Available online at
www.mdl.mazedan.com

www.mazedan.com

MCET
www.mazedan.com/mcet

MAZEDAN COMPUTER ENGINEERING TRANSACTIONS
e-ISSN: 2583-0414
Article id: MCET0602022
Vol.-6, Issue-2
Received: 24 May 2025
Revised: 15 Jun 2025
Accepted: 30 Jun 2025

GENERATION USING BLOOM'S TAXONOMY

PROF. P. G. KANADE*, JAYESH GUDAGHE, SANJASVI SONAWANE, MAHEK SHAIKH, SARTHAK GHUMARE

DOI: https://doi.org/10.5281/zenodo.16743497

Citation: Kanade, P. G., Gudaghe, J., Sonawane, S., Shaikh, M. & Ghumare, S. (2025). Generation Using Bloom's Taxonomy. *Mazedan Computer Engineering Transactions*, 6(2), 87–91.

Abstract

Designing question papers manually is often time-consuming and inconsistent, especially when trying to align with academic standards like Bloom's Taxonomy and Course Outcomes (COs). This project introduces a smart and user-friendly system that automates the question paper generation process, allowing educators to upload syllabus PDFs, set custom configurations, and instantly generate descriptive-only papers along with answer keys. The system ensures cognitive balance, supports customizable templates, and leverages natural language processing for relevant question selection. Designed for efficiency and outcome-based evaluation, it streamlines the assessment process in academic settings.

Keywords— Automatic Question Generation, Bloom's Taxonomy, Course Outcomes, Stream lit, Descriptive Questions, Educational Technology.

1. INTRODUCTION

Assessment is a cornerstone of education, serving not only as a measure of student learning but also as a tool to shape curriculum design, teaching strategies, and academic standards. One of the most traditional yet essential forms of assessment is the written question paper. Despite its importance, the process of manually crafting question papers remains a time-consuming and repetitive task for educators. It often involves ensuring alignment with institutional objectives, maintaining consistency in difficulty levels, and achieving a balanced coverage of the syllabus all while trying to reduce redundancy and personal bias.

In recent years, educational institutions have increasingly adopted outcome-based education (OBE) frameworks, where the focus lies on achieving well-defined learning outcomes. Within this paradigm, Bloom's Taxonomy has emerged as a widely accepted classification system to design and evaluate learning activities. It provides a structured way to assess students at various cognitive levels from remembering and understanding to analyzing and creating.

In parallel, institutions also define Course Outcomes (COs) that represent specific learning goals associated with each subject or module. Mapping question papers to both Bloom's levels and COs ensures that assessments are not only rigorous but also aligned with academic goals.

This paper introduces a smart, customizable system for automated generation of descriptive-only question papers, built to reduce manual effort and enhance the quality of assessment design. Educators can upload the syllabus in PDF format and configure parameters such as exam type (Mid/Final), subject name, topic, mark distribution, and difficulty level. Based on these inputs, the system uses predefined templates to generate structured question papers and corresponding answer keys. Notably, the tool avoids multiple-choice questions and focuses purely on descriptive formats, including short and long answer types.

The system is developed using Streamlit, an open-source Python framework that supports rapid creation of web applications. It provides an intuitive interface for educators, eliminating the need for technical expertise. Internally, the application manages a dynamic question bank, maintains a history of generated papers, and supports downloading outputs in professional PDF format.

Overall, this project aims to improve the reliability, consistency, and efficiency of question paper generation, especially in institutions implementing outcome-based assessment strategies. By automating routine academic workflows, it empowers educators to invest more time in content quality and pedagogical improvement.

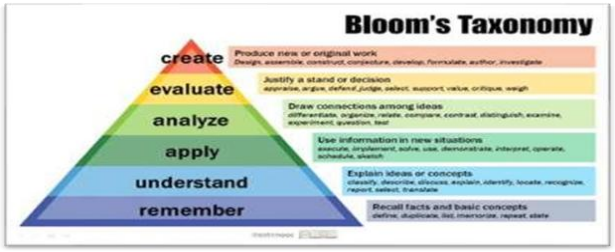

Figure 1 Blooms Taxonomy Levels

S.N.D. College of Engineering & RC, Yeola, Maharashtra, India
*Corresponding author email-kanadepg87@gmail.com

Related Work

The automation of question paper generation has received growing attention in recent years, particularly in the context of reducing repetitive academic workload and improving alignment with educational objectives. Traditional systems rely heavily on manual input from faculty members to draft questions, balance difficulty levels, and ensure comprehensive syllabus coverage. This process is not only time-intensive but also subject to inconsistencies across different instructors and institutions.

