

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi-590 018



A Project Work Report on

"Transforming Student Evaluation with Adaptive Intelligence and Performance Analytics"

A Dissertation work submitted in partial fulfillment of the requirement for the award of the degree

**Bachelor of Engineering
in
Information Science and Engineering**

Submitted by

Abhishek Mangalur	1AY21IS003
Ketan Hegde	1AY21IS046
Chetan Badachi	1AY21IS028
Mohammad Aamir	1AY21IS054

Under the Guidance of
Prof. Pushpalatha K S
Assistant Professor



DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING ACHARYA INSTITUTE OF TECHNOLOGY

(AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI. ACCREDITED BY NAAC, RECOGNISED BY AICTE, NEW DELHI)

Acharya Dr. Sarvepalli Radhakrishnan Road, Soldevanahalli, Bengaluru - 560107

2024-2025

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING
ACHARYA INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University, Belagavi, Recognized by NAAC, New Delhi)
Acharya Dr. Sarvepalli Radhakrishnan Road, Soldevanahalli, Bengaluru-560107

2024-2025



Certificate

This is to certify that the **Project Work (21ISP76)** entitled "**Transforming Student Evaluation with Adaptive Intelligence and Performance Analytics**" carried out by **Abhishek Mangalur (1AY21IS003)**, **Ketan Hegde (1AY21IS046)**, **Chetan Badachi (1AY21IS028)** and **Mohammad Aamir (1AY21IS054)** are bonafide students of **Acharya Institute of Technology, Bengaluru** in partial fulfillment for the award of the degree of **Bachelor of Engineering in Information Science and Engineering** of the **Visvesvaraya Technological University, Belagavi** during the year **2024-25**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The Project Work report has been approved as it satisfies the academic requirements in respect of Project Work prescribed for the said degree.

Signature of the Guide
Prof. Pushpalatha K S

Signature of the HOD
Dr. Kala Venugopal

Signature of Principal
Dr. Rajeswari

External Viva-Voce:

Name of the Examiners:

Signature with Date

1. _____
2. _____

DECLARATION

ABHISHEK MANGALUR (1AY21IS003), KETAN HEGDE (1AY21IS046), CHETAN BADACHI (1AY21IS028) and MOHAMMAD AAMIR (1AY21IS054), students of B.E. Information Science and Engineering, Acharya Institute of Technology, Bangalore, hereby declare that the capstone project entitled "**Transforming Student Evaluation with Adaptive Intelligence and Performance Analytics**" is an authentic work carried out under the supervision and guidance of **Prof. Pushpalatha K S**, Assistant Professor, Department of Information Science and Engineering, Acharya Institute of Technology, Bangalore. We have not submitted the matter embodied to any other university or institution for the award of any other degree.

Place:

Date:

Name	USN	Signature
Abhishek Mangalur	1AY21IS003
Ketan Hegde	1AY21IS046
Chetan Badachi	1AY21IS028
Mohammad Aamir	1AY21IS054

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of **Project Work** report would be incomplete without the mentioning the people who made it possible through their constant guidance and encouragement.

We would take this opportunity to express our gratitude to **Sri. B. Premnath Reddy**, Founder Chairman, Acharya Institutes, **Dr. Rajeswari**, Principal, and **Dr. Marigowda C K**, Vice Principal, Acharya Institute of Technology for providing the necessary infrastructure to complete this **Project Work** report.

We wish to express our deepest gratitude to **Dr. Kala Venugopal**, Head of the Department, Information Science and Engineering, and also would like to thank the Project Coordinators **Prof. Yogesh N** and **Dr. Pankaj Kumar** for their constant support.

We wish to express our sincere thanks to our guide **Prof. Pushpalatha K S**, Assistant Professor, Department of Information Science and Engineering for helping us throughout, and guiding us from time to time.

A warm thanks to all the faculty of Department of Information Science and Engineering, who have helped us with their views and encouraging ideas.

Abhishek Mangalur (1AY21IS003)

Ketan Hegde (1AY21IS046)

Chetan Badachi (1AY21IS028)

Mohammad Aamir (1AY21IS054)

ABSTRACT

The Intelligent Assessment and Evaluation System is an AI-powered student assessment platform designed to revolutionize the evaluation process. By automating question generation and grading, the platform supports descriptive formats, enabling educators to create, schedule, and manage exams with ease. The system offers a personalized interface for students, ensuring a seamless and engaging user experience. Its advanced AI algorithms enhance question variety, ensuring a more comprehensive assessment of student knowledge.

To uphold test integrity, the system incorporates robust anti-cheating measures, including enforced full-screen mode, timers, and real-time monitoring. These features create a secure environment for examinations, fostering fairness and reliability in student evaluations. The platform also provides educators with AI-driven insights into student performance, enabling data-informed decisions to improve teaching effectiveness.

This scalable solution streamlines test administration, making it an indispensable tool for modern education. By combining adaptive intelligence with performance analytics, it empowers educators to deliver high-quality assessments and equips students with a fair and efficient evaluation system.

TABLE OF CONTENTS

Acknowledgement	i
Abstract	ii
Table of Contents	iii
List of Figures	vi
List of Tables	vii

Chapters	Page No
1 INTRODUCTION	1
1.1 Overview	1
1.2 Objective	2
1.3 Scope	5
1.4 Existing System	6
1.5 Proposed System	7
2 PROBLEM STATEMENT	10
2.1 Problem Statement	10
2.2 Motivation	10
2.3 Objectives	11
3 DETAILED SURVEY	12
4 SURVEY SUMMARY TABLE	16
5 SYSTEM REQUIREMENT SPECIFICATION	18
5.1 Functional Requirements	18
5.2 Non-functional Requirements	19
5.3 Hardware Requirements	20
5.4 Software Requirements	20

6	SYSTEM DESIGN	22
6.1	System Design	22
6.1.1	System Architecture	22
6.2	Detailed Design	23
6.2.1	Class Diagram	23
6.2.2	Activity Diagram	24
6.2.3	Use Case Diagram	25
7	IMPLEMENTATION	26
7.1	Query Module	26
7.1.1	Title and Description Management	26
7.1.2	Question Generation with Gemini API	29
7.1.3	Question Storage	31
7.2	Web Module	34
7.2.1	Teacher Interface	34
7.2.2	Student Interface	37
7.3	Evaluation Module	42
7.3.1	Submission Storage	42
7.3.2	Answer Evaluation	43
8	TESTING	44
9	TEST CASES AND RESULTS (SNAPSHOTS)	47
	CONCLUSION AND FUTURE ENHANCEMENTS	52
	BIBLIOGRAPHY	53

LIST OF FIGURES

Figures	Page No.
6.1 Architecture Diagram	22
6.2 Class Diagram	23
6.3 Activity Diagram	24
6.4 Use Case Diagram	25
9.1 Teacher Dashboard	49
9.2 Student Dashboard	49
9.3 Creation of Groups	50
9.4 Managing Student Groups	50
9.5 Creating Assessment	51
9.6 Detail Report	51

LIST OF TABLES

Tables	Page No.
4.1 Survey Summary Table	16
9.1 Unit Testing Test Cases	47
9.2 Integration Testing Test Cases	48
9.3 System Testing Test Cases	48

CHAPTER-1

INTRODUCTION

1.1 Overview

The traditional methods of student evaluation have relied heavily on manual processes, often resulting in inefficiencies, time-consuming tasks, and challenges in scaling assessments. With the rise of digital learning environments and technological advancements, there is a growing need for more efficient, automated, and secure methods of assessing student performance.

The AI-driven Student Assessment Platform seeks to address these challenges by incorporating artificial intelligence (AI) to automate key aspects of the student evaluation process. This platform is designed to automate question generation, grading, and performance tracking, reducing manual effort and enhancing efficiency. The system supports both multiple-choice and descriptive assessments, and employs advanced AI to generate and evaluate questions based on instructional content.

Additionally, the platform ensures exam integrity by incorporating anti-cheating features such as full-screen mode and timers, preventing unauthorized actions during the assessment process. The platform offers a personalized user experience for both teachers and students, allowing for real-time insights into student performance and progress.

This project aims to revolutionize the way educational institutions handle assessments by providing a scalable, efficient, and secure system that benefits both educators and students. By automating key administrative tasks, it enables teachers to focus on more strategic aspects of learning, while providing students with a fair and consistent assessment experience. Moreover, the platform's scalability allows it to cater to institutions of varying sizes, from small schools to large universities, ensuring its usability across diverse educational settings.

The AI-driven Student Assessment Platform reduces the evaluation burden through automation of questions, grading, and performance tracking to allow educators to concentrate on improving learning. High levels of anti-cheating guarantee the credibility of the results, and individual analyses help both teachers and students. Furthermore, the system supports multilingual capabilities, making it accessible and inclusive for a global audience. This innovative solution helps turn assessments into an easy, effective, and equitable process for everyone.

1.2 Objective

1.2.1 Automate Question Generation

The use of AI to automatically generate questions based on the instructional content helps teachers save valuable time by eliminating the need for manual question creation. This technology ensures that the questions generated align with the curriculum and are diverse, covering different levels of difficulty and various topics. It can create multiple types of questions, such as multiple-choice, short-answer, and descriptive, which further enhances the variety of assessments. This approach not only reduces the workload on teachers but also maintains a high level of consistency and fairness across assessments.

AI-driven question generation also provides a dynamic approach to assessments by ensuring the questions are continually updated to match the evolving curriculum. The system can adapt to the specific needs of a course and the preferences of the instructor, ensuring that assessments are always relevant. This innovation in exam creation makes the assessment process more efficient and ensures that students are tested comprehensively across all necessary areas of their learning.

1.2.2 Simplify Test Scheduling

AI can significantly simplify the test scheduling process by automating the planning of assessments. Educators no longer need to manually schedule exams, set reminders, or ensure that they do not overlap with other exams. The system can automatically select the best time slots based on availability, prevent conflicts with other tests, and ensure that students and teachers are aware of all deadlines. This reduces administrative burden, enabling teachers to focus more on content delivery than on logistical tasks. Additionally, the platform can accommodate varying time zones for remote or international students.

By streamlining test scheduling, the platform makes it easier for institutions to manage assessments efficiently. With AI handling the scheduling, it ensures that the assessment process is consistent and timely. It also provides real-time updates, allowing for adjustments when necessary, ensuring that students and teachers are well-prepared. The platform also accommodates diverse schedules, reducing conflicts and promoting accessibility for all students. Furthermore, automated reminders keep everyone informed, minimizing the chances of missed assessments.

1.2.3 Enhance Exam Security

AI-based anti-cheating mechanisms are critical for maintaining the integrity of online assessments. The system can enforce security measures like full-screen mode, preventing students from accessing unauthorized resources during the exam. It can also incorporate AI-driven techniques such as facial recognition to ensure that the person taking the exam is the registered student. Behavioural analysis can be used to track unusual patterns that might suggest cheating, such as looking away from the screen or excessive mouse movements.

These advanced security features provide a robust framework for monitoring students during the assessment. By preventing cheating and ensuring a fair testing environment, the system increases trust in the evaluation process. It offers institutions a reliable and secure platform for conducting exams while safeguarding academic integrity. This creates a level playing field for all students and gives educators confidence in the results.

1.2.4 Automate Grading Process

AI-driven grading systems can automatically grade both multiple-choice and descriptive questions, significantly reducing the manual effort involved in the evaluation process. For multiple-choice questions, the AI instantly provides scores, ensuring accurate and timely results. For descriptive answers, the system uses natural language processing (NLP) to analyze and grade based on predefined criteria, such as grammar, content quality, and relevance. The system also ensures scalability by efficiently handling large volumes of assessments, making it ideal for institutions with many students. Furthermore, it can incorporate subject-specific nuances, allowing for more accurate evaluation in technical or creative disciplines.

By automating the grading process, the platform not only saves teachers considerable time but also eliminates human errors that can occur during manual grading. This results in a more consistent, fair, and timely feedback process, ensuring that students receive accurate assessments of their work. Teachers can quickly identify areas where students may need improvement, and the AI system can even generate personalized recommendations to guide further learning. Additionally, the platform supports detailed analytics, allowing educators to track performance trends across entire classes or specific individuals. This data-driven approach helps in tailoring instruction to meet diverse student needs. Moreover, automated grading enhances transparency, giving students a clear understanding of how evaluated.

1.2.5 Provide Real-Time Analytics

The AI platform generates real-time insights into student performance, identifying trends and patterns in learning. By analysing data from assessments, the system provides educators with detailed reports on student strengths and areas for improvement. This data-driven approach helps teachers make more informed decisions regarding their teaching strategies and ensures they can adapt to meet the specific needs of their students.

Students also benefit from real-time analytics, as the system can offer personalized learning recommendations based on their performance. These insights help students focus on areas that need improvement, enhancing their learning experience and boosting academic success. Educators can track the effectiveness of their teaching methods, allowing them to refine their approach and ensure better outcomes for their students.

1.2.6 Develop a Scalable Platform

The platform is designed to be highly scalable, ensuring that it can support educational institutions of varying sizes, from small schools to large universities. The system can handle an increasing number of users, assessments, and data without compromising performance. This scalability is essential as educational institutions grow, providing a seamless experience for both educators and students.

By being scalable, the platform can accommodate the changing needs of educational institutions, ensuring long-term usability. As schools expand or as student populations grow, the system can scale up to meet demand. This flexibility ensures that the platform remains efficient and reliable, regardless of the institution's size, enabling a broader range of institutions to benefit from the technology.

1.2.7 Enhance Student Experience

The platform is designed to improve the overall student experience by offering a personalized, efficient, and fair assessment system. By adapting to each student's learning needs, it provides tailored assessments that match their skill level and progress. Students receive immediate feedback, allowing them to learn and improve continuously. This instant feedback also helps in reducing anxiety and stress related to waiting for exam results. The platform enhances engagement by offering interactive assessments that keep students motivated.

1.2.8 Support Multiple Assessment Formats

The platform supports various assessment formats, including multiple-choice, short-answer, and descriptive questions, offering a comprehensive evaluation of student understanding. By accommodating different types of questions, the system ensures that students are tested on a wide range of skills, including knowledge recall, critical thinking, and problem-solving abilities.

Supporting multiple formats also allows educators to create diverse assessments that can evaluate different aspects of student learning. This flexibility makes the platform more adaptable to various subjects and teaching styles, helping educators create well-rounded assessments that reflect the full scope of the curriculum. It provides a holistic view of student capabilities, ensuring that every student is assessed fairly and comprehensively.

1.3 Scope

The scope of the AI-driven student assessment platform covers several critical aspects of modernizing the assessment process in educational institutions. The platform aims to automate key elements of the assessment workflow, such as question generation, test scheduling, grading, and security, to enhance both efficiency and fairness in evaluations. This system will be applicable across various educational levels, including schools, colleges, and universities, supporting a wide range of subjects and courses. It will cater to both multiple-choice and descriptive assessments, providing a comprehensive evaluation of student performance.

The platform will also support real-time analytics, allowing educators to track and analyze student progress, identify learning patterns, and provide personalized feedback. Additionally, it will offer advanced anti-cheating features such as AI-based facial recognition and behavioral analysis to ensure the integrity of assessments. Furthermore, the platform will be scalable to handle growing user bases, enabling its deployment in institutions of any size. It will also provide predictive analytics, helping educators anticipate potential learning challenges and intervene proactively to support student success.

