WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

AI Framework for Automated and Balanced Examination Question Paper Generation System

B. Jyothsna[1], G. Tulasi[2], Ch. Roshini[3]

[1] (23341A0519) Dept. of Computer Science and Engineering GMRIT Deemed To Be University jyothsnabangaru4@gmail.com
[2] (24345A0508) Dept. of Computer Science and Engineering GMRIT Deemed To Be University
[3] (23341A0543) Dept. of Computer Science and Engineering GMRIT Deemed To Be University

ABSTRACT:

Question paper preparation is a time-consuming task in educational institutions, requiring balanced syllabus coverage, appropriate difficulty levels, and alignment with course outcomes and bloom's taxonomy. Existing automated systems mainly rely on manually prepared question banks or simple AI-based question generation, which often require human intervention to create a balanced examination paper. This paper proposes an AI-Framework for Automated and Balanced Examination Question Paper Generation System, which generates optimized question papers directly from course syllabus based on a predefined examination blueprint. The faculty can submit the syllabus of their respective courses in the form of a PDF file, and the text content of the PDF file is extracted and processed to extract important topics and concepts. Using RAG and LLMs, relevant examination questions are generated based on the uploaded file. The optimization process then selects and organizes the questions based on certain criteria, such as weightage, marks distribution, course objectives, Bloom's taxonomy, difficulty level, and type of question. The generated question paper can be exported in PDF format for examination purposes. Overall, the proposed system reduces the time and effort involved in question paper preparation, improves consistency in assessment, and provides an efficient, scalable, and intelligent solution for automated examination paper generation.

Keywords: Automated Question Paper Generation, Retrieval-Augmented Generation (RAG), Large Language Models (LLMs), Bloom's Taxonomy, Course Outcomes, Constraint Optimization, Document Processing.

1. INTRODUCTION

Question paper preparation is an important academic task because an examination should represent the prescribed syllabus, learning outcomes, and intended cognitive skills. In a manual process, faculty members must read course materials, identify important topics, prepare questions, allocate marks, select difficulty levels, and check whether the paper covers all required units. It is both time-consuming and may result in either inconsistent coverage of topics, redundant information, or inconsistencies with COs. AQG and the current generation of LLMs have means of producing questions directly out of learning materials. RAG may help to further improve the grounding by retrieving relevant excerpts from the uploaded documents before generation. However, an efficient institutional solution should do more than just create individual questions; it should also build a full-fledged exam paper following the required blueprint, validate marks and weights distribution, balance levels and difficulty in accordance with Bloom's Taxonomy, avoid duplication, and provide means of review and approval.

The suggested project combines document processing, RAG, LLM-based question drafting, outcome mapping, and multi-constraint optimization into one pipeline. Teachers upload syllabus documents and define the examination pattern. The system creates a draft, validates it on the basis of the required constraints, and makes it available for teachers' review and export in PDF format. The suggested architecture also includes role-based features for academic configuration, approval, archival management, and evaluation records.

2. PROBLEM STATEMENT AND OBJECTIVES

The fundamental issue is that there is no single and documentation-backed approach that allows for creating full-fledged exam papers while also meeting the institutional demands. In most cases, existing approaches involve the creation of questions, MCQs, or rule-based question selection. The goal of this project is to automatethe entire process while retaining the faculty's ability to validate the generated paper.

- Create an AI-based examination question paper generator based on the course documents and syllabus.
- Extract and structure the textual data from the uploaded PDF/DOCX files into units, topics, and segments.
- Obtain syllabus-related segments via RAG approach and create questions through an LLM.
- Create papers according to a given blueprint including sections, marks, internal choice, and question type.
- Provide the balance between unit-level weightage, Course Outcomes, Bloom's Taxonomy, and difficulty level.
- Avoid the duplication and semantic repetition of questions in the paper.
- Facilitate faculty validation, HOD validation, versioning, reports, and printing as a PDF file.

