RAJIV GANDHI INSTITUTE OF TECHNOLOGY

GOVERNMENT ENGINEERING COLLEGE

KOTTAYAM-686501

DEPARTMENT OF COMPUTER APPLICATIONS

20MCA245 - MINI PROJECT REPORT

OCTOBER 2024

AI-POWERED MCQ GENERATION

Submitted By

VISHNU PRASAD C P

(KTE23MCA-2060)

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

THIRUVANANTHAPURAM

Contents

CHAPTER 1

INTRODUCTION

The AI-powered MCQ Generation is a web application for creating and managing multiple-choice questions (MCQs). Using advanced AI and natural language processing (NLP) question setters can easily add questions including options and the answers, or let AI generate them. Built with Python Flask and MySQL, it efficiently manages data and categorizes questions by subject. Moderators verify questions for accuracy and Test takers can log in to take quizzes, view scores, and report errors. Additionally, they can input text in various formats including PDF to generate questions with the help of AI models and add these questions to their local pool.

1.1 SCOPE OF THE PROJECT

The AI-powered MCQ Generation is a free resource for test takers preparing for entrance exams, offering a cost-effective alternative to existing systems. By automating MCQ generation, the platform assists Question Setters in adding questions quickly and accurately. It also enables moderators to validate and ensure the quality of questions added to the system. The system supports test takers by allowing them to solve MCQs, track their performance, and generate custom MCQs from their study materials. This can be done using various text inputs, such as PDFs, leveraging NLP techniques to create meaningful practice questions. The project also offers a global pool of questions for all test takers, continuously improved and expanded by moderators, making it a dynamic and ever-evolving educational resource. The application's ability to categorize questions by subject and handle user-generated content from PDFs makes it versatile for various educational levels and subjects

1.2 RELEVANCE OF THE PROJECT

The project's relevance lies in its potential to democratize education by providing an AI-driven platform accessible to all. Many students preparing for competitive exams face significant financial barriers to accessing quality study resources. This application aims to bridge this gap by offering an effective, interactive, and personalized learning experience for free. By automating MCQ creation, it reduces the workload of educators while maintaining content quality through moderation. The AI component, including NLP and PDF content extraction, enables the system to efficiently convert textual data into useful practice questions, enhancing students' preparation. Additionally, the platform's emphasis on personalized learning, where students can create their local pools of questions, ensures that study sessions can be tailored to individual needs and knowledge gaps, ultimately improving learning outcomes.

1.3 ORGANISATION OF THE REPORT

This report is structured into several chapters to provide a clear understanding of the "AI-POWERED MCQ GENERATION" project. Chapter 1: Outlines the scope and relevance of the project. Chapter 2: Analyzes the existing system and introduces the proposed system along with the development methodology. Chapter 3: Designs include WireFrame model, Usecase Diagram. Chapter 4: Covers the development and deployment process. Chapter 5: Results Analysis presents the outcomes and evaluates the system's performance. Chapter 6: Conclusion Future Scope summarizes the project and suggests future improvements. The report concludes with References and an Appendix containing additional materials like Git history, sample code, and screenshots.

CHAPTER 2

SYSTEM STUDY

The system study evaluates the current shortcomings of traditional MCQ creation and explores how the proposed AI-driven solution addresses these gaps. The traditional method of creating MCQs is labor-intensive and lacks adaptability. Subject categorization is done manually, which takes time and is prone to errors. Moreover, there is no seamless integration for students to extract content from PDFs or other text sources to create questions, which limits the flexibility of study resources. The proposed system tackles these issues by offering an automated way of generating and categorizing MCQs using NLP, thus streamlining the entire process and making it significantly more efficient and user-friendly. This automation supports both question setters in quickly building a comprehensive question pool and test takers in accessing diverse questions tailored to their study needs.

2.1 EXISTING SYSTEM

The existing system for MCQ creation relies heavily on manual processes, which involve educators crafting each question, along with its options and correct answer, by hand. This method is not only time-consuming but also susceptible to inconsistencies and human error. Another significant limitation is the lack of integration for converting study materials, such as PDFs, into MCQs. This prevents students from interactively utilizing their existing resources, limiting their ability to engage in personalized learning. These shortcomings underscore the need for a more dynamic and automated system that can cater to both educators and students efficiently.

2.2 PROPOSED SYSTEM

The proposed system introduces several key improvements to overcome the limitations of the existing approach. It supports both automated and manual creation of MCQs, giving question setters the flexibility to either input questions themselves or let the AI generate them. This dual capability ensures a larger and more diverse question pool. One of the standout features of the proposed system is its ability to convert study material from PDFs into MCQs. By using advanced NLP techniques, students can extract content from their study materials and turn it into practice questions, thereby creating a more interactive and personalized learning experience. This automation and integration help streamline the process for both educators and learners, making the system far more scalable and effective in delivering quality education.

