BlOOMQUEST: A PERSONALIZED LEARNING PLATFORM

Senaweera T.I.S.

(IT20123468)

BSc (Hons) in Information Technology
Specializing in Software Engineering

Department of Computer Science and Software Engineering

Sri Lanka Institute of Information Technology
Sri Lanka

February 2023

BlOOMQUEST: A PERSONALIZED LEARNING PLATFORM

Senaweera T.I.S.

(IT20123468)

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

Department of Computer Science and Software Engineering

Sri Lanka Institute of Information Technology
Sri Lanka

February 2023

Declaration

“I declare that this is my own work, and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to 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. I retain the right to use this content in whole or part in future works (such as articles or books)”.

10.09.2023

Signature: Date:

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

Signature of the supervisor: Date

Abstract

In today's competitive environment, engaging in self-studies is important for students, particularly for university students who want to get better grades. This research was geared towards providing university students with a method to obtain insights into their current academic performance level. Drawing inspiration from Bloom's Taxonomy levels, a predictive model was formulated. Beyond mere performance prediction, a primary goal was to create a dashboard where students could visualize their progress in order to increase motivation.

The empirical data underpinning this study was collected from three quizzes along with mid-exam and final exam results related to the database management system module. Each question from the quizzes and mid-exam was meticulously mapped to Bloom's Taxonomy levels, ensuring a solid foundation for the model's predictions.

The research emphasizes that by leveraging Bloom's Taxonomy for performance predicting and integrating it with a visual dashboard, students can achieve a deeper understanding of their academic performance level, thereby promoting self-recognition and active learning.

Keywords: Student performance, Self-studying, Machine learning, Bloom’s taxonomy, Performance predicting

Acknowledgement

Embarking on this research journey, I found myself surrounded by an immense pool of support, wisdom, and encouragement. It is with profound gratitude that I acknowledge those who played a pivotal role in shaping my thesis.

First and foremost, I express my deepest appreciation to my supervisor, Mr. Prasanna Sumathipala, whose unyielding guidance, patience, and expertise steered me through the complexities of my research. Your insightful feedback was the cornerstone of my journey. To my co-supervisor, Mr. Sathira Hettiarachchi, for all the invaluable advice.

To all the members of my research group, I extend my heartfelt gratitude. Your camaraderie, discussions, brainstorming sessions, and collaborative spirit made the intricate research process an enriching and enjoyable one.

I am also deeply indebted to the Research Panel for their time, expertise, and invaluable feedback. Your keen observations and suggestions have immensely contributed to refining my research work.

Lastly, to my beloved family, your unwavering love and belief in me served as my pillar of strength. A special mention to my brother, Oshada Senaweera – your assistance, encouragement, and constant support provided me with the determination to see this journey through. It is to all of you that I dedicate this work, for it would not have been possible without your boundless support.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Abbreviation	Description
EWMA	Exponentially Weighted Moving Average

1. INTRODUCTION

Self-studying is essential for university students to achieve academic success while also managing their personal lives, jobs, and extracurricular activities. Many students do self-studies for hours and hours going through their study materials. Despite this, some students may still get lower grades for subjects than they anticipated. "BloomQuest," an innovative web-based tool designed exclusively for students dedicated to self-directed learning was created to help students with efficient and productive self-studying. One of the most significant obstacles to self-study is the lack of formal direction, which BloomQuest strives to fill.

At its core, BloomQuest is not just a tool, but an ecosystem designed to create a holistic learning environment. It offers an array of features, including:

1. Generating detailed mind-maps.
2. Formulating questions aligned with Bloom's taxonomy levels with answers.
3. Tracking and predicting academic performance.
4. Suggesting extra study resources.

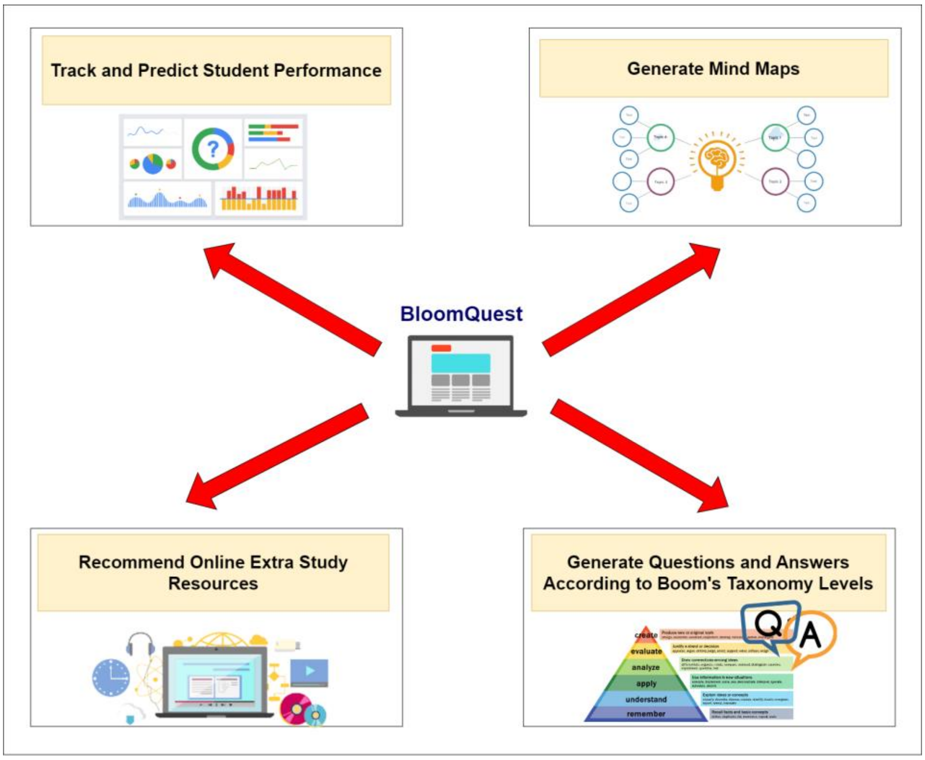


Figure 1: BloomQuest overall system.

Figure 1 depicts the overall overview of the BloomQuest system. While BloomQuest is undeniably multifunctional, this thesis will primarily focus on its third feature: the methodology and efficacy behind how students can track and predict their academic performance.

This study ventures into the potential of Bloom's taxonomy [1] as not just a tool for curriculum design or assessment framing, but as a predictor of student performance. Utilizing a uniquely curated dataset, the study delves into scores derived from assessments anchored in Bloom's taxonomy levels. The primary objective is to ascertain the efficacy of this educational framework as a nuanced and reliable predictor of student outcomes. Through this exploration, it is aimed to contribute valuable insights to the academic landscape and pave the way for more informed evaluative methodologies.

1.1. Background Literature

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.

A. Machine Learning Approaches in Predicting 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 [2], [3]. Ma'sum [4], 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 [5] 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 [6]. The investigations by [7] and [8] are particularly noteworthy, as they synergized assessment scores with variables like student engagement, departmental links, and faculty information to refine their predictions.

B. The Role of Bloom's Taxonomy in Educational Assessment.

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 [9], have emphasized the value of Bloom's taxonomy in predicting student performance.

1.2. Research Gap

According to the above literature review, predicting students' performance has been a topic of great interest and extensive research in the field of education. Many researchers have invested their time and effort in exploring various factors that can affect students' academic performance. It has been widely acknowledged that predicting students' performance can provide valuable insights to educators and policymakers. By identifying at-risk students and providing them with additional support, dropout rates can be reduced, and academic achievement can be enhanced. Thus, researchers continue to investigate and develop effective methods for predicting students' academic performance.

However, the scope of prior research efforts primarily remained centered on aggregated performance metrics without catering to the individual needs of the students. The detailed comparison provided in Table 1 underscores this observation, which signifies a glaring gap in the literature.

Table 1: Summary of research gap.

Characteristic	Paper [4]	Paper [5]	Paper [6]	Paper [7]	Paper [8]	Paper [9]	This System
Use of Machine Learning Approach	✓	✓	✓	✓	✓	X	✓
Use of Bloom’s Taxonomy	X	X	X	X	X	✓	✓
Predict Student Performance	✓	✓	✓	✓	✓	✓	✓
Visualization of Performance	X	X	X	X	X	X	✓
Synthesis of Bloom's Taxonomy with Predictive Algorithms	X	X	X	X	X	X	✓

Most notably, the majority of the researched papers did not focus on identifying students' learning levels using Bloom’s taxonomy. Bloom's taxonomy serves as a pivotal framework in educational settings, offering structured categorization of learning objectives. The omission of this categorization in most research underscores a missed opportunity for granularity and specificity in understanding and predicting student performance.