3. REVIEW OF EXISTING RESEARCH

Works on automatic question generation include rule-based selection, NLP approaches, language model approaches, retrieval-augmented generation (RAG), and question paper composition. The following review provides valuable insights that may be used to develop a system capable of generating questions out of the course content while checking the generated paper according to academic criteria.

3.1 RAG and LLM approaches

Pradeesh et al. [1] applied the RAG approach for the generation of multiple-choice questions and answers based on educational documents. This study demonstrates the usefulness of document retrieval before creating the question, even though it does not form a full paper that satisfies all institutional criteria. Kumar et al. [2] integrate syllabus processing, Gemini, Bloom's Taxonomy, and rule-based balancing. Chaube et al. [5] provide information about the PaperPilot approach that applies curriculum documents and
configurable blueprint. These works are valuable starting points, but checking is needed for coverage, marks, repetition, and faculty-related criteria.

3.2 Survey and Foundations of Question Generation

Kurdi et al. [3] survey automatic question generation in education including methods, knowledge bases, and evaluation approaches. Mulla and Gharpure [11] discuss datasets and metrics used in AQG research, while Anwar et al. [7] describe the move from rule-based MCQ generation toward machine-learning and LLM approaches. Mitroulias and Sioutas [4] examine research trends in automated MCQ generation. Together, these reviews show that question quality and evaluation remain important concerns. They also show that a review of generation methods is different from a complete workflow for preparing and approving an examination paper.

3.3 Learning outcomes, question selection, and classification

Historical-examination-data-based methods like those of Kumar et al. [6] and Praveen et al. [10] involve the use of past questions and language models for the sake of creation, classification, and paper assembly. Narayana et al. [12] talk about NLP-based automatic generation of question papers, and Bhirangi and Bhoir [14] present a question paper generator based on stored questions and randomization. Though the former provide automation support, the latter may not necessarily have understanding of new syllabus contents and also cannot guarantee the simultaneous satisfaction of all constraints of the blueprint.

3.4 Question-level generation and MCQ quality

Mishra et al. [13] investigate question generation from long answers and mention the challenge of dealing with longer contexts. Tran Vu and Blake [16] investigate question formation for English learners. Zhang et al. [8] consider the quality of MCQ distractors, and Ling and Afzaal [9] generate question-answer pairs for higher education. Zhao et al. [17] employ multiple agents for MCQ generation and validation. Sairaj and Balasundaram [15] apply RAG for reducing the repetition of question topics. All of the above research works touch upon some useful aspects of the problem which is intended to be addressed in this paper together with syllabus, marks, Course Outcomes, difficulty level, reviewing, and final paper generation.

3.5 Related work summary

It can be observed that the above studies employ various data sets and evaluation measures. In case no number is reported for the accuracy of a particular paper, it is mentioned explicitly in the table. Preparation time reduction, grammatical correctness, and duplication removal are considered as separate results and must not be considered as accuracy values.

4. RESEARCH GAP

There have been advancements in question generation, NLP, RAG, LLMs, Bloom's Taxonomy, and question paper preparation but these abilities are handled independently. Some approaches generate only MCQs or single questions.

Some depend on pre-prepared question banks or use randomization without full semantic understanding of newly uploaded course documents. Syllabus coverage by unit and the mapping of questions to COs may still require manual effort.

The lack of multi-constraint optimization approach is another research gap. Factors like marks, question type, unit weightage, difficulty, Bloom's level, internal choices, and concept repetition need to be optimized together and not individually. There is also no end-to-end solution for question paper preparation that covers both generation and subsequent processes like faculty correction, HOD approval, versioning, security, and PDF export. These gaps lead to a document-based system that brings generation, academic suitability, optimization, validation, and institution-specific workflow.

5. PROPOSED SYSTEM AND METHODOLOGY

The proposed system takes course documents and exam requirements as inputs, extracts relevant syllabus content, generates questions using an LLM, validates each generated question, and prepares a paper using blueprint constraints. In case of any unmet constraint during validation process, the

system will either select new questions or modify generation prompts before showing the generated paper to faculty members for review.

