BlOOMQUEST: A PERSONALIZED LEARNING PLATFORM

Wijayasena W.D.N.D.T

Senaweera T.I.S

Weerasekara N.N

Colambage K.G

BSc (Hons) in Information Technology

Specializing in Data Science and Software Engineering

Department of Information Technology, Computer Science and Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

February 2023

BlOOMQUEST: A PERSONALIZED LEARNING PLATFORM

Wijayasena W.D.N.D.T

Senaweera T.I.S

Weerasekara N.N

Colambage K.G

Dissertation submitted in partial fulfillment of the requirements for the Bachelor of Science in Information Technology Specializing in Data Science and Software Engineering

Department of Information Technology, Computer Science and Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

February 2023

Declaration

"We declare that this is our 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, We hereby grant to Sri Lanka Institute of Information Technology, the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. We retain the right to use this content in whole or part in future works (such as articles or books)."

Signatures:

IT20133368	Wijayasena W.D.N.D.T.	
IT20123468	Senaweera T.I.S.	
IT20133504	Weerasekara N.N.	
IT20126438	Colambage K.G.	

The above candidate has carried out research for the bachelor's degree Dissertation under my supervision.

Signature of the supervisor: Date:

Abstract

Creating high-quality and balanced exams for undergraduates that align with their diverse cognitive capabilities is a vital undertaking. Bloom's Taxonomy serves as a widely utilized framework among university lecturers to accomplish this task and evaluate student performance. However, it is concerning that most students are unaware of Bloom's Taxonomy and its impact on their learning experience, leading to difficulties in comprehending the depth and complexity of the subjects they study. As Bloom's Taxonomy forms the basis for many university papers, it is crucial for students to possess at least a basic understanding of each level within the taxonomy. This research paper addresses the research problem of how undergraduate students can identify and understand their cognitive level as classified by Bloom's Taxonomy, as well as develop the skills necessary to progress to higher levels. To tackle this issue, a personalized self-learning system is proposed, guiding students through the application of Bloom's Taxonomy to their learning process. By integrating Bloom's Taxonomy, students can identify areas where they need to concentrate their efforts to achieve their learning objectives. The system comprises four components: generating comprehensive mind maps using knowledge graphs, generating categorized questions and answers, tracking, and predicting student performance, and providing online extra study resources. By leveraging knowledge graphs, the system generates mind maps that visually represent the interconnectedness of topics, aiding students in understanding the hierarchical structure of knowledge. The categorized questions challenge students at different cognitive levels, helping them identify their current level and work towards higher levels. The system tracks performance, offering insights for improvement, and provides tailored online resources. This integration aims to empower undergraduate students, bridge the knowledge gap, and enhance the overall learning experience, ultimately enabling students to achieve their learning objectives effectively.

Keywords: Personalized self-learning system, Bloom’s taxonomy, Knowledge graphs, Machine learning.

Acknowledgement

We embarked on this research project with an abundance of support, knowledge, and inspiration. We express our heartfelt gratitude to everyone who was instrumental in shaping our research.

At the forefront, our deepest gratitude is directed towards our supervisor, Mr. Prasanna Sumathipala. His unwavering guidance, patience, and profound expertise were the compass guiding us through the multifaceted aspects of our research. His invaluable feedback laid the foundation for our research's success. Equally, our co-supervisor, Mr. Sathira Hettiarachchi, earns our sincere thanks for his invaluable advice and mentorship.

To the Research Panel, we extend our heartfelt thanks for their depth of expertise, dedication, and insightful feedback. Their keen insights and valuable recommendations were instrumental in refining and elevating the caliber of our research.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Description
EWMA	Exponentially Weighted Moving Average
NLP	Natural Language Processing
ML	Machine Learning
NER	Named Entity Recognition
RC	Relation Classification
KB	Knowledge Base
HTML	HyperText Markup Language
NLTK	Natural Language Tool Kit
TF-IDF	Term Frequency - Inverse Document Frequency
ER	Entity Relationship
FRE	Flesch Reading Ease
LDA	Latent Dirichlet Allocation
URL	Uniform Resource Locator
API	Application Programming Interface

1. INTRODUCTION

University students face enormous pressure and stress when it comes to their academics today. Many students find themselves struggling to keep up with a wide range of subjects to cover and an enormous number of assignments and exams. Some students do not have enough time to refer to all lectures, some students may not know how to study efficiently in order to achieve good grades. Many students try to do self-learning to stay on the correct track. But unfortunately, despite all the hard work they may still get lower grades for subjects than they anticipated. There can be many reasons for this, but through our research, we found that one of the main reasons may be that the student did not fully understand their study material and are unaware that the exam papers are made according to Bloom's Taxonomy, which is a framework for classifying objectives and goals in education.

University lecturers utilize Bloom's taxonomy to design and develop standardized exam papers [1]. Bloom's taxonomy offers a hierarchical framework for organizing educational objectives and goals based on cognitive levels. By incorporating the taxonomy, lecturers ensure that examination questions align with the desired cognitive skills students should demonstrate. However, many students may not be aware of this approach and are unable to respond to questions based on Bloom's Taxonomy. As a result, students may struggle to perform well on their exams and end up with low grades for exams. It is especially important for students to become familiar with Bloom's Taxonomy, as it can help them understand the type of questions being asked and how to approach them.

This research addresses the research problem presented by the knowledge gap surrounding Bloom's Taxonomy and its impact on undergraduate learning. By incorporating Bloom's Taxonomy into a personalized self-learning system, we seek to provide students with the tools and guidance needed to identify their cognitive level and develop the skills required to progress. While there are existing systems that facilitate self-learning by allowing students to search for topics of interest [2], our system goes a step further by empowering students to take control of their learning experience. Our platform enables students to upload their own study materials, which serves as the foundation for generating mind maps, questions, and answers based on Bloom's taxonomy. This personalized approach enables students to track their performance for each module, offering valuable insights into their current standing and progress. Moreover, our system incorporates advanced search functionality, empowering students to explore extra study materials using long queries or even paragraphs.

In this personalized self-learning system, mind maps are generated using knowledge graphs. This approach leverages the interconnectedness of concepts and topics within a subject, creating visual representations that aid in comprehension and retention. By utilizing knowledge graphs, our system captures the relationships between various study materials, allowing for the generation of comprehensive mind maps that facilitate a holistic understanding of the subject matter.

Through the amalgamation of Bloom's Taxonomy, personalized content generation, performance tracking, and advanced search functionality, the system empowers students to excel academically. By adopting this approach, students can develop a deeper understanding of the study material, enhance critical thinking skills, and effectively address their weaknesses. This self-learning system provides students with the means to achieve their academic goals and succeed in their examinations.

A. Initial Survey

To assess the level of awareness regarding Bloom's Taxonomy and its application in exam papers, a survey was conducted among undergraduate students at the Sri Lanka Institute of Information Technology (SLIIT). The survey involved the distribution of a questionnaire, which required the students' participation in providing answers. The aim of the survey was to determine the extent of students' familiarity with Bloom's Taxonomy (Figure 1) and their awareness of its utilization in the creation of exam papers (Figure 2).

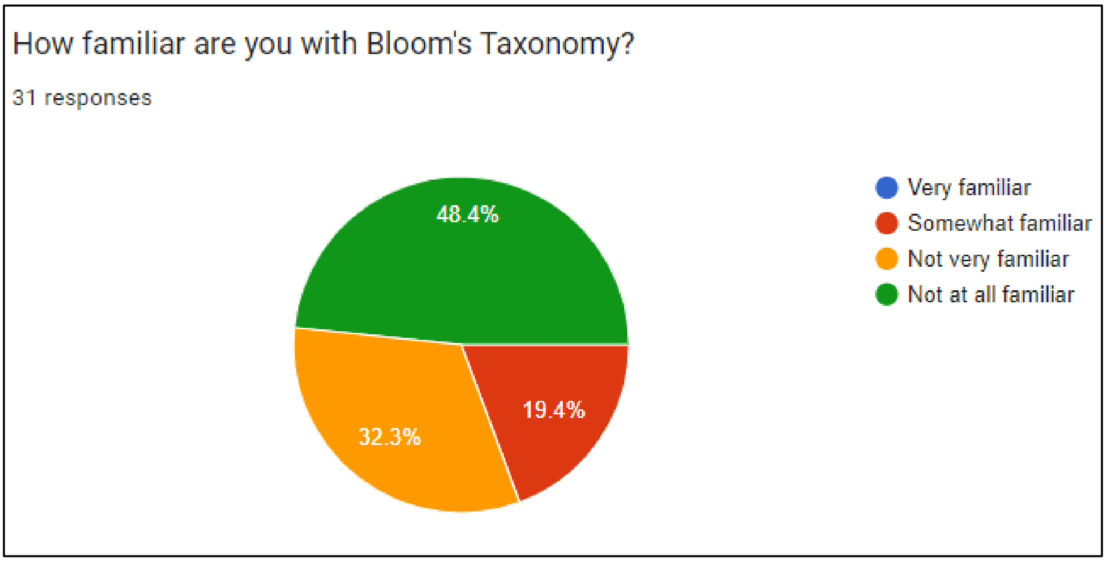


Figure 1: Students' Familiarity with Bloom's Taxonomy.

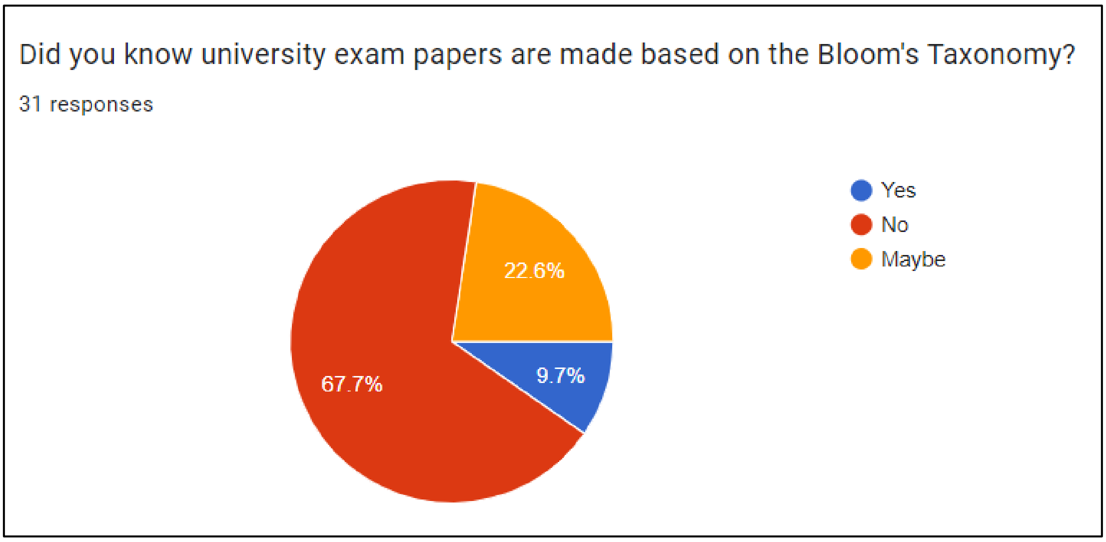


Figure 2: Awareness of Utilizing Bloom's Taxonomy in Exam Papers.

To supplement our research on the identified issues, we sought additional testimonials by analyzing the responses obtained from the survey. These testimonials were instrumental in validating the problems highlighted in our study. In addition to the testimonials, we also conducted an extensive literature review to gather relevant information from existing scholarly sources. The combination of testimonials and the literature review provided comprehensive insights into the challenges faced by students in relation to Bloom's Taxonomy and its incorporation into exam papers.

1.1. Background Literature

In mind-map generation aspect, a research study for creating mind maps from articles using machine learning was completed in 2019 by M.F. Kuroki, L.S. Riza, and Rasim at the department of computer science education at Universitas Pendidikan Indonesia. [3] For the data collection they have used articles. The topic sentence of a paragraph is chosen using the information retrieval approach, pre-processing, core NLP, and feature extraction approaches in model creation. Application development uses a stage-by-stage, linear process model. The system's output is contrasted with the average values produced by two human experts, whose accuracy rate averages 53.55%. This indicates a moderate level of system correctness.

In 2013, Ayu Purwarianthi, Athia Saelan, Irfan Afif, Filman Ferdian, and Alfan Farizki conducted research on developing an autonomous mind map generator in Indonesian language at the school of electrical engineering and informatics institution in Indonesia. [4] Indonesian Mind Map Generator utilizes Indonesian natural language understanding tools such as a POS tagger, syntactic parser, and semantic analyzer to facilitate easy creation of Mind Map objects. The tools' accuracy rates for the POS tagger, syntactic parser, and semantic analyzer are 96.5%, 47.22%, and 62.5%, respectively. They were created with the aim of addressing the dearth of Indonesian language resources. The Mind Map generator also employs radial drawing visualization and an editor for modifications. In evaluation, the Mind Map object was easily understood for simple sentences by five respondents.

In 2011 Robert, Mirko and Mladen from university of Zagreb, Croatia has implemented a mind map generating software model by using text mining algorithm. [5] This software is compatible with desktop, laptop, PDA, and mobile devices. A web service based on SOA is advised because PDAs and mobile phones may execute slowly. The algorithms will be run by the web service, which will then produce a mind map that will be saved on a database server. All mind maps can be searched for and downloaded by users, and the database can be used to integrate mind maps and conduct additional research. This software has accuracy related to other systems.

In an aspect of question-and-answer generation, research done by how question difficulty can be estimated in the context of community question and answering services. There the Research in the DB and Semantic Web communities has investigated how structured queries formulated in SQL [6] or SPARQL [7] can be paraphrased in natural language processing use and generate questions and answers.

This research done by Question answering is an indispensable link in high school teaching.[8] The system, which is the first innovative application in the field of high school teaching, integrates knowledge graph technology and intelligent question answering technology, introduces big data technology. It can solve students' questions in time and accurately, link the knowledge points related to the questions to help students construct knowledge network graph, and the big data technology is used to analyze the students' questioning behavior and to predict students' learning behavior in order to feedback the teaching effect.

Predicting university students' academic performance has been a popular research area among researchers and educators. Several approaches have been undertaken in the quest to comprehend and predict student performance.

In the recent decade, machine learning has significantly influenced the domain of academic performance prediction. With its robust ability to process large datasets and unveil hidden patterns, researchers have eagerly explored diverse machine learning techniques [9], [10]. Ma'sum [11], for example, applied regression methodologies in a computer science module by applying models such as linear regression, support vectors, and decision trees on various assessment performance measures.

Further expanding this arena, Uthej and Lokesh [12] amalgamated regression and classification techniques, incorporating an expansive set of variables from assessment scores to engagement data. Their primary aim was to discern a binary outcome: the pass or fail status of students in a module.

Delving deeper, the research community has shown interest in more intricate algorithms, such as decision trees and random forests [13]. The investigations by [14] and [15] are particularly noteworthy, as they synergized assessment scores with variables like student engagement, departmental links, and faculty information to refine their predictions.

While machine learning continues its ascendancy in educational prediction, cognitive frameworks also offer promise. Among them, Bloom's Taxonomy remains a foundational construct. However, its integration into predictive modelling is scarcely seen. Recent studies, such as those by Prasad [16], have emphasized the value of Bloom's taxonomy in predicting student performance.

In online resource recommendation aspect, these systems have drawn a lot of attention from researchers in recent years who are investigating various technologies to propose the best resources to users. The methodologies and tools that researchers used to create online resource recommendation systems are described in the section that follows.

In research done by Xin Wei et al in 2021 investigate the creation of a personalized extra learning resource recommendation system that enhances students' learning results by utilizing educational psychology theory and artificial intelligence technologies [17]. It discusses the difficulties in delivering relevant learning resources in online learning and suggests an algorithm for recommending learning resources based on LinUCB. The algorithm modifies the proportion of exploration and exploitation during recommendation using a unique exploration coefficient.

In research done by Hongtao Sun et al in 2022 proposed a learner-model-based collaborative filtering recommending approach for online study resources, which considers the traits and actions of learners when making recommendations in order to improve the accuracy of the content that is suggested [18]. In order to build learner models and to calculate the user similarity this paper proposed methods such as machine learning algorithms and learner modeling algorithms.

Another research done by Danyang Shen in 2020 proposed a neural network-based recommendation model that uses concept maps to represent knowledge points, learning behavior based on learning style scales, and Bloom's taxonomy to rank educational materials [19]. Then the score is calculated using the multilayer perceptron network based on the created embedding vectors. Studies conducted for this study's experiments showed that this recommendation model produced excellent results.

1.2. Research Gap

Referring to the research papers that we found, we learned that while there have been significant improvements and research efforts in the domain of educational technology and self-learning systems, there is still a significant gap in the integration of Bloom's Taxonomy, a core framework in education, into these systems. This crucial research gap is highlighted by our detailed analysis of existing literature.

Bloom's Taxonomy provides a structured hierarchy for categorizing educational objectives and goals based on cognitive levels, offering a profound insight into the cognitive development of learners. Despite its pivotal role in shaping curriculum design and assessment strategies, the existing literature falls short in exploring how this taxonomy can be effectively embedded into self-learning platforms.

According to our findings, there is a scarcity of individualized self-study solutions that seamlessly incorporate Bloom's Taxonomy. While some self-learning systems enable students to search for topics of interest, our approach takes a significant step forward by placing control and understanding in the hands of students. Our platform empowers students to upload their study materials, which then serve as the foundation for generating personalized mind maps, questions, and answers, all mapped to Bloom's Taxonomy. Moreover, our system integrates performance tracking and advanced search functionalities to provide a holistic learning experience. The literature review does not reveal substantial research efforts in this specific direction.

Thus, the research gap identified pertains to the limited exploration of how Bloom's Taxonomy can be practically and effectively integrated into self-learning systems to enhance personalized learning experiences. Future research endeavors in this area could focus on the development and evaluation of self-learning platforms that leverage Bloom's Taxonomy to categorize and tailor learning resources according to students' cognitive levels and learning objectives. Such research holds the potential to revolutionize educational technology by providing more context-aware and effective self-learning systems that cater to the diverse needs of students and promote deeper understanding and critical thinking.

1.3. Research Problem

It is crucial to create balanced and high-quality exams for undergraduates that cater to various cognitive levels. As a result, lecturers rely on Bloom's Taxonomy cognitive domain, a popular framework developed to assess students' intellectual abilities and skills [1]. Despite its widespread use, many students are not aware of Bloom's Taxonomy and how it affects their learning experience. This lack of understanding can lead to students missing important opportunities for growth and development in their academic pursuits. To address this issue, a personalized self-learning system that helps students understand and apply Bloom's Taxonomy in their own learning process can be implemented.

And by personalizing the system according to the student may help them learn from the type and level of the content they seem to fit in [2].

Many undergraduate students find it difficult to advance above the lower levels of the taxonomy (such Remembering and Understanding) and into the higher levels (such as Analyzing, Evaluating, and Creating). This means that instead of exhibiting a deeper comprehension of the subject and its ramifications, how they approach exams may be limited to mere memory and description.

Therefore, this presents a research problem as it raises questions about how the undergraduate students can identify and understand their cognitive level as classified by Bloom's Taxonomy, and how they can develop the skills they need to advance to the higher levels. By addressing this research problem, we can create educational environments that better support student learning and development, leading to more effective and efficient learning outcomes.

2. RESEARCH OBJECTIVES

2.1. Main Objective

The main objective of this research is to create a self-study platform to support undergraduate students. Unlike a typical self-study system, this platform distinguishes itself by incorporating the students' study materials and helping them evaluate their subject-specific performance through the application of Bloom's Taxonomy levels.

2.2. Sub Objectives

To achieve the main objective, it is further divided into four sub-objectives. They are,

1. Generate a comprehensive mind map for a given study material.
2. Generate a set of questions and answers from the study material and categorize them according to Bloom’s taxonomy.
3. Track and predict student performance in a specific subject.
4. Provide online extra study resources related to the uploaded study material.

Below chart shows user feedback on having a full-fledged self-learning system consisting above objectives.