This platform is designed to improve the learning experience for students by providing timely feedback and personalized learning recommendations, making it an adaptable and secure solution for modern educational environments. It will also integrate with various learning management systems (LMS), ensuring seamless data exchange and compatibility with

existing tools. Moreover, the platform will include multilingual support, catering to diverse linguistic needs and fostering inclusivity across global institutions.

1.4 Existing System

- **Manual Effort in Question Creation:** In the current systems, teachers are responsible for manually creating exam questions, which consumes significant time and effort. This manual approach makes it difficult to ensure diversity and alignment with the curriculum. Teachers often find it challenging to generate a wide range of questions, especially when teaching large classes.
- **Limited Automation:** Existing tools like Google Forms or basic Learning Management Systems (LMS) provide some automation, but these platforms do not offer advanced features like AI-driven question generation or grading. As a result, teachers are forced to spend considerable time not just in question creation but also in evaluating student responses manually.
- **Lack of Advanced Anti-Cheating Features:** Many of the systems in place today, especially online exam platforms, lack robust anti-cheating mechanisms. While some tools allow for timers, they do not offer features like full-screen mode, browser lockdown, or AI-driven detection of cheating behaviours such as facial recognition, eye-tracking, or behavioural analysis to prevent cheating during online exams.
- **Inefficiency in Grading:** Manual grading is prone to human error and is often time-consuming, especially when evaluating descriptive or subjective responses. Current systems generally automate only multiple-choice grading, leaving descriptive assessments to be graded manually, which slows down the entire process.
- **Basic Functionality and Limited Assessment Formats:** Existing systems mainly support basic assessment types like multiple-choice questions or simple true/false formats. These tools do not support diverse formats such as multimedia-based questions (audio, images, or video), limiting the ability to test a range of skills or cater to students with varying learning needs.
- **Scalability Issues:** Many current systems struggle with scalability. As educational institutions expand or when many students take exams simultaneously, performance issues and system crashes become common. Existing platforms are not always capable of scaling to meet the needs of larger educational environments without experiencing lag or

downtime. Existing systems may be vulnerable to data breaches, compromising sensitive student information during assessments.

- **Lack of Real-Time Analytics:** Current systems typically do not provide in-depth analytics about student performance. They lack the capability to track trends over time or offer personalized insights into students' strengths and weaknesses. This limitation makes it harder for educators to adjust teaching methods based on individual student progress or learning patterns.
- **Poor User Experience:** Many existing systems do not focus on student experience. They offer basic interfaces without personalization, and students may not receive immediate feedback. This can lead to dissatisfaction, as students do not have access to timely information on their performance, hindering their ability to improve.
- **Limited Integration with Other Tools:** Current systems often work in isolation and fail to integrate effectively with other tools like Learning Management Systems (LMS), gradebooks, or communication platforms. This lack of integration can lead to fragmented workflows, requiring educators to manually transfer data between multiple systems, increasing the administrative burden.
- **Lack of Support for Diverse Student Needs:** Existing assessment systems do not always cater to diverse learning needs. They often fail to provide personalized learning experiences or support alternative assessment formats, like voice-based or image-based questions, which can help students with disabilities or those who benefit from different types of assessments.

1.5 Proposed System

The proposed system aims to revolutionize the traditional assessment process by leveraging cutting-edge artificial intelligence (AI) to automate and streamline key aspects of student evaluation. This system will focus on improving efficiency, scalability, security, and personalization in educational assessments. Additionally, it will provide detailed analytics dashboards, enabling institutions to gain actionable insights into overall academic performance and trends.

The integration of AI ensures that assessments are tailored to individual learning needs, offering a fair and accurate evaluation for every student. Advanced security features, such as real-time monitoring and fraud detection, uphold the integrity of the process. Furthermore, the

system supports adaptive question generation, ensuring that the difficulty level aligns with the student's proficiency, promoting a more engaging and challenging learning experience.

1.5.1 Key Features of the Proposed System

- **Automated Question Generation:** The system will employ advanced AI algorithms to automatically generate a variety of question types, including multiple-choice, descriptive, and multimedia-based questions. The questions will be aligned with the instructional content and tailored to the curriculum's specific requirements. Educators will have the flexibility to define parameters such as difficulty level and topic coverage.
- **Enhanced Test Scheduling and Management:** The platform will automate the test scheduling process, eliminating the need for manual intervention. It will provide real-time notifications to students and educators and prevent scheduling conflicts, ensuring seamless planning for all stakeholders.
- **AI-Driven Grading and Feedback:** The system will integrate natural language processing (NLP) to automate grading for both objective and subjective questions. Instant feedback will be provided to students, along with detailed insights into their performance. This will help students identify areas for improvement and educators to adjust their teaching strategies.
- **Robust Exam Security:** To ensure the integrity of online assessments, the platform will implement anti-cheating features such as browser lockdown, full-screen mode, facial recognition, and behavioural analysis. These measures will create a secure and trustworthy environment for conducting exams.
- **Real-Time Analytics and Insights:** The system will generate comprehensive analytics for both educators and administrators, providing insights into student performance, learning trends, and assessment effectiveness. This data-driven approach will support informed decision-making and personalized learning interventions. Educators can track progress over time, identify knowledge gaps, and adapt teaching strategies to meet student needs. Administrators can use the analytics to evaluate curriculum effectiveness and allocate resources more efficiently. These insights ultimately contribute to a more dynamic and responsive educational environment.

- **Scalability and Accessibility:** Designed to cater to institutions of all sizes, the platform will be highly scalable, ensuring consistent performance regardless of the number of users. Additionally, it will offer a user-friendly interface with multilingual support, making it accessible to diverse educational institutions worldwide.
- **Integration with Existing Systems:** The platform will seamlessly integrate with existing Learning Management Systems (LMS), gradebooks, and communication tools. This will ensure a unified workflow for educators and administrators, minimizing the need for redundant data entry or manual processes.

1.5.2 Advantages of the Proposed System

- **Efficiency:** By automating manual tasks such as question generation, grading, and test scheduling, the system significantly reduces educators' workload.
- **Personalization:** Real-time analytics and feedback provide tailored learning experiences for students, addressing their unique needs and challenges.
- **Integrity:** Advanced security measures ensure that the assessment process remains fair and credible.
- **Scalability:** The system can grow with the institution, accommodating increasing numbers of users and assessments without performance degradation.
- **Adaptability:** Support for various question formats and integration with existing tools makes the platform adaptable to different educational contexts.

CHAPTER-2

PROBLEM STATEMENT

2.1 Problem Statement

Traditional assessment methods, such as using tools like Google Forms, require teachers to manually create questions and evaluate answers, making the process time-consuming and repetitive. This approach becomes increasingly inefficient with larger classes or frequent assessments, as it demands significant time for question creation and grading. The manual grading process also leaves room for human error, particularly with subjective questions, and delays in providing feedback. Furthermore, these systems struggle to scale effectively, making it difficult to manage assessments for larger student populations.

2.2 Motivation

The motivation behind the development of the AI-driven Student Assessment Platform stems from the growing need for more efficient, scalable, and secure assessment systems in educational institutions. Traditional assessment methods, such as using paper-based exams or tools like Google Forms, require significant manual effort from educators, which can lead to inefficiencies, increased chances of error, and time-consuming grading processes. Furthermore, ensuring the integrity of assessments in online environments remains a challenge, with minimal safeguards against cheating.

The increasing reliance on digital tools and platforms in education presents an opportunity to leverage technology to overcome these challenges. The motivation is to create a system that reduces the manual workload of teachers, ensures the security of online exams, and provides a scalable solution that can accommodate institutions of varying sizes. By integrating artificial intelligence (AI), this platform aims to automate the generation of questions, the grading process, and offer personalized feedback, making the assessment process more efficient, reliable, and fair.

Moreover, with the rapid shift toward online learning and digital classrooms, there is a pressing need for tools that can offer real-time insights into student performance, track progress, and tailor learning experiences based on individual needs. The AI-driven platform motivates the need to bridge this gap, enhancing the overall educational experience for both

students and educators. By automating assessments and ensuring security, the system promotes a more streamlined, accurate, and equitable evaluation process.

2.3 Objectives

2.3.1 Automate Question Generation

- Utilize AI to automatically generate diverse, curriculum-aligned questions, including multiple-choice, short-answer, and descriptive formats.
- Ensure consistency and fairness across assessments by providing a wide range of difficulty levels and topics.

2.3.2 Simplify Test Scheduling

- Automate the scheduling of assessments, avoiding conflicts and ensuring optimal time slots for all stakeholders.
- Provide real-time updates and notifications for both educators and students regarding upcoming tests and deadlines.

2.3.3 Enhance Exam Security

- Implement AI-driven anti-cheating measures, including full-screen mode, facial recognition, and behavioral analysis.
- Ensure the integrity of online assessments by preventing unauthorized access and detecting suspicious behaviors during exams.

2.3.4 Automate the Grading Process

- Develop an AI-powered grading system capable of evaluating both objective and descriptive questions.
- Utilize natural language processing (NLP) to provide accurate, criteria-based grading for descriptive answers.

2.3.5 Provide Real-Time Analytics

- Generate detailed, real-time insights into student performance, identifying trends and areas needing improvement.
- Enable predictive analytics to forecast potential learning challenges and recommend proactive interventions for enhanced student outcomes.

CHAPTER-3

DETAILED SURVEY

[2024] A Comparative Analysis of Large Language Models with Retrieval-Augmented Generation-Based Question Answering System, H. N. Patel, P. Goel, A. Surti, and B. Patel

This paper provides a comprehensive evaluation of large language models (LLMs) integrated with retrieval-augmented generation (RAG) systems for question-answering tasks. The authors detail the advantages of combining LLMs with external knowledge retrieval mechanisms to significantly enhance the relevance, accuracy, and diversity of generated questions. The integration of RAG enables the system to access extensive datasets, ensuring the generated questions are contextually appropriate and aligned with specific learning objectives. Furthermore, this study emphasizes the importance of scalability, demonstrating how these systems can handle vast and diverse datasets across multiple domains. For educational applications, this research underlines the utility of LLMs in dynamically generating questions tailored to curriculum standards, reducing the manual workload on educators. Additionally, by leveraging advanced retrieval mechanisms, the system can ensure that assessments are consistently updated with the latest and most relevant information, enhancing the overall quality and effectiveness of educational evaluations.

[2024] Development of a Web System with an Automated Question Generator Based on Large Language Models, L. Hrváčević, V. Jocović, J. Cincović, V. Matvejev, A. Milaković, and D. Drašković

This paper introduces a web-based platform that employs large language models to automate question generation. The system supports dynamic question creation tailored to the instructional content provided by educators and accommodates multiple question formats, including text-based, image-based, and multimedia-based questions. By focusing on scalability and user-friendliness, the authors demonstrate how the system can adapt to the requirements of diverse educational environments. The study highlights innovative features such as customization options for educators, allowing them to define parameters like difficulty level,

question type, and alignment with specific learning objectives. Additionally, the integration of automated feedback mechanisms ensures that students receive timely and constructive evaluations of their performance. The methodologies and findings of this paper provide a practical framework for designing an efficient and adaptable question-generation module within the proposed system. This ensures that assessments remain relevant, engaging, and capable of addressing diverse educational needs.

[2024] How Teachers Can Use Large Language Models and Bloom's Taxonomy to Create Educational Quizzes, S. Elkins, E. Kochmar, J. C. K. Cheung, and I. Serban

This research explores the application of large language models in creating quizzes aligned with Bloom's Taxonomy. By categorizing questions into different cognitive levels—ranging from basic knowledge recall to advanced critical thinking—the study provides a structured approach to designing educational assessments. The framework proposed by the authors allows educators to specify the desired cognitive level, enabling the system to generate questions that foster both foundational understanding and higher-order analytical skills. For example, lower-order questions might involve simple recall of facts, while higher-order questions could require students to apply concepts, analyze scenarios, or evaluate outcomes. The paper also discusses the importance of balancing question types to create comprehensive assessments that cater to diverse learning needs. These insights are critical for the project, as they ensure that the question-generation process aligns with established pedagogical standards. By integrating these principles, the system can deliver assessments that are not only automated but also pedagogically robust and conducive to student learning.

[2023] Using ChatGPT for Generating and Evaluating Online Tests, V. M. Ionescu and M. C. Enescu

This research investigates the application of ChatGPT in automating the creation and evaluation of online tests. The study demonstrates the model's ability to generate questions across various formats, including multiple-choice, short-answer, and descriptive types, tailored to specific curricula. Furthermore, it explores the potential of natural language processing (NLP) techniques in automating grading processes, particularly for subjective answers. The research highlights the model's capability to provide immediate and accurate feedback, a

feature crucial for enhancing student learning outcomes. Despite its numerous advantages, the paper identifies limitations such as biases in the model and the necessity for human oversight to validate outputs. These insights are particularly relevant to the development of AI-driven assessment platforms, as they emphasize the importance of integrating validation mechanisms and providing educators with tools to monitor and adjust AI-generated content. This ensures fairness, accuracy, and alignment with educational objectives, making the system a reliable and supportive tool for educators.

[2023] Assessing the Use of OpenAI Chat-GPT in a University Department of Education, J. Prentzas and M. Sidiropoulou

This paper evaluates the practical implementation of OpenAI's ChatGPT in academic settings, particularly within a university education department. The research focuses on the model's utility in generating educational content, automating grading, and providing personalized feedback to students. The authors highlight the benefits of using ChatGPT for streamlining routine tasks and improving efficiency. For instance, ChatGPT can assist educators by generating quizzes, summarizing lecture materials, and even offering real-time assistance to students. However, the paper also addresses challenges such as ethical concerns, potential biases, and the importance of maintaining human oversight to ensure quality and reliability. The study emphasizes the necessity of a collaborative approach, where AI tools complement, rather than replace, human educators. These considerations are essential for the current project, as they provide a balanced perspective on the opportunities and limitations of integrating AI-driven tools into educational workflows.

Summary of Contributions to the Project

The referenced papers collectively provide a strong foundation for the development of the AI-driven Student Assessment Platform. Key insights and contributions include:

8.1 Automated Question Generation

- Leveraging LLMs and retrieval-augmented generation to create diverse and curriculum-aligned questions ([Patel et al., 2024], [Hrváčević et al., 2024]).
- Employing frameworks like Bloom's Taxonomy to ensure pedagogical soundness in assessments ([Elkins et al., 2024]).

- Enhancing question variety by incorporating multimedia formats, such as image-based and audio-based questions, to cater to diverse learning preferences.

8.2 Grading and Feedback

- Utilizing NLP for automated grading of both objective and subjective questions ([Ionescu and Enescu, 2023]).
- Offering personalized feedback and insights based on real-time analytics ([Prentzas and Sidiropoulou, 2023]).
- Reducing teacher workload by automating routine grading tasks while ensuring consistency and accuracy.

8.3 Security and Integrity

- Adopting methods from surveillance applications for behavioral analysis and integrity checks during online assessments ([Loy, 2017]).
- Implementing anti-cheating measures such as gaze tracking, full-screen lockdown, and activity monitoring to maintain assessment integrity.
- Utilizing AI-driven tools to analyze real-time audio and video feeds to detect anomalies such as multiple voices, noise, or presence of unauthorized individuals.

