AI-Generated Academic Assesment Portal with Performance Tracking

by

MD. Saiful Islam Dipu 24141279
Aminul Islam 23341055
Rakibul Hasan 20301300
MD. Asif Shahriar 20301328

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
June 2025

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

MD. Saiful Islam Dipu
24141279

Aminul Islam
23341055

Rakibul Hasan
20301300

MD. Asif Shahriar
20301328

Approval

The project titled "AI-Generated Academic Assesment Portal with Performance Tracking" submitted by

1. MD. Saiful Islam Dipu (24141279)
2. Aminul Islam (23341055)
3. Rakibul Hasan (20301300)
4. MD. Asif Shahriar (20301328)

Of Spring, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on June 20, 2025

Examining Committee:

Supervisor:
(Member)

Jawaril Munshad Abedin

Lecturer
Department of Computer Science and Engineering
BRAC University

Co-Supervisor:
(Member)

Atanu Roy

Adjunct Lecturer
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam

Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi

Associate Professor & Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

Artificial intelligence revolutionized numerous digital education operations yet the assessment of academic performance continues to prove especially difficult to overcome. Unfortunately, static question banks and rigid answer matching systems, currently used in making the assessment, can't provide personalized learning experiences. These systems have difficulty acknowledging proper semantic responses and frequently misidentify them. This paper describes an academic assessment portal developed by AI technology which combines performance tracking features to solve existing evaluation problems. The system uses the multilingual mT5 model to automate the production of questions that match different domains and contextual requirements. The Bangla Transformer system dedicated to evaluation answers detects properly paraphrased responses that improve testing precision. Student performance directs the platform to automatically adjust questions until each student experiences a suitable learning challenge for their current level. The AI system evaluates student responses by analyzing context which enables it to improve both accuracy and fairness of the assessment process. Students obtain performance-related data about their areas of expertise through performance tracking while automated question generation frees educators to teach without additional paperwork. The platform delivers both robustness and user-friendly interface through the use of Flask, React.js. Initial test results indicate that self-assessment performance tracking platforms outperform other methods, showing an accuracy improvement of 40%-50% due to personalized tracking, adaptive learning, and data-driven feedback mechanisms.

Keywords: BOOTSTRAP, AI assessment, Quiz Generation ,EduAI, Autograding, mT5, BanglaNLP, SmartQuiz, DeepEval, PerfTrack, AdaptiveTest, Self-Assess, AIQuiz, NLP, AnswerMatch, EduTransformer, AutoTest, ScoreSync, CognitiveEval, AI learning, Flask.

Table of Contents

List of Figures

Chapter 1

Introduction

Academic assessment stands as a challenging aspect which proves difficult to enhance despite digital learning's fast evolution of education. The static nature of traditional assessment tools limits students from correctly gauging their comprehension because they use restricted bank of questions with strict grading criteria. Self-evaluation presents challenges for numerous learners because these assessment systems do not accommodate individual student learning requirements. These systems lack both capabilities for adapting itself to students' unique learning needs and the ability to give credit to correct answers presented in different wording patterns. The development of assessments uses substantial educational resources which prevents teachers from focusing on dynamic teaching methods and student involvement. An AI-enabled Academic Assessment Portal with Performance Track- ing serves as the proposed solution because it brings smarter adaptive features to student and educational evaluation methods. The system over one AI system where mT5 generates context-matched questions and another one for optimized grading which understands meaning of student message and not just word-based approaches. This testing system is unlike traditional tools in that it adjusts question difficulty levels based on student responses, creating an individually customized learning course. The platform permits both online assessment and also real-time student performance tracking which comes up with students with specific knowledge. Educational staff achieve better administrative outcomes through automated grading which frees them to concentrate on their teaching tasks instead of manual mark review. The system uses contemporary web technologies for construction to deliver stability alongside scalability features. The system's backend runs on Flask which executes data processes and maintains API connectivity together with the frontend which uses React.js with Bootstrap for offering an intuitive user experience. The intended study will transform academic exams through its creation of adaptive tests and its provision of better fairness and perceptual value to students. We have voice testing on our future development list, multilingual capabilities, and advanced Artificial Intelligence functions that will optimize the learning process. The addition of intelligent automation to education establishes a modern standard for delivering data-based learning assessment processes.

1.1 Problem Statement

Effective academic assessment is essential for both students and educators because traditional evaluation methods are outdated and ineffective. Most educational institutions still rely on stiff, pretrained question banks which follow manual grading, which fail to apply diverse learning styles and student progression. These obsolete systems often misjudge scoresheets which penalize students for correct answers phrased differently. As a result, grading lacks precision, and students miss out on valuable insights into their actual understanding.

First of all, self-assessment plays a critical role in modern education, allowing students to actively engage in their learning process. Research shows that students who regularly assess their own progress develop better metacognitive skills, leading to improved retention and problem-solving abilities. According to a survey by University of Tirana, formative assessments, including self-evaluation can increase student achievement by up to 25% . Moreover, in traditional quiz systems it is not possible to offer feedback only after a response, so students are unable to keep a record of their progress, understand points of weakness, and plan their study accordingly.

Secondly, There are a few chances of self-assessment for the inefficient learning tendency. For example, since the students have no instrument to measure if they are going to understand the subject they keep memorizing the subjects from beginning to end. Self-study, in contrast, is characterized as an iterative and systematic activity through which the teacher learns, collecting data about instruction, learning and the school setting in order to find, validate, and evaluate new knowledge.

Moreover, the absence of automated self-assessment also increases the burden on educators, who must manually design, grade, and analyze quizzes. This time-consuming process takes away valuable teaching hours that could be used for innovative learning. Studies indicate that teachers spend nearly 30% of their weekly work time on grading and assessments and tasks [1]. This managerial workload hampers their ability to provide timely, personalized feedback to students.

To point out these challenges, there is a great demand for AI-generated self-assessment tools that provide quick feedback, performance tracking, and adaptive question generation. AI-driven assessments can analyze student responses beyond keyword matching, ensuring fair and accurate grading while adapting quiz difficulty based on individual progress reports. A report by HolonIQ predicts that AI-powered education technologies will grow by 32% annually, emphasizing the increasing demand for automated, intelligent assessment solutions [2].

1.2 Solution Approach

To find out the described challenges in the problem statement, we propose an AI-enhanced self-assessment platform that transforms self evaluation through real-time feedback, innovative learning and noteworthy grading processes. Traditional assessment methods fail to apply diverse student learning styles and generate a heavy workload on educators. Our AI-powered solution offers automated quiz generation, semantic answer evaluation and performance tracking, enabling a personalized

learning experience for students while reducing grading burdens for educators.

One of the key features of our platform is real-time answer evaluation, which allows students to receive immediate feedback on their responses. Unlike traditional quiz platforms that rely on predefined keywords, our AI model, based on the Bangla Sentence Transformer and mT5, analyzes responses semantically. This ensures that students are not unfairly penalized for using different phrasing while still conveying the correct meaning. According to a 2022 report by Educause, students who receive immediate feedback improve their concentration rates by up to 26% compared to those receiving delayed feedback. Our AI-driven grading system eliminates waiting times and empowers students to adjust their learning strategies instantly, enhancing self-directed learning [3].

Educators spend 25%-30% of their time on grading and assessment-related tasks which reduces opportunities for innovative and student centered teaching. Our AI-driven grading system automates evaluation, freeing up teachers to focus on mentoring and personalized instruction based teaching. Additionally, data analytics and performance reports generated by our platform allow educators to identify student weaknesses and track progress over time. For example, instead of manually analyzing quiz results, teachers receive AI-generated insights that highlight areas where students need improvement.

Human grading errors are a common problem with traditional assessments, as different teachers can mark subjective responses differently. Our AI-powered grading system uses natural language processing (NLP) models to evaluate responses based on meaning rather than word choice, ensuring fair and consistent evaluation. This is especially helpful for open-ended questions, where bias may result in unequal scoring. Our AI-enhanced self-assessment platform encourages accuracy and fairness in academic assessment by removing manual errors and inconsistencies.

1.3 Objective

The primary goal of this paper is to present an AI-enhanced self-assessment platform that improves the accuracy, efficiency, and adaptability of academic evaluations. The following objectives outline the core aims of our research and system development:

- Develop an AI-enhanced self-assessment platform with real-time feedback.
- Ensure accurate answer evaluation using Bangla Sentence Transformer and mT5.
- Implement adaptive question generation based on student performance.
- Improve self-evaluation efficiency through instant feedback and analytics.
- Advance intelligent automation in education for personalized learning.

Chapter 2

Related Work

The progress of artificial intelligence technology in education assessment led to the creation of several automated quiz generation and evaluation systems. Many approaches were introduced throughout time to handle quiz development and evaluation while enabling adaptive learning. Recently, the educational assessment procedure shifted from rule based question generation towards adoption of NLP and deep learning models to make context based evaluation and student specific learning pattern.
Current models face challenges when it comes to automatic difficulty level adjustment of test questions, assessment of responses written in different words and their inability to handle multiple languages at once. The analysis of automated assessment tools from past research takes shape around two categories which consist of established models together with their applicable solution.

2.1 Existing Models

AI-driven assessment tools develop their frameworks by implementing different computational strategies starting from rule-based systems through transformer-based designs. Professor Michael Heilman created the initial systems through his question generation framework that built foundational concepts by making use of dependency parsing and linguistic structures to develop factual question-answer pairs. [4]
The system creates question-answer pairs with the help of dependency parsing and linguistic structures. The first versions of these models depended too extensively on predefined syntactic guidelines which restricted their potential to generate context-specific questions and numerous question types. [5]
The systems were introduced using Machine Learning are Automated Quiz Generator (AQG) and Enhanced Automatic Question Creator (EAQC). Statistical and semantic analysis methods were added to traditional methods through the development of these tools. The quiz distractor generation of AQG depended on semantic similarity with help from DBpedia but EAQC strengthened flexibility by extracting educational concepts from materials. These improvements failed to address the inadequate question diversity and domain customization needs of the models as better solutions were needed. [6]
The introduction of large language models (LLMs) generated a shift in automated assessment methods via GPT-3, GPT-4 and T5-based architectural models which delivered considerable advancements in question creation accuracy. Research showed

GPT-4 reached 75.3% success rate when generating multiple-choice questions whereas GPT-3 succeeded only 30.8% of the time. The models showed limited success in developing relevant distractors because human intervention was needed to overcome their challenges in distractor generation. [7]
Researchers developed hybrid systems to deal with the original shortcomings. Short-answer questions generated better through the Hybrid Automatic Question Generation model that used BERT semantic analysis together with Stanford CoreNLP syntactic capabilities. Both Automated Exam Question Set Generator (AEQSG) along with Bloom's Taxonomy enabled experts to sort questions based on cognitive levels which tailored tests for different skill levels. The developments proved the ability of AI technology to function effectively. [8]
Computers through AI successfully generated questions but faced issues regarding maintaining high-quality content standards and precise evaluation of responses which had received paraphrasing.