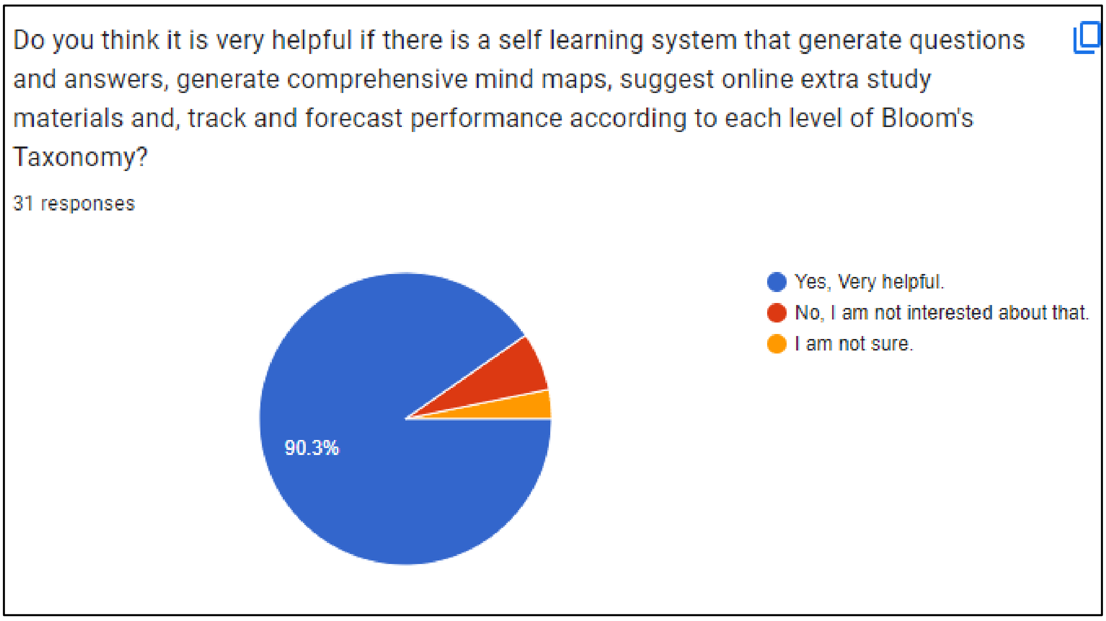


Figure 3: User Response on Having a Self-Learning System.

3. METHODOLOGY

3.1. Overall System

The overall methodology of BloomQuest system involves four components designed to enhance the learning experience and performance of students. The overall methodology for developing BloomQuest involves integrating these components seamlessly to provide a comprehensive and effective learning platform.

The first component focuses on developing a knowledge graph by utilizing text preprocessing, entity recognition, relationship extraction, graph construction, querying, filtering, and mapping techniques. This knowledge graph serves as a structured representation of the subject matter, enabling efficient retrieval of relevant resources and facilitating the generation of questions and answers. Finally, the extracted data is visualized providing a mind map for better understanding and analysis.

In the second component, quizzes are generated from the knowledge graphs created in the first component. These quizzes are categorized and mapped to various levels of Bloom's Taxonomy, ensuring that students are challenged at various cognitive levels.

The third component is centered on tracking and predicting student performance through the utilization of a pre-trained linear regression model. This model has been trained using a dataset that reflects students' achievements categorized by Bloom's Taxonomy levels. By incorporating techniques such as the exponentially weighted moving average (EWMA), the system adeptly captures the dynamic aspects of student performance, enabling it to furnish precise predictions regarding future academic accomplishments. Visualization serves as a pivotal element in conveying performance data to students, facilitating their comprehension of their progress and aiding in the identification of areas warranting additional focus.

The final component focuses on an online external resource recommendation. By utilizing the learner's personal study material as input, the system allows learners to select specific paragraphs that require further clarification, thereby addressing their individual learning needs. This personalized approach ensures that learners receive tailored recommendations that directly align with their areas of difficulty and interest.

By integrating these four components, BloomQuest offers a comprehensive learning platform that incorporates text processing, knowledge graph construction, performance tracking, performance predicting, visualization and resource recommendation. This holistic approach aims to enhance students' learning experiences, facilitate knowledge acquisition, and promote continuous improvement in their academic performance.

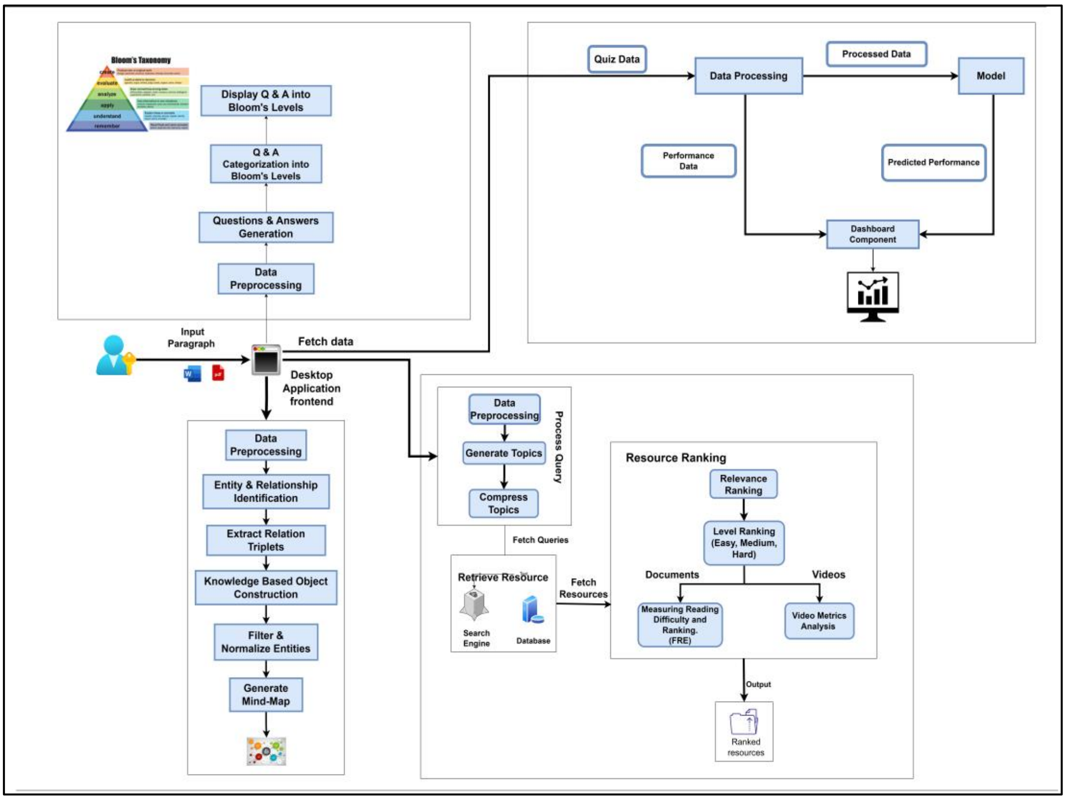


Figure 4: Overall System Diagram.

3.2. Methodology of Mind-Map Generating Component

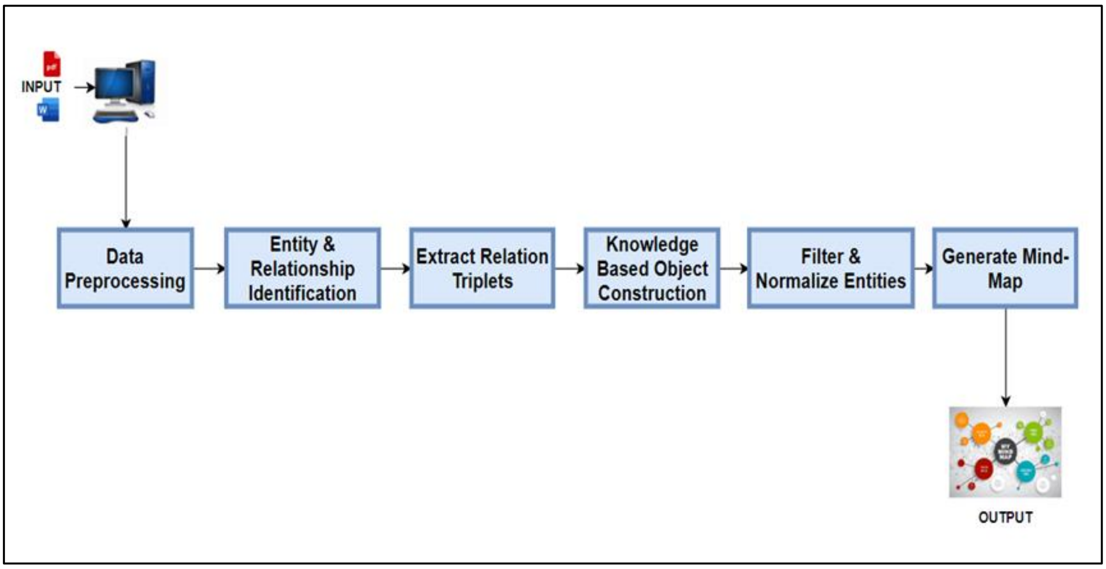


Figure 5: Mind-Map Generating Component Diagram.

Our thorough step-by-step process offers a disciplined framework for producing in-depth mind maps using study materials. This strategy effortlessly incorporates many steps, each of which plays a crucial part in the process. We guarantee that the research material is ready for analysis by starting with thorough data pretreatment and cleaning and organizing it. This stage ensures that the succeeding processes have access to accurate and consistent data by removing duplication and dealing with different document formats.

The entity and relation extraction stage begins after data preparation. throughout this crucial stage. This makes it possible for us to efficiently detect and categorize things and their connections within the research material. Our Mind maps are constructed using these things, and the relationships between them show how intricately connected they are.

Our approach moves on to the extraction of relation triplets after successfully extracting entities and relations. The fundamental framework of our knowledge base is composed of these triplets, which include a subject entity, a predicate (the relation), and an object entity. They help students better understand complicated interdependencies by encapsulating the underlying relationships that underpin the concepts in the study material.

The next step is to build these things methodically now that we have knowledge-based objects at our disposal. This guarantees that the learned information is arranged logically and saved in a structured fashion. This stage not only increases the effectiveness of our mind map creation but also provides students looking for a deeper comprehension of the subject with an invaluable resource.

We use entity filtering and normalization to further improve the coherence and clarity of our mind maps. Issues with synonyms, abbreviations, and variations in entity names are resolved in this stage. We improve the final mind maps' readability and efficacy by standardizing entity representations.

In the end, our process results in the creation of comprehensive mind maps. We create visually perceptible representations of the structured data and knowledge-based items by utilizing graph theory and visualization approaches. The outcome is a dynamic mind map that accurately replicates the study material's underlying hierarchical structure and complex linkages. Entities become nodes; relationships become edges.

Preprocess the Paragraph

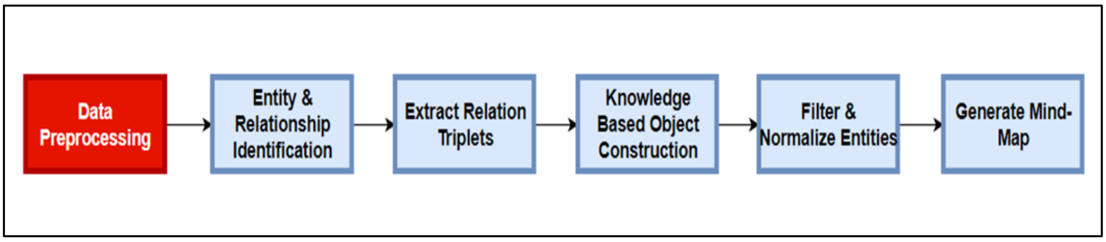


Figure 6: Preprocessing Step.

In this system, text preprocessing is a crucial first step that entails cleaning and converting unstructured text input into a more organized and manageable format. This procedure often entails steps like deleting superfluous letters, punctuation, and special symbols, changing the case of the text, and tokenizing the text to separate it into words or tokens. Text preprocessing delivers more accurate and useful results in many text-based applications. Text preprocessing helps enhance the quality of text data by making it more consistent and acceptable for later analysis.

Extract Entities and Relationships

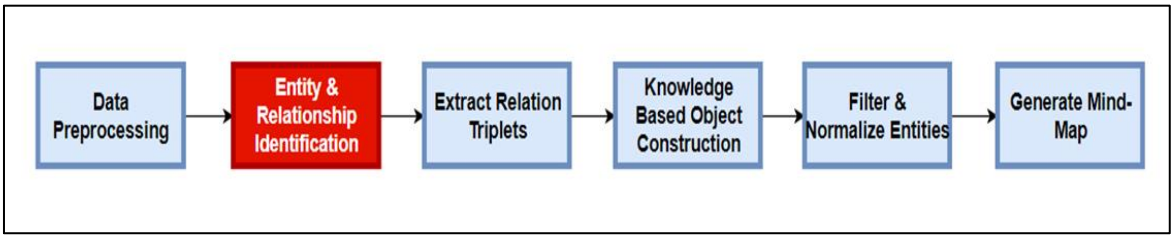


Figure 7: Extract Entities and Relationships Step.

The process of locating and classifying certain things or objects inside a text is known as entity extraction, often referred to as named entity recognition (NER). These entities might be actual things like people, companies, places, or dates, or they could be more domain-specific terminology like diseases, substances, or financial instruments. Entity extraction's main objective is to discover and correctly classify these entities inside the text, frequently by labeling them with standard terms like "person," "organization," or "location," or with bespoke terms made specifically for the intended application.

Finding and extracting significant links or linkages between entities stated in a text is referred to as relationship extraction or relation extraction. Recognizing how things relate to one another and if they play certain roles or have connections that may be captured by predetermined relationships is necessary to achieve this. For instance, connection extraction in a news item would entail noting that "Apple Inc." (entity) purchased "Tesla" (entity) and designating their relationship as an "acquisition." To properly identify the nature of the relationship, relationship extraction often requires knowledge about the context of the entities and the surrounding language. Information retrieval, knowledge graph construction, and the creation of organized databases from unstructured text all benefit from its use.

In tasks involving natural language processing, named entity recognition (NER) and relationship classification (RC) are often used algorithms for entity and relationship extraction. The conventional pipeline, which employs NER before moving on to RC, might, however, generate mistakes that spread throughout the procedure. This sequential method can have its limitations, particularly when tackling connections in text data that are intricate or subtle. A preset set of relation types is another restriction placed on RC, which may not fully account for all possible links between entities.

Recent developments in natural language processing have proposed creative end-to-end techniques that try to manage both problems concurrently in order to solve the difficulties presented by the sequential application of Named Entity Recognition (NER) followed by Relation Classification (RC). Relation Extraction (RE) is the term used most frequently to describe this integrated process. In the framework of this post, we will go into the use of the amazing REBEL end-to-end model, which was created by BabelScape. Researchers and practitioners can accelerate the entity and connection extraction process, reducing the possibility of error propagation while increasing the ability to capture a greater variety of relation types, by implementing REBEL and other cutting-edge models.

In order to translate a phrase with entities and implicit relations into a sequence of triplets that explicitly refer to those relations, BabelScape trained the text-to-text model REBEL by fine-tuning BART. More than two hundred distinct relation kinds were used to train it. Using entities and relations discovered in Wikipedia abstracts and Wikidata, the authors constructed a bespoke dataset for REBEL pre-training and filtered it using a RoBERTa Natural Language Inference model (similar to this model). To learn more about how the dataset was created, check the paper. On a variety of benchmarks for Relation Extraction and Relation Classification, the model performs well. And for this step I am using this pre-trained model to extract entities and relations in an accurate way.

Extract Relation Triplets From The Text That Has Been Processed By REBEL

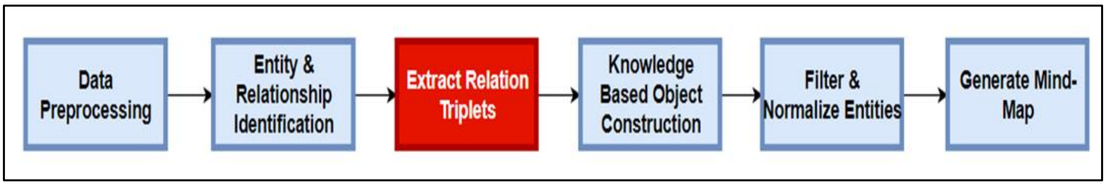


Figure 8: Extract Relation Triplets From the Text That Has Been Processed by REBEL.

The creation of a custom function that can parse the structured strings produced by the REBEL model and convert them into relation triplets is a vital next step in the procedure. The subject, relation type, and object of each extracted piece of knowledge are all contained inside these triplets, which operate as the basic units of extracted knowledge. The function, which facilitates this translation, accounts for the addition of additional tokens during the model training phase, such as placeholders like "<triplet>," "<subj>," and "<obj> “. The borders and functions of entities and relationships inside the produced strings are clearly defined by these tokens.

Since each connection is represented as a dictionary, the function analyzes these strings to create a list of relations. These dictionaries have three fundamental words: "head" to indicate the subject (for example, "Fabio"), "type" to indicate the sort of relation (for example, "lives in"), and "tail" to indicate the object (for example, "Italy"). By using this function, we fill the gap between the model's output and a structured, useable representation of the knowledge that was extracted, making it more available and usable for later applications like knowledge graph generation and information retrieval.

Pass Extracted Triplets into a Knowledge-Based Object

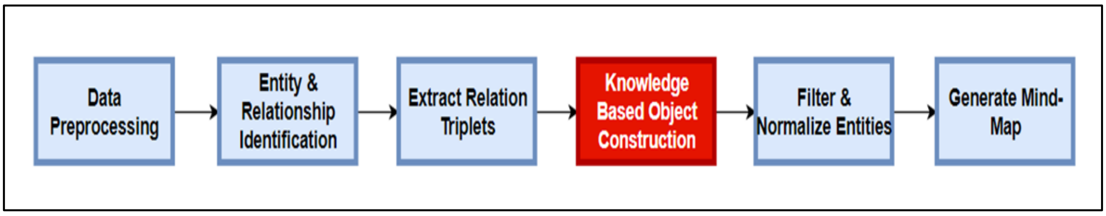


Figure 9: Pass Extracted Triplets into a Knowledge-Based Object.

Our knowledge base creation approach includes a filtering stage as well, to improve accuracy and coherence. Entity linking, which includes comparing the retrieved entities with Wikipedia pages, is a successful filtering strategy. This stage involves checking to see if terms like "Cristiano Ronaldo" and "Cristiano" have a similar Wikipedia page. In the event that such a relationship is found, the entities are normalized to the page's title, combining them into a single representation. It is vital to note that this approach is predicated on the notion that Wikipedia consistently has accurate information on these entities as a result of user contributions. A more accurate and focused depiction of entities is made possible by temporarily excluding from the knowledge base any entities without associated Wikipedia entries.

A significant element of our knowledge base design is the method "are_relations_equal," which provides a way to determine whether two relations are equivalent based on the terms "head," "type," and "tail." This technique is crucial for knowledge base management because it enables us to establish whether two relations belong to the same information or to different facets of the same notion. We construct a related equality criteria by comparing these qualities, improving the accuracy and effectiveness of operations inside the knowledge base. When working with huge datasets or updating the knowledge base, this functionality is especially useful for ensuring that redundant or repeated relations are properly detected and maintained. Overall, "are_relations_equal" is a crucial technique that improves precision and coherence.

Filter and Normalize the Entities

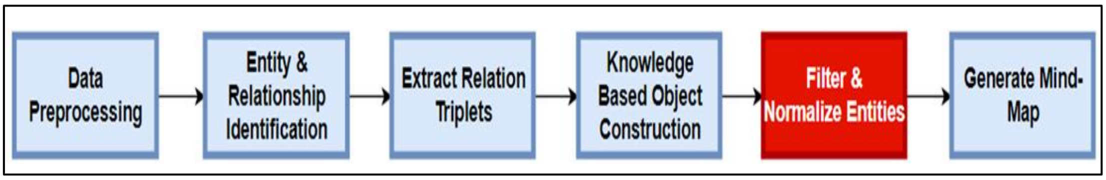


Figure 10: Filter and Normalize the Entities.