Moreover, the apparent lack of studies providing a visual representation of student performance is striking. Visualization tools are instrumental in making data interpretable and actionable. In the context of student performance, such tools can help educators identify trends and outliers, facilitating timely interventions.

This observed gap highlights a tremendous potential: a combination of cognitive understanding, as exemplified by Bloom's Taxonomy, with the algorithmic prowess of machine learning. The current study navigates this uncharted intersection. It suggests that combining Bloom's Taxonomy with predictive models could reveal deeper insights into student performance, paving the way for a paradigm shift in educational research.

1.3. Research Problem

Self-studying is essential when it comes to achieving good grades. It is a learning style in which students take care of their own studies outside of the classroom and without direct supervision. Students are expected to take responsibility for their education by actively seeking knowledge outside of the classroom, especially when they study at university. It can help students to be more independent and self-reliant, which is an important skill to have in the professional world. However, self-studying can be difficult for many university students, especially for those who have trouble managing their time or staying focused. These students require personal assistance in order to be motivated to overcome difficulties and do self-study. This raises the research question of "How can university students assess their academic performance level while doing self-studies?"

One potential solution to this issue is to develop a method to predict student performance utilizing Bloom's Taxonomy levels. Bloom's Taxonomy is a widely used framework, especially in universities for categorizing educational goals into six cognitive complexity levels: Remembering, Understanding, Applying, Analyzing, Evaluating, and Creating. This system tracks student progress across each level of Bloom's taxonomy, providing students with a better understanding of their strengths and weaknesses in different areas of cognitive development. This will enable them to focus their efforts on areas where they need improvement.

In summary, the proposed system will address the gap in personalized learning experiences by providing students with a tool to track their progress, predict their performance, and visualize their data in a clear and understandable format. Thereby giving students a more personalized and effective learning experience.

1.4. Research Objective

1.4.1. Primary objective

To provide a personalized self-study system for undergraduate students that can process student-provided study materials and assist them with their academic performance using the concept of Bloom's Taxonomy levels.

1.4.2. Sub objectives

1. Mind-map Creation: Develop an automated system within BloomQuest that generates comprehensive mind-maps from student-provided study materials to provide a structured view of the content.
2. Bloom’s Taxonomy-based Q&A Generation: Create a feature that formulates questions in accordance with Bloom's taxonomy levels and also provides answers, ensuring a broad spectrum of cognitive engagement.
3. Performance Analysis & Prediction: Integrate a performance monitoring system that not only tracks the current academic performance of the students but also employs a predictive model to help them assess their academic performance level.
4. Resource Recommendation: Implement a system that suggests supplementary study resources relevant to the uploaded study materials, enhancing students' understanding and grip on the subject.

1.4.3. Specific objectives

To develop a component within the BloomQuest platform that can track and predict undergraduate students' current academic performance according to Bloom's Taxonomy levels. And visualize those data in a visually appealing way fostering enhanced self-awareness and motivation.

1. Data Collection for Model Training: Data collection from students' assessments to train the predictive model, ensuring its accuracy and relevance.
2. Model Development for Performance Prediction: Design and develop a predictive model tailored to evaluate students' current academic performance based on Bloom's Taxonomy levels.
3. Dashboard Visualization for Performance Metrics: Construct a user-friendly dashboard that dynamically displays the predictive analytics results derived from the model. This dashboard should not only represent a student's performance metrics but also highlight their standing according to Bloom's Taxonomy levels.
4. Motivation Enhancement via Data Visualization: Ensure that the design and representation of data in the dashboard are intuitive and motivating. Visualization should serve as a catalyst, encouraging students to recognize areas of improvement and take proactive measures to enhance their learning outcomes.

2. METHODOLOGY

2.1. Performance Predicting Model

2.1.1. 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.

2.1.2. 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 carried out, 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 dataset 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 [10].

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

2.1.3. 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 2, 3, 4, and 5 provided below illustrate the scatterplots for dataset 1, which contains the final marks for each assessment.

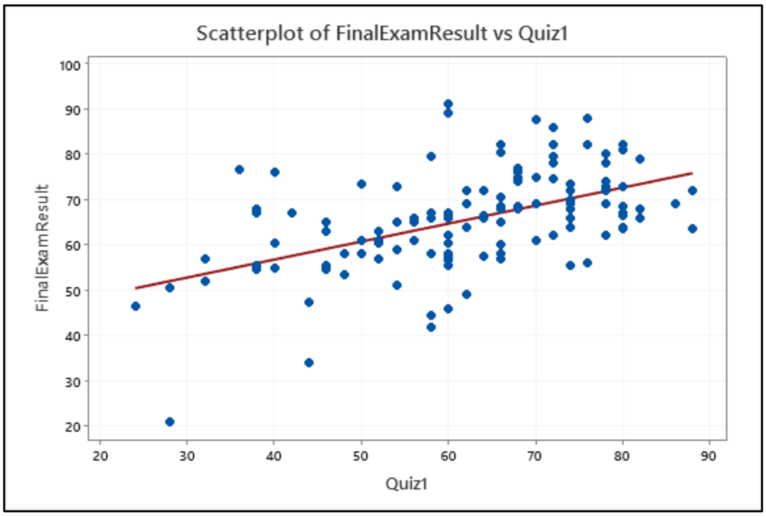


Figure 2: Scatterplot of final exam vs quiz 1.

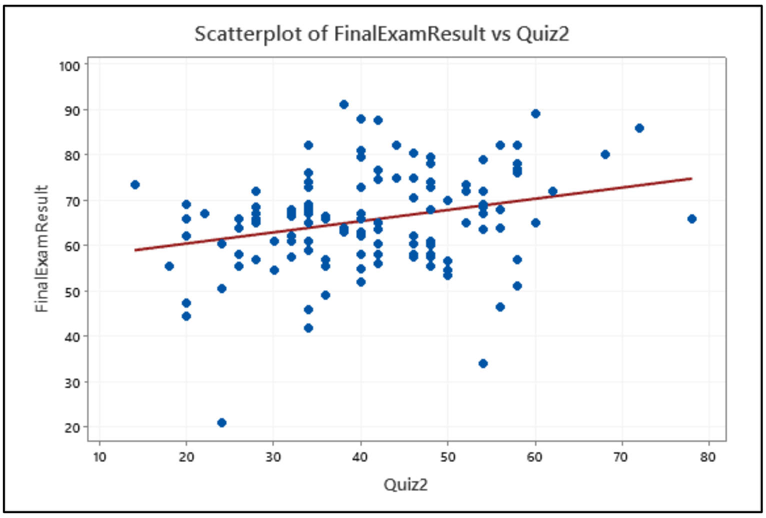


Figure 3: Scatterplot of final exam vs quiz 2.

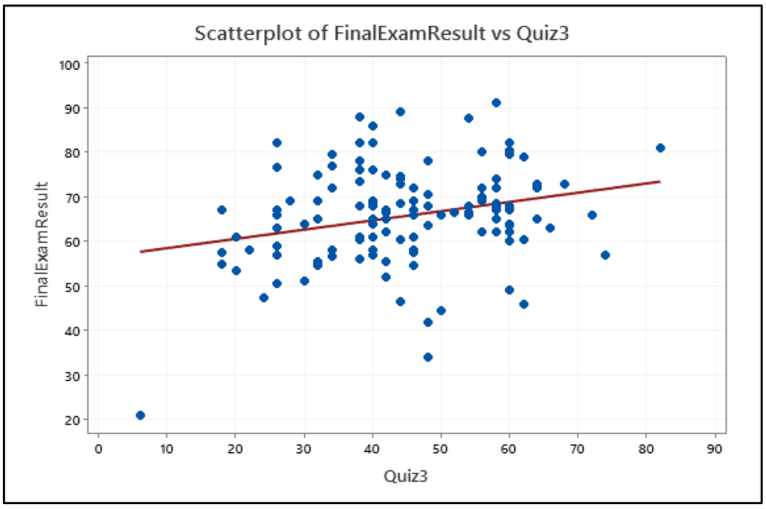


Figure 4: Scatterplot of final exam vs quiz 3.