2.2 Existing Software Solutions

Researchers developed new innovative solutions through deep learning methods and NLP technologies and transformer-based architectures while correcting traditional model limitations. Proficient grading obtained its biggest advance by adopting context-aware algorithms which went further than basic keyword processing to provide smart assessment. Bangla Sentence Transformer used cosine similarity scoring as a flexible evaluation approach which assessed the concept-based equivalence between student responses and correct answers. [9]
The adaptive learning paradigm introduced an AI model system which automatically modifies question complexity according to instant student achievement results. EduQuiz brought innovative features through SWQG and EEQG technologies that produced adaptive quizzes adaptable to individual student learning. EEQG proved its capability to create adaptive quizzes for student cognitive levels by achieving 90.5% accuracy during its implementation. [10]
Multilingual Natural Language Processing has proved essential for enlarging the accessibility of automated assessment algorithms through its integration. The pre-training process of mT5 (Multilingual T5) enables fine-tuning across 100 languages that includes Bangla to deliver quiz systems supporting learners from multiple linguistic backgrounds. The encoder-decoder framework of mT5 allows current systems to develop questions with precise grammatical structure and contextual meaning across different subjects and languages to serve NLP applications which have limited resources. [11]
A vital advancement takes place with answer validation through AI-powered grading systems. The process of answer evaluation transforms with vector embeddings since they outperform traditional keyword-based assessments in detecting sentence similarities. QG-Net proved the capability of RNN networks by demonstrating their ability to create domain-specific quiz questions through an RNN-powered system. The system suffered from being inflexible and newer hybrid models using BERT and GPT-4 attempted to fix this problem. [12]
Continued research needs to solve the three main problems related to domain-specific AI model calibration and educational AI ethics and balanced question difficulty progression. Future research about multimodal AI seeks to combine text, speech, and

image recognition for additional improvements in automated assessment technologies.

Chapter 3

Methodology

3.1 Data Collection

The completion of an AI-driven assessment platform entirely depends on the quality and diversity of its dataset. To create this project, we create a dataset of 4 different topics and question-answer pairs specifically written in the Bangla language. Each topic has more than 100 question-answer pairs. These questions also have difficulty levels as easy, medium, and hard.

Each pair in the selected dataset contains a question coupled with its suitable answer. It is also covered in four subjects, including ICT, Bangla literature, General knowledge, and economics. We took certain measures to train machine learning models like the dataset's size and being in a well-decorated format which will ensure a strong foundation. In every case, this dataset came out to be a worthy one. It carries academic assessments which is one of the prime advantages of this dataset. It also ensures that an AI-built system can generate topic-based questions and respond with answers that are suited to those questions. By implementing this dataset, the system acquires the ability to understand the variation of Bangla sentence structure, improving its performance in evaluating user-submitted answers.

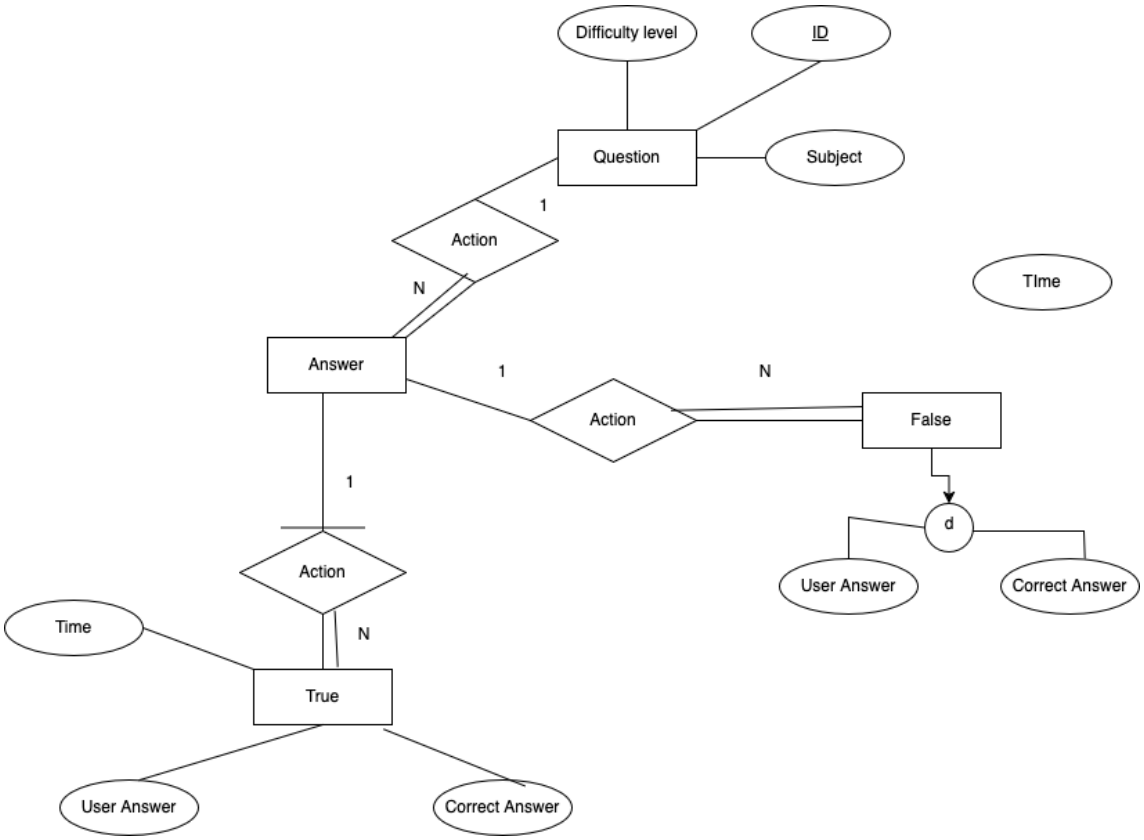

Figure 3.1: ER Diagram

3.2 Data Preprocessing

When the dataset was prepared for deployment, it went through several pre-processing procedures to ensure accuracy, consistency, and compatibility with the AI model. Since raw data often contains inaccuracies, inconsistencies, or irrelevant information that can degrade the model's performance, data pre-processing is an essential step in machine learning.

Data validation was the initial stage during which the accuracy and completeness of the dataset were checked. Making sure that every entry contained a legitimate query and a pertinent response was crucial. To ensure dataset integrity, any missing, indisputable, or tainted data points were either fixed or eliminated. This step stopped errors from arising during the training phase, greatly enhancing the quality and dependability of the dataset.

Unnecessary white spaces and duplicate entries were removed by data cleaning and frequent validation process. The model's ability of generalization gets negative impact when duplicate records occur bias behavior in training. Additionally, invalid or nonsensical question-answer pairs were identified and excluded to enhance dataset relevance.

After cleaning, the data was encoded for compatibility with machine learning models. Since AI models cannot process raw text directly, all textual information was converted into numerical representations. This was accomplished through the use of

tokenization, which divides sentences into smaller units (tokens) that AI can understand. To convert the text data into a vectorized format for the mT5 and Bangla Sentence Transformer models, embeddings were required. Through the capture of each sentence's semantic meaning and contextual relevance, these embeddings allowed the AI model to assess responses based on meaning rather than word choice. The dataset was refined using this multi-step pre-processing technique to maximize AI performance, eliminate anomalies, and guarantee smooth integration with the evaluation system.

Figure 3.2: Word Frequency Visualization from Dataset

3.3 Model Description

The assessment system has been significantly improved by incorporating two leading artificial intelligence models: Google's mT5 (Multilingual T5) and the Bangla Sentence Transformer. These models work together to improve the accuracy of processing and evaluating user input, especially when it comes to matching answers to a given data set.

The Bangla Sentence Transformer plays a central role in assessing student responses by accurately assessing the similarity of sentences. Unlike traditional systems that rely on keyword matching, this model understands the meaning of words. It transforms sentences into high-dimensional vectors and uses collinear similarity to determine how closely a student's response matches the expected response, even if different phrases or vocabulary are used.

Meanwhile, Google's mT5, a versatile, multilingual text-to-text translation transformer, helps analyze and interpret complex, open-ended responses. This model

excels at identifying key concepts and handling various language idioms and paraphrases, allowing for a deeper understanding of user input in relation to a given dataset.

The decision to combine the Bengali Sentence Transformer with mT5 was based on their complementary capabilities. While the Bengali Sentence Transformer specializes in accurate sentence-level intelligibility, which is crucial for similarity-based grading, the broader scope of mT5 helps interpret a variety of linguistic inputs, ensuring more accurate comparisons with the dataset. Together, these models provide a comprehensive, flexible assessment system that significantly improves the reliability and accuracy of the assessment process.