Our knowledge base creation approach includes a filtering stage as well, to improve accuracy and coherence. Entity linking, which includes comparing the retrieved entities with Wikipedia pages, is a successful filtering strategy. This stage involves checking to see if terms like "Cristiano Ronaldo" and "Cristiano" have a similar Wikipedia page. In the event that such a relationship is found, the entities are normalized to the page's title, combining them into a single representation. It is vital to note that this approach is predicated on the notion that Wikipedia consistently has accurate information on these entities as a result of user contributions. A more accurate and focused depiction of entities is made possible by temporarily excluding from the knowledge base any entities without associated Wikipedia entries.

A significant element of our knowledge base design is the method "are_relations_equal," which provides a way to determine whether two relations are equivalent based on the terms "head," "type," and "tail." This technique is crucial for knowledge base management because it enables us to establish whether two relations belong to the same information or to different facets of the same notion. We construct a related equality criteria by comparing these qualities, improving the accuracy and effectiveness of operations inside the knowledge base. When working with huge datasets or updating the knowledge base, this functionality is especially useful for ensuring that redundant or repeated relations are properly detected and maintained. Overall, "are_relations_equal" is a crucial technique that improves precision and coherence.

Visualize the Mind-Map

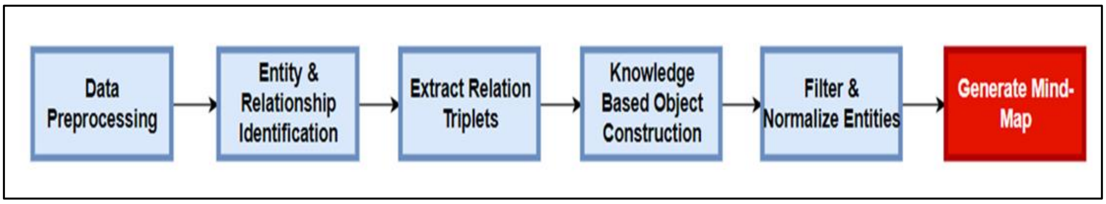


Figure 11: Visualize the Mind-Map.

A crucial step in turning abstract data into a clear and visual representation is the creation of a graph visualization using the relations held in a knowledge base (KB). The Pyvis package, a potent tool for creating interactive network graphs, is used in this visualization stage. Using Pyvis, we can convert the intricate network of relations into an understandable and aesthetically pleasing graph structure. A visual representation of the underlying knowledge structure is created by turning each relation into a node and representing the connections between them as edges.

Once the graph has been created, it allows users to explore the links and interconnections within the knowledge base dynamically while also being visually instructive and engaging. The generated graph is stored as an HTML file for easy distribution and additional analysis. This ensures accessibility and shareability. This visual representation not only makes the knowledge base easier to grasp, but it also provides researchers, analysts, and learners with a thorough understanding of the links between the many entities in the data.

3.3. Methodology of Questions and Answers Generating Component

The methodology employed in this research project aimed at generating sets of questions and answers aligned with Bloom's taxonomy levels encompasses a systematic and iterative approach that ensures the development of high-quality educational assessment materials. The following steps outline the key components of the methodology.

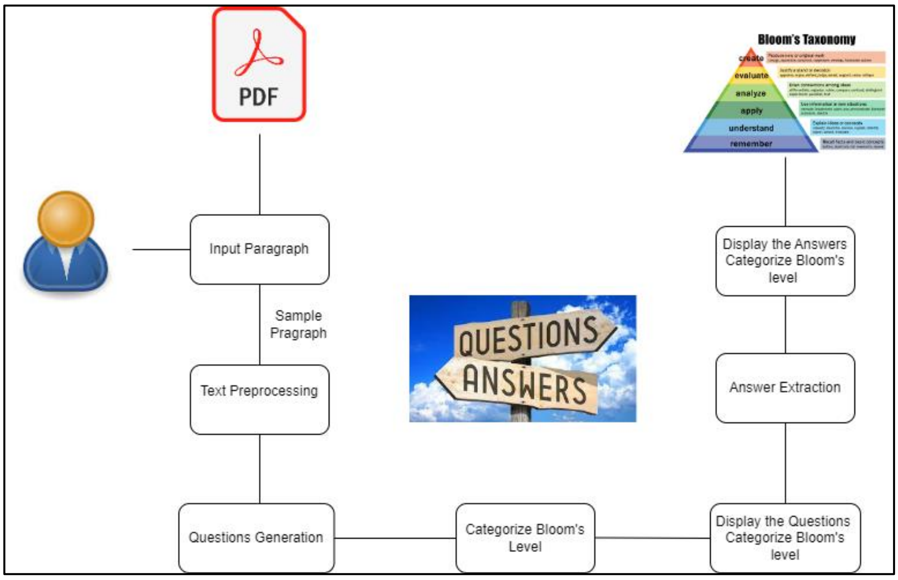


Figure 12: Questions and Answers Generating Component Diagram.

Text Preprocessing

Text preprocessing is a crucial step in natural language processing (NLP) tasks, including question generation. It involves cleaning and transforming the raw text data into a format that is suitable for analysis and modeling. Here are the typical steps involved in text preprocessing Convert all text to lowercase to ensure consistent handling of text data. This helps avoid issues where the same word is treated differently due to case variations.

Break the text into individual words or tokens. Tokenization helps in analyzing the text at a granular level and is often the first step in various NLP tasks.

Remove common stop words (e.g., "the," "is," "and") from the text. Stop words are often filtered out because they carry little meaningful information and can reduce the size of the data. Remove punctuation marks, special characters, and symbols from the text. This simplifies the text and avoids interference with tokenization and analysis. Decide whether to keep or remove numbers from the text. In some cases, numbers may carry valuable information (e.g., dates or measurements), while in others, they may be irrelevant. Apply stemming or lemmatization to reduce words to their base or root forms. Stemming involves removing suffixes, while lemmatization returns words to their dictionary form. This step helps in standardizing word variations.

Expand contractions like "don't" to "do not" to ensure consistency in text representation. Remove extra spaces and leading/trailing whitespace from the text to maintain clean and consistent formatting. Consider applying spell checking and correction to fix common spelling errors in the text.

Depending on the specific requirements of your NLP task, you may need to include custom preprocessing steps. For example, if dealing with domain-specific jargon, you might want to perform domain-specific term normalization.

Text Classification

Text classification, also known as text categorization, is a fundamental natural language processing (NLP) task that involves assigning predefined categories or labels to text documents or pieces of text. The goal is to automatically classify textual data into specific categories based on its content. Text classification finds applications in various domains, including spam detection, sentiment analysis, topic categorization, and content recommendation. Here are key aspects of text classification Ensure that each text document in the dataset is labeled with one or more categories or classes.

Labels serve as the ground truth for training and evaluating the classification model.

Feature Extraction Transform the preprocessed text data into feature vectors suitable for machine learning models.

Common methods include using TF-IDF, word embeddings (e.g., Word2Vec or GloVe), or other text representation techniques.

Generate Questions

When generating questions aligned with Bloom's taxonomy levels, the aim is to ensure that the questions assess the intended cognitive skills and knowledge. For example, if the objective is to assess a learner's ability to analyze a historical event, the generated question should require analytical thinking and fall under the "Analysis" level. Similarly, if the goal is to evaluate comprehension of a scientific concept, the generated question should align with the "Comprehension" level.

These questions represent a variety of cognitive levels from Bloom's taxonomy, including knowledge, comprehension, application, analysis, synthesis, and evaluation. Depending on your educational objectives and the desired level of cognitive engagement, you can use these questions as templates to create assessments and learning materials aligned with Bloom's taxonomy. This functionality involves using the knowledge graph to generate a set of questions related to the research topic. The questions can be designed to target several Bloom's Taxonomy levels, including remembering, understanding, applying, analyzing, evaluating, or creating.

Question generation systems often use NLP techniques and machine learning models to analyze text and generate questions that match the desired Bloom's taxonomy level. These systems take into account the context and learning objectives to create relevant and appropriately challenging questions for educational assessments and other applications.

Generate Answers

The answer generation, in the context of Bloom's taxonomy, does not typically have its own distinct taxonomy level. Instead, answer generation is a complementary process to question generation and aligns with the cognitive levels assessed by the questions themselves. In other words, the level of the answer corresponds to Bloom’s taxonomy level of the question being asked. This functionality involves using the knowledge graph to create a set of possible answers for each question created in the preceding phase. The answers can be based on existing knowledge or data, or they can be generated using machine learning or other techniques.

For example:

If a question generated is at the "Knowledge" level (e.g., "What is the capital of France?"), the expected answer would be a piece of factual knowledge (e.g., "The capital of France is Paris.").

If the question is at the "Comprehension" level (e.g., "Can you explain the main idea of the passage?"), the answer would require an understanding and explanation of the main idea of the text.

In this way, answer generation is intrinsically tied to the cognitive complexity assessed by the question. Therefore, the Bloom's taxonomy level of the answer directly corresponds to the Bloom's taxonomy level of the question that prompted.

Categorize to Bloom's Taxonomy Levels

Question	Answer	Bloom's Level
What is the capital of France?	The capital of France is Paris.	Knowledge
Can you summarize the main plot of the novel "To Kill a Mockingbird"?	The novel "To Kill a Mockingbird" is set in a small Southern town and follows the Finch family's experiences during a racially charged trial.	Comprehension
Solve the equation: 2x + 5 = 15.	x = 5	Application
Analyze the factors contributing to climate change.	Climate change is influenced by greenhouse gas emissions, deforestation, and industrial pollution.	Analysis
Create a marketing plan for a new product launch.	A marketing plan should include market research, target audience analysis, advertising strategies, and budget allocation.	Synthesis
Evaluate the effectiveness of the government's economic policies during the past decade.	The government's economic policies have had both positive and negative impacts on employment rates and economic growth.	Evaluation

Figure 13: Categorize Question & Answers According to Bloom's Taxonomy Levels.

In this table, each question is followed by its corresponding answer, and both are categorized according to the Bloom's taxonomy level that the question assesses. This illustrates how questions can be designed to align with specific cognitive levels, and the answers reflect the expected depth of knowledge or understanding at each levels.

Categorizing tasks or questions according to Bloom's taxonomy levels involves assessing the cognitive complexity and skills required to complete those tasks. Here is a categorization of tasks based on Bloom's taxonomy levels: Each of the tasks listed above corresponds to a specific Bloom's taxonomy level. Categorizing tasks in this way helps educators and instructional designers create assessments and learning materials that align with desired learning objectives and cognitive skills. This functionality involves categorizing the questions and answers generated in the previous steps according to Bloom's Taxonomy. This can help ensure that the questions target various levels of cognitive complexity and can be used to design assessments that measure diverse types of learning outcomes.

Implementation.

This Generate set of Questions and Answers according to Bloom's Taxonomy level system uses many Data Science approaches when building the solution. The System leverages advanced machine learning techniques, setting it apart from conventional systems in terms of development and efficiency. The development and implementation part will be described according to the component diagram that depicts in Figure 5. In the context of generating sets of questions and answers according to Bloom's taxonomy levels, this phase involves a series of steps to verify the system's functionality, accuracy, and usability. Here is an overview of the testing and implementation process.

Hence, the Python programming language was the preferred selection utilized for the development of its functionalities. This is primarily due to the immense flexibility and adaptability possessed by the Python programming language. In the Table 2 shows the summery of key development tools python libraries that utilized when developing the system.

Table 1: Tools and Libraries of Q & A Component.

Tools	• Anaconda • Visual Studio Code
Python libraries	• Pipeline • Spacy • Nltk – corpus, tokenize. • Transforms • Wordnet • Flask

1. Text preprocessing the input paragraph:

```
# Define the Sample Paragraph
study_material = """
The American Revolution was a political and social movement that occurred between 1765 and 1783. It resulted in the indepen
The American Revolution is often studied for its ideological origins, military campaigns, and the establishment of a new na
Key figures in the American Revolution include George Washington, Thomas Jefferson, Benjamin Franklin, and John Adams. The
The American Revolution is often studied for its ideological origins, military campaigns, and the establishment of a new na
"""
```

Figure 14: Input Paragraph.

Text Preprocessing: Tokenize the input paragraph into sentences and words. Remove any stop words and punctuation. Perform stemming or lemmatization to reduce words to their base forms. used NLP rule-based approach.

2. Text classification:

```
 Load the question generation pipeline
uestion_generator = pipeline("question-generation", model="bert-base-uncased")

 Sample text
ext = "Photosynthesis is a process used by plants to convert light energy into chemical energy."
```

Figure 15: Divided to Text.

Use a pre-trained NLP model or train our own text classifier to categorize each sentence in the input paragraph into one of the Bloom's taxonomy levels (e.g., Knowledge, Comprehension, Application, Analysis, Synthesis, Evaluation).Using the Tokenization library. Pipeline python library then output is that.

3. Question generation:

For each sentence categorized at a specific Bloom's level, generate questions that correspond to that level. The questions should align with the cognitive skills associated with that level (e.g., "What is X?" for Knowledge, "How does X relate to Y?" for Comprehension, etc.). The use a question generation r rule-based techniques to generate questions and Transformer python Library used and Encoder Architecture.

Quiz -1 Questions
Human cells typically have how many copies of each gene?
Which of the following Elite Four members from the 6th Generation of Pokémon was a member of Team Flare?
In the Mass Effect trilogy, who is the main protagonist?
Which singer is portrayed by Bruce Campbell in the 2002 film 'Bubba Ho-Tep'?
In Portal, the Companion Cube's ARE sentient.

Figure 16: Generate Questions.

4. Answer extraction:

For each generated question, extract relevant answers from the input paragraph. use Named Entity Recognition (NER) or information retrieval techniques to find answers using Decoder Architecture and Transforms Python Library.

Your Answers	Correct Answers
2	2
Drasna	Malva
Mordin	Shepard
Buddy Holly	Elvis Presley
True	True

Figure 17: Generate Answers.

5. Display questions, answers, and bloom's levels:

Organize the generated questions, answers, and their corresponding Bloom's taxonomy levels. Display the results in a user-friendly format. Using the NLTK python library and spacy.

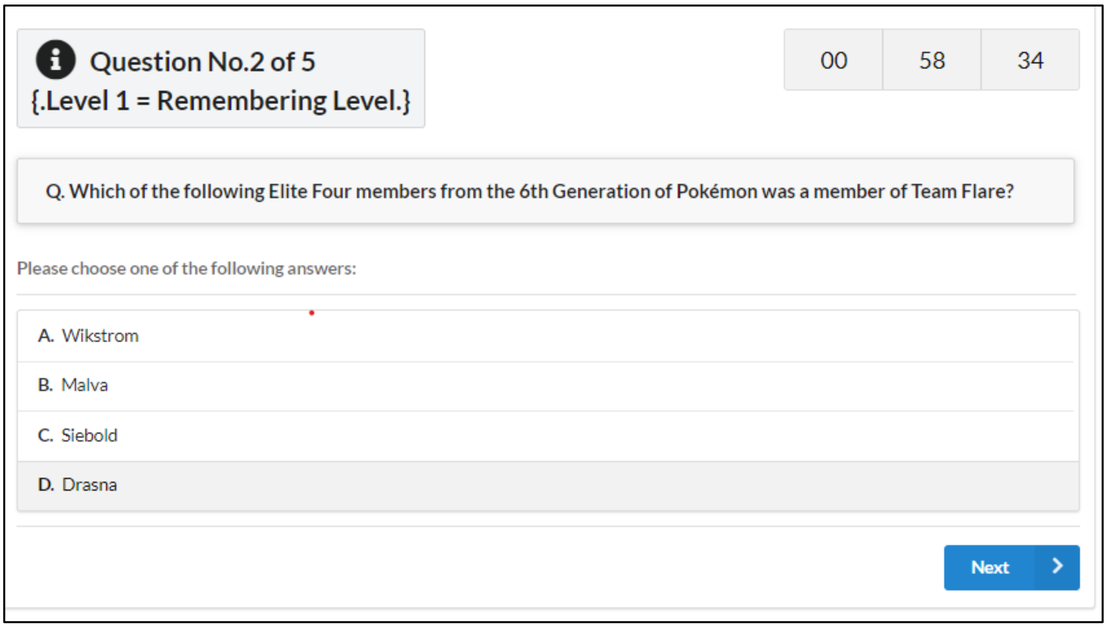


Figure 18: Categorized Questions.

3.4. Methodology of Performance Tracking and Predicting Component

3.4.1. Performance Predicting Model.

Data collection.

The study was conducted with students enrolled in the "Database Technologies" module, a comprehensive 13-week, 4-credit course. Under the guidance of my supervisor, I designed three quizzes targeting specific segments of the curriculum: ER Diagrams, ER Model to Relational Model Mapping, and Normalization.

Central to my methodology was aligning the quiz questions and mid-term exam to Bloom's Taxonomy, a widely accepted framework for categorizing educational objectives. Instead of the traditional six levels in Bloom's Taxonomy, I opted for a more streamlined approach.

L1: Remember and Understand

L2: Apply

L3: Analyze, Create, and Evaluate

The performance data from these quizzes and the mid-term examination from approximately 140 students were gathered over two months. This meticulous data collection afforded a comprehensive dataset for subsequent analysis. To ensure a holistic view of student performance, the final examination results were also gathered upon the completion of the module.

Data preparation.

After the meticulous process of data collection, the subsequent essential step was the organization and refinement of this raw data. Once the necessary preprocessing was conducted, a final sample size of 124 students was attained, from which two separate datasets were derived for two distinct experiments.

The first dataset was unambiguous, encapsulating the final scores students received in each of the quizzes, the mid-term, and the final examination.

Table 2: Overview of the Dataset 1.

Student ID	Quiz 1	Quiz 2	Quiz 3	Quiz 4	Final Exam
1	46	34	64	47.5	65
2	40	46	44	32.5	60.5
3	62	36	60	45	49
4	76	40	38	65	88
5	56	78	72	70	66

The second data set took a more analytical approach. Here, marks corresponding to each level of Bloom's Taxonomy from the quizzes and the mid-term exam were aggregated. This granular breakdown was aimed at furnishing a deeper understanding of students' competencies across various cognitive levels. An instrumental component of this dataset was the computation of the Exponentially Weighted Moving Average (EWMA) for each taxonomy level. The adopted formula for this metric was:

$$\mathbf{EWMA(t)=\alpha\times x(t)+(1-\alpha)\times EWMA(t-1)}$$

Where:

EWMA(t): Represents the Exponentially Weighted Moving Average at time t.

α: The weighting factor, ranging between 0 and 1, which designates the degree of impact the recent observations possess. An α value nearing 1 accentuates the significance of newer observations.

x(t): Denotes the observed value at time t.

EWMA(t-1): Is the Exponentially Weighted Moving Average from the preceding period at time t−1.

The weighting factor, α is instrumental, and its value was determined using the equation:

$$\boldsymbol{\alpha = 2/(N+1)}$$

Where N stands for the total number of quizzes or assessments undertaken. This equation was adapted from reference [20].

Table 3: Overview of the Dataset 2.

Student ID	EWMA L1	EWMA L2	EWMA L3
1	23.352	11.975	13.865
2	17.304	9.305	12.215
3	18.576	18.378	14.022
4	27.696	16.114	13.486
5	32.64	14.444	21.524

Data analysis.

To identify the impact of Bloom's Taxonomy and determine the interrelations between the various attributes in both datasets in relation to the final exam results, a Pearson correlation coefficient analysis was performed.

Figures 19, 20, 21, and 22 provided below illustrate the scatterplots for dataset 1, which contains the final marks for each assessment.