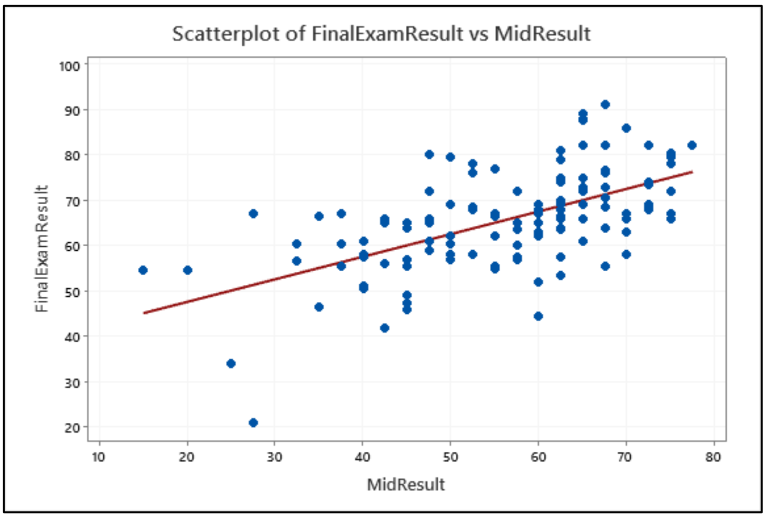


Figure 5: Scatterplot of final exam vs mid-exam.

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 isn't 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.

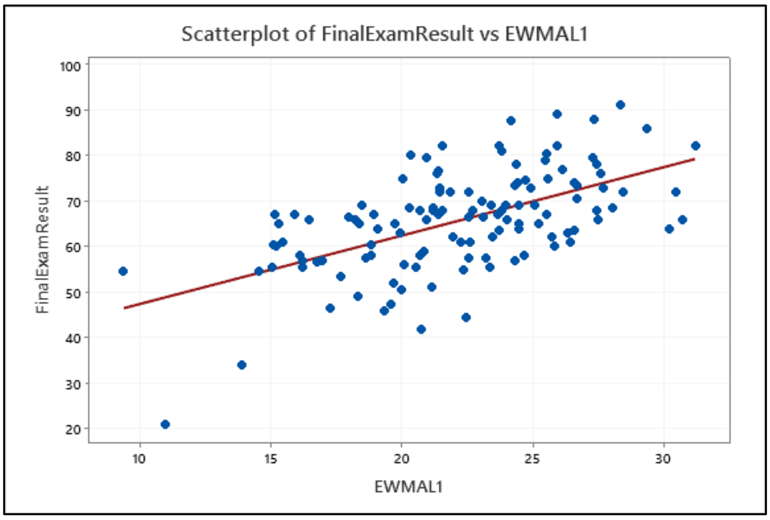


Figure 6: Scatterplot of final exam vs bloom's level L1.

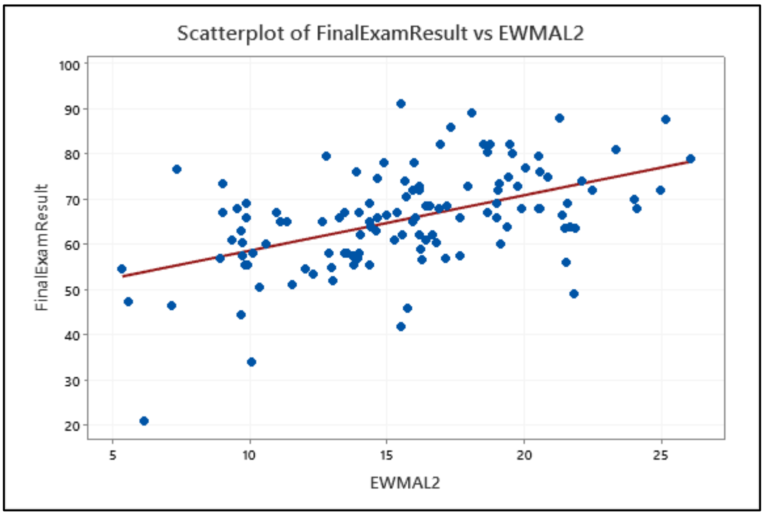


Figure 7: Scatterplot of final exam vs bloom's level L2.

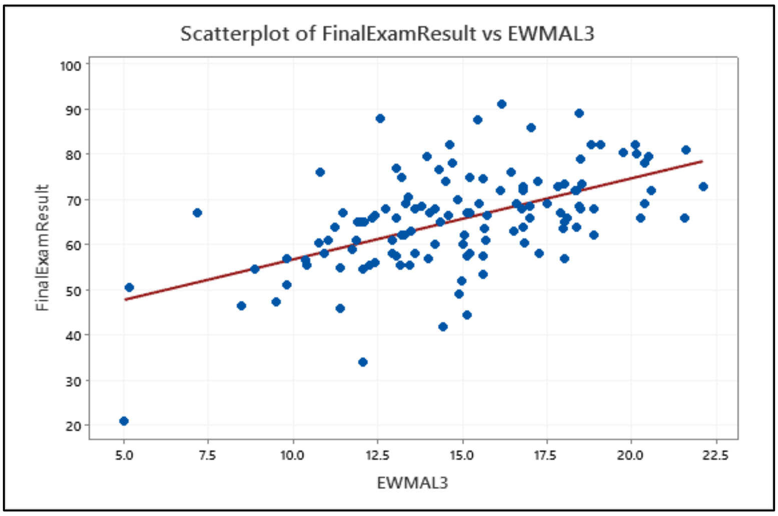


Figure 8: Scatterplot of final exam vs bloom's level L3.

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 [11] 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.

2.1.4. 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 don't 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**

2.2. 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, tested, and maintained, leading to more straightforward debugging, easier updates, and enhanced scalability.

Also, data security and privacy are an utmost important 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.

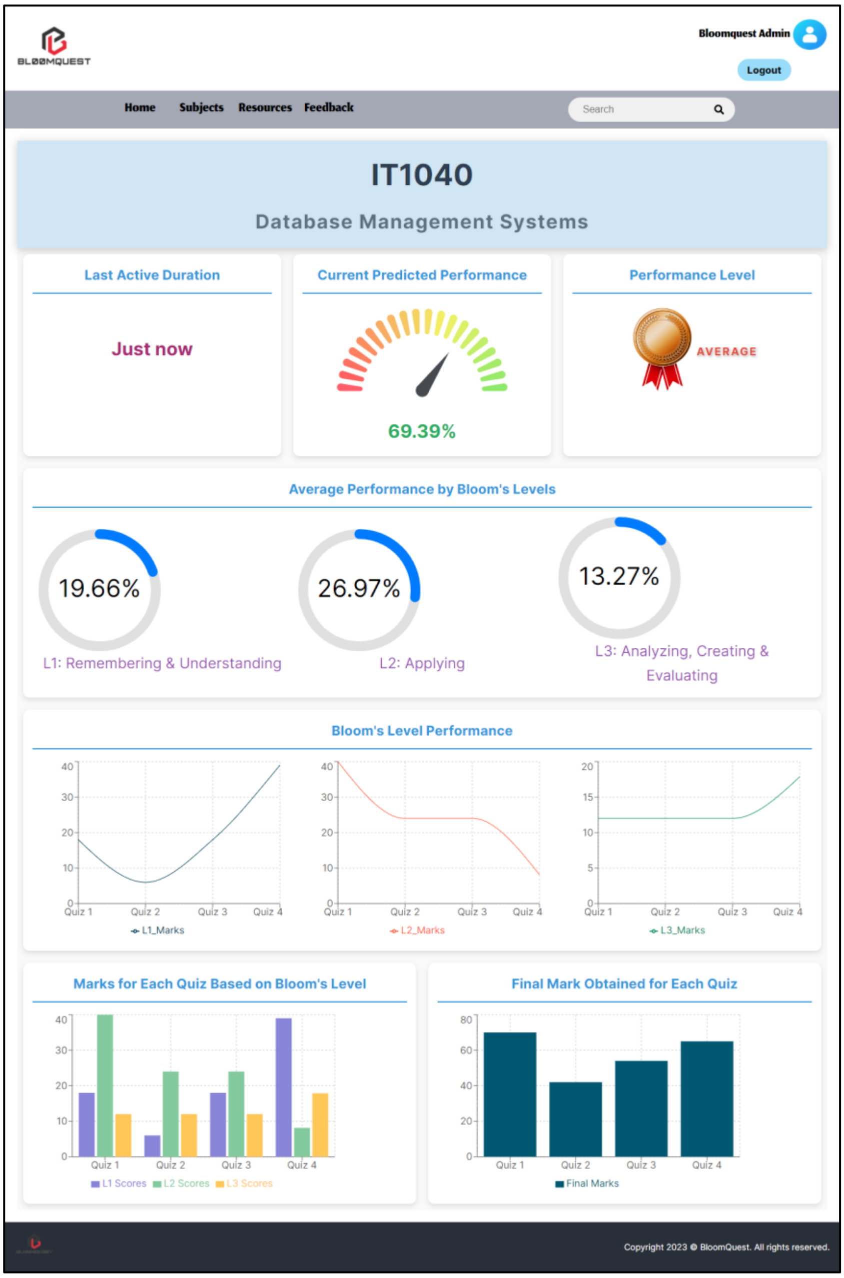


Figure 9: Dashboard design.