8.4 System Design and Scalability

- Designing scalable and user-friendly platforms to cater to diverse educational environments ([Hrvačević et al., 2024]).
- Ensuring that the platform can adapt to increasing user demands and institutional growth, making it suitable for large-scale deployments.

8.5 Ethical Considerations

- Addressing challenges like biases and the need for human oversight in AI-driven educational tools ([Ionescu and Enescu, 2023], [Prentzas and Sidiropoulou, 2023]).
- Emphasizing transparency and accountability in AI-based systems to maintain trust among educators and students.

CHAPTER-4**SURVEY SUMMARY TABLE****Table 4.1: Survey Summary Table**

Sl No.	Title of the Paper	Problem Addressed	Authors Approach / Method	Results
1	Activity Understanding and Unusual Event Detection in Surveillance Videos	Challenges in identifying unusual events and activities in surveillance videos efficiently and accurately.	Proposed algorithms integrating computer vision and deep learning for real-time anomaly detection in surveillance systems.	Achieved high accuracy in detecting abnormal activities and demonstrated adaptability to various applications.
2	A Comparative Analysis of Large Language Models with Retrieval-Augmented Generation-Based Question Answering System	Need for accurate and relevant automated question-answering systems in diverse domains.	Combined large language models with retrieval-augmented generation (RAG) to improve contextual accuracy and scalability.	Demonstrated significant improvement in question diversity, relevance, and adaptability to dynamic datasets.
3	Using ChatGPT for Generating and Evaluating Online Tests	Inefficiencies in manual test generation and grading processes in online education.	Utilized ChatGPT for automated question generation and grading using natural language processing (NLP) techniques.	Reduced workload for educators, provided accurate grading for subjective answers, and offered real-time feedback.

4	<p>Development of a Web System with an Automated Question Generator Based on Large Language Models</p>	<p>Lack of scalable, dynamic systems for generating diverse question formats in educational assessments.</p>	<p>Designed a web-based platform incorporating large language models for customizable and automated question generation.</p>	<p>Enabled dynamic and scalable question generation, supporting multimedia and diverse formats tailored to curriculum needs.</p>
5	<p>How Teachers Can Use Large Language Models and Bloom's Taxonomy to Create Educational Quizzes</p>	<p>Difficulty in aligning assessments with educational objectives across different cognitive levels.</p>	<p>Developed a framework for question generation aligned with Bloom's Taxonomy, targeting various levels of cognitive learning.</p>	<p>Generated diverse, pedagogically sound assessments fostering both basic understanding and higher-order thinking.</p>
6	<p>Assessing the Use of OpenAI Chat-GPT in a University Department of Education</p>	<p>Limited integration of AI tools in educational workflows and ethical concerns in their application.</p>	<p>Evaluated ChatGPT's use for generating content, automating grading, and offering personalized feedback while addressing ethical issues.</p>	<p>Improved efficiency in educational tasks, highlighted the need for human oversight, and addressed biases in AI outputs.</p>

CHAPTER-5

SYSTEM REQUIREMENT SPECIFICATION

5.1 Functional Requirements

The functional requirements define the specific behaviours and functionalities that the AI-driven Student Assessment Platform must support. These include:

5.1.1 Automated Question Generation

- Generate diverse types of questions (multiple-choice, short-answer, descriptive) aligned with instructional content.
- Support multimedia-based questions, including image, video, and audio formats.
- Enable customization of question difficulty and format by educators.

5.1.2 Test Scheduling and Management

- Automate test scheduling, ensuring no conflicts with other assessments.
- Notify students and educators of upcoming tests and deadlines.
- Allow real-time adjustments to schedules as necessary.

5.1.3 Grading and Feedback

- Automate grading for multiple-choice and descriptive questions using natural language processing (NLP).
- Provide immediate feedback to students with detailed performance insights.
- Generate personalized learning recommendations based on individual performance.

5.1.4 Exam Security

- Implement anti-cheating mechanisms, including full-screen mode, browser lockdown, and facial recognition.
- Monitor behavioural patterns such as gaze tracking and unusual activities during online assessments.
- Ensure secure data encryption and authentication methods to protect exam content and student information from unauthorized access.

5.1.5 Real-Time Analytics and Reporting

- Provide educators with real-time insights into student performance and learning trends.
- Generate detailed reports highlighting strengths and areas for improvement.
- Enable export of analytics data for integration with other Learning Management Systems (LMS).

5.1.6 Scalability and Multi-User Support

- Handle multiple concurrent users and exams without performance degradation.
- Support varying institutional sizes, from small schools to large universities.

5.1.7 User Management

- Allow role-based access for students, educators, and administrators.
- Enable account management, including registration, updates, and secure authentication.

5.2 Non-functional Requirements

The non-functional requirements define the quality attributes and constraints of the system:

5.2.1 Performance

- Ensure response times under 2 seconds for key functionalities, even under heavy loads.
- Support concurrent access for at least 10,000 users without performance degradation.

5.2.2 Scalability

- Ensure scalability to accommodate growing user bases and data volumes.
- Support integration with third-party tools and additional modules.

5.2.3 Security

- Protect sensitive data with encryption and secure authentication protocols.
- Ensure compliance with educational data protection regulations, such as GDPR.

5.2.4 Usability

- Provide an intuitive and user-friendly interface for both students and educators.
- Offer multilingual support to cater to diverse user groups.

5.2.5 Reliability

- Guarantee 99.9% system uptime with minimal disruptions.
- Provide failover mechanisms to ensure system availability during server issues.

5.2.6 Maintainability

- Ensure modular code design for easy updates and feature additions.
- Provide detailed documentation for developers and system administrators.

5.3 Hardware Requirements

The hardware requirements define the infrastructure needed to deploy and operate the system.

5.3.1 Server Requirements

- Processor: Intel Xeon or AMD Ryzen (8-core, 3.5 GHz or higher).
- Memory: 32 GB RAM or higher.
- Storage: SSD with at least 1 TB capacity.
- Network: 1 Gbps network connection.

5.3.2 Client Requirements

- Processor: Intel Core i5 or equivalent.
- Memory: 8 GB RAM.
- Storage: 256 GB SSD.
- Display: 1080p resolution or higher.
- Network: Stable internet connection (10 Mbps or higher).

5.4 Software Requirements

The software requirements specify the tools and platforms needed for development, deployment, and operation.

5.4.1 Server Software

- Operating System: Linux (Ubuntu 20.04 LTS or higher) or Windows Server 2019.
- Web Server: React.
- Database: MongoDB.
- AI Frameworks: Gemini AI

5.4.2 Client Software

- Operating System: Windows 10/11, macOS, or Linux.
- Browser: Latest versions of Chrome, Firefox, or Edge.

5.4.3 Development Tools

- Programming Languages: React, NodeJS.
- Frameworks: ReactJS for front-end, NodeJS for back-end.
- Version Control: Git with GitHub.
- IDE: Visual Studio Code or PyCharm.

CHAPTER-6

SYSTEM DESIGN

6.1 System Design

6.1.1 System Architecture

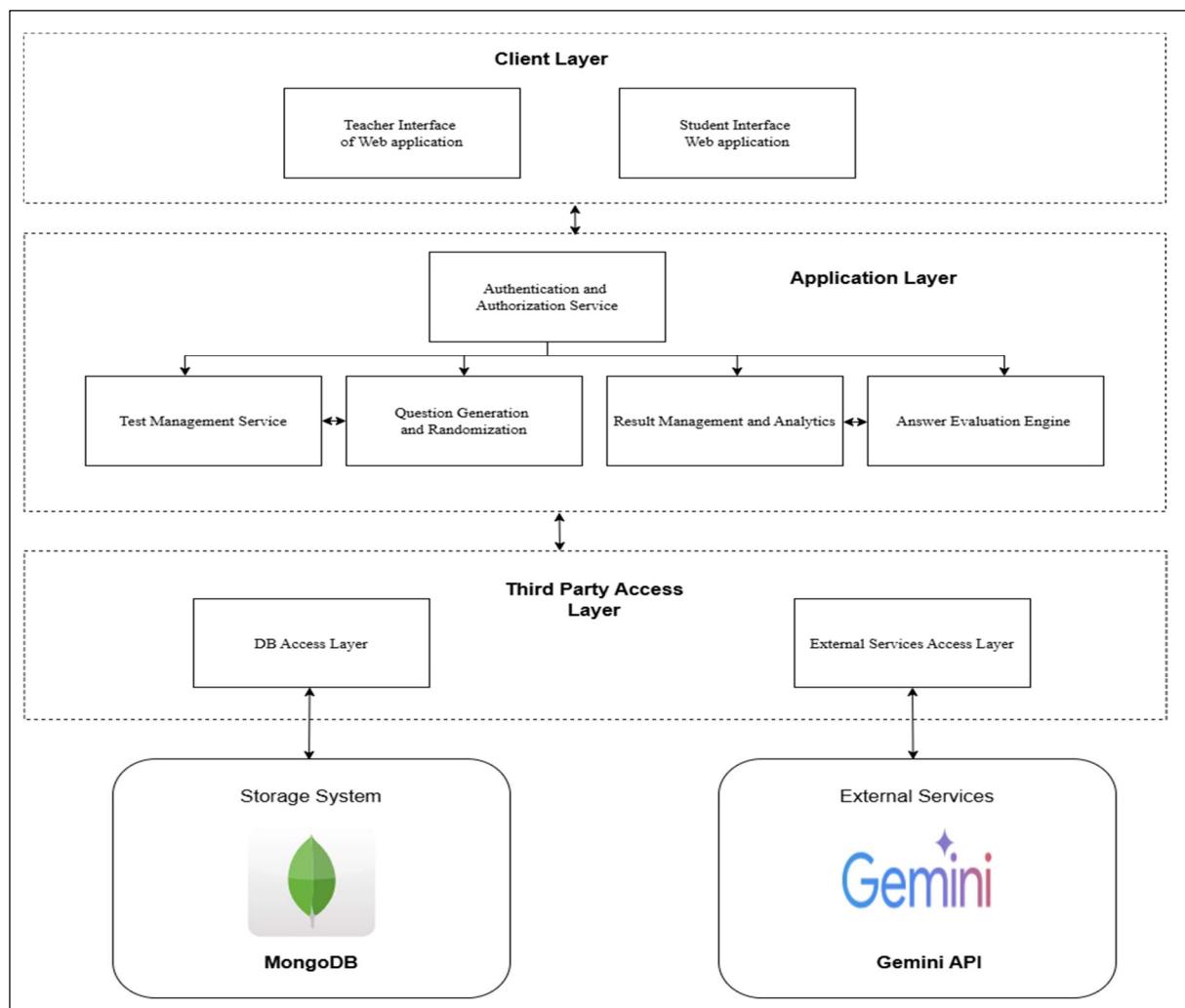


Fig. 6.1: Architecture diagram

This diagram illustrates a 3-tier web application architecture featuring a client layer (teacher/student interfaces), an application layer (core services like test management and AI question generation), and a data/external access layer (database and Gemini API). It highlights the system's modular design with distinct components for key functionalities.

6.2 Detailed Design

6.2.1 Class Diagram

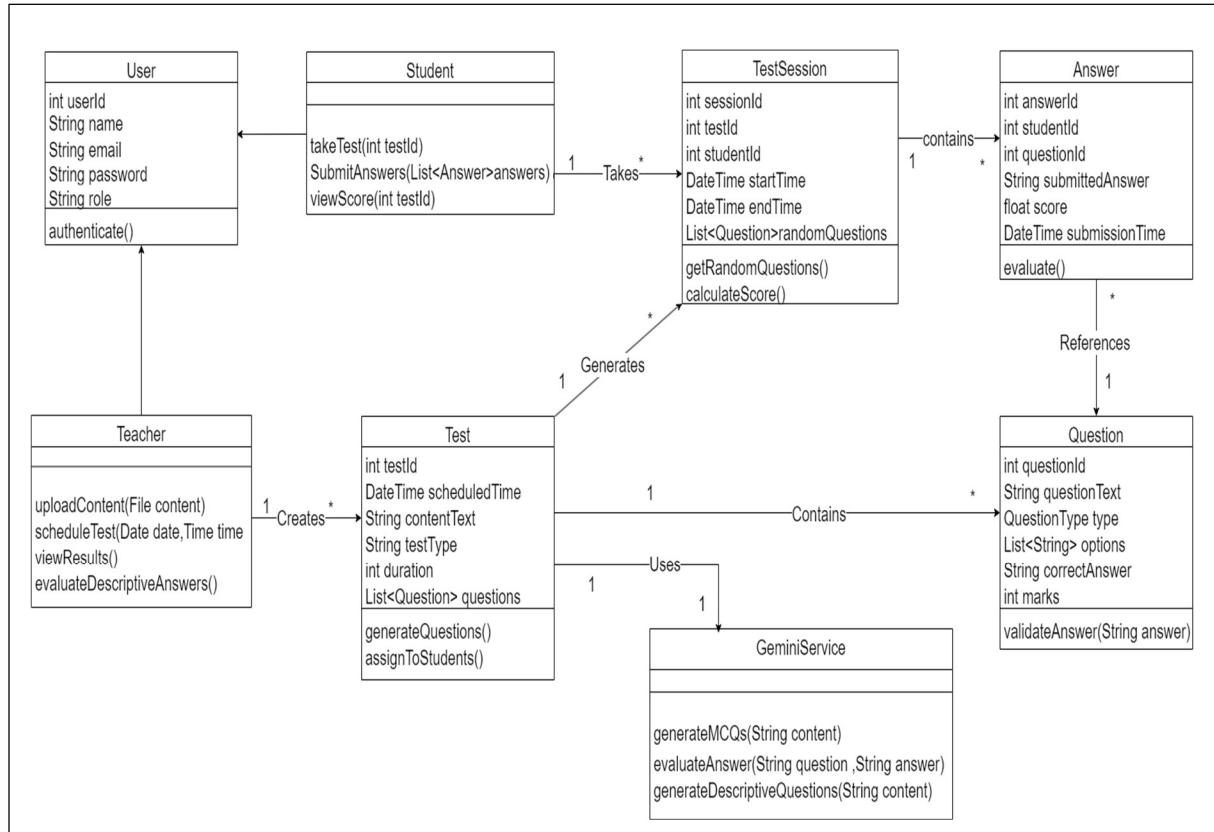


Fig. 6.2: Class Diagram

This class diagram depicts the structure of an online testing system, outlining classes like User (with subclasses Teacher and Student), Test, Question, Answer, and TestSession, and their relationships, including inheritance, composition, and usage dependencies, to model the system's entities and their interactions. It also includes a GeminiService class likely for AI-powered features.