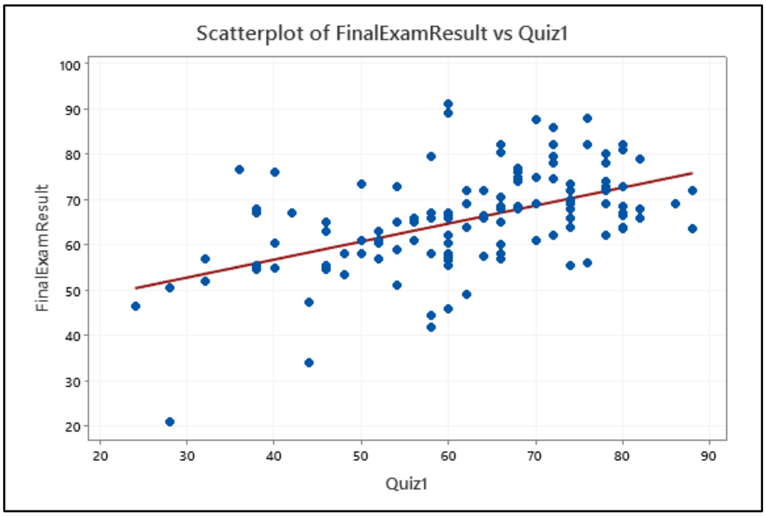


Figure 19: Scatterplot of Final Exam vs Quiz 1.

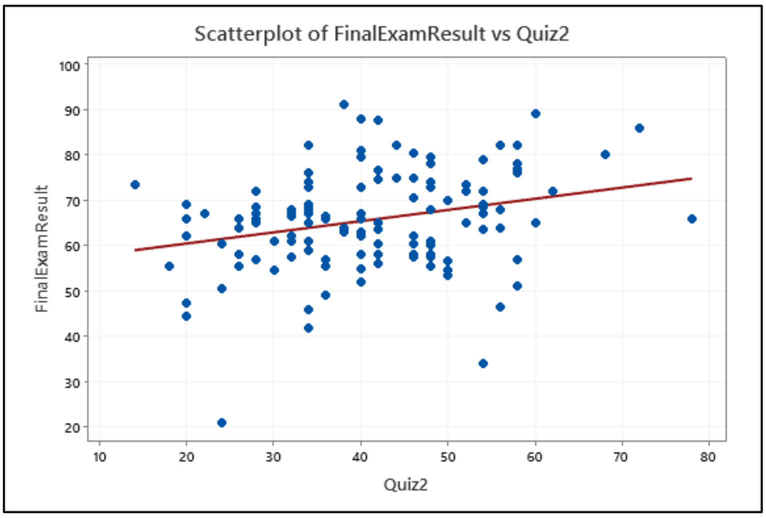


Figure 20: Scatterplot of Final Exam vs Quiz 2.

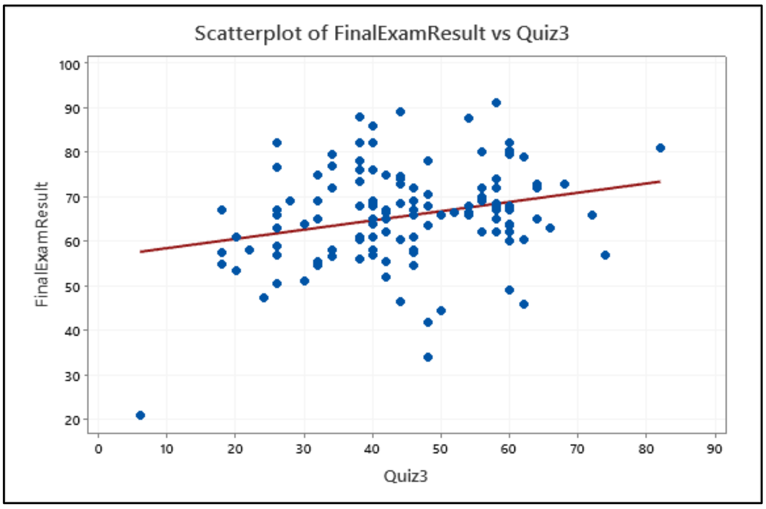


Figure 21: Scatterplot of Final Exam vs Quiz 3.

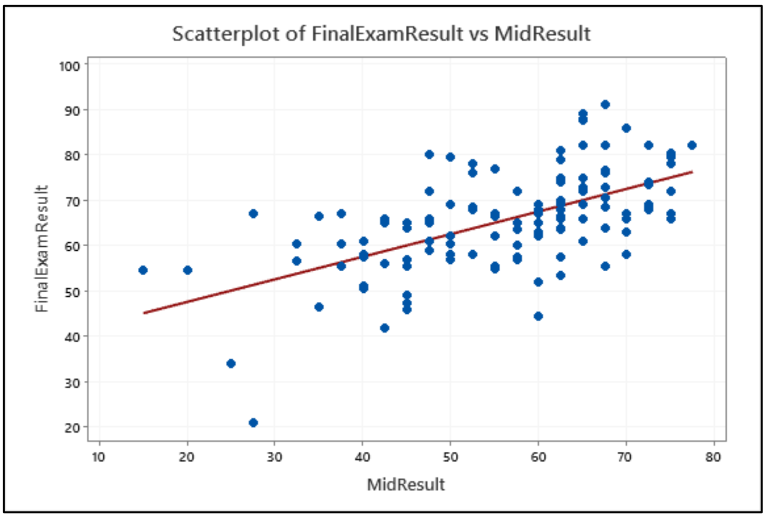


Figure 22: Scatterplot of Final Exam vs Mid-Exam.

Subsequent to the scatterplots, a Pearson correlation analysis was conducted on dataset 1.

Table 4: Pearson Correlation Table of Dataset 1.

Dependent Variable	Independent Variable	Sample Size	Correlation	P-Value
Final Exam Result	Quiz 1	124	0.509	0.000
Final Exam Result	Quiz 2	124	0.275	0.002
Final Exam Result	Quiz 3	124	0.256	0.004
Final Exam Result	Mid Exam	124	0.591	0.000

According to table 4, a correlation was evident between the scores of all quizzes, mid-exam results, and the final exam result. This suggests that performance in these assessments can potentially foreshadow the results of the final exam.

The correlation between scores in Quiz 2 and Quiz 3 with the final exam result was found to be moderate. This implies that while there is a connection between these quizzes' scores and the final exam outcome, it is not as pronounced as with other assessments.

In the second dataset, which incorporated the EWMA values of each Bloom's level across all quizzes and the mid-term exam, distinct patterns emerged.

Below are the scatterplots of dataset 2, which consist of the EWMA values for L1, L2, L3, and the final exam mark.

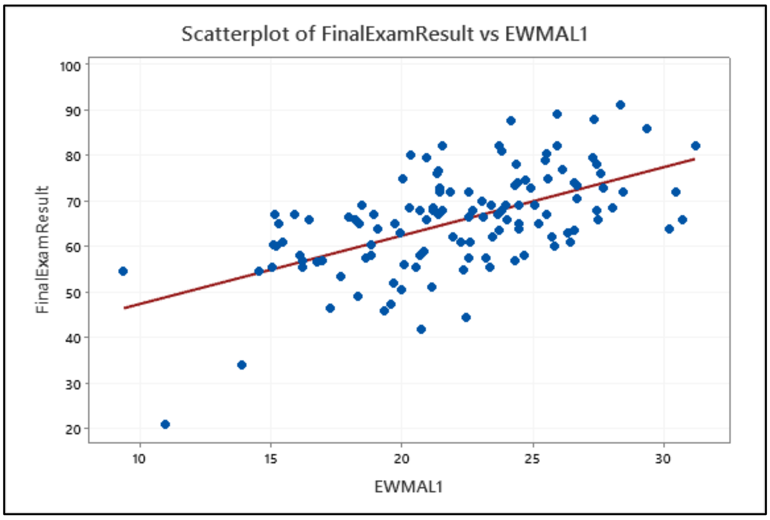


Figure 23: Scatterplot of Final Exam vs Bloom's Level L1.

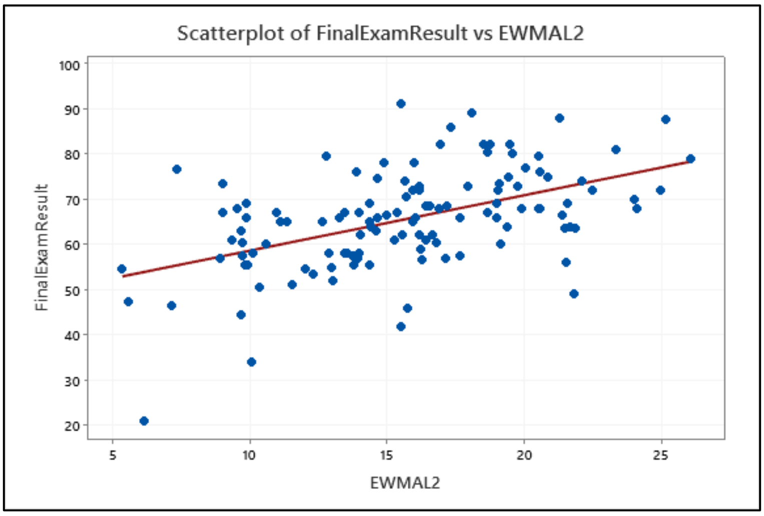


Figure 24: Scatterplot of Final Exam vs Bloom's Level L2.

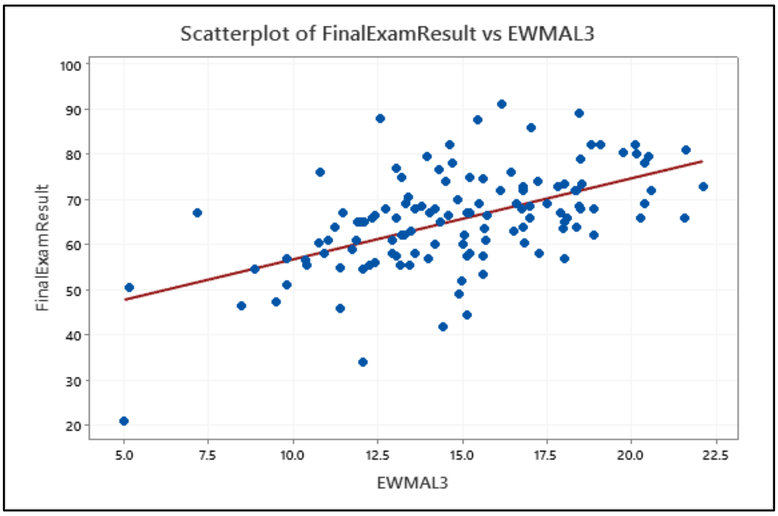


Figure 25: Scatterplot of Final Exam vs Bloom's Level L3.

Similar to dataset 1, a Pearson correlation analysis was conducted on dataset 2.

Table 5: Pearson Correlation Table of Dataset 2.

Dependent Variable	Independent Variable	Sample Size	Correlation	P-Value
Final Exam Result	EWMA L1	124	0.576	0.000
Final Exam Result	EWMA L2	124	0.507	0.000
Final Exam Result	EWMA L3	124	0.547	0.004

Table 5 shows that there was a significant correlation [21] between Bloom's Taxonomy levels L1, L2, and L3 with the final exam results. This observation underscores the significance of using Bloom's Taxonomy as a guiding framework in academic performance prediction.

Given these compelling findings, the decision was made to utilize the second dataset as the foundation for subsequent model training.

Model selection.

Upon recognizing the linear relationship between the attributes (EWMA L1, EWMA L2, EWMA L3) and the final exam results, linear regression was deemed the most suitable model for this study. The primary reasons to select this were:

1. Simplicity and Interpretability: Linear regression, being a foundational statistical method, offers clear interpretability, making it easier to understand and explain the relationship between independent and dependent variables.
2. Predictive Efficiency: When relationships between variables are linear, this model provides efficient and accurate predictions.
3. Ease of Implementation: Linear regression models are straightforward to implement and do not require extensive computational resources.

Table 6: Linear Regression Model Summary.

R-sq	**R-sq(adj)**	**R-sq(pred)**
55.89%	54.07%	50.53%

Given these considerations and outcomes, **linear regression** was adopted as the primary model for this research. The derived regression equation from the study is:

***Current Performance* = 25.99 + 0.879 EWMAL1 + 0.627 EWMAL2 + 0.694 EWMAL3**

3.4.2. Dashboard Implementation

In the digital age, data without visualization can often seem like a story without a narrative. To bridge this gap and provide students with a more intuitive, immediate, and impactful understanding of their performance, a dashboard was developed.

The dashboard, serving as the front-end interface for students, showcases their performance metrics derived from the linear regression model. More than just a collection of numbers, this platform transforms raw data into insightful graphs, charts, and other visual aids. The core features and functionalities of the dashboard are:

- Dynamic Data Presentation: As students continue their academic journey, the dashboard updates in real-time, ensuring that learners always have access to the most current data regarding their performance.
- Taxonomy-Level Breakdown: Drawing upon the significance of Bloom's Taxonomy in the study, the dashboard provides a granular breakdown of performance at each cognitive level, enabling students to pinpoint areas of strength and opportunities for improvement.
- Predictive Analytics Integration: Beyond just presenting past performance, the dashboard integrates the predictive capabilities of the linear regression model. This gives students an understanding of their current academic performance, allowing for proactive academic strategies.
- User-Friendly Interface: Recognizing that not all students may be tech-savvy, special emphasis was placed on making the dashboard intuitive. Clear labels, coherent color schemes, and interactive elements ensure ease of use.

- Motivational Insights: One of the primary goals of the dashboard is not just to inform, but to inspire. To this end, it includes features that highlight improvement areas, celebrate achievements.

To develop the dashboard modern and efficient technologies were used.

Frontend Framework: React.js was employed as the primary framework for frontend development. Known for its efficiency and flexibility, React.js provides an interactive user interface with a smooth user experience.

Data Visualization Libraries: To visualize the data in a comprehensive and aesthetic manner, popular chart libraries like 'React Charts' and 'Recharts' were incorporated. These libraries facilitated the representation of data in various formats, including line charts, bar graphs, and gauge charts, offering students a holistic view of their performance metrics.

A significant design principle adopted was the modularization of dashboard components. This approach ensures that each component can be individually developed, evaluated, and maintained, leading to more straightforward debugging, easier updates, and enhanced scalability.

Also, data security and privacy are an utmost crucial factor when it comes to students' data. To ensure this, a rigorous authorization mechanism was set up. Only authorized users, primarily the students themselves, are granted access to view their personal performance details. This mechanism shields students' data from unauthorized access and potential breaches, maintaining data integrity and privacy.

The choice of React.js, known for its component-based architecture, ensures the dashboard remains maintainable. By having distinct components for various dashboard elements, any required changes or updates can be smoothly implemented without disrupting the entire system.

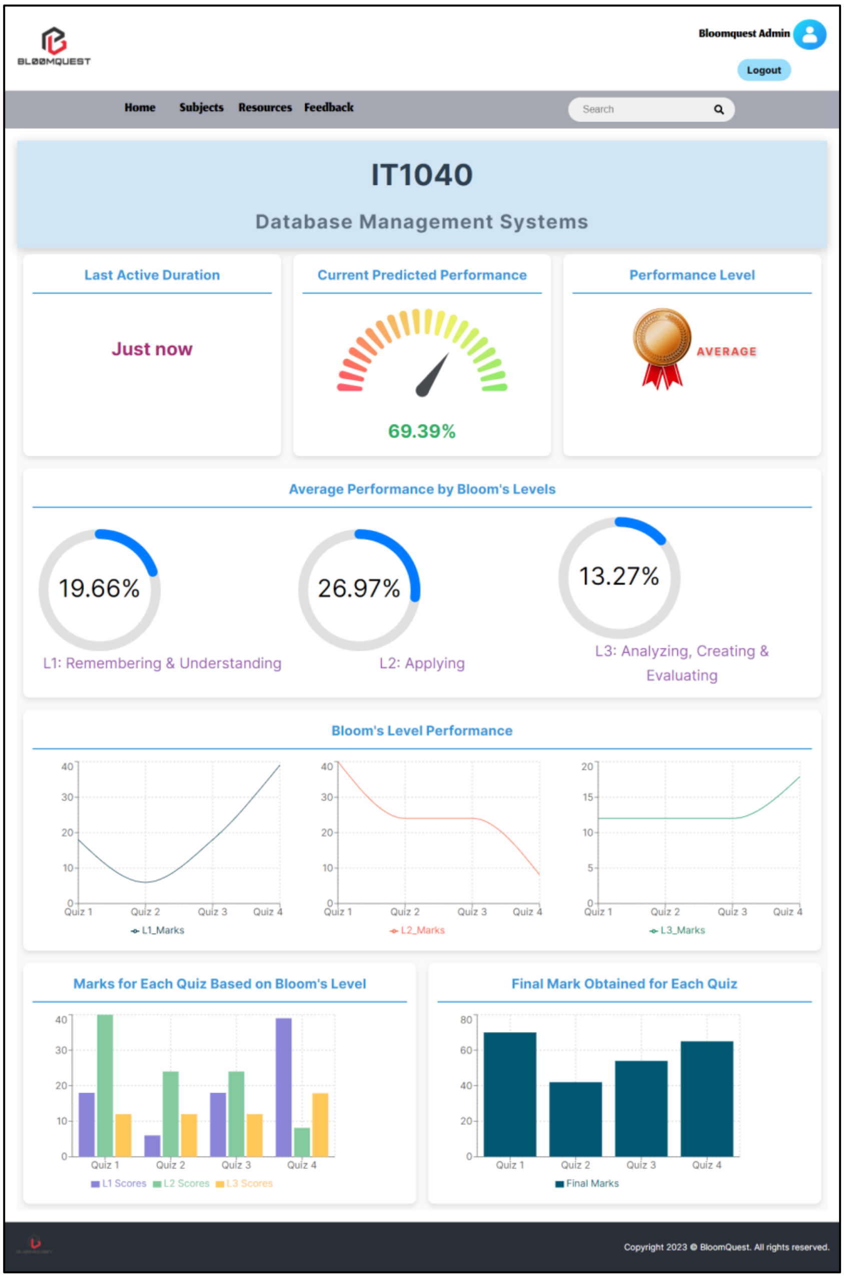


Figure 26: Dashboard Design.

The successful deployment of the dashboard is a testament to the effective amalgamation of cutting-edge technology and user-centric design principles, ensuring students receive a secure, insightful, and seamless interface to track their academic journey.

Component overview.

Once the performance tracking and prediction component is developed, it can be integrated into BloomQuest's quiz component. Here, students can undertake quizzes with questions aligned to the L1, L2, and L3 levels of Bloom's taxonomy. Data gathered from this component will be utilized for tracking and predicting student performance.

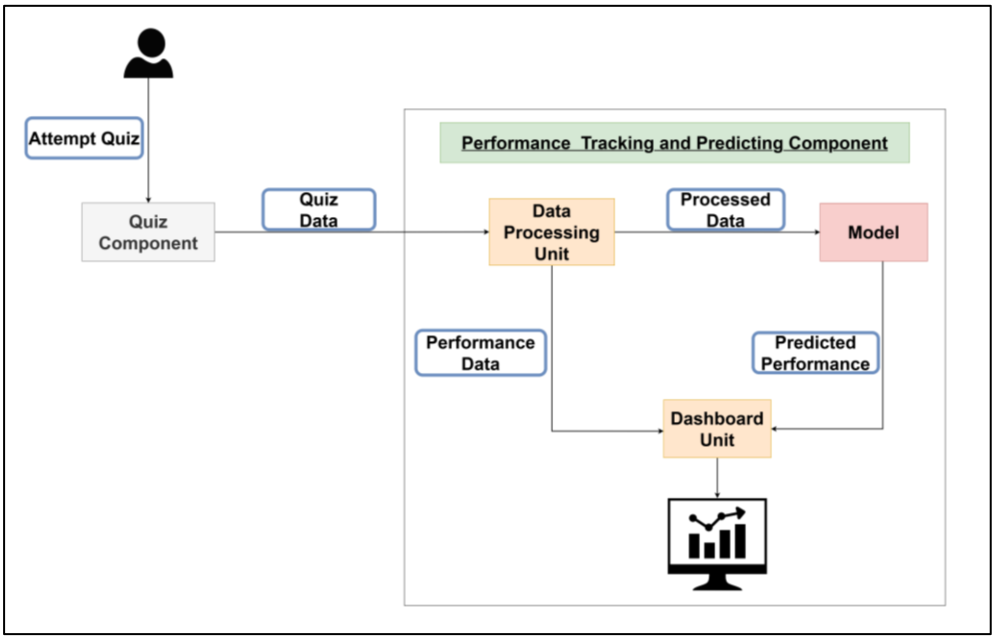


Figure 27: Performance Tracking and Predicting Component Diagram.

