International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 2, Mar-Apr 2026
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

AI Quiz Generator: An Intelligent Quiz Creation System Using Flutter and Large Language Models

Shravya Sharma[#1], Prof. Rajanikant Palwe[*2]

[#]*Department of Computer Engineering, Marathwada's Mitra Mandal's Polytechnic, Thergaon Pune-33, India*

[1]shravya_2303116@mmpolytechnic.com, [2]palwerm@mmpolytechnic.com

Abstract— The increasing demand for interactive and personalized learning tools has created a need for intelligent systems capable of generating educational content automatically. Traditional quiz creation methods require significant manual effort from educators and lack flexibility for rapid content generation across multiple topics and difficulty levels. This paper presents AI Quiz Generator, a mobile application developed using Flutter that automatically generates quizzes using a Large Language Model through the Groq API. The proposed system allows users to generate quizzes by specifying a topic, difficulty level, and quiz type. The application dynamically produces questions in multiple formats such as multiple-choice questions, short answer questions, and question banks. The system also integrates features such as a timer-based quiz system, performance evaluation with correct and incorrect answer analysis, review answer functionality, quiz retake options, and a quiz history dashboard for tracking user performance over time. User authentication and local data management are implemented using the Hive database to ensure fast and secure storage. The mobile interface is designed with multiple UI themes to enhance user experience and engagement. The application architecture combines Flutter's cross-platform capabilities with the intelligence of AI-driven content generation to deliver a flexible and scalable learning solution. Experimental usage of the application demonstrates that AI-generated quizzes significantly reduce the time required to create educational assessments while maintaining relevance and variability in question generation. The system provides an efficient solution for students, educators, and self-learners seeking automated quiz generation and interactive knowledge evaluation.

Keywords— ***AI Quiz Generator, Flutter Mobile Application, Large Language Models, Groq API, Educational Technology, Automated Quiz Generation, Mobile Learning Systems.***

Introduction

A. Background

With the rapid growth of digital learning platforms, educational systems are increasingly incorporating technology to enhance teaching and learning experiences. Traditional methods of quiz creation require educators to manually prepare questions, which can be time-consuming and repetitive. As the number of subjects and learning topics increases, manually generating quizzes for every topic becomes inefficient. Artificial Intelligence (AI), particularly Large Language Models (LLMs), has recently shown significant potential in generating educational content automatically. These models are capable of understanding context, generating relevant questions, and adapting the complexity of content according to difficulty levels. By integrating AI into learning applications, quizzes can be generated dynamically based on user input, reducing the effort required from educators while providing learners with personalized assessments. Mobile application frameworks such as Flutter allow developers to create cross-platform applications that work on multiple devices with a single codebase. By combining Flutter's mobile development capabilities with AI-powered content generation, it becomes possible to develop intelligent educational applications that are both interactive and scalable. The **AI Quiz Generator** system is designed to automate quiz generation using AI technology. The application allows users to generate quizzes based on topics and difficulty levels while also providing additional features such as timers, quiz results, answer review, and quiz history tracking.

B. Problem Statement

Generating quizzes manually for different subjects and topics can be time-consuming and inefficient, especially when large numbers of questions are required.

- **Manual Quiz Creation** — Traditional quiz systems require teachers or content creators to manually design questions, which is time-consuming and difficult to maintain for multiple subjects and topics.
- **Limited Question Variety** — Many existing quiz applications rely on fixed question banks, which reduces variability and may lead to repetitive learning experiences for users.
- **Lack of Dynamic Topic Support** — Most systems cannot automatically generate quizzes for newly entered topics, limiting flexibility for students who want practice on specific subjects.
- **Limited Performance Tracking** — Several quiz applications do not provide detailed result analysis such as correct, incorrect, and unanswered questions, making it harder for users to evaluate their learning progress.

To address these challenges, a system that can generate quizzes automatically with flexible topics and difficulty levels is required.

C. Motivation

The motivation behind developing the AI Quiz Generator application is to simplify the process of quiz creation while making learning more interactive and engaging. By leveraging AI technologies, quizzes can be generated instantly for any topic without requiring manual preparation.

Students often need quick practice tests when studying new topics. An AI-powered system that generates quizzes on demand allows learners to test their knowledge instantly. Additionally, educators can benefit from automated question generation when preparing assessments or practice materials.