Early digital systems primarily served as repositories of static question banks. These platforms allowed educators to select questions from categorized lists based on subject or chapter. While this approach streamlined the retrieval of questions, it still required considerable manual effort to assemble a well-balanced paper. Moreover, many of these systems lacked a formal structure or framework to guide cognitive-level distribution or outcome mapping.

To address these limitations, some academic tools introduced randomization and rule-based selection. In these models, questions could be tagged with metadata such as marks, difficulty, topic, and type (MCQ, short answer, long answer), allowing for semi-automated generation of question sets based on filters. However, these systems often lacked intelligent templates or did not explicitly support education frameworks like Bloom's Taxonomy, which classifies learning objectives across hierarchical cognitive levels (e.g., Remember, Understand, Apply, Analyze, Evaluate, Create).

Further, a significant shortcoming in many commercial and institutional tools is their over-reliance on objective-type questions particularly multiple-choice questions (MCQs) due to the ease of evaluation. While this might suite large-scale assessments, it limits the depth and scope of cognitive measurement. Descriptive assessments, on the other hand, are essential for evaluating higher-order thinking and analytical skills. Yet, very few systems are dedicated solely to the structured generation of descriptive-only question papers.

There are also efforts where tools integrate question paper generation with student performance analytics or learning management systems (LMS). While these offer broader assessment ecosystems, they are typically not open source and often come with steep learning curves or subscription fees. Additionally, they may not support fine-tuned customization or integration with course-specific outcomes (COs), which is increasingly important in institutions implementing Outcome-Based Education (OBE).

In contrast to these prior approaches, the system proposed in this paper is a lightweight, template-driven application focused entirely on the generation of descriptive question papers. It uses pre-defined templates that map to question types and Bloom's cognitive levels, allowing educators to choose exam types (e.g., Mid or Final), set the number of 5-marks, 6-marks, 7-mark questions up to 9 define difficulty, and upload syllabus documents. Questions are then selected from a structured question bank aligned with the subject and topic, ensuring systematic and outcome-aligned paper construction.

Moreover, the proposed system emphasizes usability by leveraging a clean interface through Streamlit, a Python-based web framework. It does not rely on machine learning, artificial intelligence, or external APIs, making it lightweight, fast, and deployable in any academic environment without the need for specialized technical infrastructure.

2. SYSTEM ARCHITECTURE

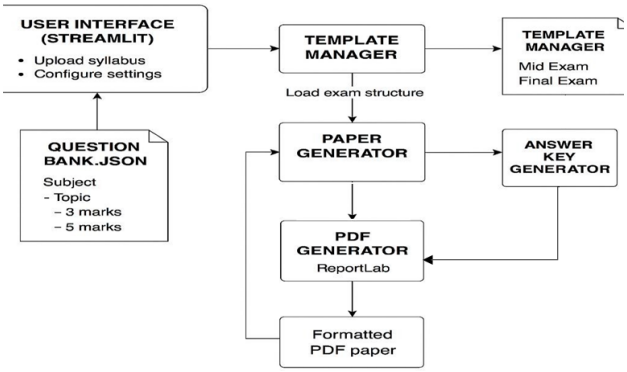

Figure 2 System Architecture

The proposed system is designed with simplicity, modularity, and educator-friendliness in mind. It follows a structured yet lightweight architecture built entirely using Python and the Streamlit framework for the user interface. The core idea is to automate the generation of descriptive question papers while maintaining alignment with Bloom's Taxonomy and Course Outcomes (COs). The architecture can be divided into the following key components:

User Interface (Streamlit)

The front-end is developed using Streamlit, which provides an interactive and intuitive interface for educators. It allows users to:

- Upload a syllabus document in PDF format.
- Select the exam type (e.g., Mid Exam, Final Exam).
- Configure parameters such as subject, topic, difficulty level, number of 3-mark and 5-mark questions, and whether to include an answer key.
- View the generated question paper on-screen and download it in PDF format

Template Manager

The system includes a customizable template mechanism stored in templates. Json, which allows for:

- Predefined structures for different exam types (e.g., number of questions per section).
- Difficulty levels (Easy, Medium, Hard).
- Total marks computation based on the configuration.

The templates.py script provides helper functions to load, save, add, and delete templates easily.