2.3 DEVELOPMENT METHODOLOGY

This project follows an agile development methodology, which emphasizes iterative progress, continuous user feedback, and the flexibility to adapt to changes. Agile allows the team to develop the application in manageable increments, ensuring that features are built, tested, and reviewed on a regular basis. The key features of the application include:

- **Authentication :** Different roles such as Admin, Moderator, Question Setter, and Test Taker each have dedicated registration and login modules. This ensures that users have access to features relevant to their roles, providing security and a structured flow of responsibilities .
- **Admin Functions :** The Admin plays a crucial role in managing the system by verifying moderators, managing all users, and ensuring that the platform remains well-regulated. Additionally, the Admin is responsible for verifying complaints and replying to them, thereby maintaining a smooth and supportive user experience for all members of the platform.
- **Question Setter Activities :** Question setters are provided with an intuitive interface where they can manually add MCQs, including multiple options and the correct answer. Additionally, AI-based models assist in automatically generating MCQs based on input text. This dual capability allows for efficient question generation and expansion of the question pool.

- **Moderator Responsibilities :** Moderators verify the quality of questions added by question setters or generated by AI. They have the ability to modify questions, correct errors, and ensure that the content added to the global question pool meets the necessary standards for quality and relevance.
- **Test Taker Capabilities :** Test takers have access to a dynamic question-solving module where they can practice MCQs, view solutions, and track their progress over time. They can also generate new questions from their study material in formats like PDFs, enabling personalized practice. This ensures that students can focus on their weaker areas and effectively enhance their performance.

2.4 REQUIREMENT ANALYSIS

The development of the AI-powered MCQ Generation project requires specific software and hardware configurations to ensure efficient performance and effective delivery of its features.

2.4.1 Software Requirements

The system relies on a range of software tools and libraries that collectively enable the different functionalities of the application. For the frontend, **HTML, CSS,** and **Bootstrap** are used to create an intuitive and responsive user interface, ensuring that all features are accessible and user-friendly to students, question setters, moderators, and administrators. The backend of the application is built using **Python Flask**, a lightweight and flexible web framework that allows for rapid development and easy integration with other components.

The project uses **NLTK** to handle the natural language processing tasks required for automatic MCQ generation. These **NLP** libraries are critical for generating meaningful questions and analyzing text inputs. The project leverages **google.generativeai** for advanced AI-driven question generation, enhancing the diversity and quality of the MCQs created, thus providing students with a comprehensive practice experience.

For converting study materials from PDF format into MCQs, **PyPDF2** is employed. This library allows students to upload their PDFs and automatically generate questions based on the content, adding

convenience and personalization to the study process. **MySQL** is used as the relational database management system to store user data, MCQs, and other essential information. This ensures data persistence and efficient retrieval for all user roles.

The application is developed using **PyCharm**, an integrated development environment (IDE) that facilitates efficient coding, debugging, and project management. For version control, **GitHub** is used, providing essential tools for collaboration, code backup, and management of changes throughout the development lifecycle.

2.4.2 Hardware Requirements

To run the application and ensure that it functions smoothly, certain hardware specifications are recommended for both development and deployment. The system requires **multi-core processors** to handle multiple processes simultaneously. This is especially important for tasks involving MCQ generation and text analysis, which can be resource-intensive due to the natural language processing (NLP) algorithms and AI components used.

The application also demands at least **8GB of RAM** to provide sufficient memory for running the NLP libraries and handling multiple users concurrently. Adequate storage capacity is needed to store user data, generated MCQs, extracted PDF content, and logs.

Lastly, the system is developed and tested on **high-performance PCs or laptops**. These development machines provide the processing power and reliability needed to implement and debug all features effectively. Ensuring the use of robust development hardware contributes to the stability and efficiency of the overall application.

CHAPTER 3

DESIGN

The design phase is a crucial step in the development of the AI-powered MCQ Generation project. It involves creating a detailed blueprint of the system's architecture, user interface, and interaction flows to ensure that the final product meets the specified requirements. The purpose of the design phase is to translate the functional requirements into a coherent structure that developers can follow during implementation. This phase includes the creation of use case diagrams, wireframes, and system architecture models, which provide a clear visualization of how different components of the system will interact with each other.

3.1 USECASE DIAGRAM

The use case diagrams play a vital role in understanding the interactions between different users and the system. They help identify the different roles, such as Admin, Moderator, Question Setter, and Test Taker, and their respective interactions with the system's features. These diagrams represent how each user interacts with functionalities like MCQ generation, moderation, user management, and question solving, making it easier for developers to understand the requirements and design a system that aligns with the needs of each role.