5.1 Academic Data Input and Document processing

The faculty members will submit syllabus/course documents in PDF or DOCX format and give the exam requirements including course name, semester, total marks, duration, section pattern, weightage of units, internal choices, question types, and difficulty distribution. PyMuPDF will be used for PDF text extraction. The extracted text will be cleaned and divided into units, modules, topics, and sub-topics for retrieval purpose. The extracted units and learning outcomes will be stored as structured academic data and not in single text block.

5.2 RAG retrieval and syllabus knowledge base

After the document is processed, the relevant syllabus content will be indexed semantically. In the process of generating a question, the system will retrieve the most relevant syllabus text or topic chunks and provide these texts as contextual input to the LLM. Top-K retrieval will help in narrowing down the scope of generation prompt to appropriate concepts of course. The retrieved texts should be sufficiently relevant to the desired unit, learning outcomes, question type, and marks. Retrieval and generation are connected processes which enable validating question statements with academic documents.

5.3 LLM-based question generation

The Google Gemini 1.5 Flash will be chosen as the primary LLM for syllabus-based question generation and structured output generation. The prompts will include desired target topic, retrieved context, question type, marks, difficulty, and Bloom's level. The generator will generate short answer, long answer, descriptive and MCQ questions as per blueprint requirement. Generated question will be seen as a candidate question and not as an automatically validated question.

5.4 Course Outome and Bloom's Taxonomy alignment

Each question is mapped with the corresponding syllabus topic and Course Outcome. The institutional CO definitions and levels in Bloom's taxonomy serve as structured reference inputs. Bloom's taxonomy contains levels like Remember and Understand to Apply, Analyze, Evaluate, and Create. The expected cognitive level may be identified using the action verb and prompt, and validated later on. The CO mapping and Bloom taxonomy assignment must be reviewed when uncertain about the classification.

5.5 Blueprint and multi-constraint optimization

The constraint-based selection algorithm checks the generated pool against the required paper blueprint. The constraints involved in the process include total marks, marks per sections, unit weightage, number and types of questions, internal choice policy, coverage of COs, Bloom's distribution, question difficulty distribution, and semantic uniqueness. The algorithm selects and arranges the questions in such a manner that the resulting paper adheres to the mandatory rules and stays as close as possible to target distribution. Concept-level deduplication is required as two questions can be phrased differently but test the same concept.

5.6 Generated paper validation and feedback

The generated draft is validated against the paper blueprint and context from the source database. The validation procedure includes checking marks, sectioning, unit-wise coverage, number and type of questions, Bloom's distribution, topic repetitions, and evidence for relevance to syllabus. In case of violation of any constraint, the workflow revises the selected questions and/or tunes the prompt and goes for another validation. This is done to reduce the chance of avoidable errors before human verification.

5.7 Human faculty review, HOD approval, and Output

Human faculties check and edit the generated paper. The HOD can review and approve the generated paper according to the process used in the academia. Once the paper is approved, a standardized print-ready PDF is generated and the paper is stored in the PostgreSQL along with its versioning information and other relevant records. Some modules allow the storage of student evaluation results and preparation of assessment reports for academic analysis. The AI generation facilitates the process; academic responsibility rests with the authorized faculty and approver.

6. SYSTEM ARCHITECTURE AND TECHNICAL STACK

The architecture has four parts – interactive faculty dashboard, backend, AI-based processing, and persistence layer. React and Vite take care of the frontend dashboard. Python Flask framework provides the backend routing and manages call to generation and retrieval algorithms. Flask-SQLAlchemy links application object to the PostgreSQL. PyJWT and Werkzeug password hashing support authentication and role-based access control.