Question Bank

All descriptive questions are stored in a hierarchical JSON structure (question bank. Json) organized by

- Subject
- Topic
- Question Type (descriptive_3, descriptive_5)

Paper Generator

The core logic of question paper creation resides in the model.py module. It:

- Reads configuration and template data.
- Selects appropriate questions from the question bank.
- Assembles the final question paper in a structured format (Section A, B, C).
- Computes total marks based on selected questions.
- Optionally, generates a matching answer key based on stored or embedded answers.

A rate limiter is also included to prevent excessive reuse or API flooding, though external APIs are not used in this version.

History Manager

The history_manager.py component maintains a log of all generated question papers in paper history. Json. It:

- Stores metadata (subject, topic, difficulty, total marks).
- Records questions and answers.
- Allows users to retrieve, search, or delete past papers.

This feature is especially useful for version control, auditing, and tracking syllabus coverage.

PDF Generation

The question paper and answer key are formatted and converted into downloadable PDF files using the Report Lab and PyPDF2 libraries. The convert to pdf () function ensures a clean and professional layout, with sections, formatting, and indentation for readability.

3. IMPLEMENTATION

The implementation of the system focuses on building a user-centric tool that can generate well-structured, descriptive-only question papers based on predefined educational standards. The system is fully developed in Python and utilizes several open-source libraries to handle various aspects such as PDF processing, data storage, and UI rendering.

Development Environment

Programming Language: Python 3.x

Framework: Streamlit (for the front-end web interface)

Libraries Used:

- report lab for generating high-quality PDFs
- PyPDF2 for extracting text from uploaded syllabus documents
- Json for reading/writing templates and question banks
- datetime, os, typing for utility operations
- streamlit to build the interactive UI

The dependencies are managed using requirements.txt, making deployment on any environment straightforward.

User Input Handling

The main application (app.py) begins by allowing users to:

- Upload a syllabus PDF file, which is parsed using PyPDF2.
- Select or define the exam type (Mid/Final), subject, and topic.
- Configure the number of 3-mark and 5-mark questions.
- Choose a difficulty level.
- Optionally include answer keys.

These inputs are passed to the generator module for processing.

Template and Configuration Management

Templates are handled by templates.py, which ensures that paper structures are consistent. These templates define:

- Question distribution (e.g., number of 3-mark and 5-mark questions)
- Total marks and difficulty level
- Whether the paper should include answers

Templates are editable through the UI and are stored persistently in templates. Json.

Question Selection and Assembly

The model.py script is responsible for selecting and assembling questions:

- It reads the selected subject and topic from question bank. Json.
- Based on the template, it pulls the required number of questions from each descriptive category.
- If sufficient questions are not available, it notifies the user or fills from similar topics.

Once the questions are assembled, they are formatted section-wise (e.g., Section B, Section C).

Answer Key Generation

If the user opts to include an answer key, predefined or static answers (entered manually in the question bank or templates) are appended section-wise. The answer format mirrors the structure of the question paper, with clear headings for ease of evaluation.

Output and PDF Export

The final content is converted to a professional-looking PDF using the report lab library. The formatting includes:

- A header with exam details
- Clear question numbering and marks
- Indented answer sections (if enabled)
- Section titles and page breaks for neatness

Users can then download the question paper and answer key through Streamlit's download buttons.

Paper History Tracking

Every generated paper is saved to paper history. Json by history_manager.py. It logs:

- Timestamp of generation
- Subject, topic, difficulty
- Full questions and answers
- Paper ID (auto-incremented)

This allows users to view, retrieve, or delete past papers for record-keeping or reuse.

4. RESULTS AND DISCUSSION

The proposed system was tested in a controlled academic environment with simulated inputs for a variety of subjects such as Data Structures, Operating Systems, and Computer Architecture. The goal was to evaluate its performance in terms of usability, flexibility, alignment with educational goals, and the quality of generated content.

Usability and User Experience

The use of Streamlit for the front-end ensured a clean and responsive interface. Educators with minimal technical experience were able to:

- Upload syllabus PDFs.
- Select exam configurations using dropdowns and sliders.
- Generate papers within a few seconds.
- Download high-quality PDF versions of the question paper and answer key.

Feedback from users highlighted the ease of use and reduced effort compared to traditional manual paper-setting workflows.

Quality of Generated Papers

Generated papers were evaluated based on:

- Cognitive balance: Questions covered a range of Bloom’s levels such as Understanding, Application, and Analysis.
- Syllabus alignment: Questions matched the uploaded syllabus content when topics were properly tagged in the question bank.
- Presentation: PDFs were professional, cleanly formatted, and ready for direct use in academic assessments.

Each section (short and long answer) was clearly organized and numbered, ensuring clarity for both students and evaluators.