6.2.2 Activity Diagram

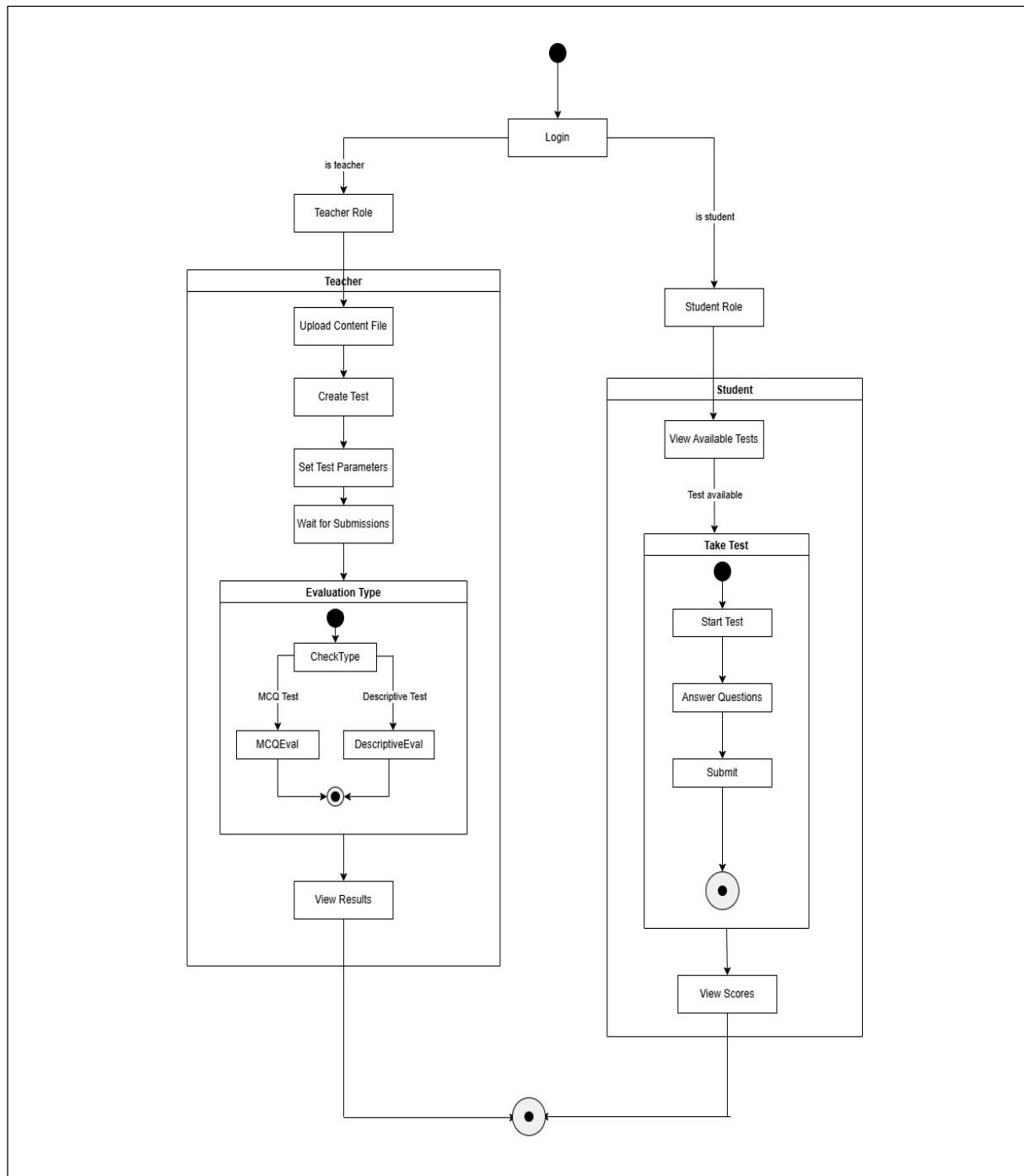


Fig. 6.3: Activity Diagram

This activity diagram illustrates the workflow of an online testing system, showing the parallel processes of teachers creating and managing tests (left) and students taking tests (right), along with decision points for evaluation types and synchronization after submission. It visually represents the sequence of actions and interactions within the system.

6.2.3 Use Case Diagram

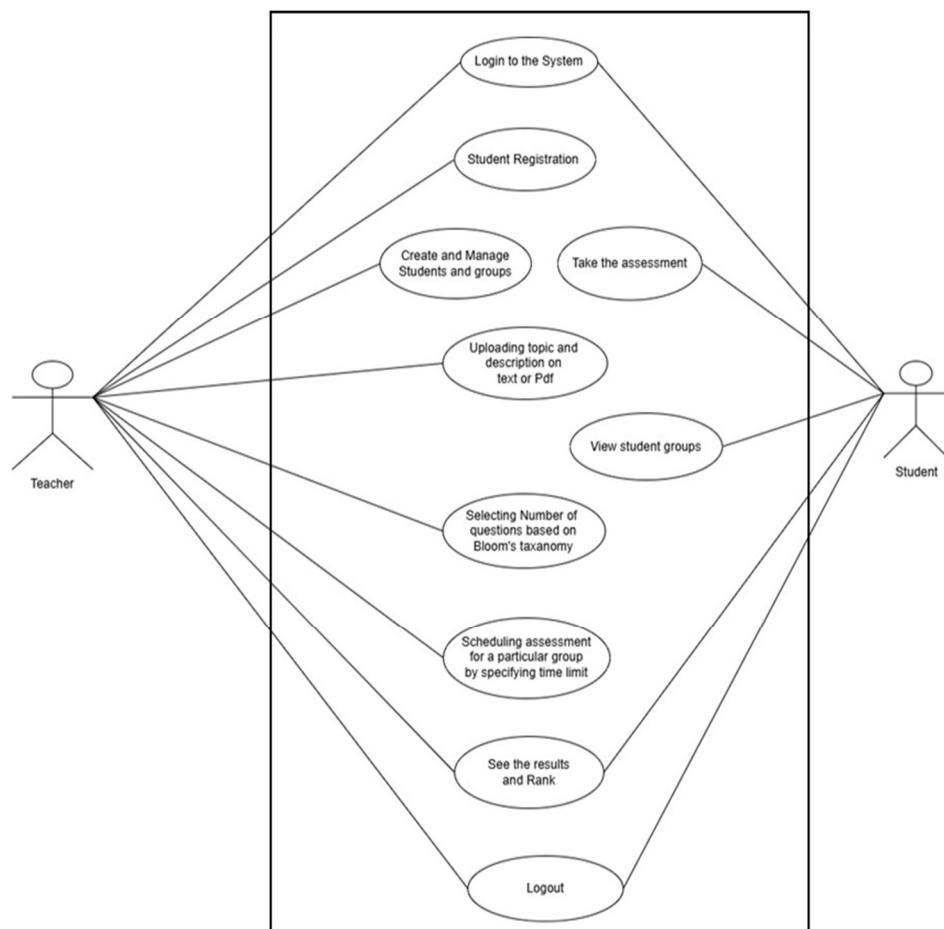


Fig. 6.4: Use case Diagram

This use case diagram outlines the interactions between a Teacher and a Student within an online assessment system, illustrating the functionalities each actor can perform, such as creating/managing content and assessments (Teacher) and taking assessments/viewing results (Student). It clearly shows the system's features from a user-centric perspective.

CHAPTER-7

IMPLEMENTATION

The platform implements a comprehensive assessment management system through three interconnected modules. The Query Module facilitates assessment creation by capturing teacher inputs, generating questions via AI integration using the Gemini API, and maintaining a structured question repository. The Web Module serves as a user-centric interface with role-based access, enabling teachers to manage student administration, schedule assessments, and monitor performance, while providing students with a streamlined experience for test-taking and result visualization. Finally, the Evaluation Module ensures accurate grading by processing student submissions through AI-powered evaluation, maintaining detailed performance records, and offering feedback mechanisms for continuous improvement.

7.1 Query Module

7.1.1 Title and Description Management

The system implements a comprehensive mechanism for storing educational content provided by teachers. This component allows teachers to input titles and descriptions of topics, which are then stored in a MongoDB database using Mongoose schemas. The implementation includes validation checks to prevent duplicate entries and supports both text descriptions and file uploads.

Code Snippet

```
// Schema definition for titles
const titleSchema = new mongoose.Schema({
  name: String,
  description: String,
  fileData: Buffer,
  fileType: String,
});
const Title = mongoose.model("Title", titleSchema);
```

```
// Save description endpoint
app.post("/api/save-description", upload.single("file"), async (req, res) => {
  const { title, description } = req.body;
  const file = req.file;
  if (!title || (!description && !file)) {
    return res.status(400).json({
      message: "Title and either a description or a file are required.",
    });
  }
  try {
    const existingTitle = await Title.findOne({ name: title });
    if (existingTitle) {
      return res.status(400).json({ message: "This title already exists." });
    }
    const dataToSave = { name: title };
    if (description) {
      dataToSave.description = description;
    }
    if (file) {
      dataToSave.fileData = file.buffer;
      dataToSave.fileType = file.mimetype;
    }
    const newTitle = new Title(dataToSave);
    await newTitle.save();
    res.json({ message: "Title saved successfully.", data: newTitle });
  } catch (error) {
    console.error("Error saving title:", error.message);
    res.status(500).json({ message: "An error occurred while saving the title." });
  }
});

// Title search and retrieval endpoint
app.get("/api/titles", async (req, res) => {
```

```
try {
  const { search } = req.query;
  let titles;
  if (search) {
    titles = await Title.find({
      name: { $regex: search, $options: "i" },
    });
  } else {
    titles = await Title.find();
  }
  res.json(titles);
} catch (error) {
  console.error("Error fetching titles:", error);
  res.status(500).json({
    message: "An error occurred while fetching titles",
    error: error.message,
  });
}
});

// Title validation endpoint
app.post("/api/check-title", async (req, res) => {
  const { name } = req.body;
  if (!name) {
    return res.status(400).json({ available: false, message: "Topic is required." });
  }
  try {
    const existingTitle = await Title.findOne({ name });
    if (existingTitle) {
      res.json({
        available: 0,
        message: "This topic is already available with us",
      });
    }
  }
});
```

```
    } else {
      res.json({
        available: 1,
        message: "This topic is new. Proceed to add description",
      });
    }
  } catch (error) {
  console.error("Error querying the database:", error);
  res.status(500).json({
    available: false,
    message: "An error occurred. Please try again."});
}
});
```

7.1.2 Question Generation with Gemini API

The system utilizes Google's Gemini API to automatically generate questions based on the provided content. This implementation supports both Multiple-Choice Questions (MCQs) and descriptive questions, with the ability to specify the number of questions and difficulty levels. The question generation process considers the content context and generates appropriate questions along with their answers.

Code Snippet

```
// MCQ Generation Function
async function generateMCQQuestions(description, topic, numQuestions) {
  const prompt = `Generate ${numQuestions} multiple choice questions about ${topic}.
  Context: ${description}
  For each question, provide:
  1. The question text
  2. Four options (A, B, C, D)
  3. The correct answer
  4. Difficulty level (easy/medium/hard)
  Format the response as a plain JSON array of objects`;
```

```
const model = genAI.getGenerativeModel({ model: "gemini-1.5-flash" });
const result = await model.generateContent(prompt);
const rawResponse = await result.response.text();
return JSON.parse(rawResponse.replace(/\`(?:json)?/g, "")).trim());
}

// Descriptive Question Generation Function
async function generateDescriptiveQuestions(description, topic, numQuestions) {
  const prompt = `Generate ${numQuestions} descriptive questions about ${topic}.
    Context: ${description}
    For each question, provide:
    1. The question text
    2. Sample Answer Text
    3. Key points that should be covered in the answer (3-5 points)
    4. Difficulty level (easy/medium/hard)
    Format the response as a plain JSON array of objects`;
  const model = genAI.getGenerativeModel({ model: "gemini-1.5-flash" });
  const result = await model.generateContent(prompt);
  const rawResponse = await result.response.text();
  return JSON.parse(rawResponse.replace(/\`(?:json)?/g, "")).trim();
}

// Content extraction from files
if (existingTopic.description) {
  topicContent = existingTopic.description;
} else if (existingTopic.fileData) {
  try {
    if (existingTopic.fileType === 'application/pdf') {
      const pdfData = await pdf(existingTopic.fileData);
      topicContent = pdfData.text;
    } else if (existingTopic.fileType === 'text/plain') {
      topicContent = existingTopic.fileData.toString('utf-8');
    } else {
      throw new Error('Unsupported file type');
    }
  } catch (error) {
    console.error(`Error extracting content from file: ${error}`);
  }
}
```

```
        }
    } catch (error) {
        console.error('Error processing file:', error);
        return res.status(500).json({
            success: false,
            message: "Failed to process topic content file",
            error: error.message
        });
    }
}
```

7.1.3 Question Storage

The system implements a robust database schema for storing generated questions. The schema is designed to accommodate both MCQ and descriptive questions, with conditional fields based on the question type. Each question is associated with an evaluation ID and includes metadata such as difficulty level and question type.

Code Snippet

```
// Base Question Fields Schema
const baseQuestionFields = {
    evaluationId: {
        type: mongoose.Schema.Types.ObjectId,
        ref: "Evaluation",
        required: true,
    },
    question: {
        type: String,
        required: true,
    },
    questionType: {
        type: String,
        enum: ["MCQ", "descriptive"],
    }
};
```

```
    required: true,  
  },  
  difficulty: {  
    type: String,  
    enum: ["easy", "medium", "hard"],  
    required: true,  
  },  
};  
  
// Complete Question Schema  
const questionSchema = new mongoose.Schema(  
{  
  ...baseQuestionFields,  
  options: {  
    type: [{ type: String }],  
    required: function () {  
      return this.questionType === "MCQ";  
    },  
  },  
  correctAnswer: {  
    type: String,  
    required: function () {  
      return this.questionType === "MCQ";  
    },  
  },  
  sampleAnswer: {  
    type: String,  
    required: function () {  
      return this.questionType === "descriptive";  
    },  
  },  
  keyPoints: {  
    type: [{ type: String }],  
  },
```

```
required: function () {
    return this.questionType === "descriptive";
},
},
},
{
timestamps: true
);
// Submitted Answer Schema
const submittedAnswerSchema = new mongoose.Schema({
usn: { type: String, required: true },
evaluationId: { type: String, required: true },
answers: [
{
questionId: { type: String, required: true },
answer: { type: String, required: true },
},
],
});
// Question Retrieval Endpoint
app.get("/api/questions/:evaluationId", async (req, res) => {
try {
const evaluation = await Evaluation.findById(req.params.evaluationId);
if (!evaluation) {
return res.status(404).json({ error: "Evaluation not found" });
}
const questions = await Question.find({
evaluationId: req.params.evaluationId,
});
res.status(200).json({
timeLimit: evaluation.timeLimit,
```

```
    questions: questions,
  });
} catch (err) {
  res.status(500).json({ error: err.message });
}
});
```

7.2 Web Module

7.2.1 Teacher Interface

The teacher web interface provides a comprehensive platform for educators to manage various aspects of the educational process. Through this interface, teachers can add and modify course titles and descriptions, manage student enrollment, and organize students into specific groups. Additionally, teachers can schedule tests, monitor student performance, and view detailed test results. The interface allows for efficient management of the learning environment, ensuring that educators can easily track progress and customize their teaching approach based on real-time data.