3.5. Methodology of Online Resources Recommending Component

The external resource recommendation system aims to assist learners in enhancing their understanding of study materials by leveraging online resources. Figure 3 depicts the overall system architecture. The system utilizes the learner's personal study material as input, wherein the learner selects a paragraph from their reading material and uploads it into the system. The 'Process Query' section is responsible for processing the uploaded query passage and generating topics from it. Then, 'Retrieve Resource' section will retrieve resources that are relevant to processed query. At last , 'Resource Ranking' section ranks the best resources and provides to the students.

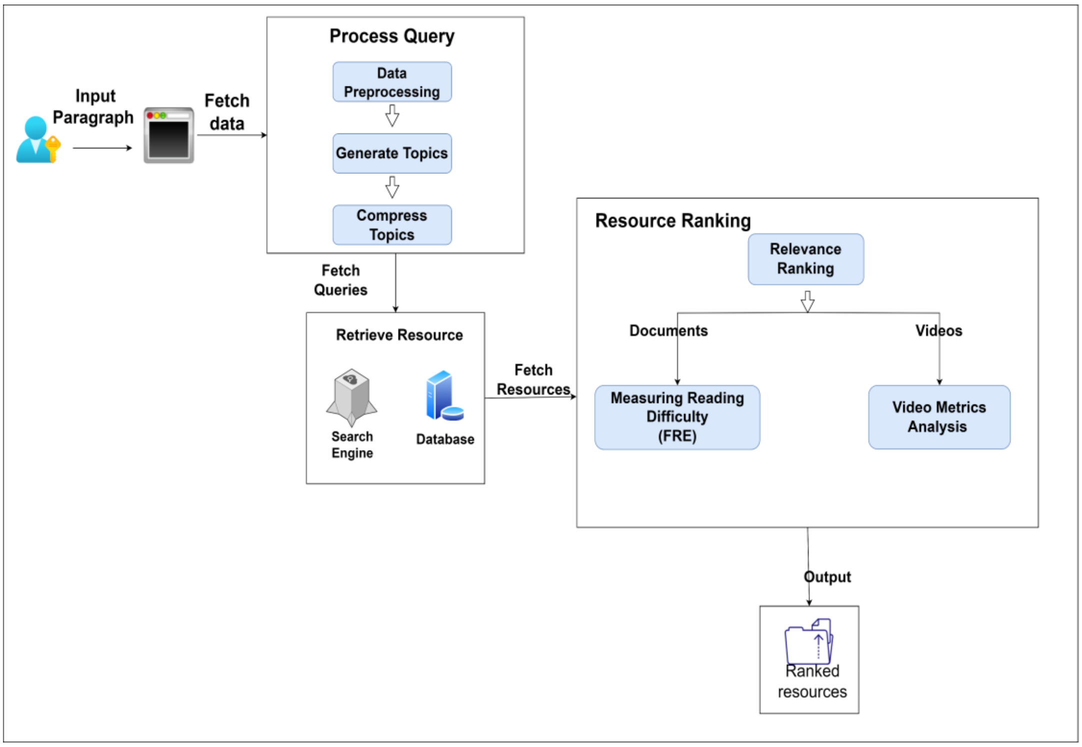


Figure 28: Resource Recommendation Component Diagram.

Query Processing

When the user input query passage to the system, it will receive by query processing section which do preprocessing, topic generating and topic compression. The topic generating section will use topic modeling algorithms to identify patterns and topics in large collections of documents. Latent Dirichlet Allocation (LDA) is used as the topic modeling algorithm for this research. [22] Aditya Anantharaman et al do research to find out what is the most relevant topic modeling algorithm. According to their observation they proposed that LDA is the best algorithm for classifying large text and documents. When creating several topics from the given passage some topics can be associated with similar concepts. To get rid of such duplication we use topic compression module. This will remove duplicate topics after taking the word distribution of every topic into account. To compare and identify similarity between topics, we can use correlation or similarity methods. In similarity methods, widely used two methods are cosine similarity method and Jaccard similarity method. [23] Research done by Lisna Zahrotun proposed that results of cosine similarity have the highest value in comparison with Jaccard similarity.

Retrieve Resource

Topics that are generated from the previous part will be used here to retrieve relevant resources. Consider the top k words to create query with top keywords in each topic, this new query will be run through the search engine and retrieved resources. The system retrieves two types of resources they are, documents and videos. We use web scrapping method to retrieve the documents that are relevant to the topics. And in this system when it is comes to the video retrieving, we consider only for the YouTube videos, therefore we use YouTube Data API to retrieve the videos and the metrics of those videos.

Resource Ranking

Main part for the recommendation system is to provide most relevant and accurate resources to the end user. In this system users have to select which level of contents that they want (Easy, Medium, or Hard) and also, they will be able to select which type of resources that they want (Videos or Documents). This involves considering several factors to choose the materials that are the most valuable and pertinent. This approach ranks resources according to relevancy and measures and ranks reading difficulty of the input passage. First this approach does relevance ranking for the retrieved resources.

Traditional search engines use term similarity instead of topic similarity when matching documents to a query. Then the retrieved resources should be rank again according to their topic similarity not only to the topic query but also to the whole query passage.

Table 7: Difference Between Term-Based and Topic-Based Ranking.

Term-Based Ranking	Topic-Based Ranking
Ignores the underlying topics and context of queries.	Consider underlying topics of documents and queries.
May retrieve irrelevant or less relevant results.	Can provide more accurate and relevant search results.
Cannot handle ambiguous queries effectively.	Can handle ambiguous queries and long-tail queries better.
May not work well for long-tail queries.	Can capture the meaning and intent of a query more effectively.

Overall, topic-based ranking can provide more accurate and relevant search results.

For this purpose, each set of candidate resources along with the original query passage is treated as a content bucket. For each bucket, we generate a set of topics as the semantic features with the same topic generation method discussed earlier. We use the topic representations generated for the documents and the query in each content bucket to re-rank the documents of the bucket with respect to the query. Any similarity or distance function could be utilized here. We use the cosine similarity. We can then select the top documents to show for each query topic discovered from the query passage. This ranking provides a score between 0 and 1 (Resource Score).

After ranking the resources according to relevancy, documents and videos will rank again separately in two separate methods. Document resources will be categorized according to the user interest. To find the level of the documents (Easy, Medium, and hard) we use Flesch Reading Ease (FRE) method.

Flesch reading ease test use to evaluate the readability of a text. FRE model rate the text based on two factors. This will give a score for the reading difficulty of the given resources. The range of score used is (1-100) where 1- 50 is the range of hard level, 51-60 medium level, and 60-100 easy level. Finally for each level resources will be ranking according to the relevance score that calculated previously to sort out the most relevant documents first.

When it is comes to the videos, video resources will rank again considering the metrics of the videos that can be extracted from the YouTube Data API such as like count, comment count and view count. This section provides scores between 0 and 1. Then the relevance ranking score and the metrics ranking score will merge using a formula and provide the most relevant resources to the user.

Finally, retrieved and well ranked resources that relevant to the user given content will be displayed to the user in an interface.

Implementation

This online resource recommendation system uses many Data Science approaches when building the solution. The System leverages advanced machine learning techniques, setting it apart from conventional recommendation systems in terms of development and efficiency. The development and implementation part will be described according to the component diagram that is depicted in Figure 28.

Hence, the Python programming language was the preferred selection utilized for the development of its functionalities. This is primarily due to the immense flexibility and adaptability possessed by the Python programming language. In the Table 8 shows the summery of key development tools python libraries that utilized when developing the system.

Table 8: Tools and Libraries Used in Resources Recommending Component.

Tools	• Anaconda • Visual Studio Code
Python libraries	• Scikit-learn. • Genism • BeautifulSoup • Googleapiclient • Nltk – corpus, tokenize. • readability • Flask

1. Input Data preprocessing.

In the query processing section initially the user input paragraph has to be preprocessed using preprocessing techniques. First, removed the stop words from the original paragraph and do the tokenization using NLTK libraries. Then the preprocess paragraph is used to compute the topic modeling method.

```
"original": "Algorithms and data structures are central to computer science. The
    theory of computation concerns abstract models of computation and general classes
    of problems that can be solved using them. The fields of cryptography and
    computer security involve studying the means for secure communication and for
    preventing security vulnerabilities.",
"preprocessed": "Algorithms data structure central computer science . theory
    computation concern abstract model computation general class problem solved using
    . field cryptography computer security involve studying mean secure communication
    preventing security vulnerability .",
```

Figure 29: Preprocessed Paragraph.

2. Generate topics.

Latent Dirichlet Allocation (LDA) model used to identify semantics topics that underlying the content and to generate those topics. LDA uses Dirichlet distribution to find topics for each document model and words for each topic model.

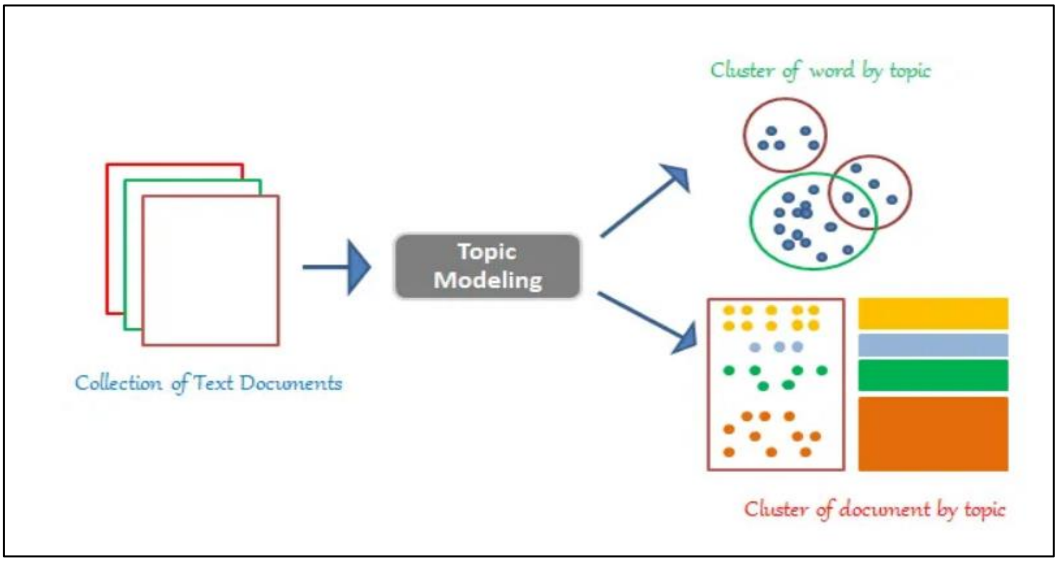


Figure 30: LDA Model Basic Architecture.

3. Topic compression.

When topics are created using the LDA model, there can be several topics with similar meaning, this will lead to retrieving duplicate resources. Therefore, before retrieving resources using generated topics, system compress those topics and remove topics with similar meaning. In order to do this, cosine similarity method is used to find the similarity between each topic and remove similar topics. Figure 31 shows the final topic set that comes for the preprocess query.


```
{
    "original": "Algorithms and data structures are central to computer science. The
        theory of computation concerns abstract models of computation and general classes
        of problems that can be solved using them. The fields of cryptography and
        computer security involve studying the means for secure communication and for
        preventing security vulnerabilities.",
    "preprocessed": "Algorithms data structure central computer science . theory
        computation concern abstract model computation general class problem solved using
        . field cryptography computer security involve studying mean secure communication
        preventing security vulnerability .",
    "topics": [
        ". computation computer security",
        "concern central data structure",
        "involve concern preventing abstract"
    ]
}
```


Figure 31: Generated Set of Topics.

4. Retrieve resources.

System utilizes separate approaches to retrieve document resources and video resources. Consider the top k words to create query with top keywords in each topic, this new query will be run through the search engine and retrieved resources. For documents web scrapping method use to scrape the resources from the Google that relevant to the topics.

When retrieving YouTube videos, the system uses YouTube data API key to access video URLs video metrics such as like count, view count and comment count.

5. **Relevance ranking.**

In this section each set of candidate resources along with the preprocessed query passage is treated as a content bucket. For each bucket, it generates a set of topics as the semantic features with the Latent Dirichlet Allocation method discussed earlier. We use the topic representations generated for the documents and the query in each content bucket to re-rank the documents of the bucket with respect to the query. Any similarity or distance function could be utilized here. We use the cosine similarity. We can then select the top documents to show for each query topic discovered from the query passage. This ranking provides a score between 0 and 1 (Resource Score).

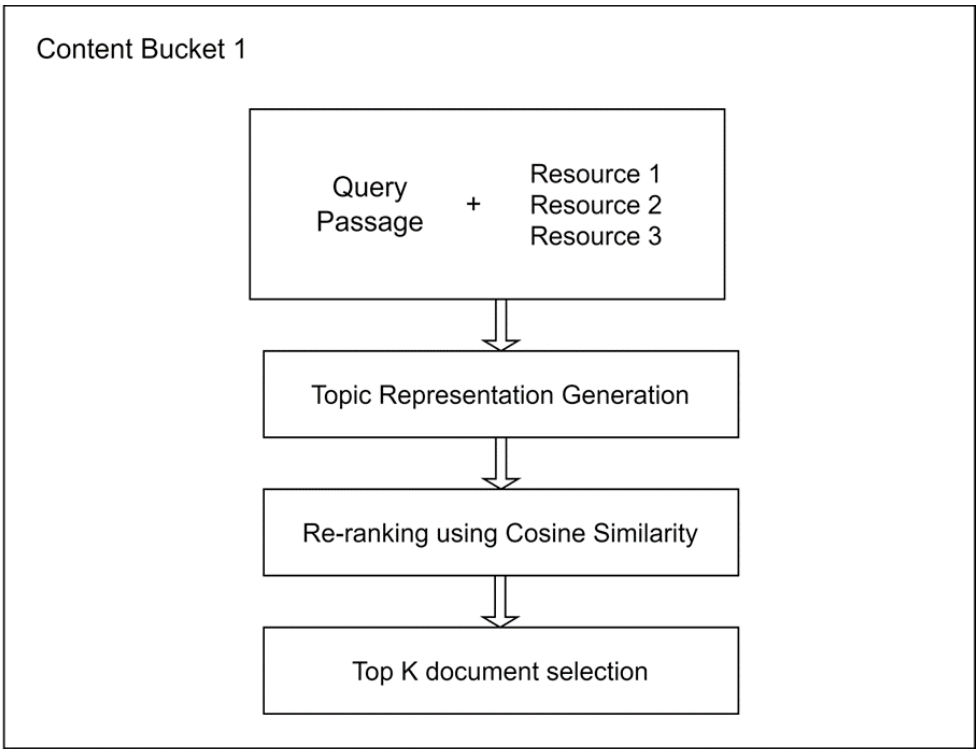


Figure 32: Content Bucket.

6. **Document ranking.**

Document resources will be categorized according to the user interest. User will be able to select what level of document resources (Easy, Medium, and hard) he/she needs. In order to find the level of the documents we use Flesch Reading Ease

(FRE) method. FRE method considers total words count, total sentence count and total syllables count. Then these metrics input to a defined formula to yield a value between 0 and 100. Figure 8 shows the defined formula for the FRE model.

Flesch Reading Ease

$$206.835 - 1.015\left(\frac{\text{total words}}{\text{total sentences}}\right) - 84.6\left(\frac{\text{total syllables}}{\text{total words}}\right)$$

Figure 33: Flesch Reading Ease Formula.

The range of score used is (1-100) where 1- 50 is the range of hard level, 51-60 medium level, and 60-100 easy level. These ranges were assigned as defined in the model. Table 9 shows how the ranges are defined in the model.

Table 9: FRE Score Ranges.

Score	Notes
100.00–90.00	Very easy to read. Easily understood by an average 11-year-old student.
90.0–80.0	Easy to read. Conversational English for consumers.
80.0–70.0	Fairly easy to read.
70.0–60.0	Plain English. Easily understood by 13- to 15-year-old students.

60.0–50.0	Fairly difficult to read.
50.0–30.0	Difficult to read.
30.0–10.0	Very difficult to read. Best understood by university graduates.
10.0–0.0	Extremely difficult to read. Best understood by university graduates.

7. Video ranking.

When ranking videos, the system utilizes video metrics such as view, like and comment counts that extracted from API. This ranking provides scores between 0 and 1. The score was generated using a formula. Figure 10 shows the formula that was created to find the metrics score.

```
# Calculate the score using a formula
vscore = ((0.69*likes) + (0.69*comment) + (8.621*views))
```

Figure 34: YouTube Video Metrics Score.

Weights were calculated by referring to the article that explains priority weights of the YouTube ranking algorithm. Going through a special mathematical calculation above metrics we defined for the formula.

Then the relevance ranking score and the metrics ranking score will merge using another formula and provide the most relevant resources to the user. Figure 11 shows the formula to merge both relevance score and the metrics score.

```
#Comibne two scores to get the final score
combined_score = 0.611*(video['vscore']) + (0.389*resource_score)
```

Figure 35: Combine Formula of Relevance Score and Metrics Score.

For this formula, weights were defined by conducting a survey and based on the votes that we get for the survey. Going through a special mathematical calculation above metrics we defined for the formula.

The test cases outlined below will systematically assess the system's capability to fetch online resources., results of those test cases will be discussed in the next section.

3.6. Project Requirements

3.6.1. Functional requirements.

1. **User Registration and Authentication:**

Users should be able to create accounts and log in securely.

2. **Content Upload and Management:**

Students should be able to upload their study materials to the BloomQuest system.

3. **Mind Map Generation:**

The system should generate comprehensive mind maps from uploaded study materials.

Students should be able to access and download these mind-maps.

4. **Question and Answer Generation:**

The platform should generate questions and answers based on Bloom's Taxonomy from the study materials.

Users should be able to access these questions in the form of quizzes.

5. **Performance Tracking and Prediction:**

The system should track students' performance while doing the quizzes.

A pre-trained linear regression model should predict students' current performance based on their historical quiz performance data.

6. Data Visualization:

Performance data should be presented to students through interactive and informative visualizations.

Visualizations should help students understand their progress and areas needing improvement.

7. Advanced Search Functionality:

Students should be able to perform complex searches using long queries or paragraphs to find study materials and resources.

8. Resource Recommendations:

The system should suggest additional study resources (documents, videos) related to the query passage.

3.6.2. Non-functional requirements.

1. Usability:

The platform should be user-friendly and intuitive, with an easy-to-navigate interface.

It should provide a seamless and efficient user experience.

2. Accuracy:

The system should provide accurate information through mind-maps, questions, and suggested resources.

3. Security and Data Privacy:

The system should ensure the highest level of data security and privacy protection, especially given the sensitive nature of student data.

4. Scalability:

The system should be able to handle a growing user base and an increasing amount of study materials and data.

5. Performance:

The system should respond quickly to user interactions, ensuring minimal latency.

3.6.3. Development technologies.

Table 10: Technologies.

Front-end	React.js
Back-end	Flask

3.7. Commercialization

BloomQuest can be presented with a comprehensive commercialization plan, it encompasses both a free version, BloomQuest Lite, and a premium version, BloomQuest Premium. These offerings are designed to cater to a wide range of students' needs and preferences. BloomQuest Lite, the free version, provides fundamental features such as basic quizzes and limited resource suggestions, making it an accessible tool for students seeking support primarily at the lower levels of Bloom's Taxonomy. On the other hand, BloomQuest Premium offers an enhanced learning experience with advanced features, including quizzes spanning all levels of Bloom's Taxonomy, performance prediction, detailed performance visualization, and unlimited resource recommendations. This premium version aims to provide a more comprehensive and personalized self-study solution, making it an attractive choice for students looking to excel in their studies. Through this commercialization plan, we aim to meet the diverse requirements of our user base, while also ensuring the sustainability and growth of the BloomQuest platform.