Furthermore, integrating features such as performance tracking, quiz history, and result analysis allows users to monitor their learning progress over time. This combination of AI-powered content generation and mobile accessibility creates a powerful educational tool that can support modern learning environments.

D. Objectives

The primary objectives of this work are:

- To develop a mobile application that can generate quizzes automatically based on user-selected topics.
- To integrate a Large Language Model through the Groq API to dynamically generate quiz questions.
- To allow users to choose quiz difficulty levels such as easy, medium, and hard.
- To support multiple quiz formats including multiple-choice questions and short answer questions.
- To implement a timer-based quiz system to simulate real examination conditions.
- To provide quiz results that display correct, incorrect, and unanswered questions.
- To implement answer review and quiz retake functionality for improved learning.
- To design a user-friendly mobile interface with multiple themes to enhance user engagement.

I. LITERATURE REVIEW

A. AI-Based Quiz Generation Systems

Artificial Intelligence has significantly improved digital learning systems by enabling automatic content generation. AI-based quiz systems use Natural Language Processing (NLP) techniques to generate questions based on educational topics or textual input. Research on AI-based quiz systems for personalized learning demonstrates that AI-generated assessments can help students practice dynamically generated questions tailored to their learning needs [1]. Such systems reduce manual workload for educators while increasing flexibility in learning environments.

B. Automatic Question Generation Techniques

Automatic Question Generation (AQG) is an active research area in educational technology. AQG systems generate questions automatically from text using natural language processing and machine learning techniques. Kurdi et al. (2020) provide a comprehensive review of question generation methodologies, datasets, and evaluation techniques used in modern AQG systems [2]. These methods include rule-based systems, template-based generation, and deep learning approaches that improve question quality and scalability.

C. AI Integration in Learning Management Systems

Several studies have explored the integration of AI into Learning Management Systems (LMS). For example, AI-powered quiz generation implemented in the Canvas LMS demonstrates how automated systems can generate quizzes based on course materials and learning objectives [3]. This integration allows instructors to create quizzes efficiently while maintaining content relevance.

D. Natural Language Processing for Educational Content

Natural Language Processing plays an important role in automated question generation. NLP techniques such as text summarization, keyword extraction, and semantic analysis are commonly used to generate meaningful questions from textual content [4]. These techniques allow AI systems to understand the context of educational materials and produce relevant quiz questions.

E. Large Language Models in Education

Large Language Models (LLMs) such as GPT-based architectures have recently been applied in educational tools for automated content generation. These models can generate explanations, summaries, and quiz questions based on user prompts. Research shows that LLMs can improve the adaptability of learning systems by dynamically generating educational content based on user input [5].

F. Research Gap

Although previous studies have explored AI-based quiz generation and automatic question generation techniques, many existing systems are designed primarily for web-based platforms and lack mobile accessibility. Furthermore, several systems do not provide interactive features such as difficulty selection, timer-based quizzes, or performance tracking. The proposed AI Quiz Generator addresses these limitations by providing a mobile-based AI quiz generation system that dynamically creates quizzes while offering interactive learning features.

II. METHODOLOGY

A. System Architecture

The AI Quiz Generator system is designed as a mobile application developed using the Flutter framework. The system integrates Artificial Intelligence through the Groq API to dynamically generate quiz questions based on user input. The application architecture consists of several modules including user authentication, quiz generation, quiz evaluation, and result analysis.

The system workflow begins when the user enters a topic and selects a difficulty level for the quiz. The application sends a request to the Groq API, which generates relevant quiz questions using a large language model. The generated questions are then displayed in the application interface, where users can attempt the quiz within a specified time limit. After completion, the system evaluates the responses and displays the quiz results including correct, incorrect, and unanswered questions.

B. Quiz Generation and Processing Module

The Quiz Generation Module automatically creates quiz questions based on the user's selected topic and difficulty level. The application sends a prompt to the Groq API, which uses a Large Language Model (LLM) to generate relevant quiz questions dynamically. If the number of questions is not specified, the system generates a default set of ten questions.

The module supports different quiz formats such as Multiple Choice Questions (MCQ), one-line questions, and question banks. The generated questions are then structured and displayed in the Flutter mobile interface, ensuring that the quiz matches the selected difficulty level and topic.