Code Snippet

```
import React, { useState, useEffect } from 'react';
import { useNavigate } from 'react-router-dom';
import './css/Title.css';
import Description from './Description';
const TitleAndText = () => {
  const navigate = useNavigate();
  const [name, setName] = useState("");
  const [message, setMessage] = useState("");
  const [isAvailable, setIsAvailable] = useState(null);
  const [debouncedName, setDebouncedName] = useState(name);
  const [showDescription, setShowDescription] = useState(false);
  const [errorMessageStatus, setErrorMsgStatus] = useState(false);
  const handleDescriptionSaved = () => {
    setShowDescription(false);
```

```
    setName("");
};

useEffect(() => {
  const handler = setTimeout(() => {
    setDebouncedName(name);
  }, 500);
  return () => {
    clearTimeout(handler);
  };
}, [name]);

useEffect(() => {
  if (debouncedName.trim() !== "") {
    checkNameAvailability(debouncedName);
  } else {
    setMessage("");
    setErrorMsgStatus(false);
    setIsAvailable(null);
  }
}, [debouncedName]);

const checkNameAvailability = async (inputName) => {
  try {
    const response = await fetch('http://localhost:5000/api/check-title', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify({ name: inputName }),
    });
    const result = await response.json();
    setIsAvailable(result.available);
  }
}
```

```
setMessage(result.message);
setErrorMsgStatus(false);
} catch (error) {
  console.error('Error checking name:', error);
  setMessage('An error occurred. Please try again.');
  setErrorMsgStatus(true);
  setIsAvailable(null);
}
};

const handleBlur = () => {
if (name.trim() === "") {
  setMessage('This field is required.');
  setErrorMsgStatus(true);
  setIsAvailable(null);
}
};

const handleButtonClick = () => {
if (isAvailable) {
  setShowDescription(true);
}
else
{
  navigate('/create-exam');
}
};

return (
<>
{!showDescription ? (
<div className="name-check-container" style={{marginTop:"20vh"}}>
<div className="name-check-content">
<h2>Check Title Availability</h2>
<input

```

```
type="text"
placeholder="Enter a topic for your test"
value={name}
onChange={(e) => setName(e.target.value)}
onBlur={handleBlur}
className="name-input"

/>

{message && (
  <p className={` message ${isAvailable ? 'available' : 'error'} ${errorMessageStatus ? 'alert1' : ''}`}>
    {message}
  </p>
)}

{name.trim() !== "" && message && (
  <button onClick={handleButtonClick} className={`action-button ${isAvailable ? 'available' : 'error'}`}>
    {isAvailable ? 'Continue to add Description': 'Schedule Test Now'}
  </button>
)}

</div>
</div>

):(
  <Description title={name} setShowDesc={handleDescriptionSaved}/>
)
</>

);

};

export default TitleAndText;
```

7.2.2 Student Interface

The student web interface is designed to provide learners with an intuitive platform to interact with course materials and track their performance. Students can attempt scheduled tests, view

their scores immediately after completion, and receive feedback on their results. The interface also allows students to view the groups they belong to, fostering collaboration and communication among peers. This interface ensures that students have easy access to important course information and progress updates, helping them stay engaged and informed throughout their learning journey.

Code Snippet

```
import React, { useEffect, useState } from "react";
import axios from "axios";
import { useNavigate } from "react-router-dom";
import Welcome from "./Welcome";
import Navbar from "./Navbar";
import "../css/StudentHome.css"; // Import CSS file
import { Link } from 'react-router-dom';
const StudentHome = () => {
  const [usn, setUSN] = useState(null);
  const [evaluations, setEvaluations] = useState([]);
  const [groups, setGroups] = useState([]);
  const [student, setStudent] = useState(null);
  const [studentName, setStudentName] = useState(null);
  const [loginSuccess, setLoginSuccess] = useState(false);
  const navigate = useNavigate();
  const token = localStorage.getItem("jwtToken");

  // Fetch data on component mount
  useEffect(() => {
    const fetchData = async () => {
      try {
        const response = await axios.get(
          "http://localhost:5000/api/evaluations/home",
        );
        headers: {

```

```
        Authorization: `Bearer ${token}`,
    },
}

setStudent(response.data.student);
setStudentName(response.data.student.Name);
setUSN(response.data.student.USN);
setGroups(response.data.groups || []);
setEvaluations(response.data.evaluations || []);
setLoginSuccess(true);

} catch (error) {
    console.error("Error fetching data:", error);
    setLoginSuccess(false);
}

};

if (token) {
    fetchData();
}

}, [token]);
// Navigation handlers

const handleTestClick = (evaluationId) => {
    navigate(`/test/${evaluationId}`);
};

const handleResultsClick = (evaluationId) => {
    navigate(`/results/${evaluationId}/${usn}`);
};

// Render logic if not logged in
if (!loginSuccess) {
    return (
        <div className="message-container">
            Please log in to continue.
    </div>
);
}
```

```

<br />
<Link to="/login">Go to Login Page</Link>
</div>
);
}

// Render logic if student info is not found
if (!student) {
    return (
        <div className="message-container">Student information not found.</div>
    );
}
return (
    <>
    {/* Navbar and Welcome Section */}
    <Navbar />
    <Welcome title="EduAssessMate" username={studentName} />
    {/* Groups Section */}
    <div className="section mx-5 mt-3">
        <h3>Your Groups</h3>
        {groups.length > 0 ? (
            <div className="card-grid">
                {groups.map((group) => (
                    <div key={group._id} className="card">
                        <h5 className="card-title">{group.groupName}</h5>
                        <p className="card-text">Members: {group.students.length}</p>
                    </div>
                )));
        ) : (
            <p className="empty-message">No groups found.</p>
        )}
    </div>
)
}

```

```
/* Evaluations Section */
<div className="section mx-5">
  <h3>Your Assessments</h3>
  {evaluations.length > 0 ? (
    <div className="card-grid">
      {evaluations.map((evaluation) => (
        <div key={evaluation._id} className="card">
          <h5 className="card-title">{evaluation.title}</h5>
          <p className="card-text">Topic: {evaluation.topic}</p>
          <p className="card-text">
            Scheduled For: {evaluation.scheduleType}
          </p>
          <p className="card-text">
            Time Limit: {evaluation.timeLimit} minutes
          </p>
          {evaluation.status === "active" && (
            <button
              onClick={() => handleTestClick(evaluation._id)}
              className="btn primary"
            >
              Take Test
            </button>
          )}
          {evaluation.status === "completed" && (
            <button className="btn secondary" disabled>
              Evaluation Pending
            </button>
          )}
          {evaluation.status === "evaluated" && (
            <button
              onClick={() => handleResultsClick(evaluation._id)}
              className="btn success"
            >

```

```
        See Results
    </button>
)
);
</div>
))}

</div>

):(
<p className="empty-message">No Assessments found.</p>
)}
</div>
</>
);

};

export default StudentHome;
```

7.3 Evaluation Module

7.3.1 Submission Storage

The system stores student submissions securely in MongoDB, including their answers for both MCQ and descriptive questions.

Code Snippet

```
app.post("/api/submit-answers", async (req, res) => {
  try {
    const { usn, evaluationId, answers } = req.body;
    const submittedAnswer = new SubmittedAnswer({
      usn,
      evaluationId,
      answers,
    });
    await submittedAnswer.save();
    res.status(201).json({ message: "Answers submitted successfully!" });
  } catch (err) {
```

```
    res.status(500).json({ error: err.message });
  }
});
```

7.3.2 Answer Evaluation

Uses Google's Gemini API to evaluate descriptive answers based on key points and sample answers. MCQs are evaluated against correct answers.

Code Snippet

```
const evaluateSubmittedAnswers = async (evaluationId) => {
  try {
    for (const submission of submissions) {
      for (const answer of submission.answers) {
        if (question.questionType === "MCQ") {
          marksAwarded = question.correctAnswer === answer.answer ? 1 : 0;
        } else if (question.questionType === "descriptive") {
          const prompt = `
            Evaluate the following descriptive answer...
          `;
          const result = await model.generateContent(prompt);
          const { marksAwarded: aiMarks } = JSON.parse(rawResponse);
          marksAwarded = aiMarks || 0;
        }
        totalMarks += marksAwarded;
      }
    }
  } catch (error) {
    return { success: false, message: error.message };
  }
};
```

CHAPTER-8

TESTING

Testing is a critical phase in the development lifecycle of the AI-driven Student Assessment Platform, ensuring that the system functions as intended and meets all requirements. Various testing methodologies will be applied to validate the platform's functionality, performance, security, and usability.

8.1 Types of Testing

8.1.1 Unit Testing

- Focuses on verifying the smallest functional units of the system, such as question generation algorithms, grading modules, and analytics components.
- Ensures that each module works independently without errors.

8.1.2 Integration Testing

- Validates the interaction between different modules, such as the integration of the AI-driven question generator with the grading and reporting system.
- Ensures seamless data flow across components.

8.1.3 System Testing

- Examines the system to ensure it meets functional and non-functional requirements.
- Includes testing end-to-end workflows, such as creating an assessment, conducting it securely, grading responses, and generating analytics reports.

8.1.4 Performance Testing

- Evaluates the system's response time, scalability, and ability to handle concurrent users.
- Includes load testing to simulate high-traffic scenarios and stress testing to determine the system's limits.

8.1.5 Security Testing

- Assesses the system's ability to protect sensitive data and maintain integrity during exams.
- Includes tests for vulnerabilities, encryption protocols, and anti-cheating mechanisms.

8.1.6 User Acceptance Testing (UAT)

- Involves end-users (educators and students) testing the system to validate its usability and effectiveness.
- Ensures the system meets real-world needs and expectations.

8.2 Testing Tools and Frameworks

- **Unit Testing Frameworks:** PyTest (Python), JUnit (Java).
- **Integration Testing Tools:** Postman for API testing, Selenium for UI testing.
- **Performance Testing Tools:** JMeter, Locust.
- **Security Testing Tools:** OWASP ZAP, Burp Suite.
- **User Feedback Tools:** SurveyMonkey, Google Forms for gathering user feedback during UAT.

8.3 Testing Process

8.3.1 Test Planning

- Define test objectives, scope, and success criteria.
- Develop detailed test cases and scenarios based on functional and non-functional requirements.

8.3.2 Test Execution

- Execute test cases across all testing phases (unit, integration, system, etc.).
- Record and document test results, including any identified defects.

8.3.3 Defect Tracking and Resolution

- Use tools like Jira or Bugzilla to track and manage defects.
- Assign issues to the development team for resolution and retest after fixes.

8.3.4 Test Reporting

- Prepare comprehensive test reports summarizing the results, defects, and overall system readiness.
- Highlight critical issues and potential risks affecting system functionality.

8.4 Expected Outcomes

- Identification and resolution of all critical defects before deployment.
- Verification that the system meets all functional and non-functional requirements.
- Validation that the system performs efficiently under expected workloads.
- Confirmation that the platform is secure, scalable, and user-friendly.

By conducting thorough testing, the AI-driven Student Assessment Platform will ensure a robust, reliable, and high-quality system that delivers an exceptional user experience for educators and students.

CHAPTER-9**TEST CASES AND RESULTS (SNAPSHOTS)****Table 9.1: Unit Testing Test Cases**

Test Case No.	Description	Input	Expected Output	Obtained Result	Status	Remarks
UT-1	Login Module	Login details	Password matched	Login Successful	PASS	Login Successful
UT-2	Login Module	Incorrect Login details	Password didn't matched	showing the message	PASS	Enter the correct credentials
UT-3	File Upload	Input: PDF file	File uploaded successfully	File uploaded	PASS	File uploaded successfully
UT-4	Evaluation	Submitted answers	Scores returned	Scores returned	PASS	NA
UT-5	View Results	Test ID	Results displayed correctly	Results displayed correctly	PASS	NA
UT-6	Timer	Exam start and end times	Accurate tracking and logging	Timings logged	PASS	NA

Table 9.2: Integration Testing Test Cases

Test Case No.	Description	Input	Expected Output	Obtained Result	Status	Remarks
IT-1	Attempting Test	Schedule test details	Test details stored in DB	Test details stored	PASS	NA
IT-2	Submitting Answers	Submitted answers	Sent to Gemini API	Sent successfully	PASS	NA
IT-3	Evaluating Answers	API call for evaluation	Evaluated scores	API returned scores	PASS	NA

Table 9.3: System Testing Test Cases

Test Case No.	Description	Input	Expected Output	Obtained Result	Status	Remarks
ST-1	Attempting Test	Student login and test taken	Test completed successfully	Test completed successfully	PASS	NA
ST-2	Security Testing	Invalid test access by unauthorized user	Access denied	Access denied	PASS	Ensure strict access control mechanism

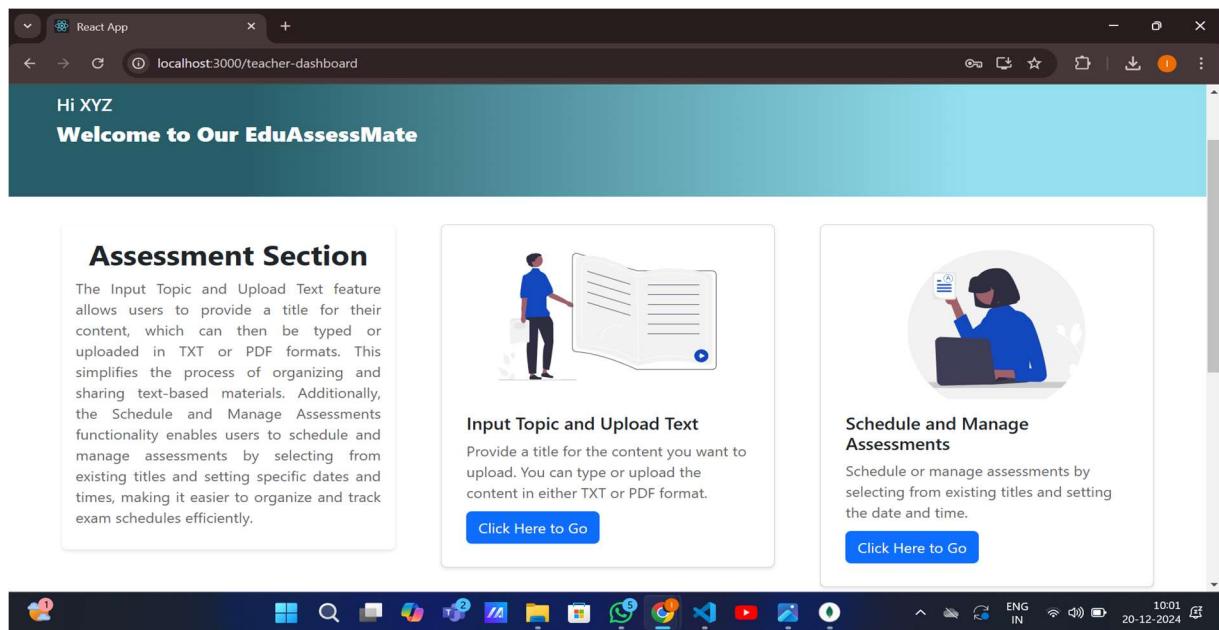


Fig. 9.1: Teacher Dashboard

The Fig. 9.1 shows a dashboard page which provides teachers with several features to manage groups, track student performance, and streamline administrative tasks.