3.3.1 mT5

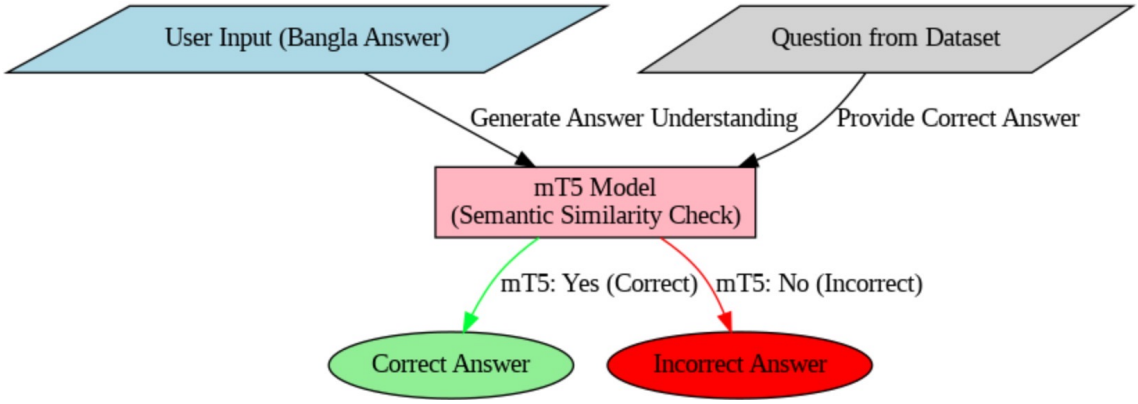

Figure 3.3: mT5 Diagarm

3.3.2 Bangla Sentence transformer

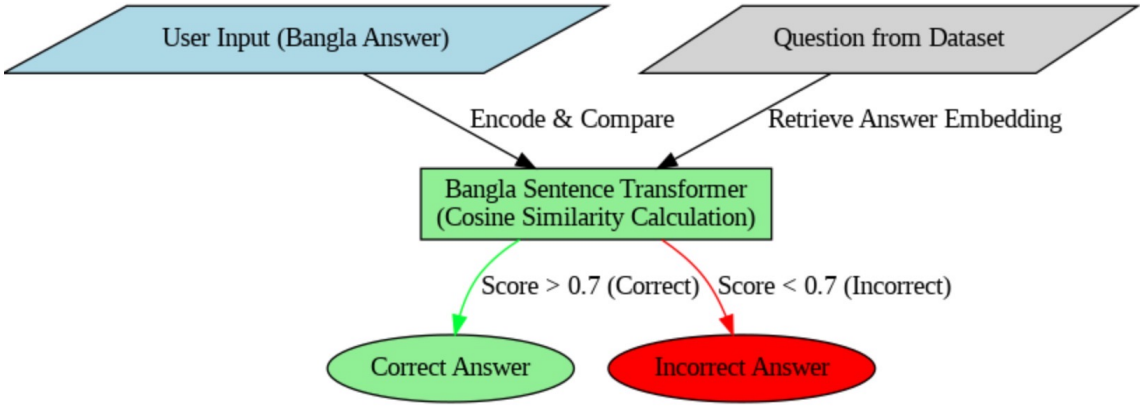

Figure 3.4: Bangla Sentence Transformer Diagarm

3.3.3 Both models integration

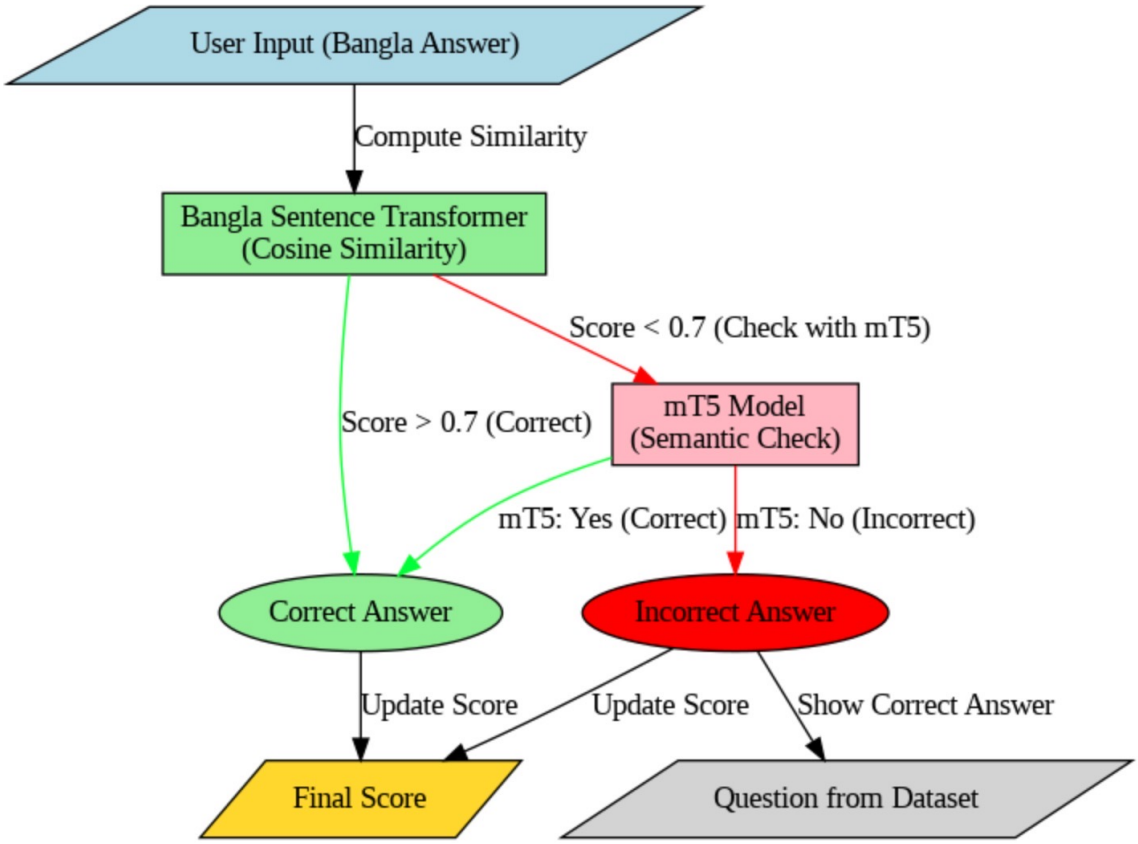

Figure 3.5: Integrated Model

3.4 Requirement Analysis

The requirements for the project titled "AI-Enhanced Quiz Platform For Self-Evaluation And Efficient Question Creation" are categorized into Functional Requirements and Non-Functional Requirements. These requirements ensure the system's functionality and performance meet the desired objectives.

3.4.1 Functional Requirements

These requirements define the core functionalities of the system and the features that must be implemented to achieve the project goals.

- The system must generate Bangla questions using the mT5 model based on a provided topic or dataset.
- Users must be able to submit answers to the generated questions for self-assessment.
- The system must validate user answers using a combination of Bangla Sentence Transformer for similarity checking.
- The system must validate user answers using mT5 model for deeper semantic validation.

- The system should evaluate user responses in real time, providing Correctness of the answer.
- Users should be presented with feedback for their answers, Correct/incorrect labels.
- A user-friendly dashboard must display Performance summaries, such as scores, time taken, and accuracy.
- A teacher can add, edit, and delete questions.
- Students can choose the difficulty level.
- Students can choose a topic.
- New users can register.
- Users can log in and log out.

3.4.2 Non-Functional Requirements

These requirements focus on the performance, usability, and other qualitative aspects of the system.

- The system must generate questions and evaluate answers within 2-3 seconds to ensure a smooth user experience.
- The backend should handle concurrent requests from multiple users without significant delays.
- The system must support an increasing number of users without performance degradation.
- The architecture should allow for the integration of additional models or datasets in the future.
- The system must maintain at least 99.5% uptime, ensuring availability for users at all times.
- Data consistency and accuracy of the generated questions and evaluations must be maintained.
- User authentication should be implemented with hashed passwords and secure login sessions.
- The user interface must be intuitive, with minimal learning curves for users.
- Language support must be optimized for Bangla, ensuring clear and accurate text display.
- Feedback must be presented in an easy-to-understand format.
- The system must follow modular design principles, making it easier to debug, update, and extend.

- Documentation must be provided for developers to understand the codebase and architecture.
- The system must integrate the 80k Bangla QA dataset for question generation and validation.
- The platform must integrate the Bangla Sentence Transformer and mT5 model for seamless operations.
- APIs must be available to connect the machine learning models with the backend for real-time question answering and validation.

3.5 Software Architecture and Flow Diagram

The system's software architecture adopts a structured, modular, and multi-layered approach designed to promote scalability, performance, and ease of maintenance. It is primarily divided into three key segments: the frontend interface, the backend server, and the AI model integration layer. The user-facing frontend, developed with React.js and Bootstrap, offers a dynamic and responsive experience, allowing both students and educators to smoothly interact with the platform — whether taking quizzes, submitting responses, or viewing detailed performance feedback. On the server side, the backend is powered by Flask, which manages data processing, user authentication, and communication through APIs. This backend seamlessly interfaces with the AI models to evaluate user inputs in real time, ensuring that answer assessments are both swift and precise. At the core lies the AI model layer, built upon Hugging Face Transformers, which delivers the essential intelligence for answer evaluation. These models communicate efficiently with the backend through RESTful APIs, allowing smooth data transfer and result generation. A flow diagram illustrates the relationships among these components, highlighting how user data is captured, processed, evaluated, and presented in real time. This well-organized design guarantees high speed, accuracy, and flexibility, providing an advanced, intelligent assessment system experience. Themodels communicate with the back end through RESTful APIs, they allow for data to be exchanged and results to be computed effectively. The flowchart illustrates a process in which the components are interconnected, and indicating how the user inputs, from initiation, evaluation and display are produced in real-time. Its architecture is organized in a way to scale and keep the system fast, accurate and flexible, thus providing a state-of-art AI-powered assessment experience.

