IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Question Paper Generation Using Bloom's Taxonomy

1Prof. Kanade P. G., 2Jayesh Gudaghe, 3Sanjasvi Sonawane,4Mahek Shaikh, 5Sarthak Ghumare

Department of Computer Engineering
S.N.D. College of Engineering & RC, Yeola, Nashik, 423401.

Abstract— Hand-writing question papers is commonly time-consuming 1and inconsistent, particularly while attempting to match academic levels such as Bloom's and Course Outcomes (COs). The project proposes an intelligent and accessible system that fully paper generation in a user-friendly manner, wherein teachers can import syllabus PDFs, establish custom settings, and automatically produce descriptive-only papers and answer keys. The system is cognitive-balanced, allows customizable templates, and uses natural language processing to enable appropriate question selection. Efficient in its design and outcome-based assessment, facilitates streamlined assessment processes within academic environments.

Keywords: Automatic Question Generation, Bloom's Taxonomy, Course Outcomes, Streamlit, Descriptive Questions, Educational Technology

I. INTRODUCTION

Evaluation is the establishment of instruction and, in expansion to measuring learning results among understudies, it too makes a difference shape the educational programs system, instruction, and instruction guidelines. Composed address paper remains one of the foremost ordinary and however inescapable assessment shapes. Whereas basic to learning, making address papers by hand proceeds to be a lumbering and labor-intensive errand for instuctors. It regularly comprises of guaranteeing regulation are in line, remaining steady in level of trouble, and achieving adjusted scope of the syllabus all with the point to play down excess and individual bias.

Over the past few a long time, instructive teach have progressively begun executing outcome-based instruction (OBE) structures, where consideration is on assembly clearly characterized learning results. In this worldview, Bloom's Scientific classification has been a commonly accepted taxonomy to arrange and survey learning exercises. It may be a orderly approach to degree understudies at diverse cognitive levels â€ ” from reviewing and comprehending to analyzing and making. At the same time, instructive teach too set up Course Results (COs) that are particular learning targets relating to each subject or module. Mapping address papers to both Bloom's levels and COs ensures that exams are not only challenging but moreover adjusted with scholarly objectives.

The paper presents a brilliant, tunable system for computerized descriptive-only address paper era, outlined to play down manual work and make strides appraisal plan quality. Instructors can input the syllabus in PDF and set parameters like exam sort (Mid/Final), subject title, theme, stamp allotment, and trouble level. The framework at that point applies predefined formats to deliver designed address papers and related reply keys based on these inputs. Interests, the instrument does not incorporate multiple-choice questions and goes as it were for graphic sorts, viz., brief and long reply types.

The program is built utilizing Streamlit, a Python web system that can be utilized for fast advancement of web applications. It offers an easy-to-use interface for teaches without requiring them to have any specialized information. Behind the scenes, the program incorporates a energetic address bank and a database for produced papers as well as permits downloading in proficient PDF output.

In all, this venture endeavors to upgrade the consistency, unwavering quality, and adequacy of address paper planning, especially in teach embracing outcome-based evaluation approaches. Through its capacity to computerize day-to-day scholarly forms, it empowers instructors to give more time to substance and educational upgrade.

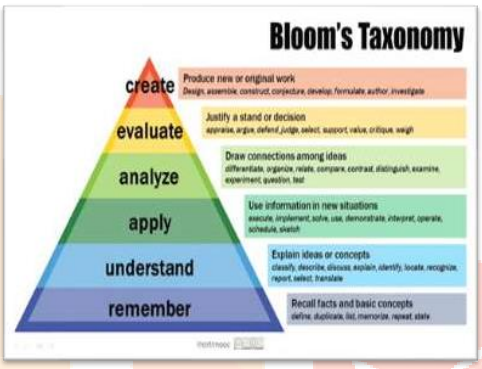

Fig [1] blooms Taxonomy levels

II. Related Work

Automation of question paper preparation has been increasingly in focus in recent times, especially as a means of mitigating redundant academic work and enhancing alignment with learning goals. Conventional systems are heavily dependent on manual efforts by instructors to prepare questions, balance question difficulty levels, and achieve full syllabus coverage. Not only is this time-consuming but also prone to inconsistencies among different instructors and institutions.

Early computerized systems were used mostly as warehouses of static bank of questions. Such systems permitted instructors to pick and choose from listed questions under given categories such as subject or chapter. This system made the question retrieval much more efficient but still needed extensive human intervention in bringing together an equitably balanced paper. Also, most such systems did not have a proper framework or model to assist cognitive-level distribution or outcome mapping.

To overcome these constraints, certain educational tools incorporated randomization and rule-based selection. In such models, questions may be labeled with metadata like marks, difficulty level, topic, and type (MCQ, short answer, long answer) to enable semi-automated generation of question sets based on filters. Yet, these systems were not intelligent template-based or did not have explicit support for education frameworks such as Bloom's Taxonomy, which categorizes learning objectives at hierarchical cognitive levels (e.g., Remember, Understand, Apply, Analyze, Evaluate, Create).