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.

2.3. 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.

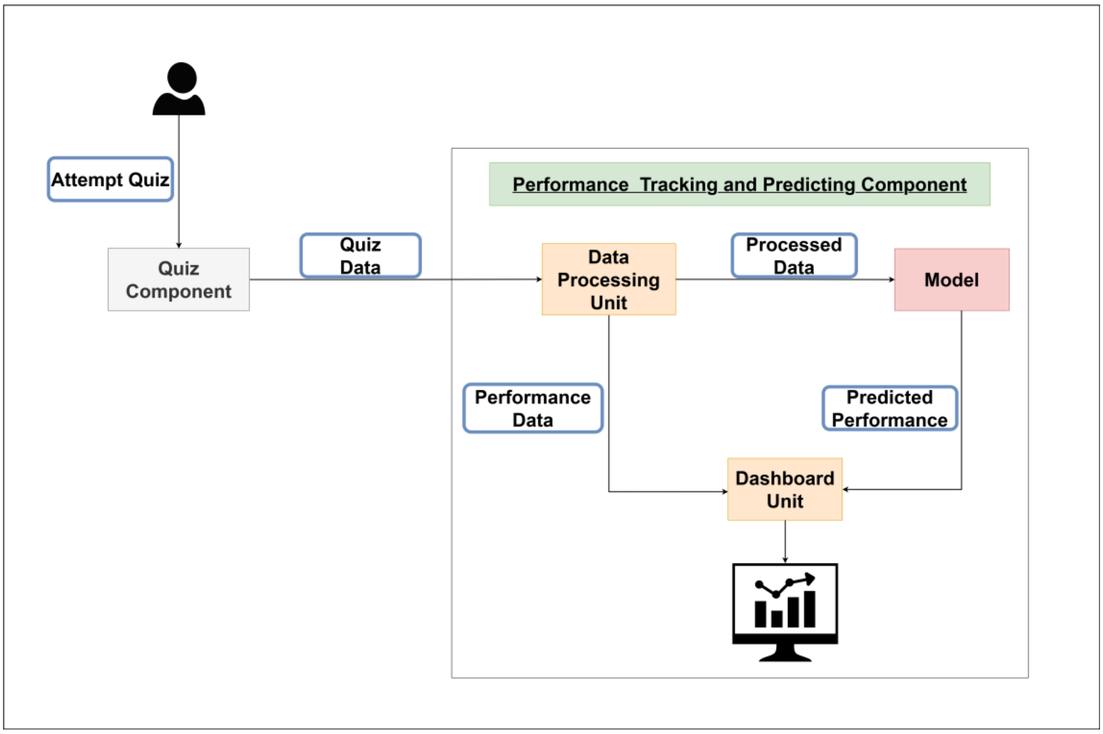


Figure 10: Component overview diagram.

2.4. Non-Functional Requirements

Specifications that define the operational capabilities, limitations of a system and the attempt to improve its functionality are the Non-functional requirements. Below are some non-functional requirements to be focused on when developing the proposed system,

1. Availability – Ability to use the system at a given time.

2. User Experience: Designed in a way that is visually appealing and engaging, while also being easy to interpret.

3. Usability: Easy to use and navigate, with a clear and intuitive interface that allows users to access the information they need quickly and easily.

4. Performance Standards: Meet specific performance standards for response time, throughput, and other metrics to ensure that it is delivering reliable and consistent results.

5. Security – Safeguards the user information and ensures the networks and resources are safe from malicious intrusion.

2.5. Commercialization Aspect of the Product

In today's digitized and data-driven educational ecosystem, tracking student performance and predicting future outcomes have become pivotal. While the tracking and predicting component was initially developed as an integral part of the BloomQuest system, its potential applications are far more expansive.

1. Universality of Bloom's Taxonomy:

Bloom’s Taxonomy serves as the bedrock of many educational systems across the globe. It provides a universally recognized framework that classifies cognitive skills, from basic knowledge recall to complex problem-solving and creativity. Given this widespread acceptance and application, our component, built upon the principles of Bloom's Taxonomy, has a natural fit across a myriad of educational settings and subjects.

2. Modular Design for Integration:

The component's modular architecture ensures seamless integration into other systems. Whether it's an advanced Learning Management System (LMS) or an e-learning platform, the component can be plugged in to provide tracking and prediction functionalities. The modular design also ensures that updates or modifications can be carried out without disturbing the host system's operations.

3. Scalability for Diverse Educational Levels:

From elementary schooling to higher education, the foundational concepts of Bloom's Taxonomy remain consistent. As such, this component can scale across various educational levels, catering to the nuanced requirements of different age groups and complexity levels.

4. Customization and Personalization:

While the component offers a generic framework for tracking and predicting based on Bloom's Taxonomy, it also retains the flexibility for customization. Institutions or educational platforms can adapt and tailor the component to align with their unique curriculum, grading system, or teaching methodology.

5. Data-Driven Insights for Stakeholders:

Beyond students, the component provides a treasure trove of insights for educators, institutions, and policymakers. By analyzing performance metrics and prediction outcomes, educators can refine their teaching strategies, institutions can optimize curriculum design, and policymakers can make informed decisions to enhance educational quality.

In conclusion, the commercial prospects for this tracking and predicting component are robust and varied. Its foundation on the universal principles of Bloom's Taxonomy, combined with its modularity, scalability, and adaptability, makes it an attractive proposition for a broad spectrum of educational environments. As educational systems globally lean into data-driven decision-making, this component stands poised to be an indispensable tool in that transformative journey.

2.6. Testing

Having a good test plan is essential when developing a component that predicts student performance. This is because the accuracy and reliability of the predictions made by the component have a significant impact on the effectiveness of the overall system. It provides assurance to users that the system is dependable and trustworthy, which is essential in any educational setting.

Test Cases:

1. 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.

2. 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.

3. 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.

4. 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.

5. 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%.

3. RESULTS AND DISCUSSION

3.1. Results

3.1.1. 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.

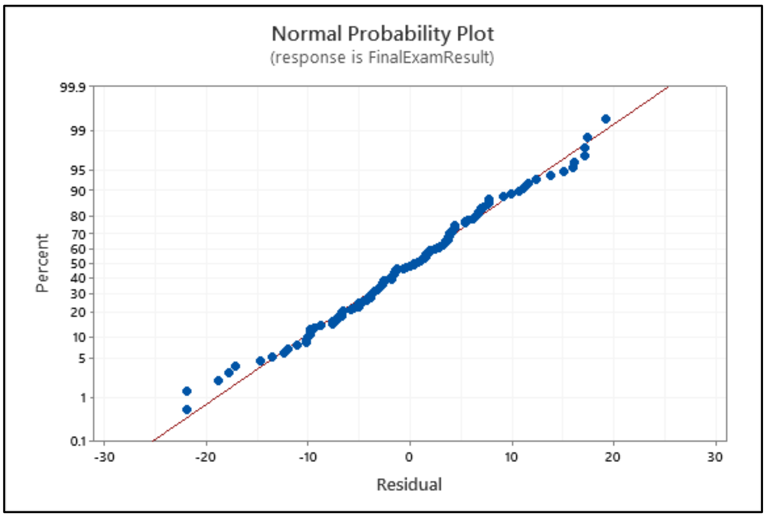


Figure 11: Normal distribution plot.

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.