Dart:

```
Future<List<QuizQuestion>> generateQuiz(
    String topic, String difficulty, int questionCount) async {
  final response = await http.post(

Uri.parse("https://api.groq.com/openai/v1/chat/completions"),
    headers: {
      "Authorization": "Bearer API_KEY",
      "Content-Type": "application/json"
    },
    body: jsonEncode({
      "model": "llama3-8b-8192",
      "messages": [
        {
          "role": "user",
          "content":
            "Generate $questionCount quiz questions about
$topic with $difficulty difficulty level."
        }
      ]
    }),
  );
  if (response.statusCode == 200) {
    final data = jsonDecode(response.body);
    return parseQuizQuestions(data);
  } else {
    throw Exception("Failed to generate quiz");
  }
}
```

C. AI-Based Quiz Generation Using Groq API

The core functionality of the system is the automatic generation of quizzes using Artificial Intelligence. The application sends a prompt to the Groq API containing the selected topic and difficulty level. The AI model processes the request and generates quiz questions dynamically.

Dart:

```
Future<String> generateQuiz(String topic, String difficulty)
async {
  final response = await http.post(
 Uri.parse("https://api.groq.com/openai/v1/chat/completions"),
    headers: {
      "Authorization": "Bearer API_KEY",
      "Content-Type": "application/json"
    },
    body: jsonEncode({
      "model": "mixtral-8x7b",
      "messages": [
        {"role": "user", "content": "Generate a $difficulty quiz
about $topic"}
      ]
```



```
    }),
  );
  return response.body;
}
```

D. Quiz Evaluation and Result Module

The Quiz Evaluation Module analyzes the user's responses after the quiz is completed. The system compares the selected answers with the correct answers and calculates the number of correct, wrong, and unanswered questions. Based on this evaluation, the final score is generated and displayed on the results screen. The application also allows users to review their answers and retake the quiz if needed. Additionally, quiz attempts are stored in the Hive database so users can track their quiz history and monitor their learning progress.

Parameter	Description
Total Questions	Number of questions in the quiz
Correct Answers	Questions answered correctly
Wrong Answers	Incorrect responses
Unanswered	Questions not attempted
Score	Final quiz result
Quiz History	Stores past quiz attempts

TABLE I. Quiz Evaluation Parameters

E. User Interface and Theme Customization Module

The User Interface Module is responsible for displaying quizzes, results, and dashboards in a clear and interactive format. The application provides multiple UI themes that allow users to customize the appearance of the app for better usability. Flutter widgets are used to design responsive screens for quiz questions, results, and history. A recommended improvement is to add accessibility options such as larger fonts and high-contrast themes for better readability.

Dart:

```
ThemeData darkTheme = ThemeData(
  brightness: Brightness.dark,
  primaryColor: Colors.blue,
  scaffoldBackgroundColor: Colors.black,
);
```

F. System Data Flow Diagram

The system processes user input to generate quizzes and store results efficiently. The user enters a topic and quiz preferences through the mobile interface. The application sends this request to the Groq API for quiz generation and displays the generated questions to the user. After the quiz is completed, the system evaluates the answers and stores the results in the Hive database. It is recommended to implement validation checks to ensure correct data flow between modules.