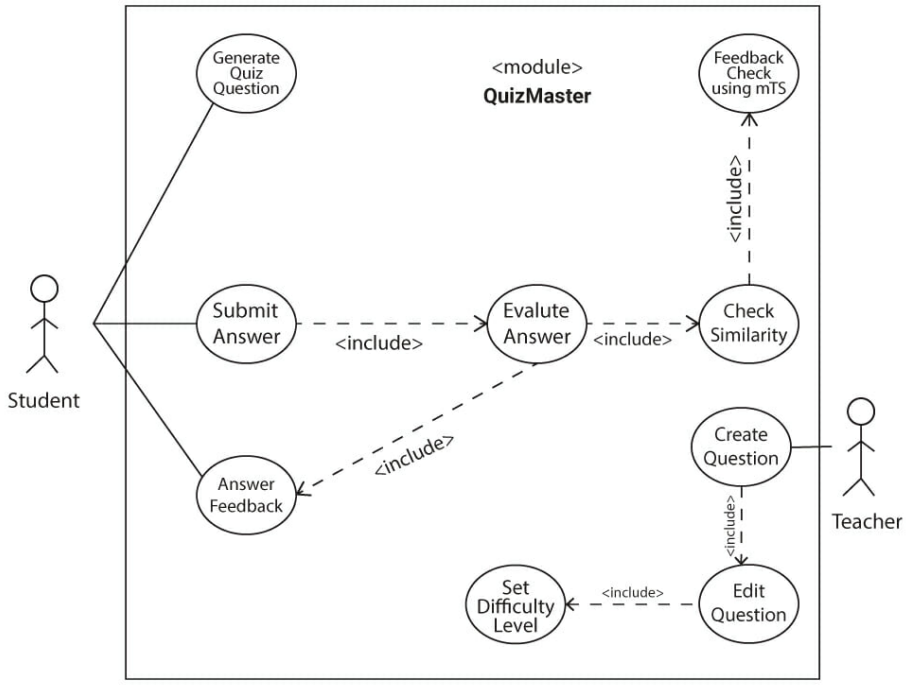

Figure 3.6: Use Case Diagram

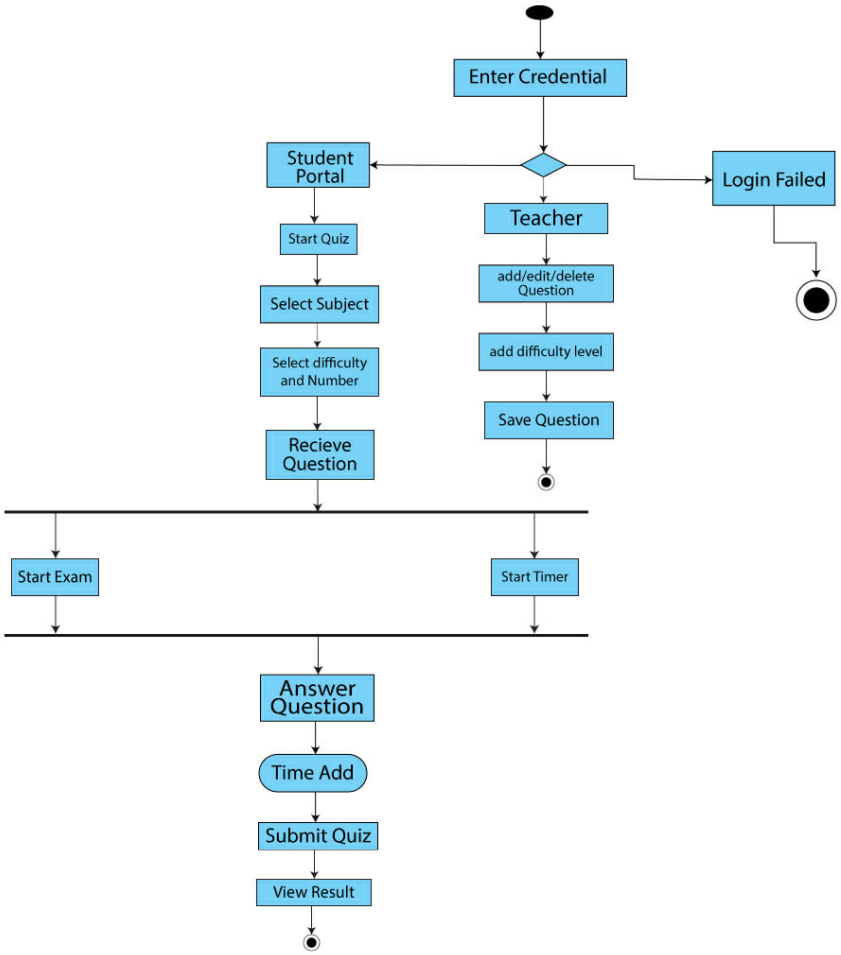

Figure 3.7: Activity Diagram

Chapter 4

Implementation

4.1 LLM Model Utilization

Instead of training new models from scratch, two pre-trained models—Bangla Sentence Transformer and mT5-small—were used from Hugging Face for their ability to handle Bangla linguistic structures, semantics, and variations in phrasing. Bangla Sentence Transformer for Similarity Calculation The Bangla Sentence Transformer (shihab17/bangla-sentence-transformer) was used to compute the cosine similarity between a student's response and the expected answer. The process involved: Converting both the user's answer and the correct answer into numerical vector representations. Computing the cosine similarity between these vectors in a multi-dimensional space. Comparing the similarity score to a predefined threshold (0.7) to determine if the answer is correct. This approach allowed the system to evaluate paraphrased answers accurately, even if they differed in wording but retained the same meaning.

4.1.1 mT5 for Semantic Verification

While the Bangla Sentence Transformer handled most similarity calculations, there were cases where the cosine similarity score was below the threshold, making it difficult to determine correctness. In such cases, the system used mT5-small (google/mt5-small) as a secondary verification step. The mT5 model, pre-trained on a multilingual dataset including Bangla, was used for natural language understanding and semantic validation of responses. To verify answers, the mT5 model was prompted as follows:

```
‘‘Do the following two Bangla sentences have the same meaning?
Sentence 1:  [User Answer]
Sentence 2:  [Correct Answer].
Respond with ‘Yes’ or ‘No’.’’
```

If the model responded with "Yes", the answer was considered correct, even if the cosine similarity score was low. This secondary check helped the system handle complex sentence structures, synonyms, and different expressions of the same idea, improving evaluation accuracy.

4.1.2 Integration and Evaluation Process

The intelligent assessment system was implemented using Python and required the following libraries:

- sentence-transformers for Bangla Sentence Transformer
- transformers for mT5-small
- torch for GPU acceleration
- pandas for dataset handling

A Bangla QA dataset (80k-bangla-qa-dataset.csv) was used for evaluation, ensuring that the dataset contained both questions and answers. The assessment system operated as follows:

- A question is displayed to the user, and they provide a textual answer.
- The Bangla Sentence Transformer first evaluates the answer using cosine similarity.
- If the similarity score is above 0.7, the answer is marked as correct.
- If the similarity score is below 0.7, the answer is passed to mT5-small for semantic validation.
- If mT5 determines that the two sentences have the same meaning, the answer is accepted.
- If neither model confirms the answer as correct, it is marked as incorrect, and the correct answer is displayed.

4.2 Software Development

4.2.1 Frontend Development

Frontend was designed with simplicity, responsiveness, and usability in mind. It is intuitive for students to navigate through its features. HTML, CSS, React.js, and Bootstrap are the technologies utilized in the frontend design. With React.js and CSS, modern, lightweight, and scalable web applications can be developed. A key feature of the frontend is the input form that enables users to enter their responses for analysis. The interface was crafted to prevent distractions. It focuses on usability and simplicity. Dynamically provided feedback was included to show results, which comprise measurements of correctness, similarity score. The frontend is optimized for users at computer(Desktop/Laptop). This guarantees that the students have the capacity to utilize the system without experiencing performance issues.