Moreover, one of the major drawbacks in most commercial and institutional software is their excessive dependency on objective-type questions—notably multiple-choice questions (MCQs)— because of ease of marking. Although this may be appropriate for large-scale evaluations, it inhibits depth and breadth of cognition measurement. Descriptive evaluations, however, are necessary to gauge higher-order thought and analytical capability. However, hardly any systems are exclusively for the systematic preparation of descriptive-only question papers.

There are also initiatives in which tools combine question paper creation with student performance analysis or learning management systems (LMS). Although these provide larger assessment ecosystems, these are generally not open source and tend to have high learning curvatures or subscription charges. They may also lack fine-tuned customization or integration with course-specific outcomes (COs), which becomes highly relevant in institutions that deploy Outcome-Based Education (OBE).

Unlike these earlier methods, the system presented in this paper is a template-based, lightweight application that is dedicated solely to the generation of descriptive question papers. It is based on pre-defined templates that correspond to question types, as well as Bloom's cognitive levels so that teachers can select exam types (e.g., Mid or Final), select the number of 3-mark and 5-mark questions, specify difficulty, and upload syllabus documents. Questions are then picked from a well-organized question bank specific to the topic and subject matter for systematic and outcome-based paper development. Additionall, the suggested system prioritizes ease of use by utilizing a clean interface with Streamlit, a web framework based on Python. It makes no use of machine learning, AI, or external APIs, thus being light, quick, and deployable in any educational setting without requiring specialized technical setup.

III. PROPOSED SYSTEM

Workflow of the Proposed System

User Login and Upload of Document:
The client logs into the framework and transfers a address bank document-for case, a syllabus or address bank record. The framework will extricate questions from the archive and handle them.

Question Classification:
It forms the extricated questions utilizing the NLP demonstrate and categorizes them concurring to Bloom's Scientific classification levels. All such questions are kept within the database in conjunction with their cognate level and difficulty.

Question Paper Pattern:
The client arranges parameters for the address paper, counting add up to marks, Bloomâ€ ™s level dissemination, and address trouble. Parameters are thus utilized by the framework whereas selecting questions from the classified address bank.

Question Paper Generation:
The framework produces a address paper based on the questions chosen and parameters set by the utilize. The questions are organized with a organized system concurring to their cognitive level and difficulty.

Bloom’s Scientific classification Report Generation:
A report is produced, showing the recurrence of questions over the cognitive levels of Bloom's Scientific categorization. This gives the report that makes a difference evaluate the cognitive adjust of the address paper. Survey, Reformulate, and Download Clients can survey the created address paper and make any alterations required. Upon finalization, the address paper and report of Bloom's Scientific categorization can be downloaded in different groups.

IV. SYSTEM ARCHITECTURE

The proposed framework has been planned with straightforwardness, measured quality, and educator-friendliness. It includes a efficient but light-weighted engineering created comprehensively utilizing Python and the Streamlit system as the client interface. The framework points to consequently produce clear address papers and at the same time guarantee arrangement with Bloom's Scientific classification and Course Results (COs). The engineering can be broken down into the taking after vital components:

A. User Interface (Streamlit)

The front-end is built with Streamlit, which is an intuitively and instinctive stage that gives teachers an easy-to-use interface. It empowers clients to:

- Up Upload a PDF record for the syllabus document.
- Choose the sort of exam (e.g., Mid Exam, Last Exam).
- Set parameters like subject, theme, trouble level, number of 3-mark and 5-mark questions, and whether an reply key ought to be included.
- Display the produced address paper on screen and download it in PDF format.

B. Format Manager

The framework contains a layout customizable component spared in templates.json to support:

- Predefined groups for different exam sorts (e.g., number of questions per section).
- Difficulty levels (Simple, Medium, Hard).
- Total marks calculation based on the setup.
- The templates.py module offers utility capacities to stack, spare, add, and expel layouts in an easy manner.

C. Question Bank

All descriptive questions are kept in a hierarchical JSON format (question_bank.json) organized by:

Subject → Topic → Question Type (descriptive_3, descriptive_5)

Teachers can populate this question bank manually to meet their syllabus and cognitive level distribution. The system retrieves questions based on the chosen subject, topic, and question type.

D. Paper Generator

The central rationale of address paper era lies within the model.py module. It:

- Reads config and layout data.
- Chooses appropriate questions from the address bank.
- Constructs the ultimate address paper in a organized arrange (Section A, B, C).
- Calculates add up to marks from chosen questions.
- Optionally, makes a coordinating reply key from put away or inserted answers.
-

A rate limiter is additionally given to dodge over-reuse or API flooding, in spite of the fact that outside APIs are not utilized in this release.

E. History Manager

The history_manager.py module keeps a record of all address papers produced in paper_history.json. It:

- Saves metadata (subject, theme, trouble, add up to marks).
- Saves questions and answers.
- Enables clients to download, look, or expel past papers.