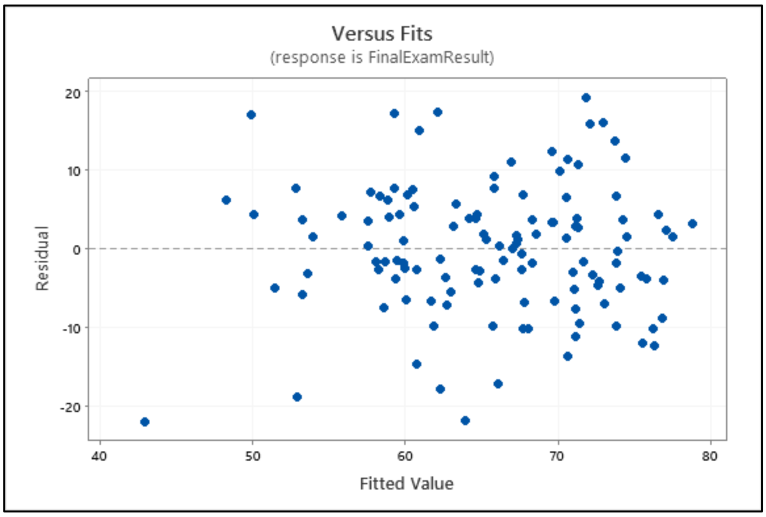


Figure 12: Residual variance plot.

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.

3.1.2. Test cases validation

1. Test case 1 : Data Ingestion

Table 6: 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%

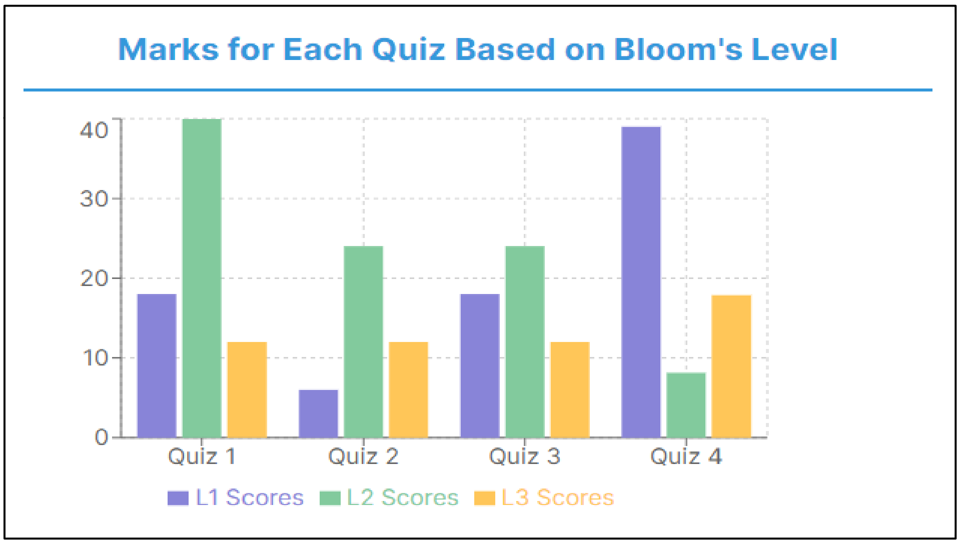


Figure 13: Test case 1 results shown in dashboard.

Test case 2 : EWMA Calculation

Table 7: 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%

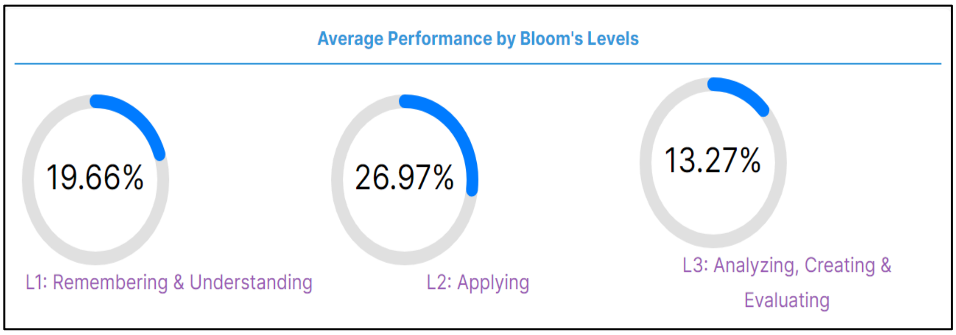
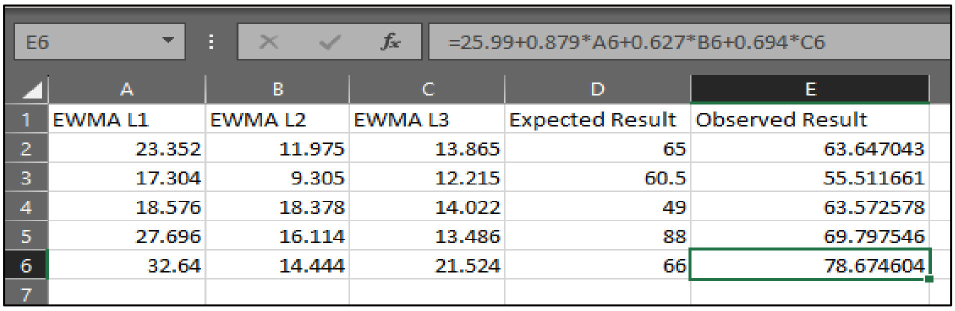


Figure 14: Test case 2 results shown in dashboard.

3. Test case 3 : Performance Prediction

Table 8: 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
7					

Figure 15: Test case 3 results calculated in excel.

4. Test case 4 : Visualization Interface

Table 9: 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 10: Test case 5 results.

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

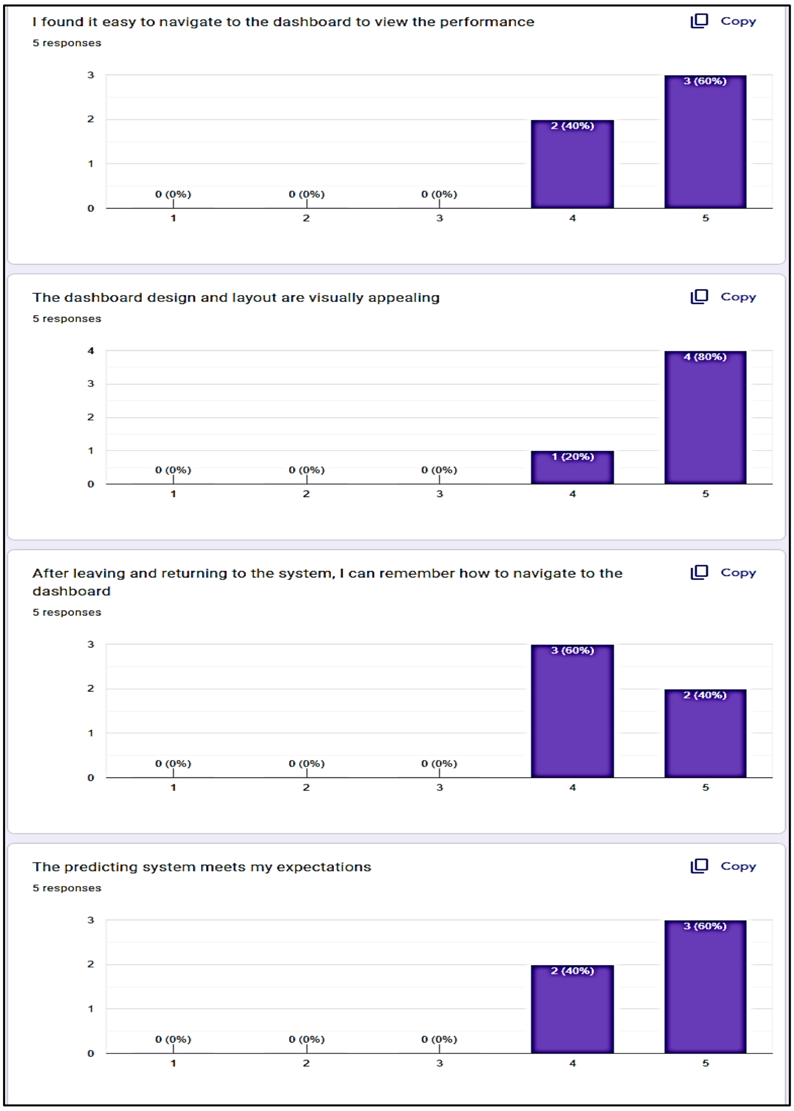


Figure 16: Test case 5 results user feedback.