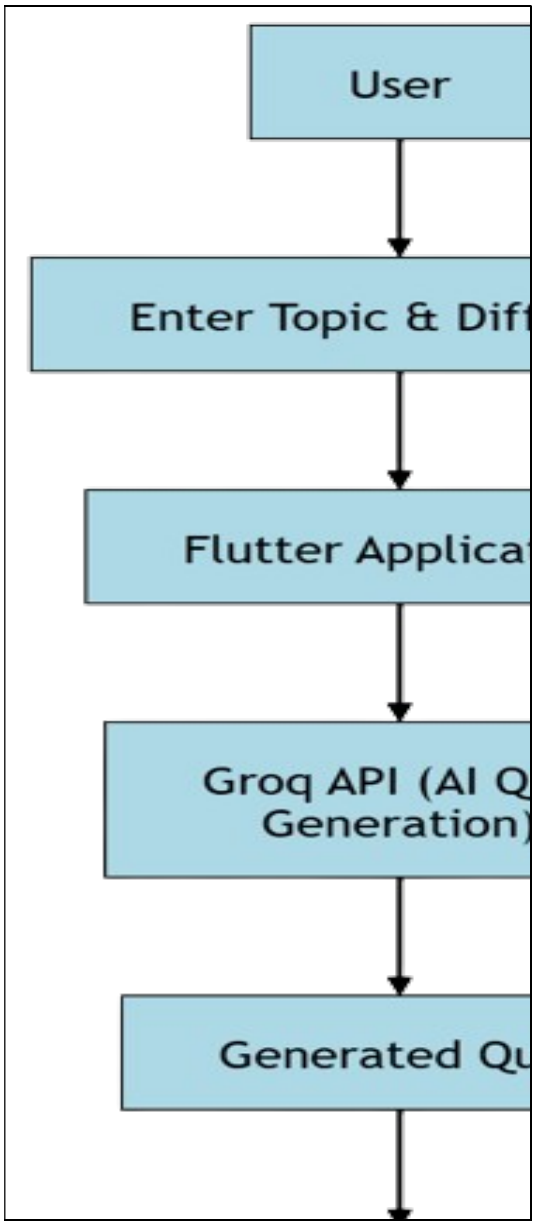

FIGURE I. Data Flow Diagram

G. Technology Stack and Implementation Environment

The AI Quiz Generator application is developed using modern mobile and AI technologies. Flutter and Dart are used for building the mobile interface, while the Groq API enables AI-based quiz generation. Hive database is used for efficient local data storage and quiz history management. It is recommended to optimize API calls and caching mechanisms to improve performance.

Component	Technology Used
Mobile Development	Flutter
Programming Language	Dart
AI Integration	Groq API
Database	Hive
API Communication	REST API

TABLE II. System Technology Stack

III. Results and Discussion

A. System Testing and Functionality

The AI Quiz Generator application was tested to verify its ability to generate quizzes dynamically and evaluate user responses. The system successfully generated quizzes based on the selected topic and difficulty level. Users were able to attempt quizzes, view results, review answers, and retake quizzes without errors. The application also stored quiz history using the Hive database.

Feature Tested	Result
Quiz Generation	Successful
Difficulty Selection	Working correctly
Quiz Evaluation	Accurate results
Quiz Retake	Functional
Quiz History Storage	Stored in Hive database

TABLE III. System Functionality Testing

B. Quiz Generation Performance

The AI Quiz Generator was evaluated based on its ability to generate quizzes quickly and accurately using the Groq API. Tests were performed with different topics and difficulty levels to verify that the system produces relevant questions. The results showed that the application consistently generated quizzes and displayed them correctly in the mobile interface.

Topic	Difficulty	Questions Generated	Final Verdict
Python Basics	Easy	10	Successful
Computer Networks	Medium	10	Successful
Data Structures	Hard	10	Successful
Operating Systems	Medium	10	Successful

TABLE IV. Sample Quiz Generation Output

C. Quiz Evaluation and Result Accuracy

The system was tested to verify the accuracy of quiz evaluation and result generation. After completing a quiz, the application correctly calculated the number of correct, wrong, and unanswered questions. The results were displayed immediately on the results screen, and quiz attempts were successfully stored in the Hive database for history tracking.

Total Questions	Correct Answers	Wrong Answers	Unanswered	Score
10	8	2	0	80%
10	6	3	1	60%
10	7	2	1	70%
10	9	1	0	90%

TABLE V. Sample Quiz Result Evaluation

These results demonstrate that the system accurately evaluates quiz responses and provides immediate feedback to users.

D. User Interface and Usability

The user interface was tested for smooth navigation and usability. Users can generate quizzes, answer questions, view results, and access quiz history, while UI themes allow customization without affecting functionality.

UI Feature	Observation
Quiz Interface	Easy to understand
Navigation	Smooth between screens
Result Display	Clear and readable
Theme Selection	Working correctly
Quiz History Screen	Accessible and functional

TABLE VI. UI Usability Testing

E. System Performance

The overall performance of the AI Quiz Generator was evaluated based on quiz generation speed, application responsiveness, and result processing efficiency. During testing, the system successfully generated quizzes using the Groq API and displayed them in the Flutter mobile interface without significant delay. The application processed user responses instantly and calculated quiz results accurately.

The Hive database handled quiz history storage efficiently, allowing users to access previous attempts without performance issues. Overall testing indicated that the system operates smoothly on mobile devices and provides a responsive user experience during quiz generation and evaluation.

Metric	Observation
Quiz Generation Time	2 - 4 seconds
Result Processing	Instant
UI Response	Smooth navigation
Data Storage	Efficient storage using Hive

TABLE VII. System Performance Metrics

F. Discussion

- The results obtained from system testing indicate that the AI Quiz Generator functions effectively as an automated quiz generation platform. The integration of the Groq API enables the system to generate topic-based quizzes dynamically, reducing the need for manual question preparation. The evaluation module accurately calculates quiz results and provides immediate feedback to users.