5.8 End-to-end workflow

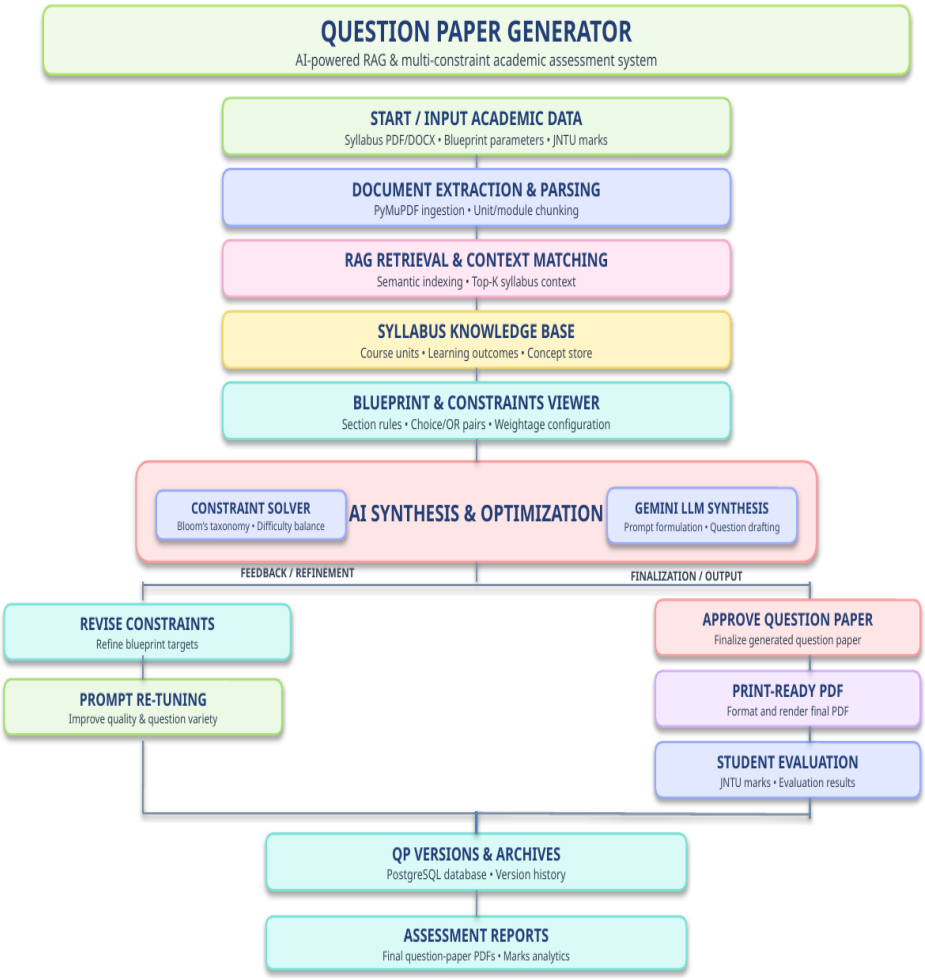


Fig. 1. End-to-end workflow of the proposed AI-based question paper generation system.

Stage	Input / operation	Output
Academic input	Syllabus document, course outcomes, blueprint and institutional rules	Configured generation request
Extraction	PDF/DOCX parsing, text cleaning, unit/topic chunking	Structured syllabus knowledge base
RAG retrieval	Semantic indexing and Top-K context retrieval	Relevant syllabus passages
Question drafting	Gemini prompt with context, type, marks, difficulty and Bloom level	Candidate question set
Optimization	Check marks, units, COs, Bloom levels, difficulty and repetition	Balanced paper draft
Review and output	Faculty editing, HOD approval and PDF generation	Approved PDF and archived version

Table 1. Workflow stages and expected outputs.

Component / technology	Role in the proposed system
React 18 + Vite	Responsive faculty, administrator, and HOD interfaces.
Python + Flask	Backend APIs, validation, workflow coordination, and AI service integration.
PyMuPDF	Extracts text from uploaded syllabus PDFs for processing.
RAG pipeline	Retrieves relevant course passages before the LLM drafts questions.
Google Gemini 1.5 Flash	Generates syllabus-grounded questions and structured responses.
Constraint/optimization engine	Checks paper structure, marks, COs, Bloom levels, difficulty, and repetition.

Component / technology	Role in the proposed system
PostgreSQL + Flask-SQLAlchemy	Stores users, courses, CO mappings, papers, versions, archives, and evaluation records.
PDF/report modules	Exports approved question papers and displays relevant assessment summaries.