3.2. Research Findings and Limitations

3.2.1. Research findings

- Linear Regression Suitability

Linear regression was found to be a suitable model for predicting student performance based on the linear relationships identified between EWMAs and final exam results. The derived regression equation is:

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

- Model Assumptions

The linear regression model met its primary assumptions of normality and homoscedasticity of residuals as evidenced by the Q-Q plot and residual variance plot (Figures 9 and 10).

- Test Case Results

Data ingestion worked seamlessly, with data being stored without discrepancies.

EWMA calculations consistently matched expected results, indicating the robustness and reliability of the model.

The model's performance prediction showed variances but remained within satisfactory ranges for most test cases.

Visualization tools provided clear and accurate visual representations of the performance data.

The user-interface evaluations resulted in a high satisfaction rating, indicating a profoundly user-friendly system layout.

3.2.2. Limitations and future work

- Dataset Constraints

One of the study's main limitations was the dataset's size. Being derived from a real-world context, it was relatively small. This constraint limited the use of potentially more sophisticated models, such as decision tree regressions or random forest regressions, due to insufficient data for robust training and testing splits.

- Future Work

With the acquisition of more data, future studies can venture beyond linear regression. By doing so, predictions can be enhanced, harnessing the potential of more intricate algorithms.

3.3. Discussion

The primary aim of this research was to develop a method for students to track and predict their performance. This prediction employed a linear regression model, integrating machine learning techniques with Bloom’s taxonomy. Although the model showed potential, the differences between expected and actual outcomes indicate areas for improvement. Such differences might be linked to factors not considered within the model, emphasizing that academic performance is a complex combination of many aspects.

The 100% satisfaction rate from the user-interface test illuminates the system's efficacy in offering an intuitive experience. Such positive feedback accentuates the importance of user-centered design in developing systems, ensuring usability and efficiency.

Overall, while the linear regression model demonstrated promise, the limitations of the dataset impede the exploration of potentially superior algorithms. Future research, backed by more extensive data, can uncover the potentials of more complex models, driving the predictive accuracy further.

4. SUMMARY OF EACH STUDENT'S CONTRIBUTION

4.1. 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.

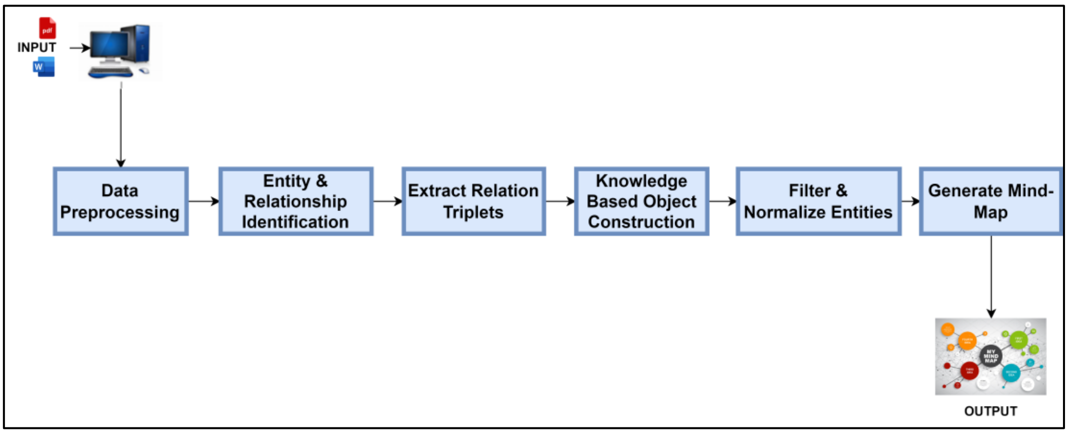


Figure 17: Overview of mind-map component.

4.2. 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.

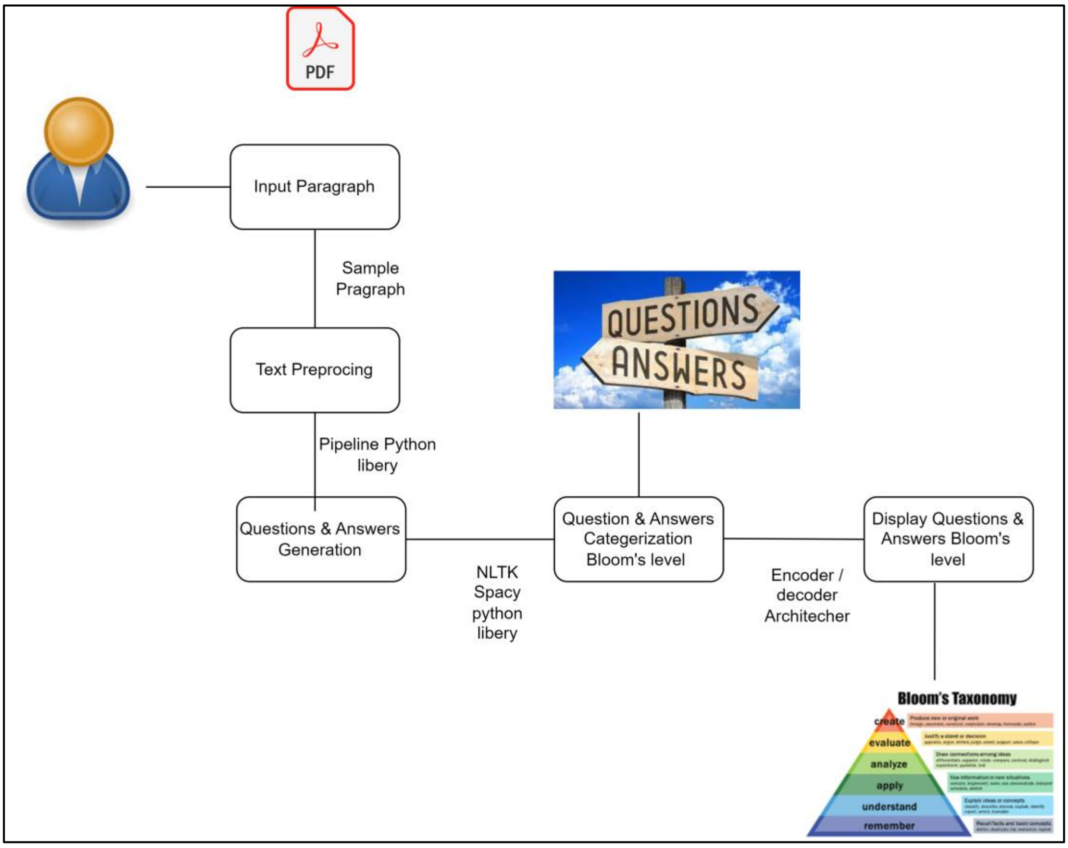


Figure 18: Overview of quiz component.

4.3. Track And Predict Student Performance. (Member3: IT20123468 - My Self)

- 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.