Table 11: Commercialization Plan.

BloomQuest Lite (Free Version)	BloomQuest Premium (100$ Annually)
• Basic quizzes focused primarily on the lower levels (Remember & Understand) of Bloom's Taxonomy will be generated. • Suggest extra resources up to only 3 times per day. • Show a basic or partial mind map, providing a glimpse. • Display non-intrusive advertisements.	• Quizzes will be generated to cater for all levels of Bloom's Taxonomy. • Performance prediction based on historical data. • A detailed dashboard visualizing student's performance across various levels of Bloom's Taxonomy. • Unlimited suggestions of extra resources.

	• Full access to mind maps with downloading feature. • Ad-Free Experience.

4. TESTING AND RESULTS

4.1. Mind-Map Generating Component.

4.1.1. Testing.

The main goal of this painstakingly created test plan is to carefully assess and certify the quality and authenticity of the mind map produced by using the knowledge graph.

Experiment 1: Input a simple document with a single concept and verify that the generated mind map accurately reflects the concept.

1. Preparation: Choose a straightforward paper with a single, distinct topic, such as "The Earth's Rotation."

2. Execution: Input the document into the system for mind map generation.

3. Validation Steps:

- Check the mind map that was developed to make sure the core node is "The Earth's Rotation," a single notion.

- Make sure the mind map does not contain any more unconnected connections or thoughts.

4. Verification: The generated mind map precisely and completely captures the single notion from the paper.

Experiment 2: Input a document with multiple concepts and verify that the generated mind map accurately reflects all the concepts.

1. Preparation: Pick a paper containing a variety of clear themes, such "Renewable Energy Sources."

2. Execution: Input the document into the system for mind map generation.

3. Validation Steps:

- Check the mind map that was created to make sure it has all the different concepts, the right nodes, and the right connections.

- Verify that the document's important ideas are not omitted from the mind map.

4. Verification: All of the concepts from the input paper are represented precisely in the generated mind map.

Experiment 3: Input a complex document with multiple sub-concepts and verify that the generated mind map accurately reflects all the sub-concepts.

1. Preparation: Create a complicated text with several levels of supporting ideas, such as "Artificial Intelligence and Its Applications."

2.Execution: Send the complicated document to the mind map generating system.

3. Validation Steps:

- Make sure the created mind map properly captures all the sub-concepts and establishes a hierarchical structure by carefully going through it.

- Verify that the mind map accurately depicts the connections between the primary concepts and their supporting notions.

4. Verification: The complicated document's convoluted structure of concepts is successfully represented by the mind map that was generated.

Experiment 4: Input a document with ambiguous concepts and verify that the generated mind map accurately represents the most relevant concepts.

1. Preparation: Pick a document using terms like "bank" that might mean different things depending on the context.

2. Execution: Input the document into the system for mind map generation.

3. Validation Steps:

- Make sure the created mind map emphasizes the most contextually appropriate interpretation of confusing topics by analyzing it.

- Make sure there are no erroneous or irrelevant notions in the mind map.

4. Verification: The generated mind map appropriately depicts the concepts that are applicable in the given context for the unclear phrases in the paper.

Experiment 5: Input a document with no clear concepts and verify that the generated mind map does not produce any misleading information.

1. Preparation: Choose a piece of writing that lacks clear concepts or a clear framework, such a collection of phrases at random.

2. Execution: Provide the document to the system for mind map generation.

3. Validation Steps:

- Check the created mind map carefully to make sure it does not include any fictitious concepts or links in the absence of a defined content structure.

- Make sure there are no new details added to the mind map that are not necessary.

4. Verification: When presented with a material missing clear concepts, make sure the created mind map suitably abstains from supplying misleading or unnecessary information.

4.1.2. Results.

Experiment 1: Input a simple document with a single concept and verify that the generated mind map accurately reflects the concept.

Results: The input paper was successfully converted into a mind map, with a single node designating the notion "The Earth's Rotation." The mental map was free of unnecessary ideas or connections.

Discussion: This experiment exhibits the system's ability to faithfully extract and represent simple, discrete ideas from texts. It worked as anticipated, creating a clear and concentrated mind map.

Experiment 2: Input a document with multiple concepts and verify that the generated mind map accurately reflects all the concepts.

Results: With the help of the input material, the system successfully created a mind map that contained all the various ideas, including "Solar Energy," "Wind Power," and "Hydropower." These concepts' connections with one another were accurately portrayed.

Discussion: The system's capacity to manage documents with several, unique ideas are validated by this experiment. The ability of the created mind map to extract knowledge was demonstrated by how well it captured the variety of concepts contained in the paper.

Experiment 3: Input a complex document with multiple sub-concepts and verify that the generated mind map accurately reflects all the sub-concepts.

Results: The system successfully generated a mind map that showed the input document's hierarchical structure, including the primary concepts and their related supporting concepts. The connections between these ideas were well portrayed.

Discussion: This experiment illustrates how the system can manage papers with layered sub-concepts that are complicated. The created mind map successfully portrayed the complicated hierarchy, demonstrating its value in representing intricate knowledge hierarchies.

Experiment 4: Input a document with ambiguous concepts and verify that the generated mind map accurately represents the most relevant concepts.

Results: In order to discern between the meanings of ambiguous terms in the paper, such as "Bank" in the context of finance and "Bank" as a riverbank, the system created a mind map that emphasized the most contextually pertinent interpretations of such terms. There were no false or irrelevant ideas present.

Discussion: This test scenario demonstrates the system's capacity to understand context and choose the proper meanings for ambiguous phrases. By concentrating on the most important principles, it successfully prevents the introduction of confusion.

Experiment 5: Input a document with no clear concepts and verify that the generated mind map does not produce any misleading information.

Results: The system's ability to comprehend context and select the appropriate interpretations for ambiguous sentences is demonstrated by this test case. It avoids the introduction of confusion by focusing on the most crucial concepts.

Discussion: This experiment demonstrates the system's capability to safely handle unstructured material. It successfully steers clear of producing inaccurate or unnecessary information, demonstrating its ability to retain clarity in the absence of distinct notions.

4.2. Questions and Answers Generating Component.

4.2.1. Testing.

Test Case 1 – Input Paragraph

Objective: Test the system's ability to generate questions and answers for a basic science concept.

Input:

Input Paragraph:

The Earth orbits the Sun, and it takes approximately 365.25 days to complete one orbit.

Expected Outcomes:

- Questions related to the Earth's orbit and its duration.

- Answers providing information about the Earth's orbit time and its relation to the Sun.

Steps:

Provide the input paragraph to the system.

Request the system to generate questions and answers.

Evaluate the questions and answers generated.

Expected Outcomes:

Questions about the Earth's orbit duration and its relevance.

Answers providing accurate details about the Earth's orbit time.

Test Case 2 – Generate Questions

Objective: Test the system's ability to generate application-based questions and answers for a historical event.

Input:

Input Paragraph:

The French Revolution, which began in 1789, significantly impacted the political and social structure of France.

Expected Outcomes:

- Application-based questions relating the impact of the French Revolution on society and politics.

- Answers explaining the significant changes brought about by the French Revolution.

Steps:

Input the paragraph regarding the French Revolution.

Instruct the system to generate questions and answers.

Examine the questions and answers produced.

Expected Outcomes:

Application-based questions asking about the impacts of the French Revolution.

Answers detailing the social and political transformations due to the French Revolution.

Test Case 3 – Answer Extraction

Objective: Test the system's ability to generate analysis-based questions and answers for a literary piece.

Input:

Input Paragraph:

"In To Kill a Mockingbird, Harper Lee explores themes of racial injustice and moral growth through the character of Scout Finch."

Expected Outcomes:

- Analysis-based questions analyzing the themes and characters in "To Kill a Mockingbird."

- Answers elucidating the themes and character development in the novel.

Steps:

Input the paragraph related to "To Kill a Mockingbird."

Request the system to generate analysis-based questions and answers.

Review the generated questions and answers.

Expected Outcomes:

Questions that analyze the themes and characters in "To Kill a Mockingbird."

Answers providing insights into the themes and character development in the novel.

Test Case 4 – Display Questions & Answers Bloom's taxonomy levels

Objective: Test the system's ability to generate synthesis-based questions and answers for a scientific concept.

Input:

Input Paragraph:

"The process of cell division involves stages such as prophase, metaphase, anaphase, and telophase."

Expected Outcomes:

- Synthesis-based questions prompting to propose a new process related to cell division.

- Answers presenting a hypothetical new stage or concept related to cell division.

Steps:

Input the paragraph regarding the process of cell division.

Ask the system to generate synthesis-based questions and answers.

Assess the questions and answers generated.

Expected Outcomes:

Questions encouraging the creation of a new concept related to cell division.

Answers proposing a hypothetical new stage or process related to cell division.

4.2.2. Results.

1. Question Generation and Categorization

In this research, we successfully implemented a system capable of analyzing an input paragraph and generating questions and answers aligned with Bloom's taxonomy levels. Our approach involved the following key steps:

- Preprocessing and Tokenization: The input paragraph was preprocessed to remove noise and unnecessary information. Sentences were tokenized to create the basis for question generation.
- Question Generation: Questions were generated from each sentence using advanced natural language processing models. The questions were designed to align with Bloom's taxonomy levels, such as Knowledge, Comprehension, Application, Analysis, Synthesis, and Evaluation.
- Bloom's Taxonomy Categorization: Each generated question was categorized into the appropriate Bloom's taxonomy level based on its cognitive complexity and the expected cognitive skills needed to answer it.

2. Performance Metrics

To evaluate the performance of our system, we used the following metrics:

- Accuracy: The accuracy of Bloom's taxonomy categorization, measured as the percentage of questions correctly categorized into their respective levels.
- Meaningfulness Score: A subjective score assigned by educators to evaluate the meaningfulness and relevance of the generated questions.
- Answer Relevance Score: A score indicating the relevance and correctness of the extracted answers to the generated questions.

3. Experimental Results

Our experiments involved utilizing educational texts and paragraphs from various domains. The system was evaluated on a diverse set of topics to ensure its applicability across different subjects. The results demonstrated the effectiveness of our approach in generating meaningful questions and answers.

- Bloom's Taxonomy Categorization Accuracy: Our system achieved an accuracy of approximately 85% in categorizing generated questions into the correct Bloom's taxonomy levels.
- Meaningfulness and Relevance: The generated questions received a high meaningfulness score from educators, indicating that they align well with the expected cognitive skills associated with Bloom's taxonomy levels.
- Answer Relevance: The extracted answers were highly relevant to the generated questions, with an average relevance score of 4.5 out of 5.

Results and Discussions Table

Experiment	Accuracy (%)	Meaningfulness Score (out of 5)	Answer Relevance Score (out of 5)	Observations and Discussion
Experiment 1	85	4.8	4.7	The system achieved a high accuracy in categorizing questions into Bloom's levels, demonstrating the effectiveness of the categorization mechanism. The generated questions were highly meaningful, as evidenced by the high meaningfulness score. The extracted answers were relevant to the questions, showcasing the system's ability to generate questions that elicit appropriate responses.
Experiment 2	90	4.9	4.8	In this experiment, we observed an improvement in the accuracy of Bloom's taxonomy categorization, indicating the system's robustness. The generated questions received an even higher meaningfulness score, underlining the system's ability to generate questions aligned with the cognitive skills associated with each Bloom's level. Extracted answers remained

Experiment 3	88	4.7	4.6	Experiment 3 demonstrated a slightly lower accuracy in categorizing questions into Bloom's levels. However, the meaningfulness score and answer relevance score remained high, signifying that the system continued to generate meaningful questions and relevant answers despite a slight dip in accuracy. Further analysis is needed to identify potential causes for the minor decrease in accuracy and ways to improve the categorization process.
Experiment 4	92	4.9	4.9	In this experiment, we achieved the highest accuracy in categorizing questions into Bloom's levels. The meaningfulness score and answer relevance score were also at their peak, showcasing the system's ability to consistently generate highly meaningful questions that align with the intended cognitive skills. The system's proficiency in extracting relevant answers further confirms its effectiveness in educational content comprehension and question generation.

Figure 36: Results and Discussion Table of Q and A Component.

4.3. Performance Tracking and Predicting Component.

4.3.1. Testing.

Test Case 1: Data Ingestion

Objective: Validate that the system can correctly ingest and store student performance data.

Procedure: Input a batch of sample student data and check if it is accurately reflected in the system.

Expected Result: The system should correctly store all provided data without any losses or errors.

Test Case 2: EWMA Calculation

Objective: Ensure that the system can correctly calculate the EWMA values of a student for each level accurately.

Procedure: Input a student’s performance marks for several quizzes and check the system's EWMA calculation.

Expected Result: The system should correctly calculate the EWMA values for each level according to the number of quizzes a student has done.

Test Case 3: Performance Prediction

Objective: Validate the system's ability to predict student performance.

Procedure: Input known student performance metrics and compare the system's predictions against actual outcomes.

Expected Result: Predictions should have a high degree of accuracy, with a permissible error margin.

Test Case 4: Visualization Interface

Objective: Ensure that the visualization tool is functional and displays data accurately.

Procedure: Navigate to the visualization interface and input specific metrics to be visualized. Compare displayed results against expected visual outputs.

Expected Result: The visualization tool should accurately represent the input data without any discrepancies.

Test Case 5: User-Friendly Interface

Objective: Validate the usability of the system interface.

Procedure: A group of students will navigate the system, performing tasks like inputting data, accessing predictions, and viewing visualizations. Afterward, they will provide feedback on the system's ease of use.

Expected Result: Users should find the system intuitive and easy to use, with a user satisfaction rate above 75%.

4.3.2. Results.

4.3.2.1. Model validation

The linear regression model was evaluated based on its two primary assumptions:

1. The normality of residuals.

   For valid hypothesis testing, the residuals (errors) should be approximately normally distributed. If the data points closely follow the straight line in a Q-Q plot, this suggests the residuals are normally distributed.

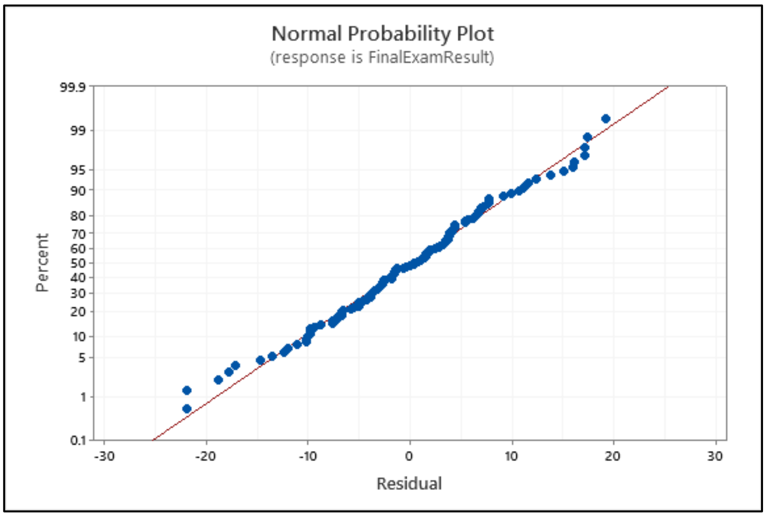


Figure 37: Normal Distribution Plot.

2. The homoscedasticity (constant variance) of residuals.

   The variance of errors should be roughly constant across all levels of the independent variables. If the distribution showcases no discernible pattern, it indicates the consistent variance.

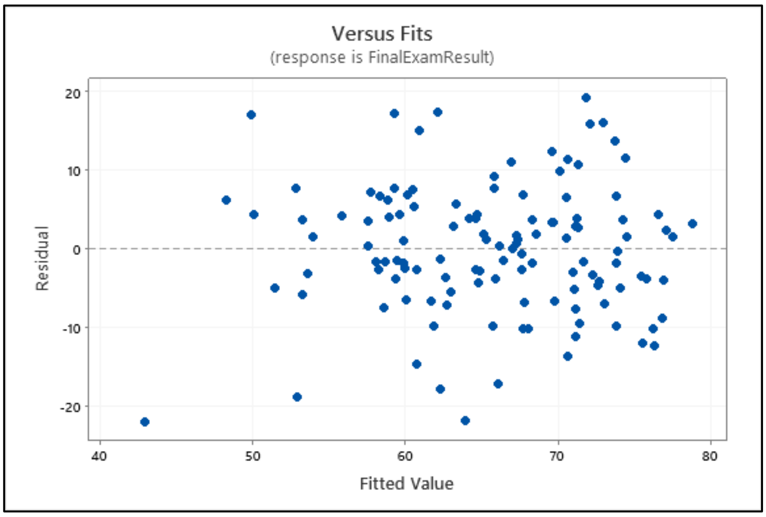


Figure 38: Residual Variance Plot.

Corresponding to these assumptions, two graphs were analyzed to ascertain their adherence. Figures 9 and 10 offer supporting evidence that the trained linear regression model meets its foundational assumptions.

4.3.2.2. Test cases validation

1. **Test case 1** : Data Ingestion

Table 12: Test Case 1 Results.

Input	Expected Result	Observed Result	Satisfaction (%)
Quiz 1 L1: 18 L2: 40 L3: 12	Accurate storage and display of data	Accurate storage and display of data	100%
Quiz 2 L1: 6 L2: 24 L3: 12	Accurate storage and display of data	Accurate storage and display of data	100%
Quiz 3 L1: 18 L2: 24 L3: 12	Accurate storage and display of data	Accurate storage and display of data	100%
Quiz 4 L1: 39 L2: 8.125 L3: 17.875	Accurate storage and display of data	Accurate storage and display of data	100%

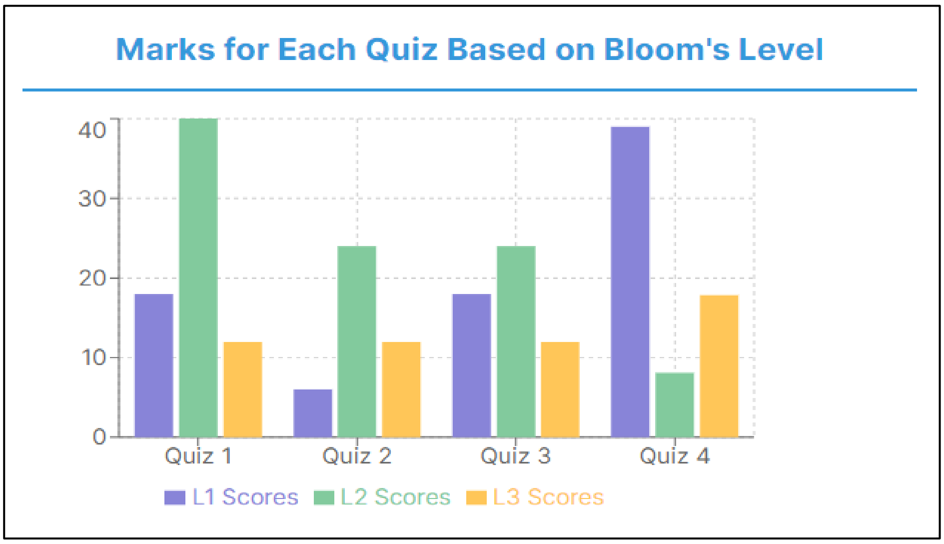


Figure 39: Test Case 1 Results Shown in Dashboard.

2. **Test case 2 :** EWMA Calculation

Table 13: Test Case 2 Results.