Table 2. Main components and their roles in the proposed system.

6.1 Role-based academic workflow

The Administrator maintains users and roles, programs, branches, regulations, courses, curriculum mappings, and system-level question settings such as Bloom's levels. The faculty assigns the courses, uploads syllabus files, configures blueprints, generates the paper, reviews and edits the paper, exports the finalized paper, and enters evaluation information if supported. The HOD reviews the generated paper and approves the final version. Role-based access ensures that each user can access only the functions assigned to that role.

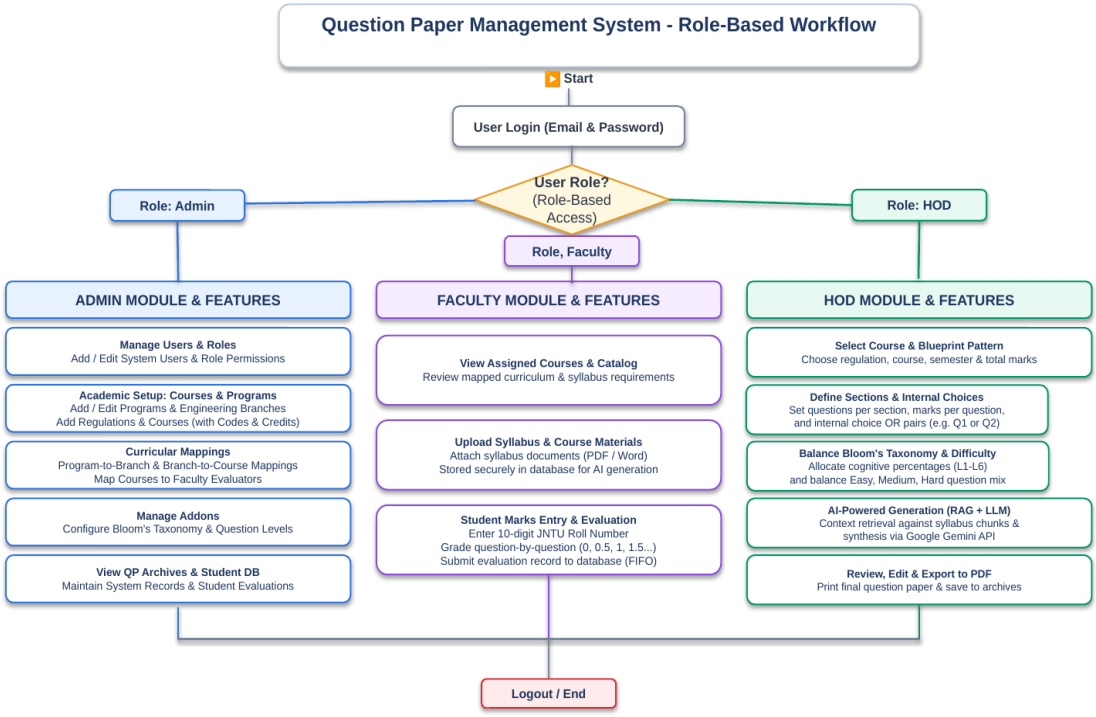


Fig. 2. Role-based workflow for Administrator, Faculty and HOD

7. DATASETS AND INFORMATION SOURCES USED

The proposed framework uses academic documents and institutional configuration records rather than relying only on one manual and one set of fixed questions. The sources of information are provided below. Previous years' examination papers may be used as per requirements for pattern recognition or any other use, but the main source of information is going to be the syllabus provided by the faculty. The actual coverage of information depends on the documents entered and the blueprint setting.

Information source	Contents	Purpose
Curriculum documents (PDF/DOCX)	Syllabus, units, topics, definitions, course material and learning objectives.	Provide source context for retrieval and generation; support unit-level coverage.
Relational academic schema	Courses, programs, syllabus topics, CO1–CO6 where applicable, Bloom levels, action verbs and mappings.	Represent the institutional curriculum and support CO/Bloom alignment.
Examination blueprint	Sections, total marks, marks per question, unit weightage, question types, difficulty targets, choice/OR pairs and duration.	Define the hard and target constraints used to assemble the paper.
Generated questions and paper documents	Question text, topic/source references, grades/level, etc.	Help with duplicate detection, review by staff, archive.