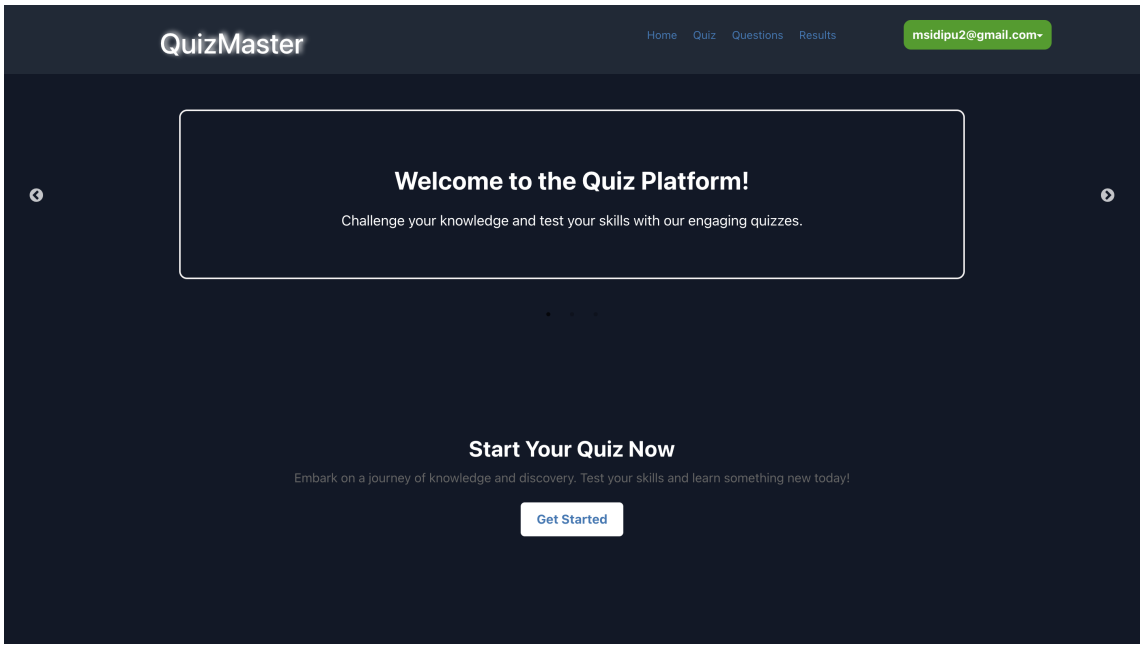

Figure 4.1: Homepage

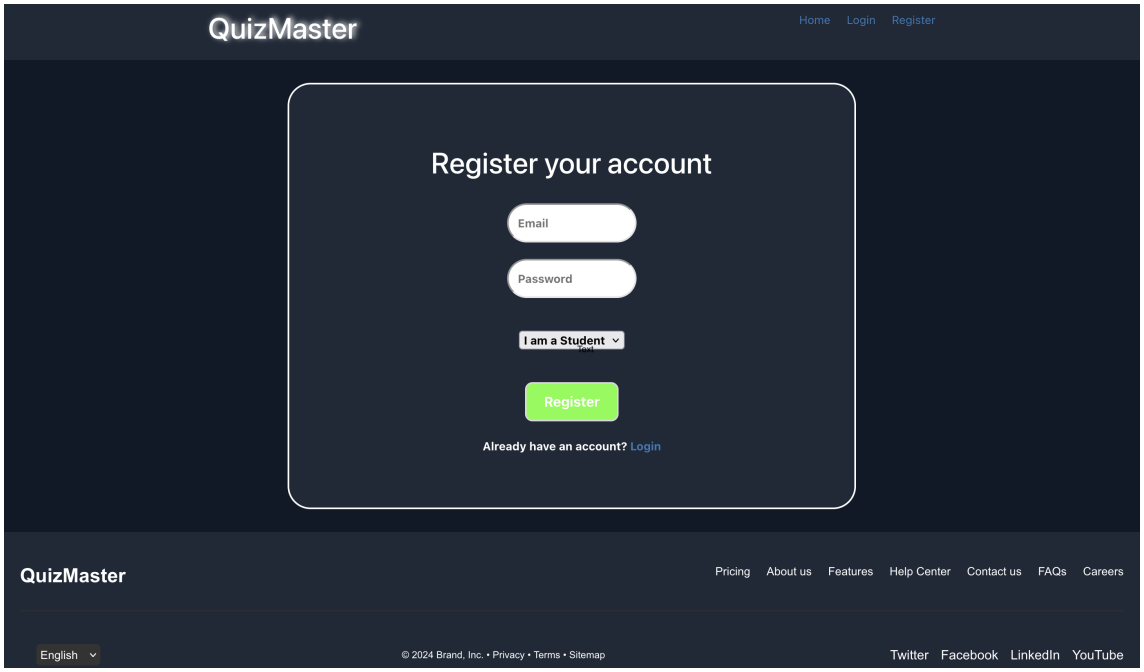

Figure 4.2: User registration

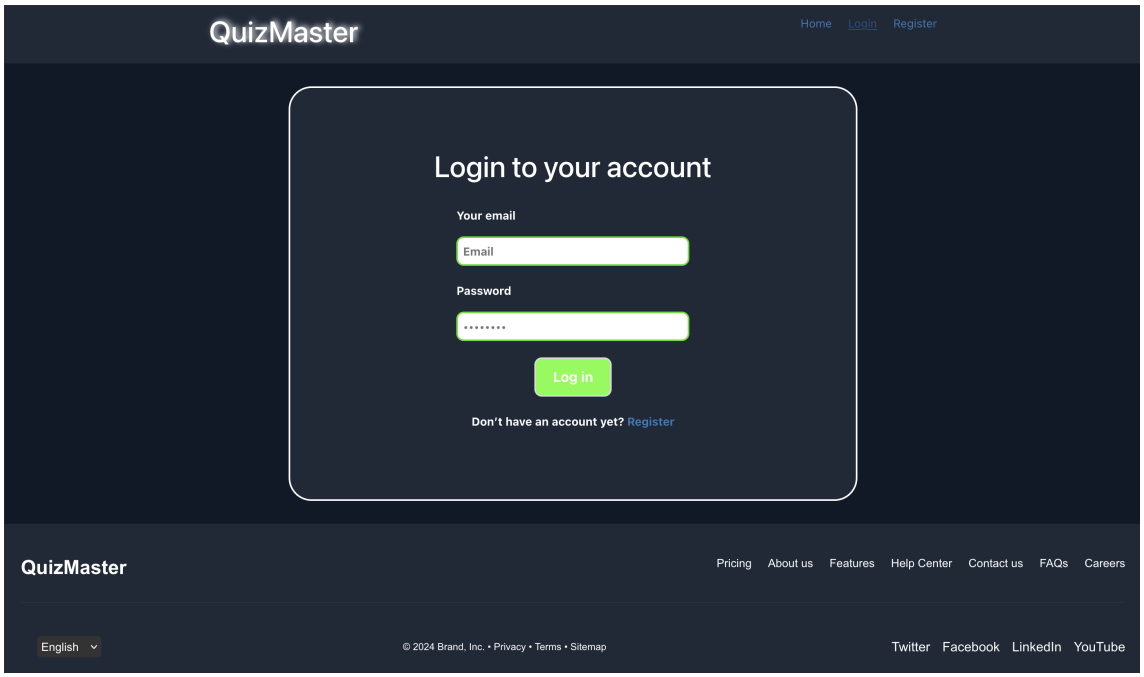

Figure 4.3: User login

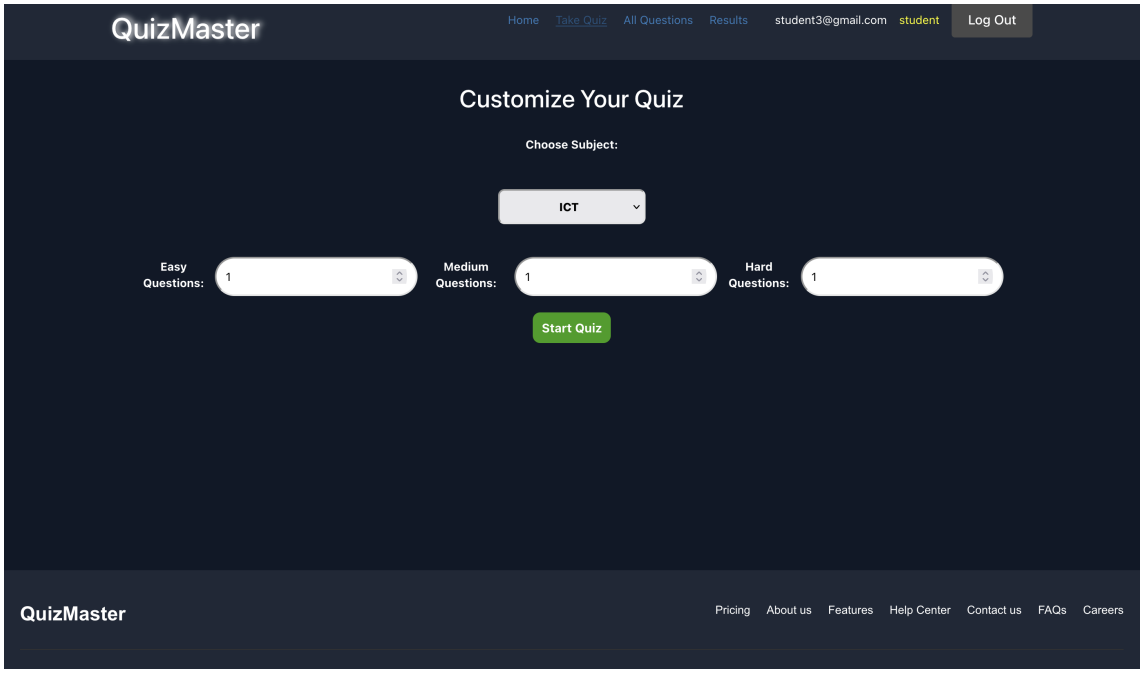

Figure 4.4: Quiz Customization

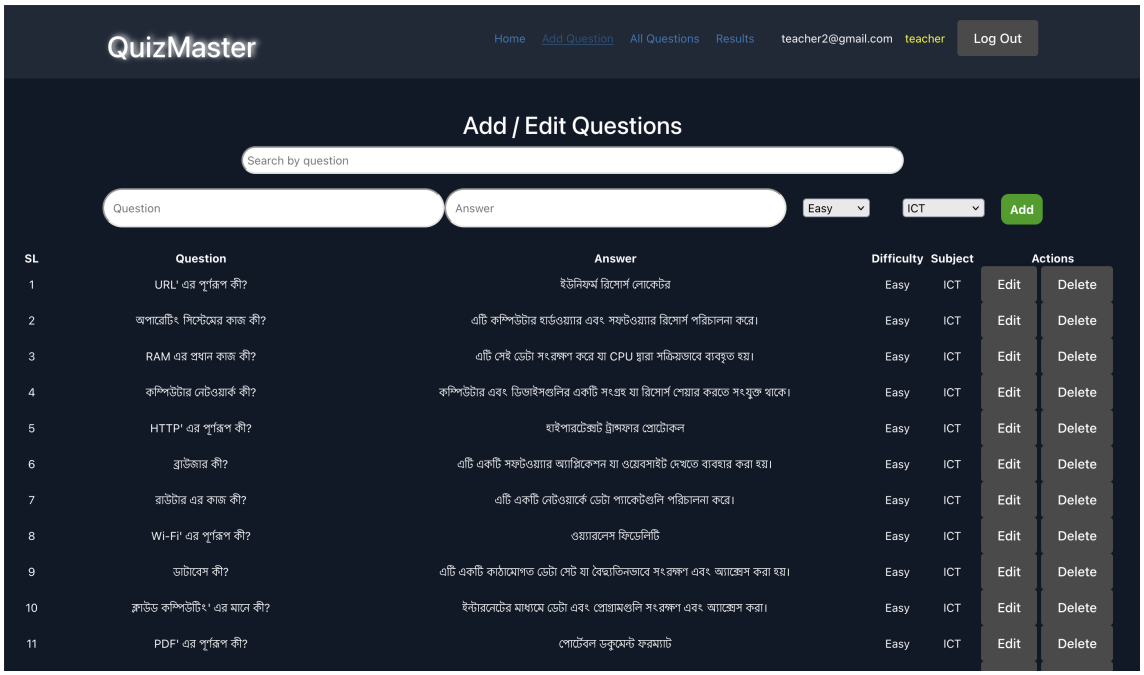

Figure 4.5: ADD, Edit and Delete questions

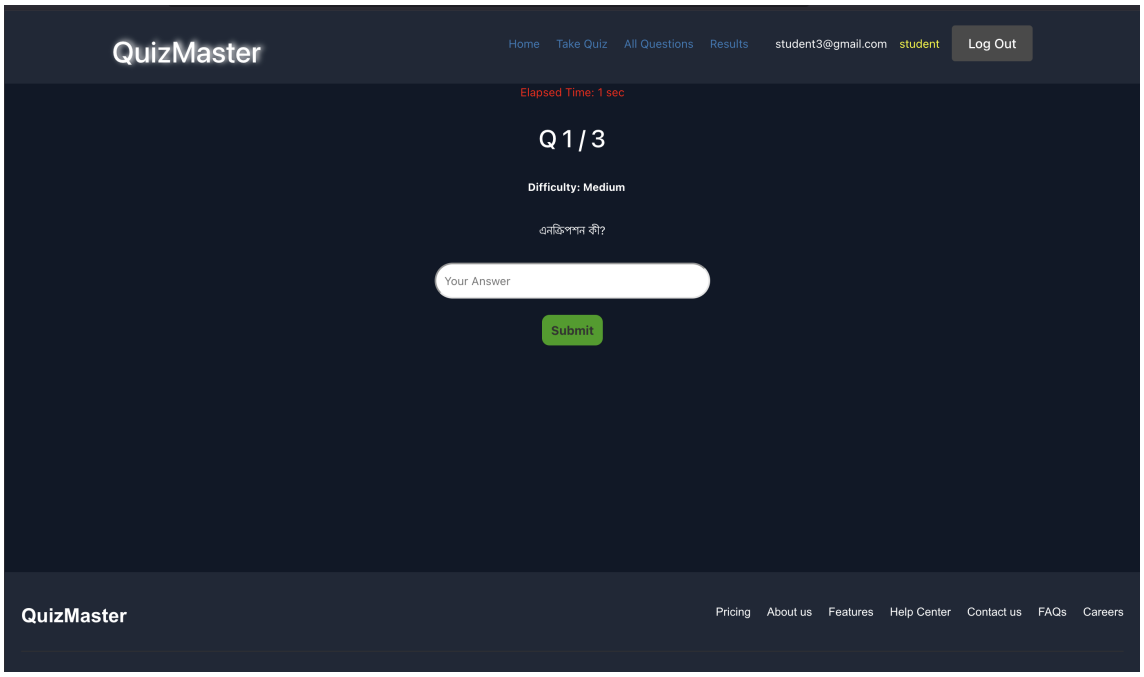

Figure 4.6: Questions

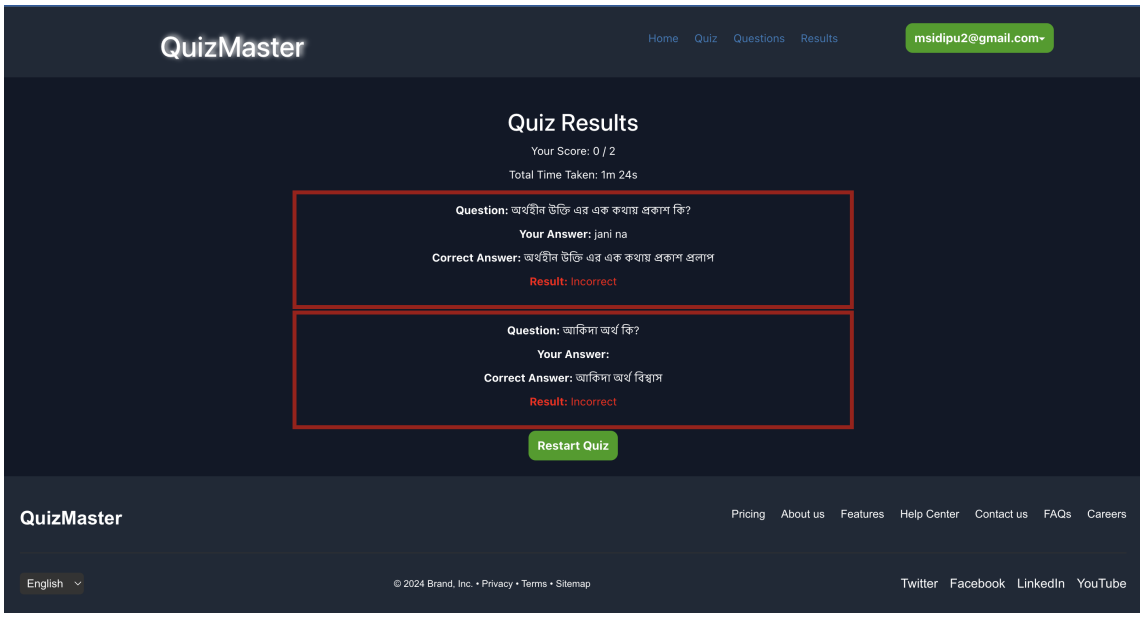

Figure 4.7: Answer validation

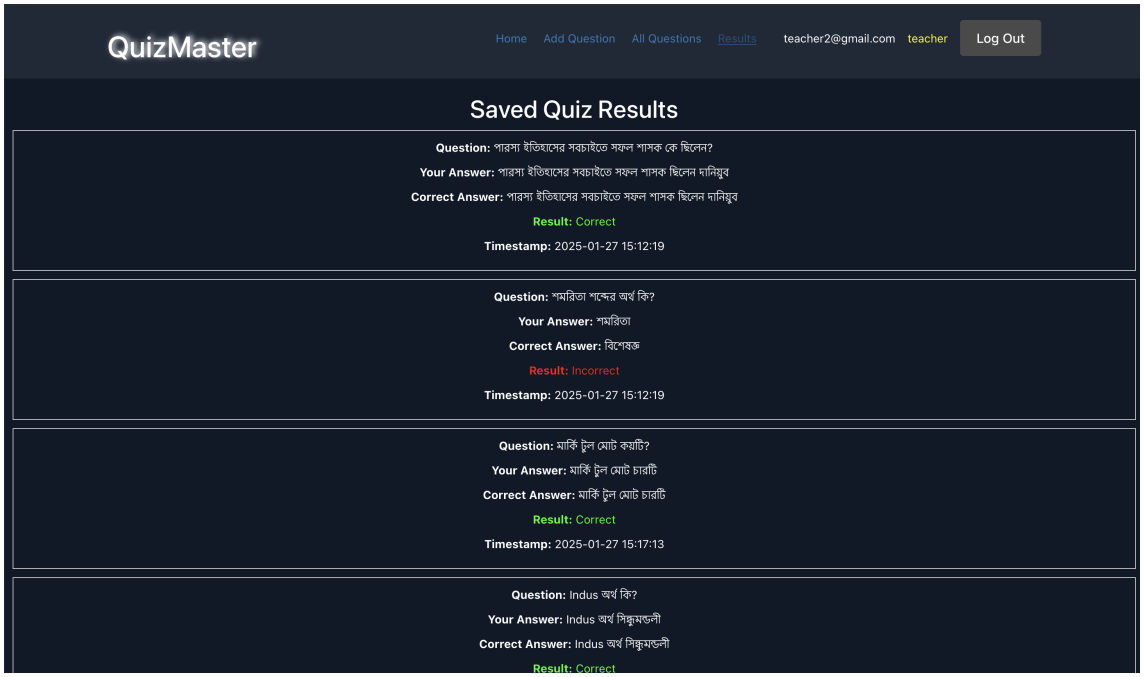

Figure 4.8: Quiz Results

4.2.2 Backend Development

The backend of the quiz platform is implemented using the Flask web framework, providing a lightweight and modular RESTful API to support core functionalities such as question generation, answer validation, and result storage. To ensure seamless integration with the React-based frontend, Cross-Origin Resource Sharing (CORS) is configured to accept requests from `localhost:3000`.

The system initializes by loading a question dataset from a public Google Sheets CSV file. This dataset serves as the primary source for generating quiz content. To

validate user answers semantically, the backend leverages the `SentenceTransformer` model (`bert-base-multilingual-cased`), which encodes text into dense vector representations. Cosine similarity is computed between the user's response and the reference answer, and a threshold of 0.7 is used to determine correctness.
The API exposes several endpoints:

- `/generate_questions`: Accepts a POST request and returns a specified number of randomly selected questions from the dataset.
- `/validate_answer`: Accepts a POST request with the user's answer and the correct answer, computes semantic similarity, and returns whether the answer is considered correct.
- `/save_results`: Stores quiz results, including user answers, correctness, and similarity scores, in a local SQLite database.
- `/get_results`: Retrieves all stored quiz results for review or analysis.

Additionally, the backend is equipped with proper error handling for invalid inputs, dataset loading failures, and model inference issues. The SQLite database schema is initialized at runtime to persist user interactions, enabling performance tracking and future evaluation.

4.3 Software Integration with AI Models