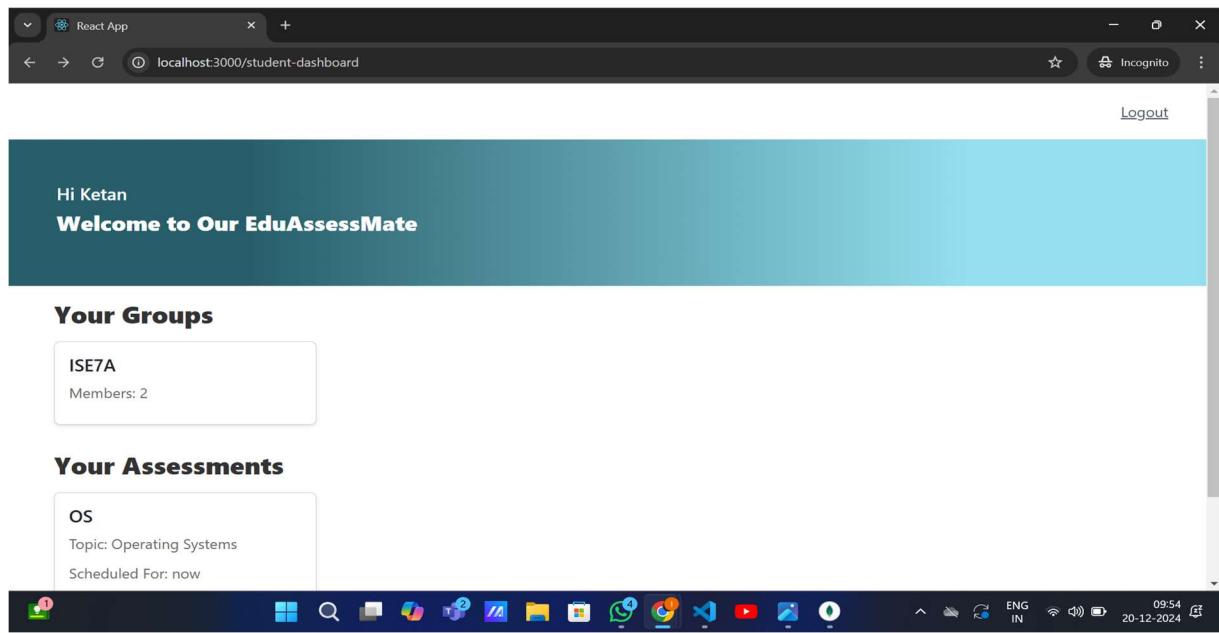


Fig. 9.2: Student Dashboard

The Fig. 9.2 shows a student dashboard page in which students can access their groups, submit their assessment test and check their results.

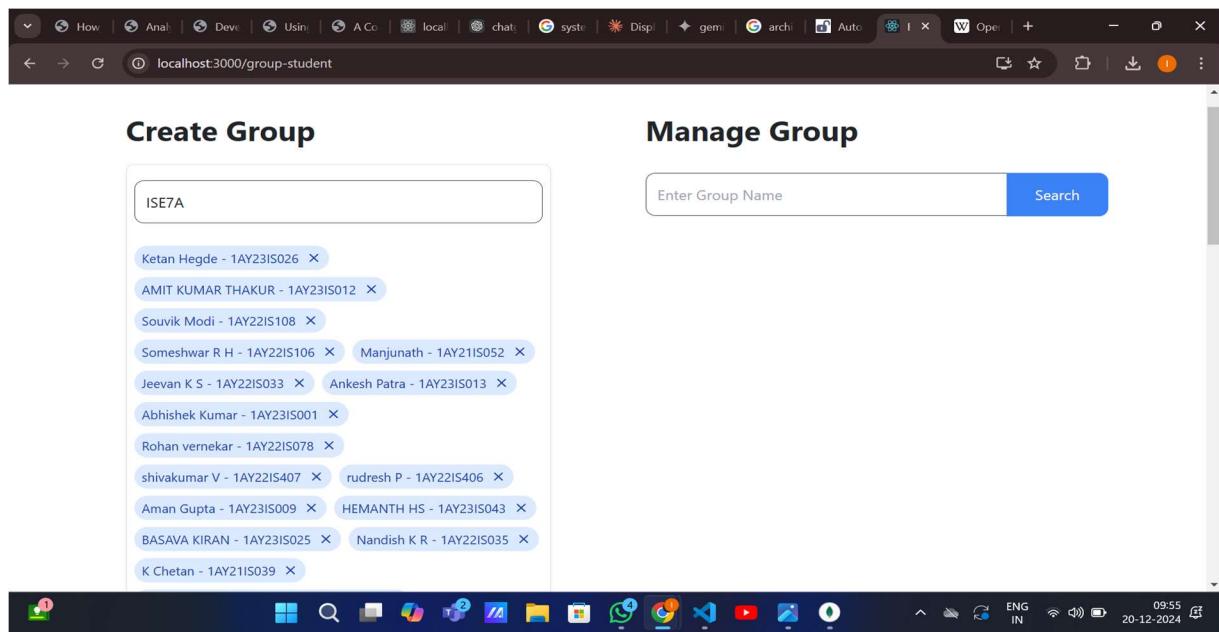


Fig. 9.3: Creation of Groups

The Fig. 9.3 shows how the teachers can create groups, manage groups and perform the CRUD operations.

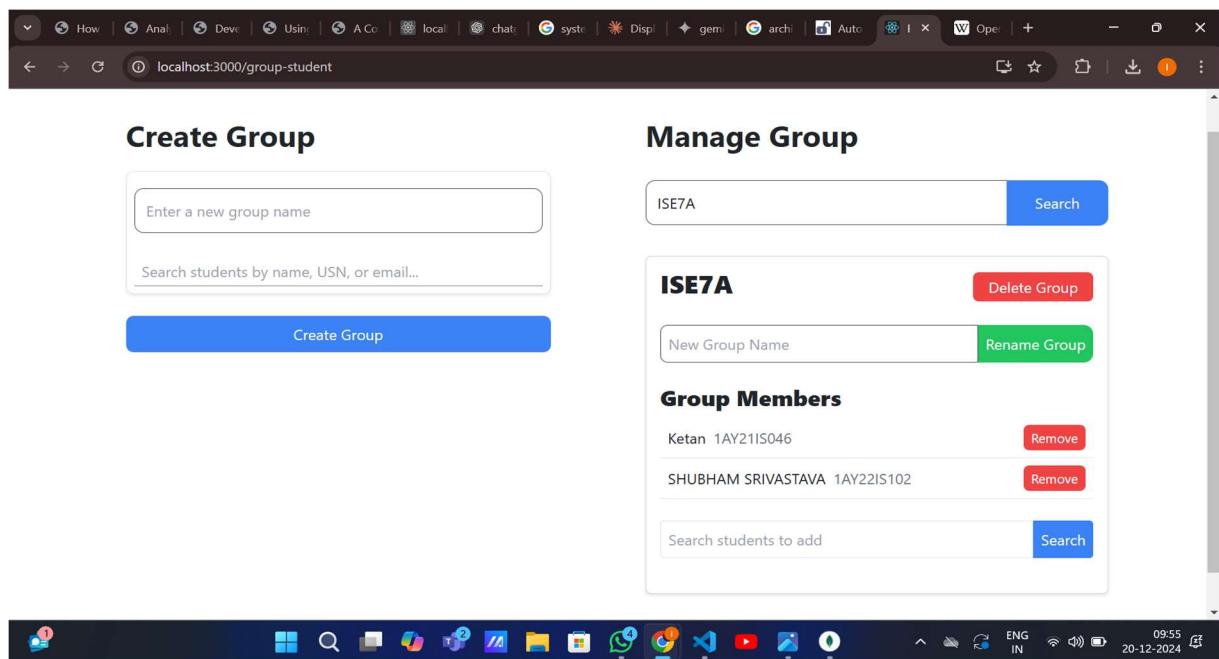


Fig. 9.4: Managing Student Groups

The Fig. 9.4 shows how the teachers can manage the student's group after creation of the groups which gave them to create, update, rename and search functionalities.

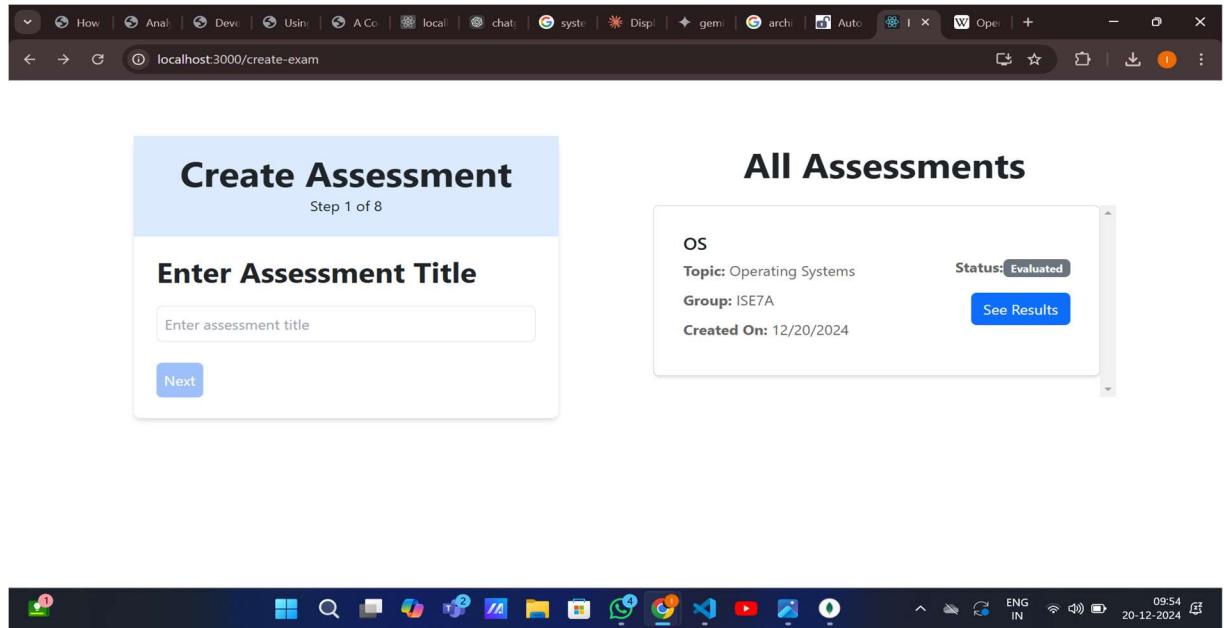


Fig. 9.5: Creating Assessment

The Fig. 9.5 shows the creation of the assessment process in 8 easy steps for the starting from the entering the assessment title till the final creation of the assessment.

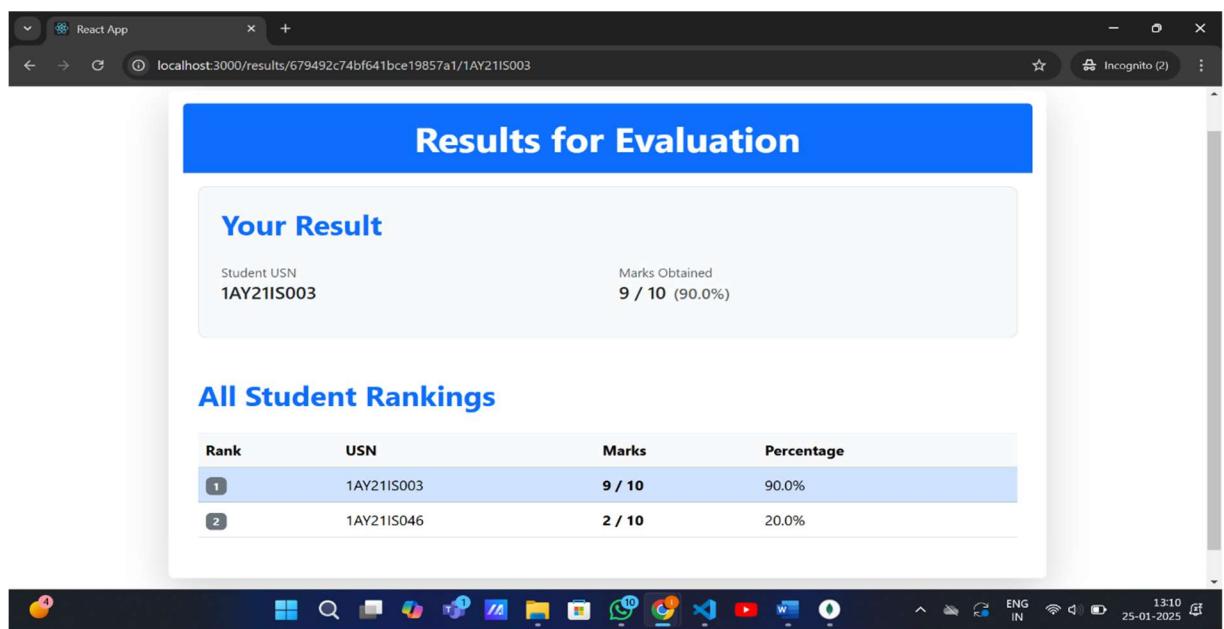


Fig. 9.6: Detail Report

The Fig. 9.6 shows the detailed report of the students of the group which includes the personalized report and overall class group rank.

CONCLUSION

The AI-driven student assessment platform offers a revolutionary approach to modernizing educational assessments by automating essential processes such as question generation, grading, and performance tracking. With the integration of AI, the platform enhances the efficiency and scalability of exams while ensuring a secure and fair testing environment. Educators benefit from reduced manual effort as the system automatically generates diverse, curriculum-aligned questions, thus saving valuable time. Furthermore, advanced anti-cheating mechanisms, including full-screen mode, timers, and AI-powered techniques, ensure the integrity of the exams. The platform also provides real-time analytics that empower educators to make data-driven decisions and offer personalized learning experiences. By streamlining these processes, the platform improves the student learning experience, provides timely feedback, and enhances the overall security of the testing environment.

FUTURE ENHANCEMENT

Looking ahead, several enhancements could further strengthen the effectiveness of the platform. One promising direction is the integration of adaptive learning algorithms, which would adjust the difficulty of questions based on individual student performance, fostering a more personalized learning experience. Additionally, incorporating real-time feedback could further support continuous student engagement by enabling instant insights into performance and encouraging improvement. Enhanced AI-powered analytics could provide educators with deeper insights into student performance, trends, and learning patterns, enabling them to make more informed decisions about teaching methods and interventions. Integration with Learning Management Systems (LMS) would streamline workflows by synchronizing data across platforms, offering a more cohesive experience for educators and students. Furthermore, integrating advanced AI-driven anti-cheating technologies such as facial recognition, eye-tracking, and behavioral analysis would significantly enhance exam security. Lastly, improving the user interface to make it more intuitive and engaging would enhance the overall user experience, promoting wider adoption of the platform across educational institutions. These enhancements would not only increase the platform's effectiveness but also ensure it adapts to the evolving needs of modern education.