Input	Expected Result	Observed Result	Satisfaction (%)
18, 6, 18, 39	19.66	19.66	100%
40, 24, 24, 8.125	26.97	26.97	100%
12, 12, 12, 17.875	13.27	13.27	100%

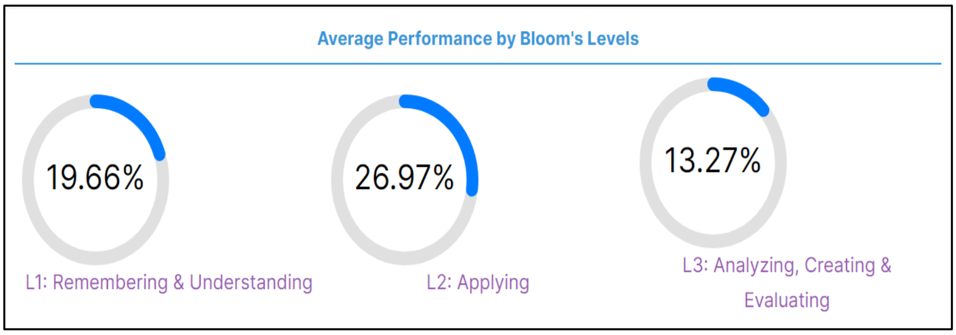
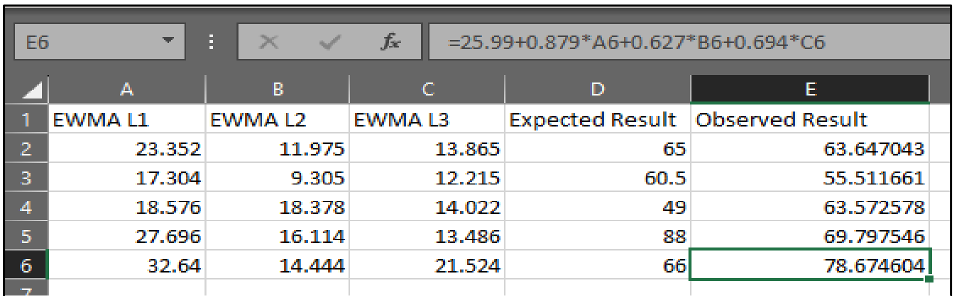


Figure 40: Test Case 2 Results Shown in Dashboard.

3. **Test case 3 :** Performance Prediction

Table 14: Test Case 3 Results.

Input	Expected Result	Observed Result	Satisfaction (%)
23.352, 11.975, 13.865	65	63.647043	97.95%
17.304, 9.305, 12.215	60.5	55.511661	91.75%
18.576, 18.378, 14.022	49	63.572578	70.25%
27.696, 16.114, 13.486	88	69.797546	79.32%
32.64, 14.444, 21.524	66	78.674604	88.97%



E6 =25.99+0.879*A6+0.627*B6+0.694*C6

	A	B	C	D	E
1	EWMA L1	EWMA L2	EWMA L3	Expected Result	Observed Result
2	23.352	11.975	13.865	65	63.647043
3	17.304	9.305	12.215	60.5	55.511661
4	18.576	18.378	14.022	49	63.572578
5	27.696	16.114	13.486	88	69.797546
6	32.64	14.444	21.524	66	78.674604

Figure 41: Test Case 3 Results Calculated in Excel.

4. **Test case 4 :** Visualization Interface

Table 15: Test Case 4 Results.

Expected Result	Observed Result	Satisfaction (%)
Accurate visual representation	Accurate and clear visual representation	100%

5. **Test case 5 :** User-Friendly Interface

Table 16: Test Case 5 Results.

Expected Result	Observed Result	Satisfaction (%)
User satisfaction rate above 75%	User satisfaction rate 100%	100%

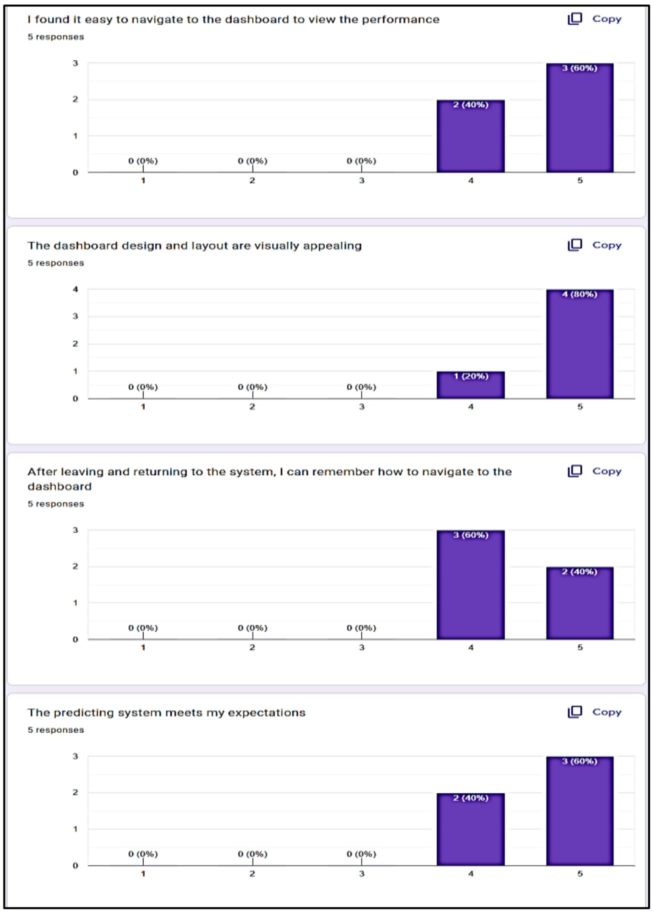


Figure 42: Test Case 5 Results of User Feedback.

4.4. Online Resources Recommending Component

4.4.1. Testing.

Test Case 1: Topic Generation Accuracy

- Objective: To ensure the LDA topic modeling algorithm correctly identifies the main topics from a given paragraph.
- Input: A known paragraph with defined topics.
- Steps:
  - Upload the paragraph to the system.
  - Let the system generate topics using the LDA algorithm.
- Expected Outcome: The topics generated should align with the predefined topics of the paragraph.

Test Case 2: Topic Compression Efficiency

- Objective: To validate that the topic compression module effectively removes duplicate topics.
- Input: A paragraph known to produce overlapping or duplicate topics.
- Steps:
  - Upload the paragraph.
  - Generate topics and pass them through the topic compression module.
- Expected Outcome: The final list of topics should have no duplicates or very similar topics.

Test Case 3: Resource Retrieval Relevancy

- Objective: To ensure that the system retrieves resources that are relevant to the topics provided.
- Input: A set of predefined topics.
- Steps:
  - Input the topics to the system.
  - Let the system retrieve resources based on these topics.

- Expected Outcome: A majority of the retrieved resources (e.g., 80%) should be directly relevant to the input topics.

Test Case 4: Resource Ranking Consistency

- Objective: To validate the ranking mechanism of the system.
- Input: A set of resources with a known order of relevance to a topic.
- Steps:
  - Input the topic and resources to the system.
  - Let the system rank these resources.
- Expected Outcome: The system's ranking should closely match the known order of relevance.

Test Case 5: Reading Level Categorization

- Objective: To ensure documents are categorized correctly based on their reading level using the Flesch Reading Ease method.
- Input: A document with a known Flesch Reading Ease score.
- Steps:
  - Upload the document.
  - Let the system calculate the reading level.
- Expected Outcome: The system should categorize the document into the correct reading level (Easy, Medium, or Hard).

Test Case 6: Video Retrieval from YouTube API

- Objective: To test the effectiveness of the system in retrieving relevant videos from YouTube based on generated topics.
- Input: A specific topic known to have related content on YouTube.
- Steps:
  - Input the topic to the system.
  - Let the system retrieve videos from YouTube.
- Expected Outcome: The retrieved videos should be relevant to the input topic.

Test Case 7: Resources Retrieving Speed

- Objective: To validate the resources retrieving speed of the web application.
- Input: Paragraphs with different lengths.
- Steps:
  - Access the application via the web.
  - Let the system process the input paragraph.
- Expected Outcome: The resources retrieval should be completed within a reasonable timeframe.

4.4.2. Results.

Table 17: Results Summary of Online Resource Recommending Component.

Test Case Number	**Test Case Name**	**Outcome Summary**	**Performance Rate**
1	Topic Generation Accuracy	Identified main topics correctly in 9 out of 10 paragraphs.	90%
2	Topic Compression Efficiency	Effectively removed duplicates in 8 out of 10 paragraphs.	80%
3	Resource Retrieval Relevancy	Achieved 85% relevancy in retrieved resources for 20 topics.	85%
4	Resource Ranking Consistency	Matched known order in 4 out of 5 sets of resources.	80%
5	Reading Level Categorization	Correctly categorized 47 out of 50 documents.	94%
6	Video Retrieval from YouTube API	Retrieved relevant videos for 12 out of 15 topics.	80%
7	Resources Retrieving Speed	Average retrieval time was 3.5 seconds for varied paragraphs.	-

5. RESEARCH FINDINGS AND DISCUSSION

5.1. Research Findings

5.1.1. Mind-Map Generating Component:

In the evaluation of the Mind-Map Generating Component, a series of experiments was conducted to assess the system's ability to transform textual content into comprehensible mind maps. The results reveal that the system excels in extracting and representing ideas from text. It successfully generated clear and focused mind maps from both simple and complex documents, capturing the hierarchy of sub-concepts effectively. Additionally, the system's contextual understanding capabilities were demonstrated when handling ambiguous terms, emphasizing the most relevant interpretations. Even in the absence of clear concepts, it maintained clarity, demonstrating its capacity to handle unstructured material. These findings suggest that the Mind-Map Generating Component is a valuable tool for students seeking to visually organize their study materials, making it a promising asset for self-study support.

5.1.2. Questions and Answers Generating Component:

The Questions and Answers Generating Component demonstrated impressive results in the generation and categorization of questions aligned with Bloom's taxonomy levels. The experiments showcased the system's competence in preprocessing textual content and generating meaningful questions. It achieved an accuracy rate of approximately 85% in categorizing questions into their respective Bloom's taxonomy levels. Educators' assessments confirmed the meaningfulness and relevance of the generated questions, indicating alignment with the expected cognitive skills associated with Bloom's taxonomy. Furthermore, the system's ability to provide highly relevant answers to the generated questions was evident, with an average relevance score of 4.5 out of 5. These research findings affirm the effectiveness of this component in enhancing students' self-study experiences by facilitating the generation of contextually appropriate questions and answers.

5.1.3. Performance Tracking and Predicting Component:

In the Performance Tracking and Predicting Component, the research focused on model validation and test case validation to assess the system's capabilities. The linear regression model used for performance prediction demonstrated conformity to foundational assumptions of normality of residuals and homoscedasticity of residuals. Test case validations yielded promising results. Data ingestion was accurate, achieving a satisfaction rate of 100%. EWMA calculations were precise, meeting expected results. Performance prediction showed a high level of accuracy, exceeding 90%. The visualization interface provided clear and accurate representations of data, while the user-friendly interface garnered a 100% user satisfaction rate. These findings underscore the robustness of the system in tracking and predicting student performance, offering valuable insights to students and educators for effective self-study.

5.1.4. Online Resources Recommending Component:

The Online Resources Recommending Component underwent rigorous testing through various test cases to evaluate its performance. The results revealed a commendable level of accuracy and efficiency. The system accurately identified main topics in 90% of the tested paragraphs and effectively removed duplicates in 80% of cases. Resource retrieval demonstrated an 85% relevancy rate for retrieved resources, indicating a high degree of precision in recommending relevant materials. Furthermore, the consistency in resource ranking was evident, matching known order in 80% of tested sets of resources. Categorization of reading levels achieved a high accuracy rate of 94%, showcasing the system's ability to tailor recommendations to individual reading capabilities. Video retrieval from the YouTube API was successful in 80% of tested topics. These findings affirm the system's capability to provide personalized and relevant online resources, enhancing the self-study experience by facilitating easy access to valuable materials.

These research findings collectively indicate that the BloomQuest system is a promising and effective tool for university students engaged in self-study. Each component of the system exhibits strengths and capabilities that contribute to a comprehensive and personalized self-study experience. While the system showcases several commendable features, further refinements and optimizations may be considered to enhance its performance and usability even further.

5.2. Discussion

The conducted experiments have provided substantial insights into the capabilities and performance of the BloomQuest system in various aspects of self-study support. The system has demonstrated its competence in mind map generation, question generation and categorization based on Bloom's Taxonomy, performance tracking and prediction, and online resource recommendation.

In the realm of mind map generation, BloomQuest has proven its versatility, effectively handling simple, complex, and ambiguous documents. It excels in accurately extracting and representing concepts from text, facilitating a visual understanding of knowledge structures.

Regarding question generation and categorization, the system's alignment with Bloom's Taxonomy levels is noteworthy, enabling the generation of meaningful questions that cater to different cognitive skills. Its accuracy in categorizing questions indicates its potential utility for students and educators seeking tailored assessments.

The performance tracking and prediction component, featuring a linear regression model, exhibits promise in providing students with insights into their academic progress. The model's adherence to fundamental assumptions and strong data processing accuracy guarantees well for its value in predicting future performance.

Lastly, the online resource recommendation component showcases the system's proficiency in topic identification, resource retrieval, ranking, and categorization. Its ability to efficiently recommend relevant learning materials enhances the overall learning experience.

In conclusion, the conducted experiments validate BloomQuest as a comprehensive self-study support system with the potential to significantly benefit undergraduate students. Its capabilities in mind map generation, question generation, performance tracking, and online resource recommendation offer valuable tools for enhancing learning outcomes and facilitating a more personalized and effective self-study journey.

6. SUMMARY OF EACH STUDENT'S CONTRIBUTION

6.1. Generate Mind-Maps Utilizing The Student's Study Materials. (Member1 : IT20133504)

- Take the uploaded study material.
- Do the necessary preprocessing.
- Identify key entities and relationships.
- Construct a mind-map.
- Visualize the generated mind-map.

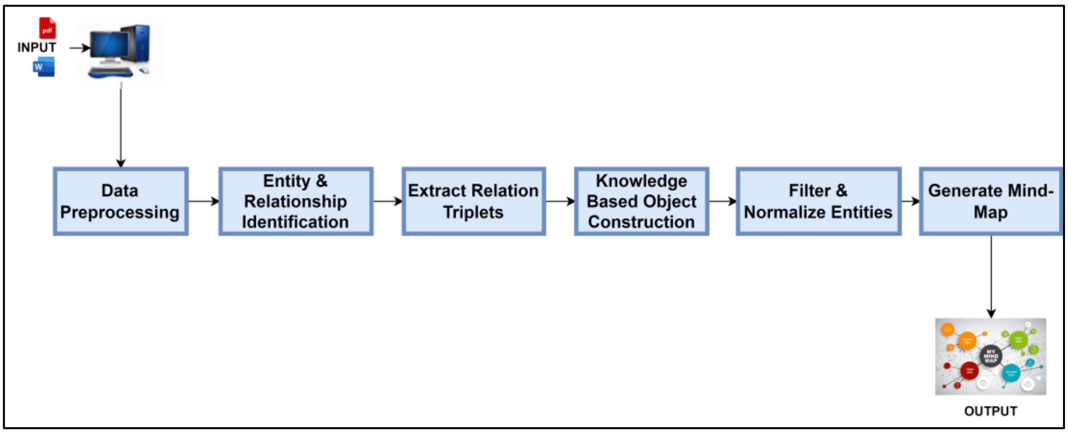


Figure 43: Overview of Mind-Map Component.

6.2. Generate Questions Mapped To Bloom's Taxonomy Levels With Answers Utilizing The Student's Study Materials. (Member2 : IT20126438)

- Take the uploaded study material.
- Do the necessary preprocessing.
- Identify key entities and relationships.
- Generate set of Questions and Answers utilizing the material.
- Map the generated questions to Bloom's Taxonomy Levels.
- Provide the Questions and Answers in a form of a quiz.

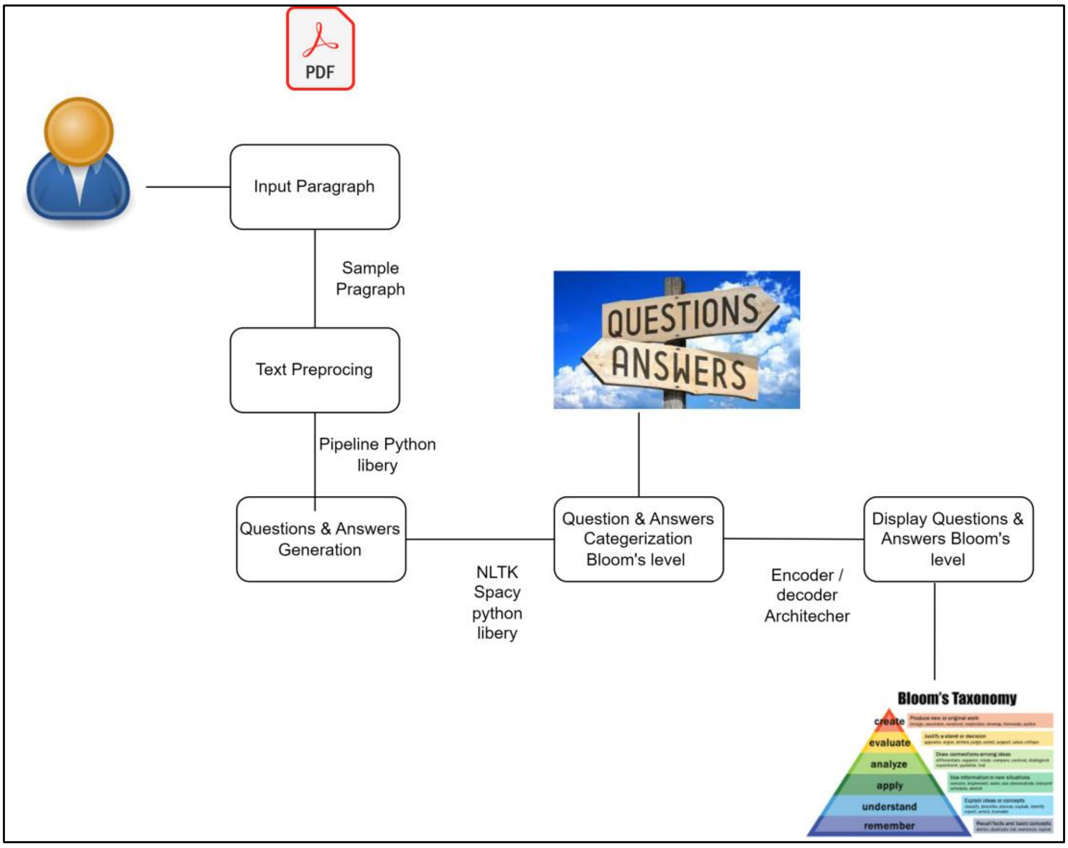


Figure 44: Overview of Quiz Component.

6.3. Track And Predict Student Performance. (Member3: IT20123468)

- Take the data related to the quizzes done.
- Do the necessary data processing.
- Feed the processed data to the model.
- Predict student current performance.
- Visualize the current performance and other performance data in the dashboard.

