INDIAN INSTITUTE OF
INFORMATION
TECHNOLOGY

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY DHARWAD

MINI PROJECT REPORT

Automatic Question Generation using Generative AI on User Context

Submitted by

GANDHIRAJ (22BDS017)
SOMASEKHAR (22BDS022)
SREESHANTH (22BDS016)
ARAVIND (22BDS014)

Under the guidance of

Dr. Krishnendu Ghosh

April 2025

Certificate

This is to certify that the project, entitled **"Automatic MCQ Generator from Text or File Upload using Flask and Google Gemini AI"**, is a bonafide record of the Mini Project coursework presented by the students whose names are given below during the academic year 2024-2025 in partial fulfilment of the requirements of the degree of Bachelor of Technology in Computer Science and Engineering.

22BDS017   Ch D S Gandhi Raj
22BDS022   Soma Sekhar
22BDS016   Sreeshanth
22BDS014   Aravind

Dr. Krishnendu Ghosh
(Project Supervisor)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Creating MCQs manually is a time-consuming and repetitive task. This project introduces a fully automated approach that enables users to generate MCQs through a web-based interface by either typing the content or uploading documents. With the integration of Google Gemini AI, the system can process and understand the context of the content, ensuring accurate and relevant question generation.

As digital education grows, the need for scalable and efficient assessment tools becomes crucial. Our system addresses this need by providing educators with a seamless experience to create quizzes, saving time while maintaining quality.

Chapter 2

Related Work

Several tools exist for educational content generation, but most rely on static question banks or manual entry. Automated question generation using AI has gained interest recently due to advancements in natural language processing.

- **Manual MCQ Creation:** Traditional methods depend on human effort, consuming considerable time.

- **Rule-Based Systems:** These systems follow predefined patterns, often resulting in generic questions.

- **AI-Based Approaches:** Models like Google Gemini AI provide context-aware question generation, which enhances the quality and relevance.

Chapter 3

Data and Methods

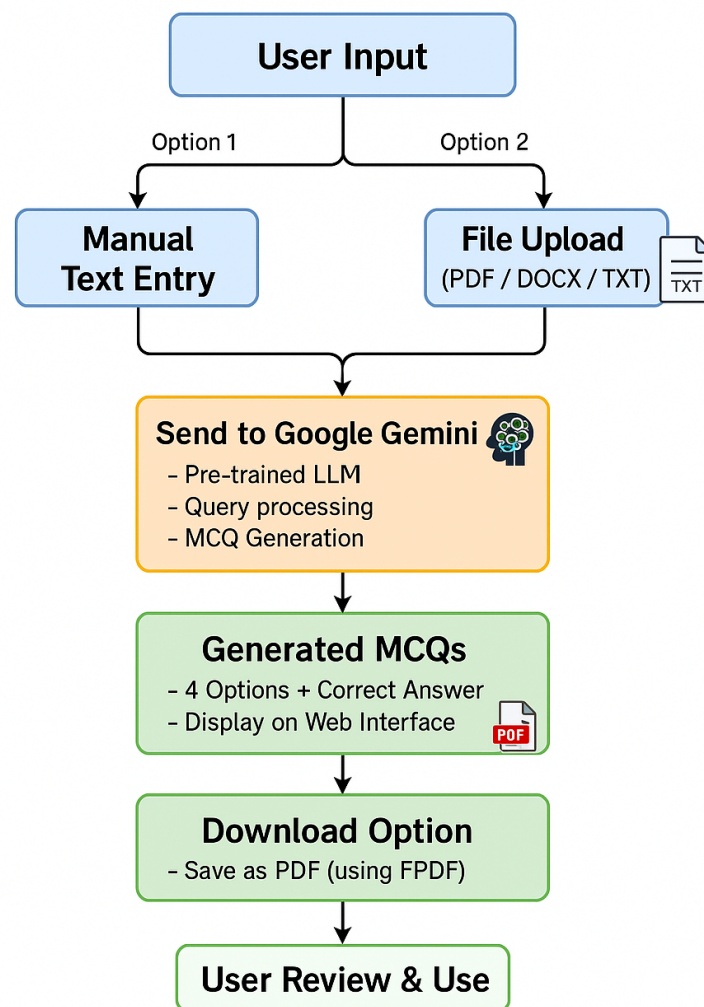## System Workflow

Workflow of Automatic MCQ Generator

Figure 3.1: Workflow of Automatic MCQ Generator

Steps Involved:

1. **Web Interface Development:** Built with Flask, allowing text input and file upload.