This usefulness is especially supportive for form control, reviewing, and syllabus scope tracking.

F. PDF Generation
The address paper and reply key are styled and rendered into downloadable PDF records through the ReportLab and PyPDF2 libraries. The change over to pdf() work gives a clean and professional-looking format, with segments, designing, and space for readability.

G. File Structure Overview
Here's a high-level outline of the project's measured structure:

```
├── app.py              # Streamlit UI logic
├── model.py            # Question generation logic
├── templates.py        # Template handling
├── history_manager.py  # Paper history management
├── styles.py           # Local CSS custom styling
├── templates.json      # Standard exam templates
├── question_bank.json  # Organized question store
├── paper_history.json  # History of created papers
├── requirements.txt    # Python package requirements
```

V. IMPLEMENTATION

Implementation of the framework rotates around creating an end-user application that can consequently make well-formed, description-only address papers against pre-defined instruction guidelines. The framework is built totally utilizing Python and employments a few open-source libraries to perform numerous perspectives such as PDF taking care of, information capacity, and UI rendering.

A. Advancement Environment
Programming Dialect: Python 3.x
Framework: Streamlit (for the front-end web interface)
Libraries Used:
pandas==2.2.3: For information control and analysis.
plotly==5.14.1: For making intuitively visualizations (e.g., pie charts).
protobuf==3.20.3: Required for compatibility with a few libraries such as Streamlit.
PyPDF2==3.0.1: For PDF record perusing and extraction of text.
reportlab==3.6.6: For composing PDF documents.
streamlit==1.43.1: For making the web-based frontend.
matplotlib==3.10.1: For chart and visualization creation.
The necessities are dealt with by requirements.txt so that sending on any environment is easy.

B. Taking care of Client Input
The center application (app.py) begins with allowing clients to:

- Upload a PDF record of a syllabus, which is examined by PyPDF2.
- Choose or indicate the exam sort (Mid/Final), subject, and topic.
- Set the number of 5-mark, 6-mark, 7-marks, 8-marks, questions.
- Select a trouble level.
- Optionally include reply keys.
- These inputs are passed to the generator module for processing.

C. Format and Setup Management
Templates are overseen by templates.py, which implements paper structures to be uniform. These formats specify:

- Question dispersion (e.g., number of 5-mark, 6-mark, 7-marks, 8-marks, questions)
- Total marks and trouble level Whether the paper ought to have answers
- Templates are editable within the UI and are endured in "templates.json"

D. Address Determination and Assembly
The model.py script is utilized for selecting and gathering questions:

- It peruses the chosen point and subject from "question_bank.json"
- Depending on the format, it recovers the fundamental number of questions from each clear category.
- If there are not sufficient questions, it illuminates the client or fills from related topics.
- After the questions are compiled, they are organized section-wise (e.g., Area B, Segment C).

E. Era of Reply Key
If the client chooses to supply an reply key, pre-defined or inactive answers (written in physically within the address bank or formats) are included section-wise. The reply arrange reproduces the organize of the address paper, with clear headings for simple marking.

F. Yield and PDF Export
The last substance is at that point rendered into a professional-looking PDF utilizing the reportlab library. Designing includes:

- A header with exam information
- Clean address numbering and marks
- Indented reply areas (in the event that enabled)
- Section headers and page breaks for cleanliness
- Users can at that point download the address paper and reply key by means of Streamlit's download buttons.

G. Paper History Tracking
Each produced paper is put away to paper_history.json by history_manager.py. It records

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

This makes it conceivable for clients to show, recover, or eradicate prior papers for reference or reuse.

VI. RESULT AND DISCUSSION

i. CONCLUSION

The "Question Paper Generator using Bloom's Taxonomy" promises much when further developed for the improvement of education assessment. This may open the Scope up to personal learning adaptive difficulty, support for multiple languages, and integration into the existing educational system by improving how assessments are designed and presented in different educational contexts. The system will probably evolve with on going advancements in AI and machine learning, aiming at delivering increasingly more complex, data-intensive solutions that make education both more effective and accessible to both educators and learners.

ii. FUTURE SCOPE

The Question Paper Generator through Bloom's Taxonomy does have a great scope for improvement and further development, particularly as regards functionality and usability. Along with changes in the way education functions, the tool could be developed to fulfill various pedagogical needs and technological enhancements that may emerge in the future. Some possible areas of improvements and expansions lie as follows:

- Advanced Question Classification
- Adaptive Learning and Question Customization
- Integration with any Learning Management System

- Support for Other Taxonomies and Frameworks
- Multi-Language Support
- Real-Time Collaborative Functionality
- Machine Learning for Question Difficulty Prediction
- Cloud-Based Collaboration and Sharing
- AI-Driven Feedback and Recommendations
- Integration Educational Analytics

VII. REFERNCES

[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.