- The use of Flutter ensures a responsive mobile interface, while the Hive database efficiently manages quiz history and user data. Overall, the system demonstrates reliable performance for generating quizzes, evaluating answers, and maintaining user interaction within the mobile learning environment.

- The system demonstrates the practical use of Artificial Intelligence in educational applications by enabling automatic quiz generation for various subjects and difficulty levels.

- The mobile-based implementation improves accessibility, allowing students to practice quizzes anytime and track their learning progress through stored quiz history.

IV. CONCLUSION

A. Summary

The AI Quiz Generator system successfully demonstrates the use of Artificial Intelligence to automate quiz creation and learning assessment through a mobile application. The system addresses the limitations of traditional quiz preparation methods by providing an intelligent platform that dynamically generates quizzes based on user-selected topics and difficulty levels. By integrating the Groq API with a Flutter-based mobile interface, the application enables users to instantly generate quizzes without requiring manual question preparation.

The system processes user input through an automated workflow in which the selected topic and difficulty level are sent to the Groq API, where a Large Language Model generates relevant quiz questions. The generated questions are then structured and displayed within the Flutter mobile interface in different formats such as Multiple Choice Questions (MCQ), one-line questions, and question banks. The system further enhances learning by incorporating features such as timer-based quizzes, answer review screens, result analysis, and quiz retake functionality. These features provide users with a more interactive and engaging learning experience compared to conventional quiz applications.

The evaluation module of the system accurately analyzes user responses and calculates quiz results by identifying correct, incorrect, and unanswered questions. The results are displayed immediately after quiz completion, allowing users to review their performance and understand areas for improvement. Additionally, the application stores quiz attempts and performance data using the Hive database, enabling users to access quiz history and monitor their learning progress over time.

The system also focuses on usability and accessibility through a responsive mobile interface designed using Flutter. Multiple UI themes allow users to customize the appearance of the application according to their preferences while maintaining consistent functionality. Testing results confirmed that the system successfully generates quizzes, processes user responses efficiently, and stores quiz data reliably without affecting application performance.

Overall, the AI Quiz Generator provides an effective solution for automated quiz generation and interactive learning assessment. By combining AI-powered question generation with mobile application technology, the system demonstrates how intelligent educational tools can simplify quiz creation while enhancing the learning experience for students and educators.

B. Future Scope

- Adaptive learning features can be integrated so that the quiz difficulty automatically adjusts based on the user's performance and learning progress over time.
- Support for multimedia-based questions such as images, diagrams, and audio can be added to make quizzes more interactive and visually engaging.
- Cloud-based data storage can be implemented to allow users to synchronize quiz history and performance data across multiple devices instead of relying only on local storage.
- A teacher or administrator dashboard can be developed to allow educators to create quizzes, manage question sets, and monitor student performance.
- Advanced analytics and progress tracking features can be added to provide detailed reports and performance insights for users.

ACKNOWLEDGMENT

The authors would like to thank their institution and department for the support and resources provided during the course of this research.

REFERENCES

[1] M. S. Rahman et al., "AI-Based Quiz System for Personalized Learning," International Journal of Educational Technology, 2023.

[2] G. Kurdi, J. Leo, and R. Parsia, "Automatic Question Generation: A Review of Methodologies, Datasets, Evaluation Metrics, and Applications," ACM Computing Surveys, vol. 53, no. 5, 2020.

[3] J. Smith and L. Brown, "AI-Powered Quiz Generation for Learning Management Systems: A Full-Stack Implementation for Canvas LMS," IEEE Access, 2022.

[4] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed., Pearson, 2021.

[5] T. Brown et al., "Language Models are Few-Shot Learners," Advances in Neural Information Processing Systems (NeurIPS), 2020.

[6] Google Developers, "Flutter: Building Cross-Platform Applications," Google Technical Documentation, 2023.

[7] J. Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," NAACL, 2019.

[8] Y. Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," arXiv:1907.11692, 2019.

[9] A. Vaswani et al., "Attention Is All You Need," Advances in Neural Information Processing Systems, 2017.

[10] H. Chen et al., "Question Generation for Educational Applications using NLP," IEEE Transactions on Learning Technologies, 2021.

[11] S. K. Gupta and R. Patel, "AI-Based Educational Applications for Smart Learning Environments," International Journal of Computer Applications, 2022.