BIBLIOGRAPHY

- [1] H. N. Patel, P. Goel, A. Surti, and B. Patel, "A Comparative Analysis of Large Language Models with Retrieval-Augmented Generation based Question Answering System," in Proceedings of the International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC 2024), Changa, India, 2024.
- [2] V. M. Ionescu and M. C. Enescu, "Using ChatGPT for Generating and Evaluating Online Tests," in Proceedings of the 15th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Pitesti, Romania, 2023.
- [3] L. Hrvačević, V. Jocović, J. Cincović, V. Matvejev, A. Milaković, and D. Drašković, "Development of a web system with an automated question generator based on large language models," in Proceedings of the 11th International Conference on Electrical, Electronics and Computer Engineering (IcETRAN), Nis, Serbia, 2024.
- [4] S. Elkins, E. Kochmar, J. C. K. Cheung, and I. Serban, "How Teachers Can Use Large Language Models and Bloom's Taxonomy to Create Educational Quizzes," 2024.
- [5] J. Prentzas and M. Sidiropoulou, "Assessing the Use of OpenAI Chat-GPT in a University Department of Education," in Proceedings of the 14th International Conference on Information, Intelligence, Systems & Applications (IISA), Alexandroupolis, Greece, 2023.



Department of Information Science and Engineering

Acharya Institute of Technology

Affiliated to VTU, Recognized by GOK and Approved by AICTE, New Delhi (Accredited by NAAC and NBA)
Acharya Dr. Sarvepalli Radhakrishnan Road, Acharya P.O., Soladevanahalli, Bangalore-560107, INDIA
www.acharya.ac.in

Project Work Outcome

Achievement of Objectives

Objective	Status	Remarks
Automate Question Generation	Achieved	Implemented AI-based question generation using Gemini API to create multiple question formats.
Simplify Test Scheduling	Achieved	Automated scheduling process with conflict prevention and real-time notifications for stakeholders.
Enhance Exam Security	Achieved	Incorporated features such as full-screen mode, browser lockdown, and facial recognition.
Automate the Grading Process	Partially Achieved	Grading automation was achieved for multiple-choice questions and descriptive answers via NLP, but human oversight is still necessary for complex responses.
Provide Real-Time Analytics	Achieved	Real-time analytics provided actionable insights into student performance and learning trends.
Develop a Scalable Platform	Achieved	The platform supports scalability and multi-user environments, ensuring seamless performance.
Enhance Student Experience	Achieved	Personalized feedback, immediate results, and an intuitive user interface improve the student experience.

Future Scope

- Integrate adaptive learning algorithms to adjust question difficulty based on student performance.
- Implement eye-tracking and behavioural analysis for advanced anti-cheating measures.
- Synchronize the platform with Learning Management Systems (LMS) for seamless integration.
- Offer continuous real-time feedback during assessments for better student engagement.
- Enhance the UI/UX design with multilingual support for a more engaging and accessible platform.



Department of Information Science and Engineering Acharya Institute of Technology

Affiliated to VTU, Recognized by GOK and Approved by AICTE, New Delhi (Accredited by NAAC and NBA)
Acharya Dr. Sarvepalli Radhakrishnan Road, Acharya P.O., Soladevanahalli, Bangalore-560107, INDIA
www.acharya.ac.in

Final Deliverables

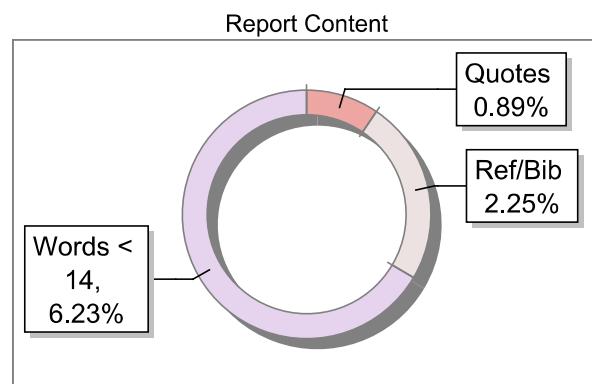
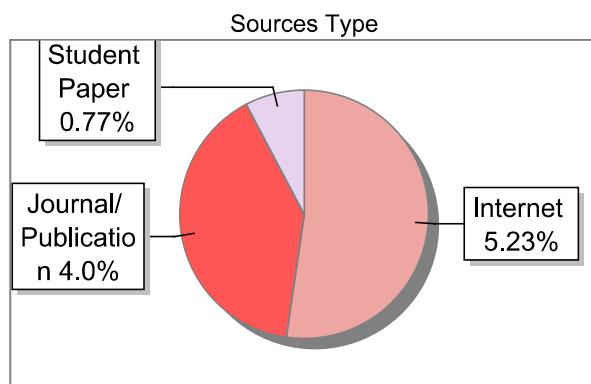
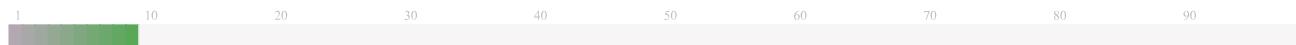
- Fully Functional Prototype
- Comprehensive Project Report
- Published Paper: Our paper is ready for submission, and we are awaiting the right journal for publication.
- Code repository (GitHub Link):
https://github.com/AbhishekMangalur/Transforming_Student_Evaluation_with_Adaptive_Intelligence_and_Performance_Analytics

Submission Information

Author Name	Chetan Maning Badachi
Title	Transforming Student Evaluation with Adaptive Intelligence and Performance Analytics
Paper/Submission ID	2992612
Submitted by	asklibrarian@acharya.ac.in
Submission Date	2025-01-17 12:52:28
Total Pages, Total Words	45, 8429
Document type	Project Work

Result Information

Similarity **10 %**



Exclude Information

Quotes	Excluded
References/Bibliography	Excluded
Source: Excluded < 14 Words	Not Excluded
Excluded Source	0 %
Excluded Phrases	Not Excluded

Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File





DrillBit Similarity Report

10

SIMILARITY %

62

MATCHED SOURCES

A

GRADE

- A-Satisfactory (0-10%)
- B-Upgrade (11-40%)
- C-Poor (41-60%)
- D-Unacceptable (61-100%)

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	www.ijraset.com	1	Publication
2	docplayer.net	<1	Internet Data
3	ijcsrr.org	<1	Publication
4	www.rajeevelt.com	<1	Publication
5	dsc.duq.edu	<1	Publication
6	www.linkedin.com	<1	Internet Data
7	www.ijraset.com	<1	Internet Data
8	www.swamivivekanandauniversity.ac.in	<1	Publication
9	REPOSITORY - Submitted to Exam section VTU on 2024-07-31 16-04 912380	<1	Student Paper
10	Submitted to Visvesvaraya Technological University, Belagavi	<1	Student Paper
11	www.dx.doi.org	<1	Publication
12	www.mdpi.com	<1	Internet Data
13	mediawebgp.ir	<1	Internet Data
14	nextgeninvent.com	<1	Internet Data

Transforming Student Evaluation with Adaptive Intelligence and Performance Analytics

Assistant Prof. Pushpalatha K S¹

Dept. of ISE

Acharya Institute of Technology
Bengaluru (Affiliated to VTU, Belagavi),
Karnataka, India
pushpalatha23691@acharya.ac.in

Abhishek Mangalur²

Dept. of ISE

Acharya Institute of Technology
Bengaluru (Affiliated to VTU, Belagavi),
Karnataka, India
abhimangalur1@gmail.com

Ketan Hegde³

Dept. of ISE

Acharya Institute of Technology
Bengaluru (Affiliated to VTU, Belagavi),
Karnataka, India
ketanhedge2003@gmail.com

Chetan Badachi⁴

Dept. of ISE

Acharya Institute of Technology
Bengaluru (Affiliated to VTU, Belagavi),
Karnataka, India
badachichetan@gmail.com

Mohammad Aamir⁵

Dept. of ISE

Acharya Institute of Technology
Bengaluru (Affiliated to VTU, Belagavi),
Karnataka, India
mohdaamir4141@gmail.com

Abstract — The development in Artificial Intelligence (AI) offers transformative potential for redefining student assessment methodologies. This paper aims to establish the idea of the advancement of Artificial Intelligence (AI) and its prospect in reshaping approaches to assessing students. It creates a system for the evaluation of students' performance using Artificial intelligence, and particularly the Gemini API for the generation of questions, grading and report on the student's performances. This is to facilitate easy use of the tools in creating, scheduling, and delivering assessments with minimal chances of cheating through options such as full screen and time limit. There are formats of questions in the system which comprises multiple choice, short answers and descriptive questions, developed by Gemini. The most conspicuous feature is the self-checking system whereby the user gets instant feedback for the correct score that each of the students would have scored instantly with explanations about wrong answers. Moreover, the platform has intelligent learning progressions where the user will be able to monitor his/her performances to be recommended a certain level of performance. It will allow students as well as educators to have real-time analytics and feedback on what they are good at and where they need to improve. Not only does it make the assessment easier, but it also improves the levels of accuracy in grading and effectively strengthens a data-based learning process for students.

Keywords — Artificial Intelligence, Student Assessment, Automation, Performance Analytics, Anti-cheating Mechanisms, Adaptive Learning, Gemini API.

I. INTRODUCTION

Traditionally, evaluation of students has been rooted in common practices that include preparation of questions, setting of examinations, grading as well as monitoring of student performance. However, they are labor intensive, time consuming and are vulnerable to development of human error [3]. For instance, there are common challenges as the consistency in grading, fairness in the assessment and the capacity to design an assessment that can meet the needs of every learner. In addition, the modern education is incredibly sophisticated to be evaluated by the existing methods in the present days.

Along the rapid advancement of Artificial Intelligence (AI) and machine learning technologies, there are learning opportunities that may reshape the educational processes. Informatics devices and programs that claim to do field work, provide individual recommendations or perspective from the vast big data. In other words, they establish this instrument as a useful solution to break the traditional student evaluation system constraints. Thanks to this, it is possible to create the environment of the adaptive, efficient and at the same time safe assessment of the results in which each participant would receive the benefit.

A. Objective

Develop an AI-powered student assessment system that would allow the implementation of the major assessment stages and would maintain student assessment integrity and information confidentiality at the highest level. The target areas are:

- Automated Question Generation

Gemini generates many forms of questions, including multiple choice, short and significant questions, and descriptions among them. These questions therefore meet the curriculum objectives and the learning standards as per curriculum set by different institutions of learning.

- Exam Scheduling and Management

It makes the process of managing the whole range of assessments easy and convenient for the educators, thus creating a smooth workflow [1].

- Automated Grading

The system delivers consistent, accurate, and unbiased grading for different question formats. Students get an instant and personalized feedback to support their learning process [4].

- Anti-Cheating Mechanisms

To eliminate any cheating on the evaluation, the platform has enabled some measures such as full screen, browser lock, and time limit for the assessments.

Additionally, this research aims to integrate real-time performance analytics and adaptive learning pathways. The above facilitate the understanding of the progress, they also provide suitable suggestions as well facilitate lifelong learning for students. This ensures that educators can analyze the student performance and be able to make some useful changes to their strategies in a bid to enhance the teaching methodology.

B. Significance

This has a great prospect to compel educators to better focus on other higher-order tasks and responsibilities, in essence, curriculum enhancement, learners support, among other functions as some of the time-consuming routine tasks can be effectively addressed through such applications. The goal of this approach is to make the assessment fair and each student would be able to give their best performance to get the best result.

Additionally, the platform's integration of real-time performance analytics fosters a data-driven approach to education. Teachers can monitor the strengths, weaknesses and learning behavior of the students, which can be easily followed and differentiated between those fast learners and those that need special focus. On the other side the platform also offers the accommodation in the context of students' continuous enhancement and students' self-

motivated learning as well as the more creative and individualized educational approach for the students.

This work is an important step towards bridging the gap in modern learning ecosystem to enable proper utilization of resources for the evaluation techniques. The proposed platform would have benefits from AI, performance analytics, and that would make the process of assessing students more efficient and impactful, as well as equitable in the way the process of evaluating students is done.

II. BACKGROUND AND RELATED WORKS

A. Development of a Web System with an Automated Question Generator Based on Large Language Models [1]

This work proposes a web system for generating questions and answers based on LLM models made for several fields and where Serbian Wikipedia serves as the knowledge source. The idea of applying the LLM model in generating questions and answers makes it quite a unique way of creating quizzes. The incorporation of Wikipedia as the source of knowledge also enhances the quality of the generated questions. Not only can the web system generate questions for learners, but it also is capable of solving knowledge quizzes. A new set of questions for the quiz can be posted every one hour. Such functionality can help users to learn something new and enhance their skills in certain areas. Users can also keep track of their progress through response statistics. The web system can find its place in numerous platforms used for knowledge assessment. In general, this application brings together the LLM model, information from Wikipedia, and engaging quiz questions that enhances user learning.

B. Using ChatGPT for Generating and Evaluating Online Tests [2]

This paper aimed to assess the latest research on the quiz generation and its evaluation based on the ChatGPT-3 NLM model. From the literature research it was found out that the Quiz generation aspect is well discussed in literature since the launch of ChatGPT papers. It was established that the most significant issues were that although the model was developed and tested using a specific format of prompt, even minor changes to the format rendered the model wordy, unfocused and directionless about the task at hand. However, the results of the quiz were discussed in far fewer papers, and the primary focus was made on the summary production. The papers, which actually employed and experimented with NLPs for assessment, were few and highlighted the uncontrollability of the model because it changes the answer pattern even to similar questions, and the restrictive bias of the answers owing to the black box policy of the algorithm.

C. A Comparative Analysis of Large Language Models with Retrieval-Augmented Generation-Based Question Answering System [3]

This paper compares how efficiently Large Language Models (LLMs) are able to acquire and incorporate specific domain knowledge into their model. The Retrieval-Augmented Generation (RAG) system mitigates this by incorporating specific data, searching for relevant information from external knowledge resources, and improving the LLM's knowledge base for better responses. This research compared GPT-3.5 turbo, Gemini-Pro, and Llama-3 using the RAGAS evaluation framework in a RAG-based Question-Answering system. The evaluation focused on the performance comparison, where GPT-3.5-turbo stood out as the most effective model in using the retrieved data to enhance the answer quality, relevance, and correctness. Subsequent studies in this area might focus on improving the means by which information gets fed into LLMs with better filter and sort routines.

D. Assessing the Use of OpenAI Chat-GPT in a University Department of Education [4]

This paper elaborates on some issues related to the application of Chat GPT especially in a University Department of Education. A current investigation is underway in which the Department's students and its members will evaluate Chat GPT. Findings presented are based on a sample of an undergraduate course so they should be viewed as preliminary. It will therefore be useful to analyze tools like Chat-GPT in a bid to identify how they can be used bearing in mind the various negative outcomes they may entail. The dissemination of the results of the assessment will be useful to tool developers, education policymakers, teachers, students, and parents of minor students.

E. How Teachers Can Use Large Language Models and Bloom's Taxonomy to Create Educational Quizzes [5]

The purpose of this paper is to assert that it is possible for LLMs to generate various question types from a given context that would be just as effective as a written quiz that the teacher writes themselves. To achieve this objective, a series of quiz-writing experiments involving handwritten, simple, and controlled quizzes were undertaken. The controlled quizzes employed questions formulated to fit Bloom's taxonomy as a guide to categorization. Broadly and conclusively, the findings indicate that teachers exhibited a preference for writing quizzes with the use of controlled generations. They also mimic more of the controlled generations than the simple generations, meaning that these questions are of higher quality, or better for a teacher's interests. This supports our hypothesis that teachers benefit from automatically generated pedagogical questions for quiz development. Furthermore, the assessment of quiz quality revealed that the quizzes with

controlled and simple generations are equally effective. Some are even indicative of their quality, especially when considered against handwritten quizzes. These findings will also guide the future of educational QG research in the direction that will benefit the targets and objectives of students and teachers.