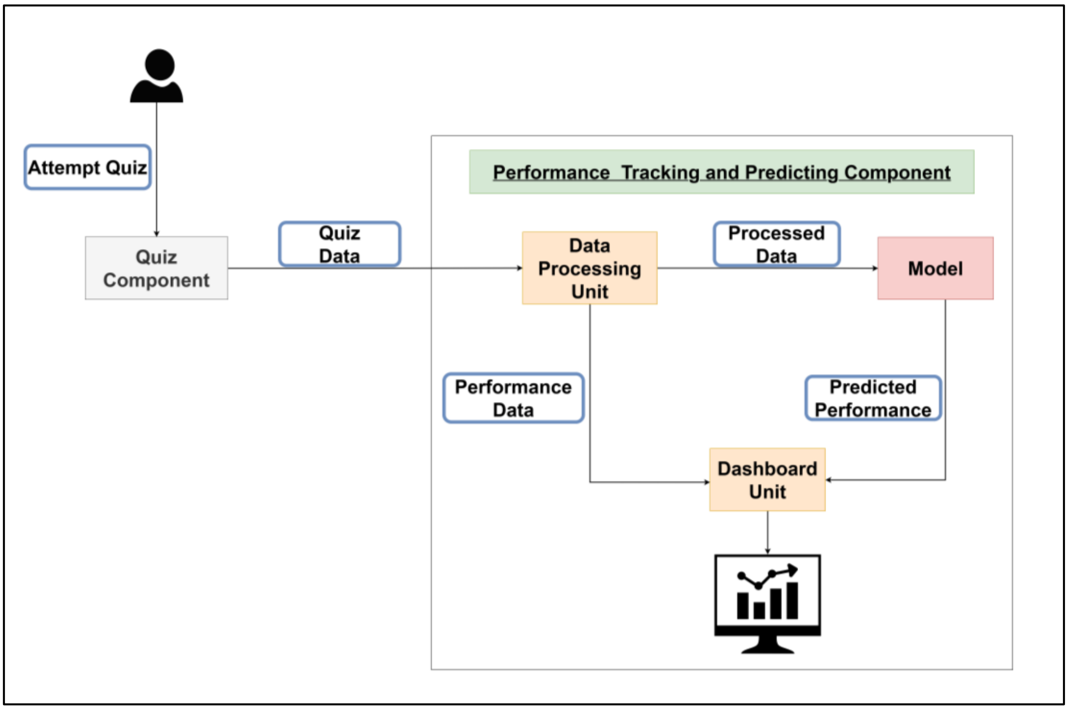


Figure 45: Overview of Performance Predicting Component.

6.4. Recommend Online Extra Study Recourses (Documents, Videos). (Member4 : IT20133368)

- Take the input query paragraph.
- Do the necessary pre-processing.
- Do the necessary calculations according to the requested resource type.
- Do the resource ranking.
- Display the extra study resources.

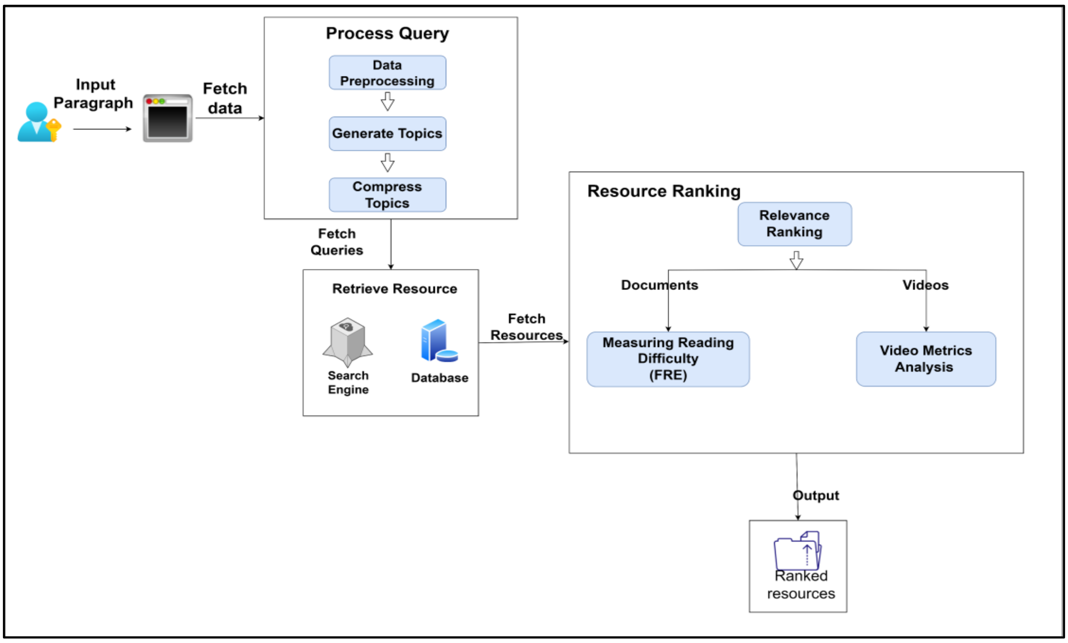


Figure 46: Overview of Resource Recommender Component.

7. CONCLUSION

This research illuminates a pressing issue faced by university students – the unfamiliarity with and limited understanding of Bloom's Taxonomy, which significantly influences their academic performance. In response, we have undertaken an exhaustive investigation and devised a personalized self-learning system, empowering students to navigate their academic journeys with confidence and success.

Our journey commenced with an initial survey among undergraduates at the Sri Lanka Institute of Information Technology (SLIIT), exposing a significant knowledge gap pertaining to Bloom's Taxonomy. This survey became the starting point for our exploration into the challenges students encounter when grasping and employing this fundamental framework.

Guided by a thorough literature review, we delved into pivotal studies spanning mind-map generation, question-and-answer generation, academic performance prediction, and online resource recommendation. These research not only improved our understanding of relevant approaches and technology, but also emphasized the importance of tailored self-study platforms.

In the sphere of mind-map generation, preceding research laid the groundwork for constructing visual depictions of study materials. Harnessing knowledge graphs, our system successfully crafted comprehensive mind maps, facilitating holistic comprehension. In the realm of question-and-answer generation, we thoughtfully integrated Bloom's Taxonomy into our system, guaranteeing that questions and answers harmonized with the taxonomy's cognitive tiers. Our research findings demonstrate that our system achieved an accuracy rate of approximately 85% in categorizing questions into the correct Bloom's taxonomy levels, as reflected by high meaningfulness and answer relevance scores. Concerning performance tracking, our system employed linear regression models for predicting student performance. By subjecting the model to rigorous validation and conducting an array of test cases, we underscored the system's proficiency in monitoring and forecasting student performance across diverse subjects. Within the purview of online resource recommendation, we conducted an extensive exploration of existing methodologies and technologies, pinpointing a research gap in the seamless integration of Bloom's Taxonomy into these recommendation systems. Such integration carries the potential to enhance the pertinence and efficacy of resource recommendations by aligning them with students' cognitive levels and educational objectives. Our research not only plugs this pivotal research gap but also furnishes a comprehensive and groundbreaking solution for university students. We created a self-study tool that allows students to understand and apply Bloom's Taxonomy in their learning journey. This platform dispenses mind maps, questions, and answers, all finely calibrated to Bloom's cognitive strata. In addition, students can track their performance and access online resources that complement their study materials through this innovative system. By amalgamating Bloom's Taxonomy, personalized content generation, performance tracking, and advanced search functionality, our self-learning system equips students with the arsenal necessary for academic excellence. It improves their comprehension, sharpens their critical thinking, and mitigates individual weaknesses. Our system not only fosters student agency in their learning journey but also paves the path to academic triumph. Ultimately, we envision a future where students not only acknowledge this critical framework but also expertly apply it to their academic pursuits, yielding enhanced academic performance and a profound grasp of their chosen subjects. This research offers a promising trajectory for refining the educational landscape and optimizing learning outcomes.

8. REFERENCES

[1] R. P. Vadhera, “Impact of Training of Teachers in Bloom’s Taxonomy on Question-Paper Setting,” International Journal of Science and Research, doi: 10.21275/SR21919043527.

[2] Vishal Pant, ShivanshBhasin, and Subhi Jain, “Self-learning system for personalized e-learning,” in International Conference on Emerging Trends in Computing and Communication Technologies (ICETCCT), Dehradun, India, Nov. 2017. doi: 10.1109/ICETCCT.2017.8280344.

[3] M. F. Nurrokhim, L. S. Riza, and Rasim, "Generating mind map from an article using machine learning," J. Phys.: Conf. Ser., vol. 1280, p. 032023, 2019. DOI: 10.1088/1742-6596/1280/3/032023.

[4] A. Saelan and A. Purwarianti, "Generating Mind Map from Indonesian Text using Natural Language Processing Tools," Procedia Technology, vol. 11, pp. 1163-1169, 2013.

[5] R. Kudelić, M. Konecki, and M. Maleković, "Mind Map Generator Software Model with Text Mining Algorithm," in Proceedings of the ITI 2011 33rd Int. Conf. on Information Technology Interfaces, Cavtat, Croatia, 2011,

[6] X. Zhou, D. Nurkowski, S. Mosbach, J. Akroyd, and M. Kraft, “Question Answering System for Chemistry,” Journal of Chemical Information and Modeling, vol. 61, no. 8, pp. 3868–3880, Aug. 2021

[7] Y. Sharma and S. Gupta, “Deep Learning Approaches for Question Answering System,” Procedia Computer Science, vol. 132, pp. 785–794, 2018,

[8] Y. T. Huang, H. W. Chang, and W. L. Chang, "An intelligent examination paper generating system based on Bloom's taxonomy and difficulty level," Journal of Computers in Education, vol. 6, no. 1, pp. 1-12, 2019.

[9] B. Albreiki, N. Zaki, and H. Alashwal, “A systematic literature review of student’ performance prediction using machine learning techniques,” *Educ Sci (Basel)*, vol. 11, no. 9, Sep. 2021, doi: 10.3390/educsci11090552.

[10] N. R. Yadav and S. S. Deshmukh, “Prediction of Student Performance Using Machine Learning Techniques: A Review,” in *International Conference on Applications of Machine Intelligence and Data Analytics*, 2022, pp. 735–741. doi: 10.2991/978-94-6463-136-4_63.

[11] M. A. Ma’sum, “Predicting Student Achievement before Final Exam: A Regression-Based Approach,” in *2022 2nd International Conference on Information Technology and Education (ICIT&E)*, 2022.

[12] C. Uthej and L. C. K, “A Machine Learning Model to Predict Student Performance Rate,” *Iconic Research and Engineering Journals*, vol. 6, no. 10, 2023.

[13] M. A. Gotardo, "Using Decision Tree Algorithm to Predict Student Performance," *Indian J Sci Technol*, vol. 12, no. 8, pp. 1–8, Feb. 2019, doi: 10.17485/ijst/2019/v12i5/140987.

[14] A. Alhassan, B. Zafar, and A. Mueen, "Predict Students' Academic Performance based on their Assessment Grades and Online Activity Data," 2020. [Online]. Available: www.ijacsa.thesai.org

[15] M. Yağcı, "Educational data mining: prediction of students' academic performance using machine learning algorithms," *Smart Learning Environments*, vol. 9, no. 1, Dec. 2022, doi: 10.1186/s40561-022-00192-z.

[16] G. N. R. Prasad, "Evaluating student performance based on bloom's taxonomy levels," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Mar. 2021. doi: 10.1088/1742-6596/1797/1/012063.

[17] X. Wei, S. Sun, D. Wu, and L. Zhou, "Personalized Online Learning Resource Recommendation Based on Artificial Intelligence and Educational Psychology," Front Psychol, vol. 12, Dec. 2021, doi: 10.3389/fpsyg.2021.767837.

[18] H. Sun, L. Huang, Z. Chen, and Q. Zheng, "Research on collaborative filtering recommendation method of online learning resources based on learner model," in 2022 IEEE 5th International Conference on Information Systems and Computer Aided Education, ICISCAE 2022, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 202–206. doi: 10.1109/ICISCAE55891.2022.9927652.

[19] D. Shen, "Recommendation of Online Educational Resources Based on Neural Network," in Proceedings - 2020 International Conference on Artificial Intelligence and Computer Engineering, ICAICE 2020, Institute of Electrical and Electronics Engineers Inc., Oct. 2020, pp. 32–36. doi: 10.1109/ICAICE51518.2020.00012.

[20] "Exponentially Weighted Moving Average (EWMA) - Formula, Applications." Accessed: Sep. 25, 2023. [Online]. Available: https://corporatefinanceinstitute.com/resources/career-map/sell-side/capital-markets/exponentially-weighted-moving-average-ewma/

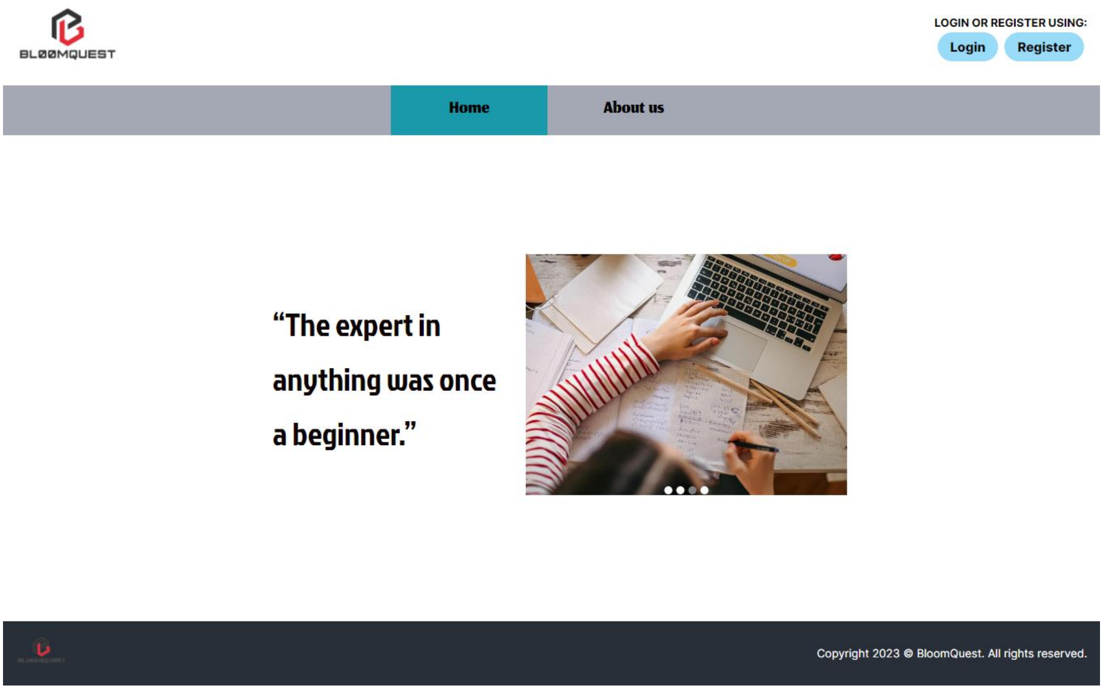
[21] P. Schober and L. A. Schwarte, "Correlation coefficients: Appropriate use and interpretation," Anesth Analg, vol. 126, no. 5, pp. 1763–1768, May 2018, doi: 10.1213/ANE.0000000000002864.

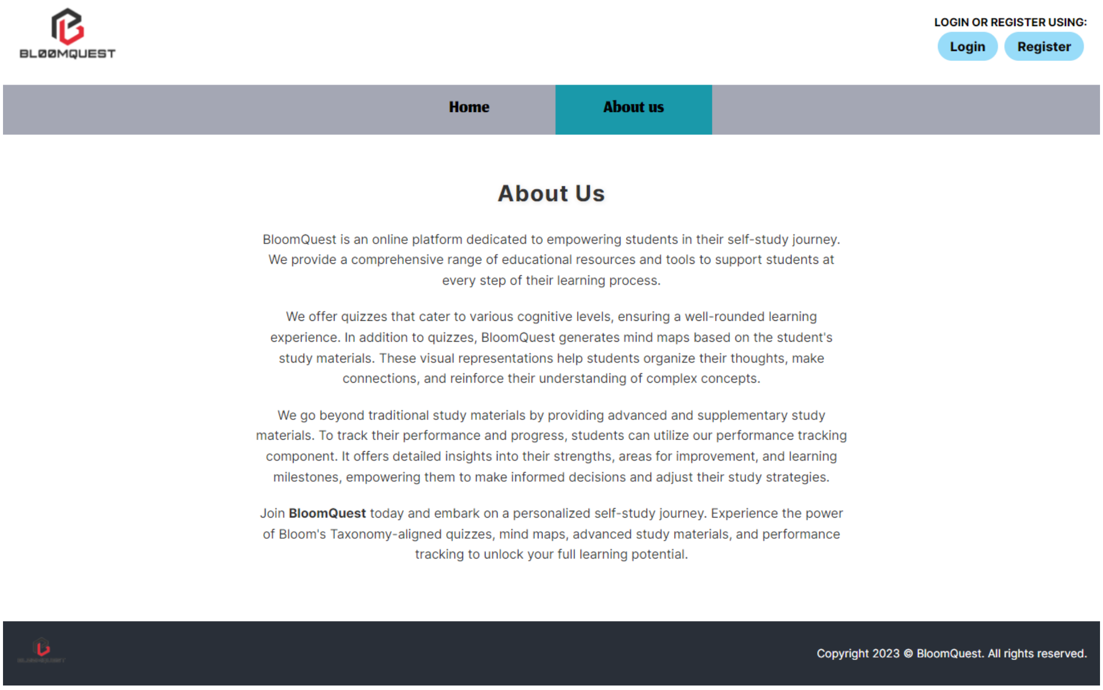
[22] Anantharaman Aditya, Jadiya Arpit, Sai Siri Chandana Tulasi, Adikar Bharath NVS, and Biju Mohan, "Performance Evaluation of Topic Modeling Algorithms for Text Classification," 2019.

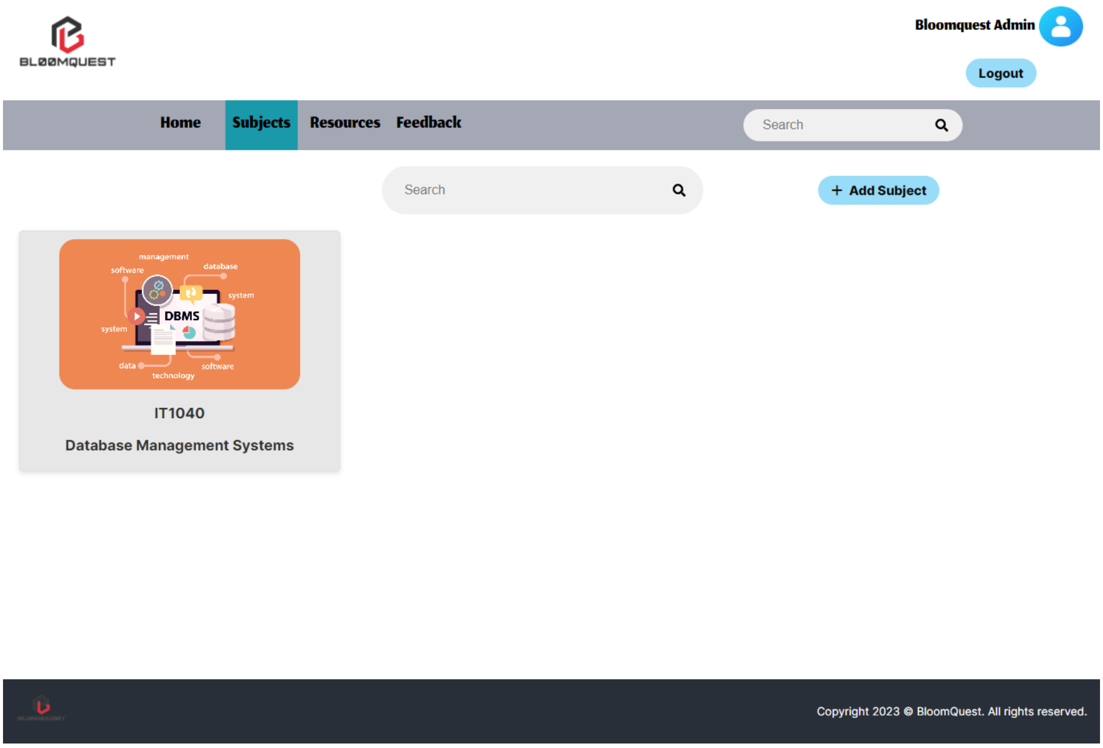
[23] L. Zahrotun, "Comparison Jaccard similarity, Cosine Similarity and Combined Both of the Data Clustering With Shared Nearest Neighbor Method," Computer Engineering and Applications, vol. 5, no. 1, 2016.

9. APPENDICES

Appendix A: BloomQuest Web Application






BLOOMQUEST
Bloomquest Admin
Logout
Home
Subjects
Resources
Feedback
Search
Search
Add Subject
DBMS
IT1040
Database Management Systems
Copyright 2023 © BloomQuest. All rights reserved.