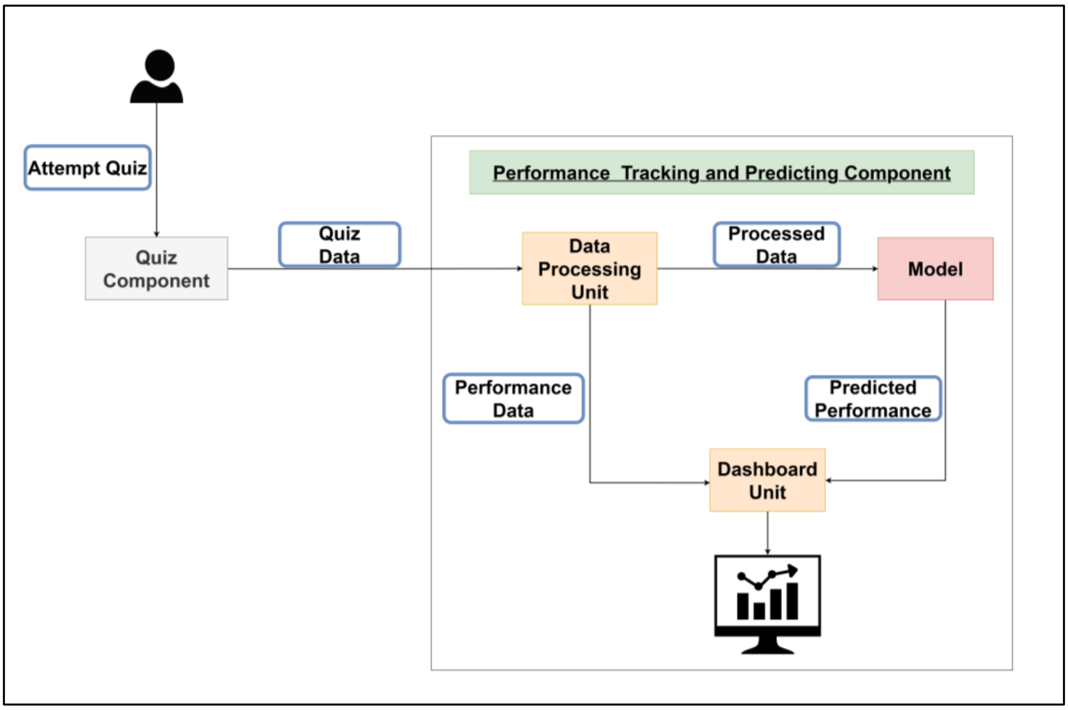


Figure 19: Overview of performance predicting component.

4.4. Recommend 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.

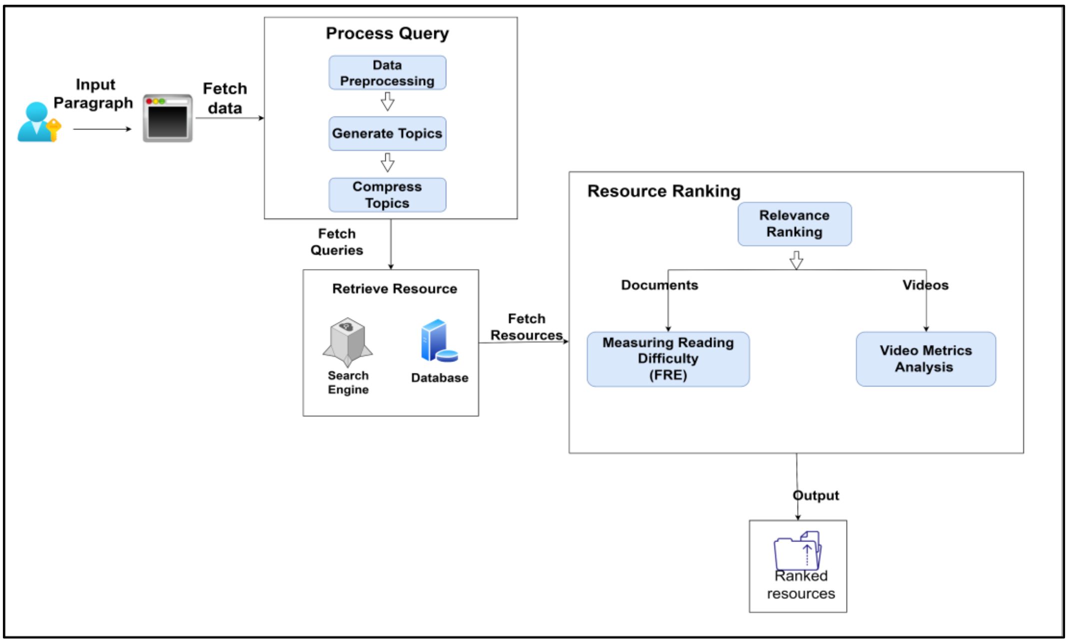


Figure 20: Overview of resource recommender component.

5. CONCLUSION

The pursuit of a predictive tool to assist students in recognizing and addressing their academic needs is an endeavor of significant importance in the educational landscape. The component developed as part of the BloomQuest system not only achieves this but surpasses traditional methods by integrating the nuanced layers of Bloom's Taxonomy. Through the integration of this taxonomy, our approach offers a granular view of a student's academic journey, pinpointing areas of strength and areas that require further attention.

Unlike the existing systems, which are primarily educator-focused, our component provides a student-centric approach. This fundamental shift places students at the helm of their academic voyage, empowering them to take charge of their learning outcomes. The visualization feature, in particular, acts as a beacon, guiding students through their areas of improvement and showcasing their progress.

However, the true strength of the system lies in its versatility. While designed for the BloomQuest system, its foundational principles allow for its integration into various other platforms where student assessments are accessible. This adaptability signifies the potential for widespread implementation and the consequent enhancement of student learning experiences across different educational systems.

In summary, the development and integration of this component underscores the potential of blending educational theory with modern predictive technology. As education evolves in the digital age, tools like these will undoubtedly be pivotal in ensuring that students aren't just learners but active participants in shaping their academic futures.

6. REFERENCES

[1] D. R. Krathwohl, "A Revision of Bloom's Taxonomy: An Overview," *Theory into Practice*, vol. 41, 2002.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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

[8] 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.

[9] 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.

[10] "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/

[11] 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.

7. APPENDICES

Appendix A: Frontend Dashboard Code.

```
/* eslint-disable react-hooks/exhaustive-deps */
import React, {
  useCallback,
  useEffect,
  useMemo,
  useState,
  Suspense,
} from "react";
import axios from "axios";
import { useLocation, useParams } from "react-router-dom";
import LastActivePanel from "./panels/LastActivePanel";
import CurrentPerformancePanel from "./panels/CurrentPerformancePanel";
import PerformanceBadgePanel from "./panels/PerformanceBadgePanel";
import AveragePerformancePanel from "./panels/AveragePerformancePanel";
import ScoresByLevelPanel from "./panels/ScoresByLevelPanel";
import LastSevenQuizzesPanel from "./panels/LastSevenQuizzesPanel";
import PerformanceBadgeSkeleton from
"./panels/PerformanceBadgeSkeleton";
import EWMAPerformancePanel from "./panels/EWMAPerformancePanel";
import "./Dashboard.css";

function Dashboard() {
  const { subjectId } = useParams();
  const { subCode = "", subName = "" } = useLocation().state || {};
  const [lastActive, setLastActive] = useState("");
  const [quizDetails, setQuizDetails] = useState([]);
  const [EWMAL1, setEWMAL1] = useState(0);
  const [EWMAL2, setEWMAL2] = useState(0);
  const [EWMAL3, setEWMAL3] = useState(0);
  const [currentPerformance, setCurrentPerformance] = useState(0);
  const [error, setError] = useState(null); // Error state

  const fetchData = useCallback(async () => {
    try {
      const [lastActiveResponse, quizzesResponse, performanceResponse]
=
        await Promise.all([
          axios.get("http://localhost:5000/last-active"),
          axios.get(
            `http://localhost:5000/get-all-
quizzes?subject_id=${subjectId}`
```

```
        ),
        axios.get(
          `http://localhost:5000/performance?subject_id=${subjectId}`
        ),
      ]);

      setLastActive(lastActiveResponse?.data?.last_active_time || "");
      setQuizDetails(quizzesResponse?.data?.details || []);
      setEWMAL1(performanceResponse?.data?.EWMAL1 || 0);
      setEWMAL2(performanceResponse?.data?.EWMAL2 || 0);
      setEWMAL3(performanceResponse?.data?.EWMAL3 || 0);
      setCurrentPerformance(performanceResponse?.data?.currentPerforman
ce || 0);
    } catch (error) {
      console.error("Error fetching data:", error);
      setError(error);
    }
  }, [subjectId]);

  useEffect(() => {
    fetchData();
  }, []);

  if (error) {
    return <div>Error loading data...</div>;
  }

  return (
    <div className="dashboard-container">
      <Suspense fallback={<div>Loading...</div>}>
        <div className="subject-header">
          <h1 className="subject-name">{subCode}</h1>
          <p className="subject-code">{subName}</p>
        </div>
        <div className="row">
          <div className="column">
            <LastActivePanel data={lastActive} />
          </div>
          <div className="column">
            <CurrentPerformancePanel data={currentPerformance} />
          </div>
          <div className="column">
            {currentPerformance ? (
              <PerformanceBadgePanel data={currentPerformance} />
            ) : (
              <PerformanceBadgeSkeleton />
```

```
          )}
        </div>
      </div>
      <div className="row">
        <div className="column">
          <AveragePerformancePanel
            EWMAL1={EWMAL1}
            EWMAL2={EWMAL2}
            EWMAL3={EWMAL3}
          />
        </div>
      </div>
      <div className="row">
        <div className="column">
          <EWMAPerformancePanel
            data={quizDetails}
            title="Bloom's Level Performance"
          />
        </div>
      </div>
      <div className="row">
        <div className="column">
          <ScoresByLevelPanel data={quizDetails} />
        </div>
        <div className="column">
          <LastSevenQuizzesPanel data={quizDetails} />
        </div>
      </div>
    </Suspense>
  </div>
 );
}

export default Dashboard;
```