2. **Text Extraction:** Handled using Python libraries like `pdfplumber` and `python-docx`.

3. **MCQ Generation:** Google Gemini AI processes the content and generates MCQs.

4. **Result Presentation:** Questions are displayed on the webpage and downloadable as a PDF using `FPDF`.

Technical Stack

Table 3.1: Technology Stack Used

Component	Technology
Frontend	HTML, CSS, Flask Templates
Backend	Python, Flask
AI Model	Google Gemini AI
Text Extraction	pdfplumber, python-docx
PDF Generation	FPDF

Chapter 4

Results and Discussions

The system successfully processes a variety of educational content and converts it into structured multiple-choice questions. Testing confirmed its versatility across subjects and formats.
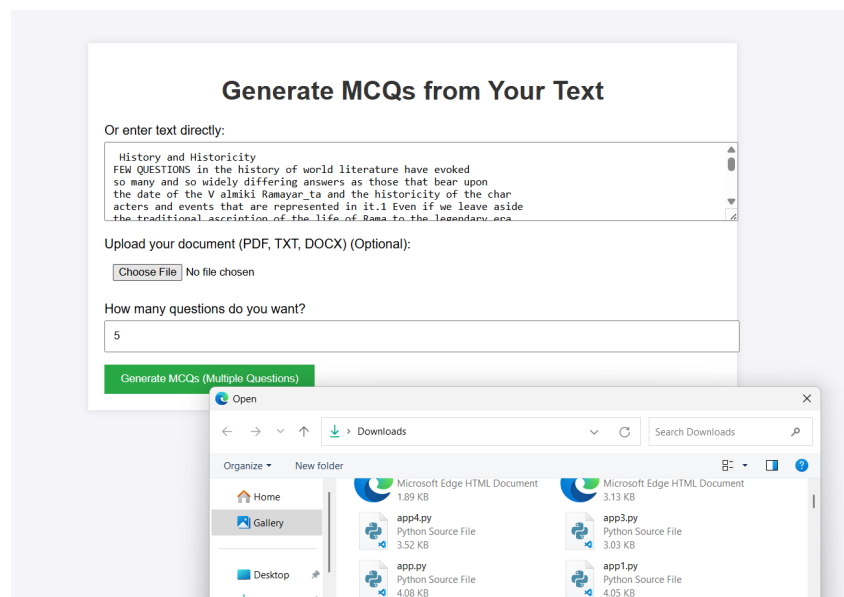

Figure 4.1: User Interface for Input

Figure 4.2: Generated MCQs Display

Figure 4.3: Input the No of Questions

Performance Comparison

Table 4.1: Manual vs. Automated MCQ Generation

Criteria	Manual Creation	Automated (Our System)
Time Required	High	Low
Consistency	Varies	High
Scalability	Limited	Excellent
Customization	Moderate	High
User Effort	High	Low

Chapter 5

Conclusion

The "Automatic MCQ Generator from Text or File Upload using Flask and Google Gemini AI" proves to be an efficient solution for automated assessment generation. By integrating advanced AI capabilities with a user-friendly interface, the system simplifies the process of creating quizzes for educators.

Future improvements may include:

- Enhanced formatting and design of generated PDFs.

- Support for additional question types (True/False, Fill in the blanks).

- Multi-language support for broader accessibility.

Chapter 6

References

1. Flask Documentation: `https://flask.palletsprojects.com/`

2. Python pdfplumber: `https://github.com/jsvine/pdfplumber`

3. python-docx: `https://python-docx.readthedocs.io/`