In this study, we employ the `mT5-small` model, a multilingual variant of the Text-to-Text Transfer Transformer (T5) architecture, pre-trained by Google on a diverse set of languages, including Bangla (Bengali). The mT5 model is particularly well-suited for natural language processing tasks involving Bangla due to its comprehensive multilingual pretraining corpus.
The implementation utilizes the Hugging Face `transformers` library, which provides streamlined access to state-of-the-art pretrained models and tokenizers. Specifically, the `MT5Tokenizer` and `MT5ForConditionalGeneration` classes are instantiated using the `google/mt5-small` checkpoint. This approach leverages transfer learning to enable downstream generation tasks such as question generation or summarization in Bangla without requiring extensive task-specific data.
A critical dependency for the tokenizer is the `SentencePiece` library, which facilitates subword tokenization. `SentencePiece` enables effective handling of out-of-vocabulary words and supports languages with complex morphology, such as Bangla. This dependency is explicitly installed to ensure compatibility with the mT5 tokenizer.
Input text sequences in Bangla are first tokenized into token ID sequences via the `SentencePiece`-based tokenizer. These token IDs serve as input to the mT5 model, which generates output token sequences corresponding to the desired task. The generated token sequences are subsequently decoded back into natural language text by the tokenizer.
The model and tokenizer are loaded once at application startup within a Flask-based backend environment to optimize inference latency and computational resource usage. Upon receiving input requests via HTTP endpoints, the backend processes the Bangla text using the mT5 model and returns the generated outputs to the client.

Overall, the use of `mT5-small` in conjunction with Hugging Face Transformers demonstrates an effective paradigm for developing multilingual NLP applications, providing robust performance on Bangla text generation tasks while benefiting from the modular and extensible Transformer library ecosystem.

Chapter 5

Testing and Evaluation

5.1 Testing

The testing covers 22 manual test cases for a Flask-based quiz backend system. These tests validate core functionalities such as dataset loading, question generation, answer validation, result saving, CORS policy, and error handling. Used Postman for testing.