Table 3. Sources of academic information used by the framework

8. EVALUATION PLAN AND EXPECTED RESULTS

Evaluation for the project is done component-by-component and not by an overall grade. Because the evaluation framework is being introduced, the below listed evaluation criteria will be applied during testing of the framework. They are not measurements as yet experimental results. These outcomes must be validated using representative courses and faculty feedback. The framework should not be described as achieving a particular accuracy or time-saving percentage until controlled experiments or a documented evaluation have been completed.

8.1 Performance Metrics

The following metrics will be used during testing. They describe the planned evaluation; they are not results already achieved by the proposed system. These outcomes must be validated using representative courses and faculty feedback. The framework should not be described as achieving a particular accuracy or time-saving percentage until controlled experiments or a documented evaluation have been completed.

8.2 Expected Outcomes

- A structured examination paper generated from syllabus content and a configurable examination blueprint.
- Improved visibility of unit-wise coverage, marks distribution, Course Outcomes, Bloom's levels, and difficulty mix.
- Fewer duplicated or conceptually repetitive questions through semantic checks and validation.
- Reduced manual effort in drafting, organizing, checking, and formatting papers.
- Traceable faculty review, HOD approval, PDF export, version history, and academic records.

9. RESULT AND DISCUSSIONS

The reviewed studies show progress in syllabus-based question generation, RAG, language-model generation, question validation and automated paper assembly. Reported results differ by task, so values from one paper should not be treated as the accuracy of this proposed system. The table above lists the methodologies and findings presented in the reviewed information; in case of absence of numerical accuracy value, it is indicated as such

Component	Metric(s)	What is evaluated
Question classification	Accuracy, Precision, Recall, F1-score	Checks Bloom's Taxonomy and question-label assignments against faculty-verified labels.
Syllabus grounding	Context Precision, Context Recall, Faithfulness	Checks whether retrieved passages are relevant and generated questions are supported by the uploaded material.
Course Outcome mapping	Accuracy, Precision, Recall, F1-score; faculty review	Checks whether each question matches its intended Course Outcome.
Blueprint compliance	Constraint Satisfaction Rate, mark deviation, section-rule pass rate	Checks total marks, section structure, unit-wise weightage, question count and internal-choice rules.
Duplicate detection	Duplicate rate, semantic similarity, deduplication rate	Checks repeated questions and questions that test the same concept using different wording.
Operational performance	Generation time, validation failures, user feedback	Measures workflow speed, reliability and ease of faculty review/export.

Table 4. Planned evaluation metrics for the proposed system.

10. LIMITATIONS OF EXISTING STUDIES

- Manual and time-consuming preparation: question selection, balancing, and final formatting still require substantial faculty effort in many workflows.
- Question-level focus: some systems generate individual questions or MCQs without assembling a full examination paper.
- Static question bank dependence: questions defined earlier may not account for newly added course material.
- Limited balance optimization: unit weightage, marks, question types, level of difficulty, Bloom's taxonomy level, and internal choices cannot be optimized simultaneously.
- Partial result mapping: CO mapping and higher levels of cognition require some labeling and correction manually.
- Semantic redundancy: new wording may conceal repetition of concepts not recognized by simple string matching.
- Need for expert review: AI-generated text can be inaccurate, ambiguous, unsupported, or inappropriate for the intended marks and learning outcome.

11. HOW THE PROPOSED PROJECT ADDRESSES THE GAP

The system performs processing of uploaded documents and extraction of evidence specific to syllabuses prior to question generation instead of depending entirely on a predefined question bank. It utilizes an LLM to generate various question types and then performs checks using a blueprint for unit-wise weighting, marking, COs, Bloom's Taxonomy, difficulty, sections and topic repetition. The feedback loop will enable a constraint unmet to cause either a revision of the selection or prompt refinement. Finally, the generated paper is reviewed by faculty members and is then approved via the HOD workflow before being exported to a print-ready PDF format and stored with the version.