A use case refers to a specific scenario that outlines how different user roles interact with the system to achieve particular goals. Each use case defines the steps involved in a user's interaction with the application. Use cases help in understanding user requirements, guiding the design and development of the application to ensure it meets the needs of all stakeholders involved.

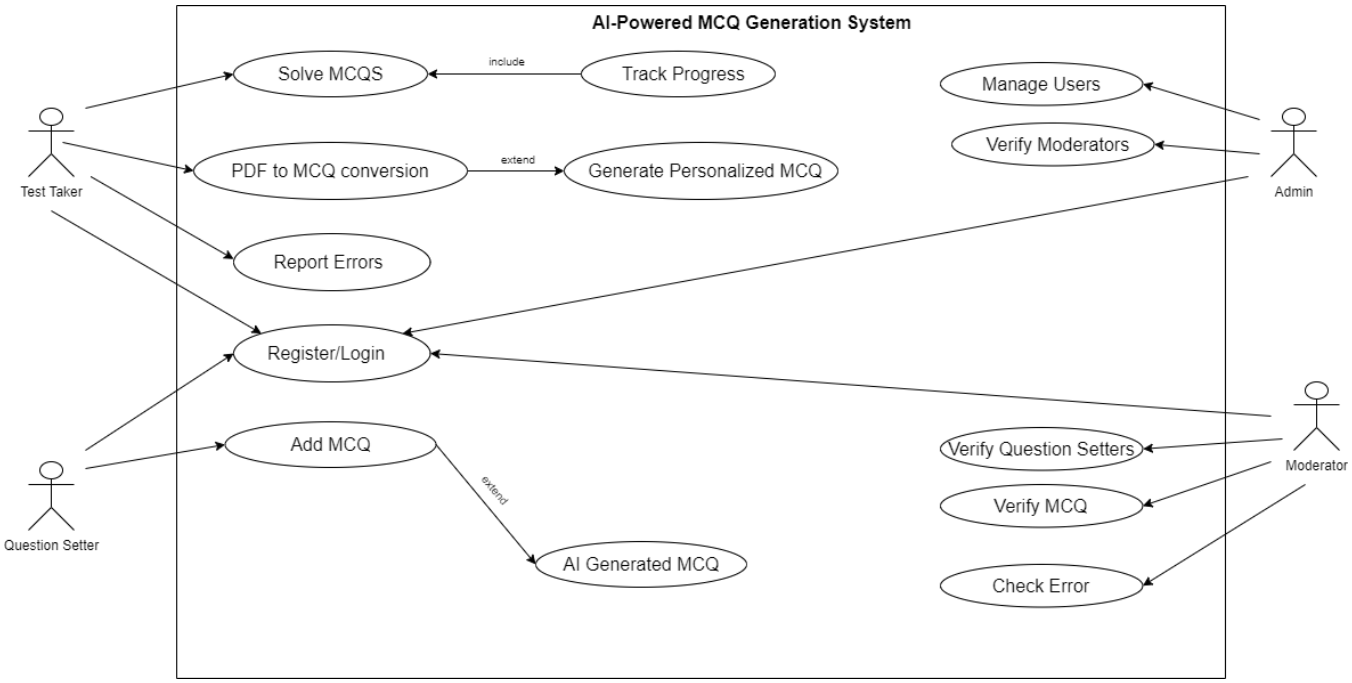

Figure 3.1: Use Case Diagram for AI-powered MCQ Generation System

The use case diagram for the MCQ-based platform illustrates the interactions between various users and the system's key functionalities. The diagram highlights the roles of different actors, which are likely categorized as Test Takers, Question Setters, Moderators, and Admin.

The left side of the diagram focuses primarily on the Test Takers and Question Setters functionalities. Test Takers interact with features such as "Solve MCQs", and "Track Progress" where they can take quizzes or assessments provided by the platform. Additionally, they can utilize the "PDF to MCQ Conversion" feature, which likely allows them to upload documents and convert them into multiple-choice questions. Other key features available to Test Takers include the ability to "Report Errors" in MCQs, which ensures that any mistakes or inconsistencies can be flagged for review. Both Test Takers and Question Setters have access to the "Register/Login" functionality to access the platform, and question setters can add new MCQs using the "Add MCQ" feature, enabling them to contribute new content to the system.

On the right side of the diagram, we see actions related to the Administrator and Moderators roles, such as "Manage Users," "Verify Moderators," "Verify Question Setters," "Verify MCQs," and "Check Errors." These functions ensure that the platform is well-regulated and that only verified content and users are allowed access.