Dataset Handling Tests verify loading from valid and invalid Google Sheet links.

API Endpoints Key routes like `/generate_questions`, `/validate_answer`, and `/get_results` are tested under different input conditions (valid, missing, or invalid).

Answer Validation Scenarios include correct, incorrect, partial, and empty answers, along with similarity threshold edge cases.

Result Storage Saving single and multiple results, along with invalid formats, is tested.

System Behavior Tests include invalid endpoint requests, CORS validation for both allowed and disallowed origins, database initialization, and simulated model failures. tabularx caption

5.2 System Evaluation

The system was also evaluated in terms of speed, scalability, and usability, beyond accuracy.

- The feedback was zoned for return within 2 to 3 seconds, to keep students out of a feedback delay zone.
- The platform could scale with no performance degradation given that the platform could operate multiple concurrent users.
- Beta testers such as students felt the system was very easy to navigate and highly responsive.

Table 5.1: Manual Test Case Results

ID	Title	Test Description	Input	Expected Output	Actual Output	Status
TC01	Dataset Load - Valid Link	Load dataset from Google Sheets (CSV)	Public Google Sheet CSV link	Dataset loaded successfully	Dataset loaded successfully	Pass
TC02	Dataset Load - Invalid Link	Load dataset from a broken/unauthorized link	Invalid/Restricted CSV URL	Error message: Unauthorized	Error loading dataset:...	Pass
TC03	Homepage Response	Check backend status	GET /	"Backend is running!" message	Same	Pass
TC04	Generate Questions - Default	Generate 5 questions with default POST	POST with empty body	JSON with 5 questions	JSON with 5 questions	Pass
TC05	Generate Questions - Custom N	Generate 3 questions	POST with num_questions=3	JSON with 3 questions	JSON with 3 questions	Pass
TC06	Generate Questions - Zero	Invalid number of questions	POST with num_questions=0	HTTP 400 + error	HTTP 400 + error	Pass
TC07	Generate Questions - Negative	num_questions = -5	POST with negative value	HTTP 400 + error	HTTP 400 + error	Pass
TC08	Validate Answer - Correct	Compare exact match	user="Paris", correct="Paris"	is_correct: true	is_correct: true	Pass
TC09	Validate Answer - Wrong	Incorrect answer comparison	user="London", correct="Paris"	is_correct: false	is_correct: false	Pass
TC10	Validate Answer - Empty	Empty user answer	user="", correct="Paris"	HTTP 400 error	HTTP 400 error	Pass
TC11	Validate Answer - Partial	Partial match	user="Paris is in France"	is_correct: false	As expected	Pass
TC12	Save Results - Valid	Save a quiz result	JSON array with 1 result	"Results saved successfully"	Same	Pass
TC13	Save Results - Multiple	Save multiple quiz results	JSON array with multiple entries	Success message	Success message	Pass
TC14	Save Results - Invalid Format	Missing keys in JSON	Incomplete JSON	HTTP 500 error	Error message in logs	Pass
TC15	Get Results	Fetch saved results	GET /get_results	JSON with results	JSON with results	Pass
TC16	CORS Policy	Allowed origin check	Origin: localhost:3000	Header included	Header included	Pass
TC17	CORS - Disallowed Origin	Disallowed origin test	Origin: malicious.com	No CORS headers	No headers	Pass
TC18	DB Initialization	Check table creation	On app start	DB and results table exist	DB file exists	Pass
TC19	Invalid Endpoint	Wrong endpoint request	GET /not_found	HTTP 404 Not Found	404 error	Pass
TC20	Simulated Model Failure	Force error in model	Mocked error case	HTTP 500 error	Traceback error	Pass
TC21	Similarity Threshold Edge	Test close to threshold	user vs correct score = 0.69	is_correct: false	As expected	Pass
TC22	Similarity Threshold Exact	Score exactly at 0.70	Embedded score = 0.70	is_correct: true	As expected	Pass

Chapter 6

User Manual

The platform was designed with ease of use in mind, ensuring that both students and educators could navigate it effortlessly. Steps to Use the Platform: Quiz Generation: Educators can generate quizzes based on custom topics. Answer Submission: Students submit answers, which are immediately evaluated by AI models. Feedback and Progress Tracking: Users receive detailed feedback, including accuracy scores, similarity percentages, and topic-based performance analysis. The system's dashboard provides a comprehensive overview of a user's learning progress, allowing them to identify strengths and areas for improvement.

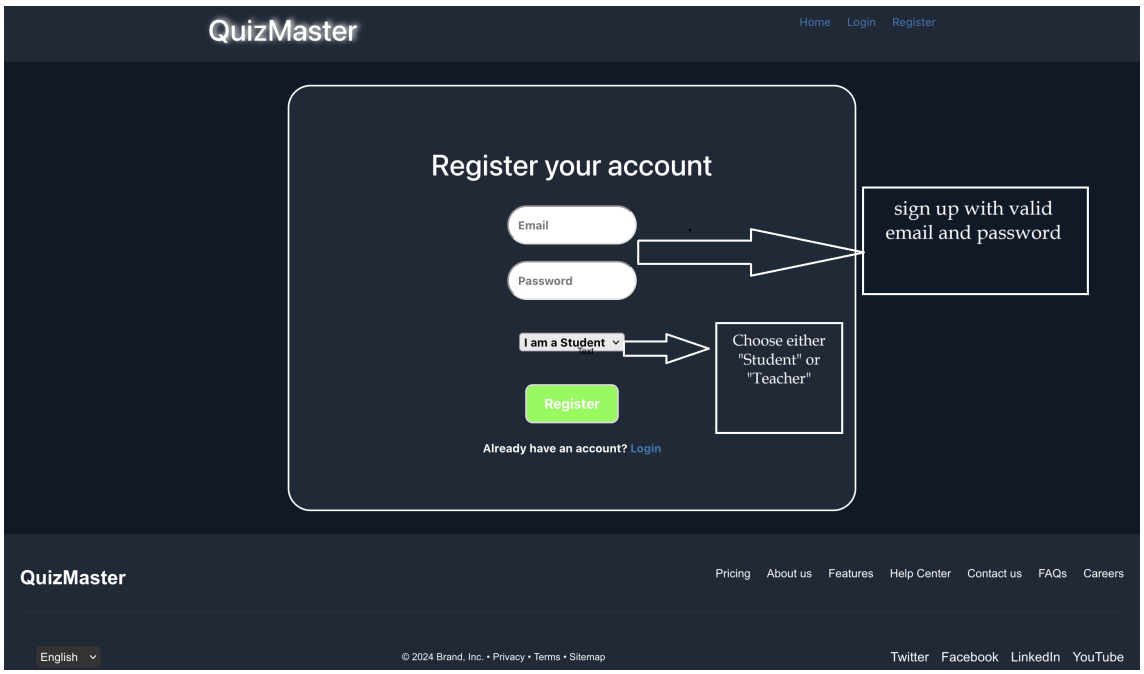

Figure 6.1: Register

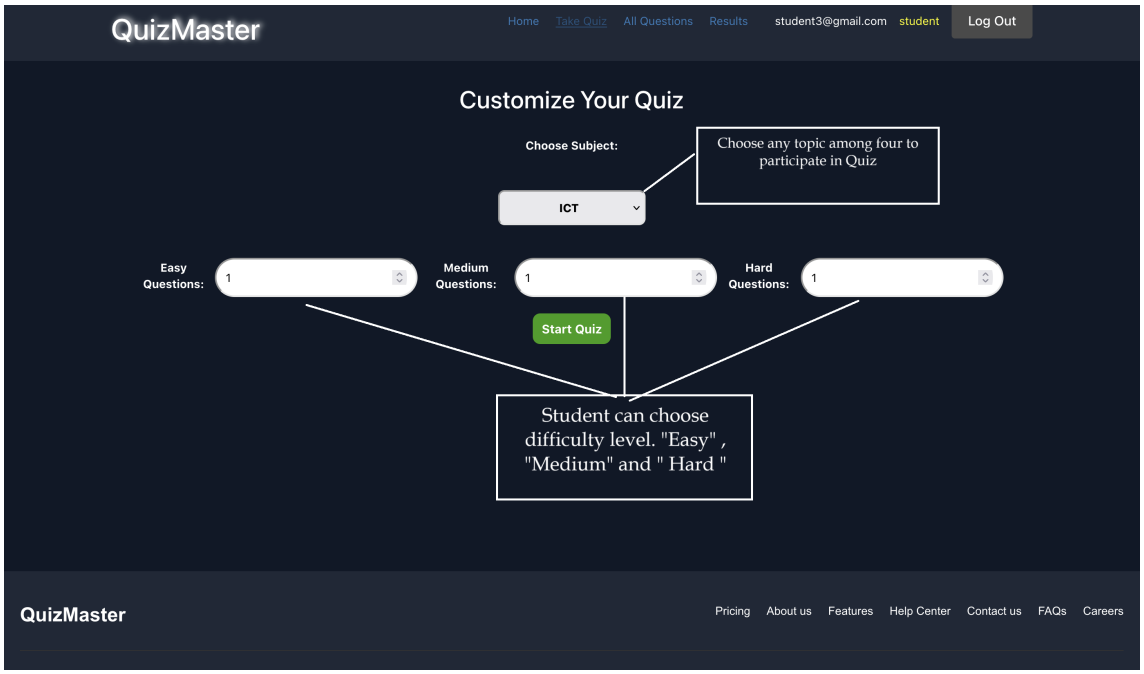

Figure 6.2: Select Questions (STUDENT)