Flexibility

Thanks to the template-based structure, the system allowed:

- Customization of exam types (e.g., midterm with fewer marks, final with extended questions).
- Variation in question patterns without modifying core code.
- Dynamic reusability of previous templates and history logs.

Educators appreciated the ability to preconfigure templates once and reuse them across different semesters or courses.

Limitation

While the system performed reliably, a few limitations were noted:

- It depends on the manual population of the question bank. Currently, there is no automatic import of questions from textbooks or lecture notes.
- Syllabus parsing from PDFs is basic and does not include semantic understanding.
- There is no tagging automation for Bloom's levels questions must be pre-classified appropriately by the user.
- The system does not yet support images, diagrams, or equations within questions or answers.

Comparative Discussion

Compared to existing commercial solutions, this system is:

- Lightweight and open source.
- Free from vendor lock-in or subscription fees.
- Tailored for descriptive question paper generation, which is often overlooked by tools focusing on MCQs.

Its architecture supports institutional adoption where academic autonomy and outcome-based education models are prioritized.

Future Scope

The current implementation provides a robust foundation for automated descriptive question paper generation; however, there are several avenues for future enhancement and broader adoption. The following directions outline the future potential of the system:

Institutional Integration

The system can be scaled and integrated into university portals or Learning Management Systems (LMS) to automate assessment workflows department-wide, with support for multiple instructors and role-based access.

Question Bank Expansion

A community-driven or institution-wide collaborative platform could be created to allow educators to contribute

and curate question banks, fostering a rich, shareable academic resource.

Natural Language Support (Optional)

While current tagging is manual, future versions could explore optional AI-assisted suggestions to classify Bloom's levels or identify topics from uploaded content to enhance speed and flexibility.

Interactive Paper Review and Feedback

Future versions can include features like previewing papers with inline editing, question reordering, and adding reviewer comments before finalizing.

Multimedia and LaTeX Support

Integration of support for mathematical formulas (LaTeX), diagrams, and inline images can greatly benefit STEM subjects and enrich the generated content.

Mobile and Offline Versions

A mobile-responsive version or downloadable desktop variant (via tools like PyInstaller or Electron) can make the system more accessible to faculty with limited connectivity.

Advanced Analytics

Future updates can include detailed analytics dashboards showing Bloom level distribution across generated papers, frequently used questions, topic heatmaps, and syllabus coverage metrics.

5. CONCLUSION

This paper presented an intelligent, customizable system for automated descriptive question paper generation aligned with Bloom's Taxonomy and Course Outcomes. Unlike AI-driven or ML-based approaches, the system is built on a practical foundation of manually tagged question banks, structured templates, and a simple, educator- friendly web interface using Streamlit. The tool enables faculty members to generate balanced, syllabus-aligned question papers within minutes, significantly reducing the time and effort traditionally required.

The project successfully addresses key educational goals including cognitive distribution, syllabus mapping, question paper structuring, and outcome-based assessment. Features such as syllabus upload, template application, Bloom's level distribution, answer key inclusion, and paper history tracking have contributed to a system that is both functional and adaptable in real academic scenarios

REFERENCES

[1] Anderson, L. W., & Krathwohl, D. R. (2001). *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. Allyn & Bacon.

[2] Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., & Krathwohl, D. R. (1956). *Taxonomy of Educational Objectives: The Classification of Educational Goals. Handbook* I: Cognitive Domain. Longmans, Green and Co.

[3] Bhardwaj, A., & Pal, R. (2012). *An Automated System for Generation of Randomized Question Papers. International Journal of Computer Applications*, 53(3), 23-27.

[4] Mishra, S., & Chandra, A. (2017). *Classification of Exam Questions Using Bloom's Taxonomy and Word Embeddings*. International Journal of Advanced Research in Computer Science, 8(8), 225-230.

[5] Kumar, A., Gupta, S., & Singh, P. (2020). *AI-Driven Question Paper Generator Using Bloom's Taxonomy for Balanced* Assessment. International Journal of Educational Technology in Higher Education,17(1), 1-14.

[6] Sharma, M., Sharma, S., & Kapoor, K. (2018). *Challenges in Generating Automated Question Papers for Educational Examinations*. In 2018 International Conference on Computing, Power, and Communication Technologies (GUCON), 467-472.

[7] Singh, N., Agarwal, S., & Singh, A. (2015). *A System for Automated Generation of Question Papers Using Bloom's Taxonomy and Topic Distribution*. International Journal of Education and Development using ICT, 11(1), 88-98.