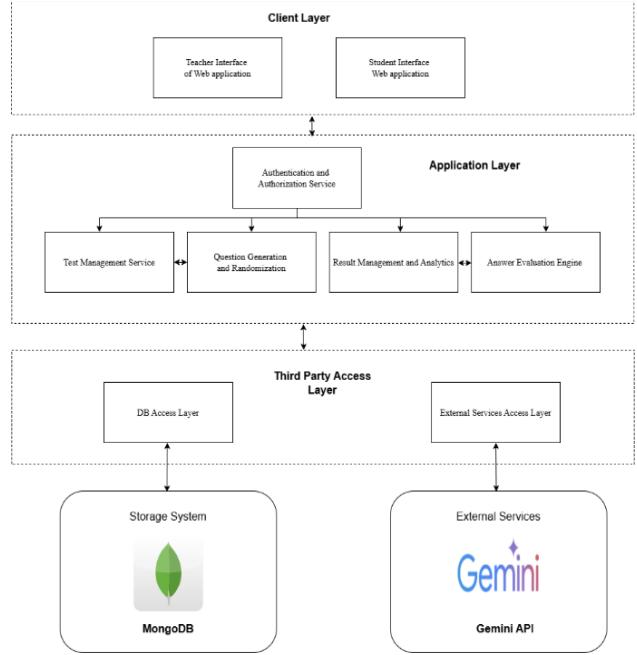


Fig. 1. System Architecture

III. METHODOLOGY

The proposed system incorporates the Gemini API in enhancing the conventional student assessment system to be more effective, efficient, secure, and structured. The approach concerns itself with the major evaluation activities and structures that affect the performance of students whilst guaranteeing each learner an equitable process. The following components and methodologies are used in formulation and implementation of the system:

A. AI-Driven Question Generation

To meet curriculum standards and specific learning objectives, the platform leverages the Gemini API to automatically generate questions. These questions may consist of multiple-choice questions, short questions, and long questions that may be descriptive in nature which are commanded by the teachers. The topics aspect of the Gemini API guarantees that the questions used are diverse, relevant, and meet the assigned levels of difficulty to enable manufacturing effective assessment detail. The system also uses contextual analysis to formulate questions that bring critical analysis and other higher orders of questions such as problem solving. A dynamic aspect

enables the variation of question level according to the identified ability level of students.

B. Anti-Cheating Mechanisms

To ensure assessment integrity and fairness, multiple anti-cheating measures are implemented to ensure that none of the students engage in cheating. To enhance the security of student assessments full-screen mode, browser locking, and disabling keys and clicking are incorporated active features, and the use of time limits to avoid excessive student engagement with material is implemented as a passive feature. By implementing these features, it provides a trustworthy means to assess the outlined assessments.

C. Performance Analytics

The platform augments the performance of the service through real-time performance tracking. With the use of predictive analysis, the system offers convenient outcome insights in the educational sector to the educators and students. It facilitates performance analysis that reveals students' strengths and all the aspects that require attention making it easy for the teachers to make the necessary decisions [2]. Students get such feedback along with the rate and pattern of performance analysis with directions for possible improvement for the specific student group. The analytics framework fosters continuous learning while also promoting transparency and accountability in the educational process.

D. Automation and Efficiency

AI-driven automation facilitates the entire evaluation process, including question development, grading, and exam scheduling. The platform can help assist educators in the way the platform provides an algorithm that makes it easier for the teachers to create lessons, thus easing their burden. This puts the teachers in a better position to perform professional assignments such as improving students' motivation, and coming up with new appropriate strategies of assessing students with a view of improving performance. Overall, the system allows for constant evaluations and causes no time lags as well as very few errors. Also, adaptive exam templates have been included to reflect on to the level of difficulty of the topic covered in the assessment as well as having the ability to be flexible in its assessment but at the same time being precise in its measures.

E. Scalability

The platform is extensible with the ability to modify and expand it in accordance with the number of users and assessments. AI helps to optimally allocate resources thus; the system can effectively cope with several hundred of users. It is flexible which enables it to be applicable in all levels ranging from individual institutions to a country's

different educational system. Furthermore, it has multilingual as well as regional adaptability that makes it useful in various learning environments. This makes the platform flexible in a way that it can accommodate other tests and introduce more domain subjects, to answer educational evolution.

F. Security and Integrity

The assessments are structured to maintain standards of credibility and fairness for the tasks assigned to the techniques in the platform. Real-time monitoring, artificial intelligence in proctoring, along with behavioral analysis allow to identify and prevent cheating during exams. There are key security measures such as toggling the full-screen mode, the disabling of right-click, tab-closing and any other shortcut that may lead to test leakage in case someone tries to cheat. To ensure and uphold the integrity of the responses submitted, plagiarism check on the written answers is strengthened. These security measures enhance confidence and integrity in assessment to students as well as the educators who assess them.

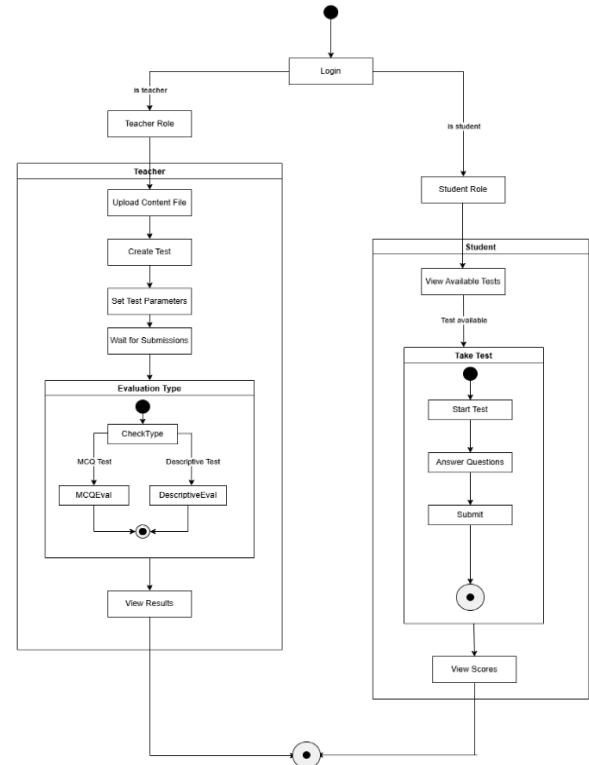


Fig. 2. Activity Diagram

IV. IMPLEMENTATION

The implementation of the online assessment system involved integrating several modules to create a seamless and interactive platform for both teachers and students.

A. Teacher Module

It allows teachers to contribute in the loading of content, creation of tests, setting of evaluation criteria and results. Some of the activities and features of the module are as follows:

- Login and Role Authentication: It can only be accessed by the Teachers through a secure login system to access administrative features.
- File upload: Teachers can upload educational content in various formats to support learning contents.
- Test creation: Test can be created as a type of dynamic form, one can select the type of the test (MCQ or descriptive), the number of questions and grading parameters.
- Setting Test Parameters: To maintain validity, the teachers can decide on the duration of an assessment, the time range within which the test must be completed.
- Evaluation Management: All the answers will be evaluated by the management suite of Gemini.

B. Student module

This involves the student who can also be able to check available tests, take up tests, and check his/her scores. The process of the flowchart for the student module are as follows:

- Login and Role Authentication: Students log in securely to access the platform.
- View Available Tests: A list of tests for courses taken is provided to the students [1].
- Test Participation: Students can initiate a selected test within the stipulated time for the test.
- Result Viewing: It allows the students to see full analysis and outcome of the evaluation they have gone through.

C. Backend Services

The backend of the system was implemented using Node.js with a MongoDB database for data storage.

D. Evaluation and Security Features

The four levels of security features, which allow the achievement of more reliable results of examination, are Full-Screen mode, Tab-Switch detector, Test Submission automation and data protection [2].

V. EVALUATION PROCESS

The evaluation of the online assessment system involves the use of the Gemini API to facilitate the correct assessment of the MCQs as well as the scoring of the descriptive questions. It is necessary to note that this method of evaluation is fully automated and can be deployed in the real-time mode, while utilizing AI technologies.

A. Automated MCQ Evaluation

The evaluation of MCQ questions is conducted entirely by the Gemini API. The process follows these steps:

- Answer Matching: The Gemini matches student answers against the correct options.
- Automated Grading: All the respective questions carry full marks if a student gets the correct answer. In cases of an incorrect or unanswered question, no marks are awarded (or negative marking, if configured).
- Instant Results: The Gemini results instantly and returns the total scores of the students, rank and time taken among others.
- Performance insights: The teachers as well as students both receive detailed feedback, including response accuracy and performance pattern.

B. Automated Descriptive Answer Evaluation

Descriptive answers, which traditionally require human evaluation, are assessed using the Gemini API's natural language processing (NLP) capabilities. The evaluation involves:

- Semantic Analysis: The Gemini API provides the analysis of the student responses and the comparison to the pre-defined rubrics with parameters: relevance, Grammar, Coherence, Completeness, and depth [5].
- Scoring Logic: The Gemini evaluates responses simply by matching the responses given with one word or a set of words given to the ideal responses.
- AI-Powered Feedback: The Gemini generated automated feedback to help students understand areas for improvement.

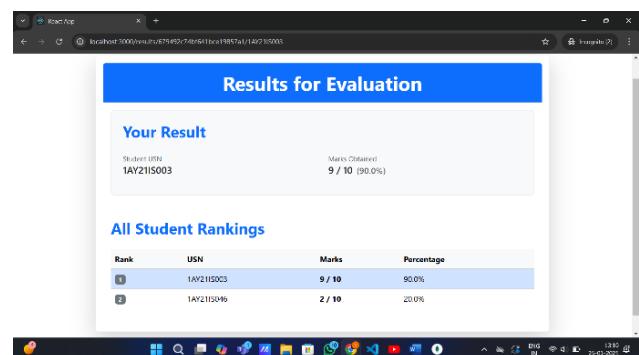


Fig. 3. Detail Report

The Fig. 3. displays above show the results of an online evaluation. It shows the user's individual score, 9/10 or 90%, along with their rank among other students. The table below lists the rankings of all participants, including their USN (University Seat Number), marks, and percentage.

VI. RESULTS AND DISCUSSION

TABLE 1. Performance Metrics

Metric	Value
Question Accuracy	98%
Grading Consistency	97%
Feedback Timeliness	Instant

A. Evaluation Efficiency

The platform's use of the Gemini API for automated evaluation significantly improved the assessment process. Multiple choice questions, short answer and descriptive question assessments were done with high reliability and comparability. This led to the minimizing of the waiting time that students used to spend waiting for results from the instructors as the system offered instant results to the students.

B. User Feedback

Students elaborated on self-test data that stated more instant data response and made the participants know their mistake and area of deficiency after submitting their tests. According to experiences of the teachers some of the features like auto scoring and fewer concerns of grading on paper was seen as good.

C. Adaptability Success

The platform has been designed in such a way that it fit perfectly with the existing educational processes and Learning Management Systems (LMS). Thus, its versatility in covering various subject fields and kinds of tests proved useful for various educational establishments, including small schools and networks.

D. Security Analysis

Strategies such as behavioral monitoring in real time as well as automated test monitoring worked to counter cheating. The secure and transparent evaluation system built trust among both students and educators [4].

E. Scalability Results

The system's architecture demonstrated robust scalability during stress tests, efficiently handling up to thousands of concurrent users with stable performance. It also included support for multilingual content and broadened its use in the educational institutions.

VII. CONCLUSION

This research adopts the Gemini API and developing an automated assessment system for multiple choice and descriptive kind of questions. It also simplifies the assessment process by eliminating the possibility of following the wrong answer paths as well as provides a means of automatically generating the results in every assessment occasion.

The conclusion of the proposed solution captures the major problems facing modern educational assessment and they include a time-consuming process of manual assessment and subjective grading. The flexibility of the system can easily fit into current education setups and accommodate multiple curricula as well as various and methods of evaluating students.

The results shows that evaluation efficiency has improved along with the reliability of the assessment with benefits that make the platform more applicable to large-scale educational application. Further development of the program will involve the improvement of the evaluation capacity, multi-lingual features, and, consequently, the presence of diversified subject areas required by learners.

REFERENCES

- [1] Hrvacević, Luka, Jelica Cincović, Adrian Milaković, Vladimir Jocović, Valerijan Matvejev, and Dražen Drašković. 2024. "Development of web system with an automated question generator based on large language models." *2024 11th International Conference on Electrical, Electronic and Computing Engineering (IcETRAN)*. Nis, Serbia: IEEE. 6. doi:10.1109/IcETRAN62308.2024.10645106.
- [2] IONESCU, Valeriu Manuel, and Madalin Ciprian ENESCU. 2023. "Using ChatGPT for Generating and Evaluating Online Tests." *2023 15th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*. Bucharest, Romania: IEEE. 6. doi:10.1109/ECAI58194.2023.10193995.
- [3] Patel, Hetul Niteshbhai, Azara Surti, Parth Goel, and Bankim Patel. 2024. "A Comparative Analysis of Large Language Models with Retrieval-Augmented Generation based Question Answering System." *2024 8th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. Kirtipur, Nepal: IEEE. 7. doi:10.1109/I-SMAC61858.2024.10714814.
- [4] Prentzas, Jim, and Maretta Sidiropoulou. 2023. "Assessing the Use of Open AI Chat-GPT in a University Department of Education." *2023 14th International Conference on Information, Intelligence, Systems & Applications (IISA)*. Volos, Greece: IEEE. 4. doi:10.1109/IISA59645.2023.10345910.
- [5] Sabina Elkins, Ekaterina Kochmar, Jackie C.K. Cheung, Iulian Serban. 2024. "How Teachers Can Use Large Language Models and Bloom's Taxonomy to Create Educational Quizzes." *The Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24)*. IEEE. 8. doi:10.1609/aaai.v38i21.30353.

Prof. Pushpalatha K S

Transforming Student Evaluation with Adaptive Intelligence and Performance Analytics

-  Articles
-  Acharya LRC
-  Acharya Institute of Technology, Bengaluru

Document Details

Submission ID

trn:oid:::1:3148569946

Submission Date

Feb 7, 2025, 6:42 PM GMT+5:30

Download Date

Feb 7, 2025, 6:44 PM GMT+5:30

File Name

IEEE_PAPER_-_MOHAMMAD_AAMIR_AIT21BEIS076.docx

File Size

184.7 KB

6 Pages

3,700 Words

21,643 Characters

5% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- ▶ Bibliography
- ▶ Quoted Text

Match Groups

-  **16** Not Cited or Quoted 5%
Matches with neither in-text citation nor quotation marks
-  **0** Missing Quotations 0%
Matches that are still very similar to source material
-  **0** Missing Citation 0%
Matches that have quotation marks, but no in-text citation
-  **0** Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

- 3%  Internet sources
- 4%  Publications
- 1%  Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.