Appendix B: Frontend Current Performance Panel.

```
import React from "react";
import GaugeChart from "react-gauge-chart";

function CurrentPerformancePanel({ data }) {
  const percentValue = Math.min(Math.max(data / 100, 0), 1); // Clamp
the value between 0 and 1

  return (
    <div className="panel">
      <h3>Current Predicted Performance</h3>
      <div className="current_performance">
        <GaugeChart
          id="performance-gauge"
          nrOfLevels={20}
          colors={["#FF5F6D", "#FFC371", "#8DE969"]}
          arcWidth={0.3}
          percent={percentValue}
          hideText
        />
      </div>
      <div className="current_performance_value">{data}%</div>
    </div>
  );
}

export default CurrentPerformancePanel;
```

Appendix C: Frontend EWMA Performance Panel.

```
import React from "react";
import {
  LineChart,
  Line,
  XAxis,
  YAxis,
  CartesianGrid,
  Tooltip,
  Legend,
} from "recharts";

const EWMAPerformancePanel = ({ data, title }) => {
  // Extract the last 7 data points for l1_marks, l2_marks, and
l3_marks
  const l1MarksData = data
    .slice(-7)
    .map((item) => ({
      date: item.quiz_name,
      l1_marks: item.l1_marks,
    }))
    .reverse();

  const l2MarksData = data
    .slice(-7)
    .map((item) => ({
      date: item.quiz_name,
      l2_marks: item.l2_marks,
    }))
    .reverse();

  const l3MarksData = data
    .slice(-7)
    .map((item) => ({
      date: item.quiz_name,
      l3_marks: item.l3_marks,
    }))
    .reverse();

  return (
    <div className="panel">
      <h3>{title}</h3>
      <div className="blooms-panel-container">
        <div className="blooms-panel-wrapper">
          <LineChart width={400} height={300} data={l1MarksData}>
```

```
            <CartesianGrid strokeDasharray="3 3" />
            <XAxis dataKey="date" />
            <YAxis />
            <Tooltip />
            <Legend />
            <Line
              type="monotone"
              dataKey="l1_marks"
              name="L1_Marks"
              stroke="#003A51"
            />
          </LineChart>
        </div>
        <div className="blooms-panel-wrapper">
          <LineChart width={400} height={300} data={l2MarksData}>
            <CartesianGrid strokeDasharray="3 3" />
            <XAxis dataKey="date" />
            <YAxis />
            <Tooltip />
            <Legend />
            <Line
              type="monotone"
              dataKey="l2_marks"
              name="L2_Marks"
              stroke="#FF5733"
            />
          </LineChart>
        </div>
        <div className="blooms-panel-wrapper">
          <LineChart width={400} height={300} data={l3MarksData}>
            <CartesianGrid strokeDasharray="3 3" />
            <XAxis dataKey="date" />
            <YAxis />
            <Tooltip />
            <Legend />
            <Line
              type="monotone"
              dataKey="l3_marks"
              name="L3_Marks"
              stroke="#008955"
            />
          </LineChart>
        </div>
      </div>
    </div>
  );
```

```
};

export default EWMAPerformancePanel;
```

Appendix D: Frontend Bloom's Levls Performance Panel.

```
import React from "react";
import {
  LineChart,
  Line,
  XAxis,
  YAxis,
  CartesianGrid,
  Tooltip,
  Legend,
} from "recharts";

const EWMAPerformancePanel = ({ data, title }) => {
  // Extract the last 7 data points for l1_marks, l2_marks, and
l3_marks
  const l1MarksData = data
    .slice(-7)
    .map((item) => ({
      date: item.quiz_name,
      l1_marks: item.l1_marks,
    }))
    .reverse();

  const l2MarksData = data
    .slice(-7)
    .map((item) => ({
      date: item.quiz_name,
      l2_marks: item.l2_marks,
    }))
    .reverse();

  const l3MarksData = data
    .slice(-7)
    .map((item) => ({
      date: item.quiz_name,
      l3_marks: item.l3_marks,
    }))
    .reverse();

  return (
    <div className="panel">
      <h3>{title}</h3>
      <div className="blooms-panel-container">
        <div className="blooms-panel-wrapper">
          <LineChart width={400} height={300} data={l1MarksData}>
```

```
            <CartesianGrid strokeDasharray="3 3" />
            <XAxis dataKey="date" />
            <YAxis />
            <Tooltip />
            <Legend />
            <Line
              type="monotone"
              dataKey="l1_marks"
              name="L1_Marks"
              stroke="#003A51"
            />
          </LineChart>
        </div>
        <div className="blooms-panel-wrapper">
          <LineChart width={400} height={300} data={l2MarksData}>
            <CartesianGrid strokeDasharray="3 3" />
            <XAxis dataKey="date" />
            <YAxis />
            <Tooltip />
            <Legend />
            <Line
              type="monotone"
              dataKey="l2_marks"
              name="L2_Marks"
              stroke="#FF5733"
            />
          </LineChart>
        </div>
        <div className="blooms-panel-wrapper">
          <LineChart width={400} height={300} data={l3MarksData}>
            <CartesianGrid strokeDasharray="3 3" />
            <XAxis dataKey="date" />
            <YAxis />
            <Tooltip />
            <Legend />
            <Line
              type="monotone"
              dataKey="l3_marks"
              name="L3_Marks"
              stroke="#008955"
            />
          </LineChart>
        </div>
      </div>
    </div>
  );
```

```
};

export default EWMAPerformancePanel;
```

Appendix E: Backend Performance Calculation and Regression Model.

```
from flask import Blueprint, jsonify, request
from models.quizzes import Quizzes
from extensions import mongo
from flask_jwt_extended import jwt_required
import logging
import numpy as np

performance_bp = Blueprint('performance', __name__)
quizzes_model = Quizzes(mongo.db)

# Helper function to calculate EWMA
def calculate_ewma(data, alpha):
    if not data:
        return 0
    ewma_value = data[0]
    for i in range(1, len(data)):
        ewma_value = alpha * data[i] + (1 - alpha) * ewma_value
    return ewma_value

@performance_bp.route('/performance', methods=['GET'])
@jwt_required()
def get_performance():
    try:
        subject_id = request.args.get('subject_id')
        if not subject_id:
            return jsonify({'message': 'Subject ID is required'}), 400

        quizzes = quizzes_model.get_all(subject_id)

        if not quizzes:
            return jsonify({'message': 'No quizzes found for the 
subject'}), 404

        N = len(quizzes)
        alpha = 2 / (N + 1)  # Calculate alpha based on the number of 
quizzes

        # Extract l1_marks, l2_marks, and l3_marks from the quizzes
        l1_marks = [quiz['l1_marks'] for quiz in quizzes]
        l2_marks = [quiz['l2_marks'] for quiz in quizzes]
        l3_marks = [quiz['l3_marks'] for quiz in quizzes]

        # Calculate EWMA for l1_marks, l2_marks, and l3_marks
        ewma_l1 = calculate_ewma(l1_marks, alpha)
```

```
        ewma_l2 = calculate_ewma(l2_marks, alpha)
        ewma_l3 = calculate_ewma(l3_marks, alpha)

        # Calculate current performance from linear regression model
        finalResultRaw = 25.99 + 0.879 * ewma_l1 + 0.627 * ewma_l2 +
0.694 * ewma_l3
        finalResult = round(finalResultRaw, 2)

        # Prepare the response
        performance_data = {
            'EWMAL1': ewma_l1,
            'EWMAL2': ewma_l2,
            'EWMAL3': ewma_l3,
            'currentPerformance': finalResult
        }

        return jsonify(performance_data), 200

    except Exception as e:
        logging.error(f"Error fetching performance data for subject
{subject_id}: {str(e)}")
        return jsonify({'message': f'An error occurred: {str(e)}'}),
500
```