue, Jinghui Lu, Irene Li, “Leveraging large language models for concept graph recovery and question answering in NLP education,” *arXiv (Cornell University)*, Feb. 2024, doi: 10.48550/arxiv.2402.14293.

[12] R. Bakker, Gino J. Kalkman, Ioannis Tolios, Dominique Blok1, Guido A. Veldhuis, Stephan Raaijmakers, and Maaike H.T. de Boer “Exploring Knowledge Extraction Techniques for System Dynamics Modelling: Comparative Analysis and Considerations,” BNAIC/BeNeLearn 2023, Netherlands Organisation for Applied Scientific Research (TNO) 2 University of Leiden, Oct. 2023.