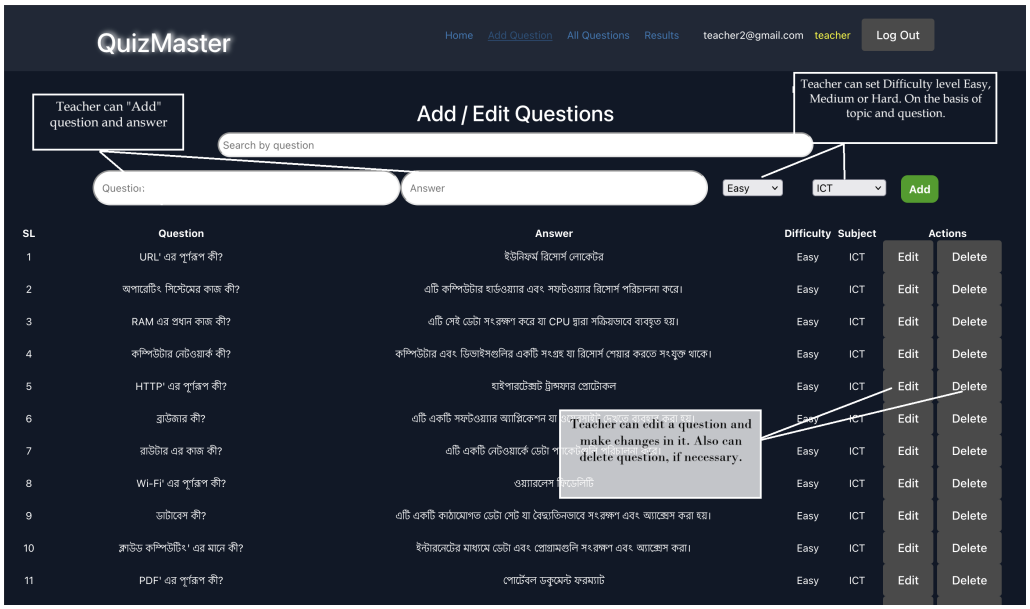

Figure 6.3: ADD, Edit and DELETE questions (Teacher)

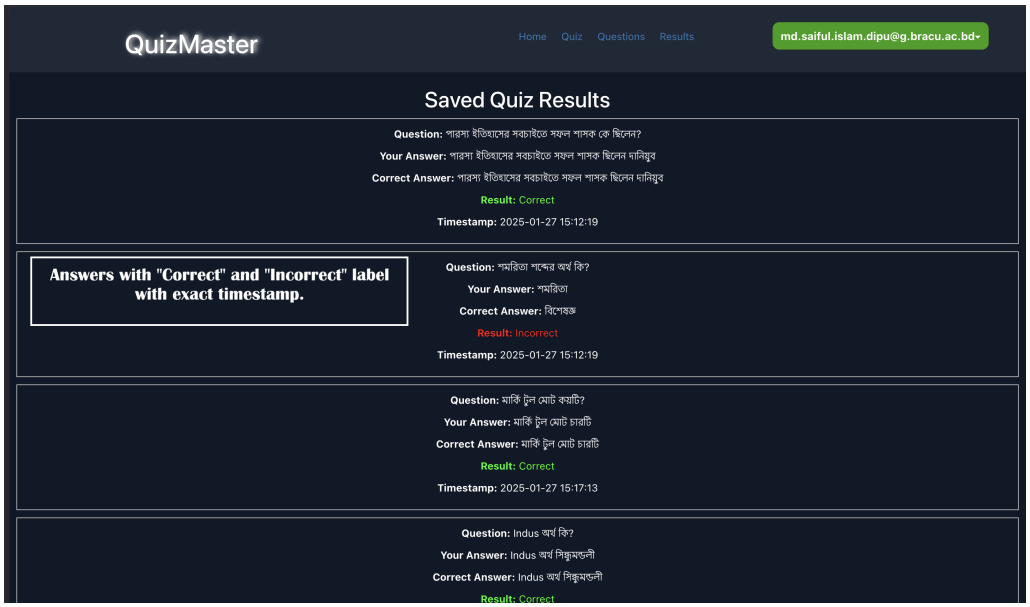

Figure 6.4: View Results

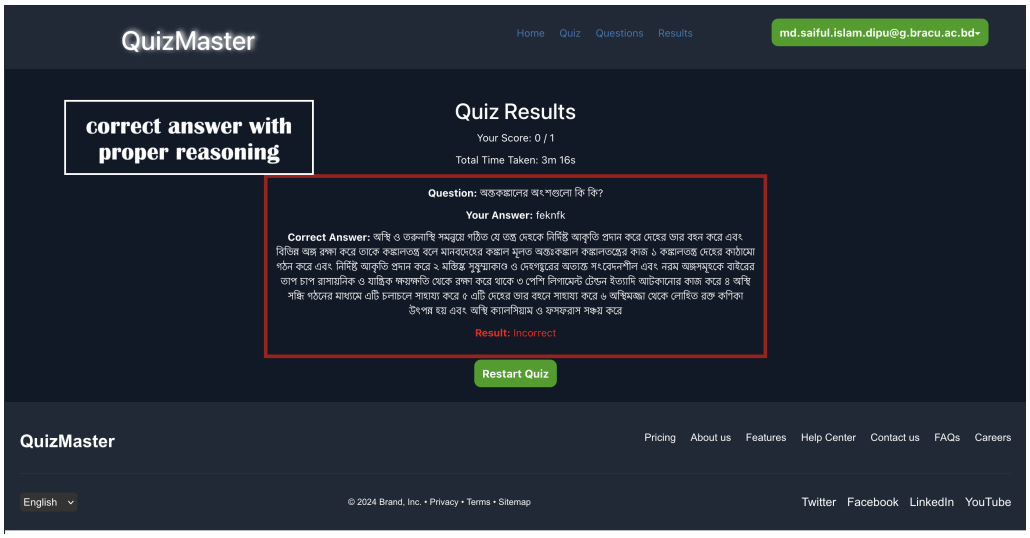

Figure 6.5: Descriptive Answer

Chapter 7

Limitations and Future Work

7.1 Limitations

The system has some major limitations in spite of being efficient.

- Relying on model Quality: The system fully depends on pre-trained reliable data.
- Faces difficulties on handling complex sentences: Model works smoothly on general questions but sometimes it struggles with high complexity and context-dependant responses.
- High traffic may cause Latency: The system performs flawlessly when it runs by single user but higher number of users will cause latency.

7.2 Future Work

We aim to make the system more better-performed and want to improve user experience for which following upgrades are planned:

- Dataset Expansion: Real world QnA pairs will make the dataset more diverse. And we will improve system's dataset by implementing that.
- By fine-tuning mT5 model we will handle more complex and context dependent sentences.
- By Implementing speech-to-text models to enable voice-based quiz submissions.
- Optimized system for mobile devices.

Chapter 8

Conclusion

Students should use the AI-created Academic Assessment Portal with Performance Tracking as a revolutionary tool that utilizes AI power and adaptive learning approaches to produce a new paradigm in assessment practices.

The mT5 system together with the Bangla Transformer produces a platform that creates a successful transition between automated assessments and individualized educational methods. The system presents diverse tests compared to standard quiz tools which only use pre-defined question pools because it continuously updates challenge levels for improved personalized student interactions.

This assessment method delivers improved accuracy through automated systems which decrease the educational staff's administrative duties. Real-time assessment tracking provides students with data-focused self-discovery which results in better learning strategies that educators can design through performance observations. A powerful academic framework results from the cohesive connection between Flask, React.js and SQLite which enables scalability for large academic implementations.

The research achieves its objectives yet recognizes two main drawbacks which stem from the sensitivity of the data sources and the systematic handling of complex linguistic answers. The identified constraints create vital opportunities for transformation of upcoming academic solutions. Developments in great demand will deploy voice capabilities along with multilingual abilities and progressive AI individualization that will increase academic testing effectiveness.

The fast-changing educational environment demonstrates that this AI system serves as proof of how intelligent automation can transform student learning. The platform promotes innovation together with adaptability to create assessments which bring both assessment optimization and develop an interactive academic environment that's both inclusive and effective. The study creates a new standard in automated academic assessment that will direct personalized learning improvements through highly precise and efficient methods of the future.

Bibliography

[1] Vartika Dutta and Sangeeta Sahney. Relation of principal instructional leadership, school climate, teacher job performance and student achievement. *Journal of Educational Administration*, 60(2):148–166, 2022.

[2] Natalia V Chicherina. Higher education futures: On the diversity of scenarios. In *International Conference on Professional Culture of the Specialist of the Future*, pages 161–170. Springer, 2024.

[3] Betsy Tippens Reinitz, Mark McCormack, Jamie Reeves, Jenay Robert, Nichole Arbino, Jeremy Anderson, John Hamman, Connie Johnson, Olivia Kew-Fickus, Rob Snyder, et al. 2022 educause horizon report® data and analytics edition. 2022.

[4] Michael Heilman. *Automatic factual question generation from text*. Carnegie Mellon University, 2011.

[5] Kaustubh D Dhole and Christopher D Manning. Syn-qg: Syntactic and shallow semantic rules for question generation. *arXiv preprint arXiv:2004.08694*, 2020.

[6] Christian Gutl, Klaus Lankmayr, Joachim Weinhofer, and Margit Hofler. Enhanced automatic question creator–eaqc: Concept, development and evaluation of an automatic test item creation tool to foster modern e-education. *Electronic Journal of e-Learning*, 9(1):23–38, 2011.

[7] Wanyong Feng, Jaewook Lee, Hunter McNichols, Alexander Scarlatos, Digory Smith, Simon Woodhead, Nancy Otero Ornelas, and Andrew Lan. Exploring automated distractor generation for math multiple-choice questions via large language models. *arXiv preprint arXiv:2404.02124*, 2024.

[8] Nicy Scaria, Suma Dharani Chenna, and Deepak Subramani. Automated educational question generation at different bloom's skill levels using large language models: Strategies and evaluation. In *International Conference on Artificial Intelligence in Education*, pages 165–179. Springer, 2024.

[9] Alibek Barlybayev and Bakhyt Matkarimov. Development of system for generating questions, answers, distractors using transformers. *International Journal of Electrical & Computer Engineering (2088-8708)*, 14(2), 2024.

[10] Ramon Dijkstra, Zülküf Genç, Subhradeep Kayal, Jaap Kamps, et al. Reading comprehension quiz generation using generative pre-trained transformers. In *iTextbooks@ AIED*, pages 4–17, 2022.

[11] L Xue. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*, 2020.

[12] Zichao Wang, Andrew S Lan, Weili Nie, Andrew E Waters, Phillip J Grimaldi, and Richard G Baraniuk. Qg-net: a data-driven question generation model for educational content. In *Proceedings of the fifth annual ACM conference on learning at scale*, pages 1–10, 2018.