The key difference is that of integration of retrieval, generation, academic mapping, optimization against constraints, validation and role-based approval into one workflow. This framework intends to help faculty members in ensuring consistency while maintaining the human authority in the process of official examination paper generation.

12. CONCLUSION AND FUTURE SCOPE

This paper introduced an AI Framework for Automated and Balanced Examination Question Paper Generation System, which integrates PDF/DOCX processing, Retrieval-Augmented Generation (RAG), question generation using LLMs, COs & Bloom's Taxonomy mapping, constraint-based optimization, semantic deduplication, review, approval and PDF output. This framework attempts to go further from question generation by generating a full-fledged paper adhering to institution's examination policies. The quality of this framework depends on the relevance of retrieved source content, accuracy of blueprint, reliability of the model and the efficiency of validation and human review.

Future works involve evaluation of the framework on other subjects and engineering departments; improvements in handling mathematical expressions, derivations, diagrams, circuits and code-based questions, calibrating the difficulty based on the feedback provided by faculty members; exploring other approaches for embeddings and language models, and integration with institutional learning management systems..

REFERENCES

[1] N. Pradeesh, T. Remya, M. G. Thushara, K. Arun Krishna, and V. Pranav, "Retrieval-Augmented Generation for Multiple-Choice Questions and Answers Generation," Procedia Computer Science, vol. 259, pp. 504–511, 2025, doi: 10.1016/j.procs.2025.03.352.

[2] K. P. Kumar, G. A. Reddy, D. Vignesh, A. Pavan, T. Vamsi, K. Rishi, and K. S. Kumar, "Automated Question Paper Generator Using LLM," International Journal of Research and Innovation in Applied Science, vol. X, no. IV, Apr. 2025, doi: 10.51584/IJRIAS.2025.10040020.

[3] G. Kurdi, J. Leo, B. Parsia, U. Sattler, and S. Al-Emari, "A Systematic Review of Automatic Question Generation for Educational Purposes," International Journal of Artificial Intelligence in Education, vol. 30, no. 1, pp. 121–204, 2020, doi: 10.1007/s40593-019-00186-y.

[4] D. Mitroulias and S. Sioutas, "A Systematic Review and Bibliometric Analysis of Automated Multiple-Choice Question Generation," Big Data and Cognitive Computing, vol. 10, no. 1, Art. no. 35, 2026, doi: 10.3390/bdcc10010035.

[5] A. Chaube, P. Chaurasia, K. Gupta, and S. Ghanate, "An AI-Driven System for Automated Question Paper Generation from Curriculum Documents," PaperPilot, 2025.

[6] A. Kumar, R. Sharma, P. Gupta, and S. Patel, "AI-Based Question Paper Generation Using Historical Examination Data," 2024.

[7] M. R. Anwar, S. Khalid, S. Alshahrani, H. S. M. Bilal, and M. Aldawsari, "MCQs Generation With Large Language Models: A Survey of Methodologies, Evolution, and Open Research Issues," IEEE Access, vol. 14, 2026, doi: 10.1109/ACCESS.2026.3652006.

[8] C. Zhang, Y. Sun, H. Chen, and J. Wang, "Generating Adequate Distractors for Multiple-Choice Questions," 2020.

[9] J. Ling and M. Afzaal, "Automatic Question-Answer Pairs Generation Using Pre-trained Large Language Models in Higher Education," Computers and Education: Artificial Intelligence, vol. 6, Art. no. 100252, 2024.

[10] P. H. S. Praveen, P. Khot, and O. Shinde, "AI-Driven Automated Question Paper Generator Using Past Exam Datasets," IRE Journals, vol. 9, no. 10, 2026, doi: 10.64388/IREV9I10-1716734.

[11] N. Mulla and P. Gharpure, "Automatic Question Generation: A Review of Methodologies, Datasets, Evaluation Metrics, and Applications," Progress in Artificial Intelligence, vol. 12, pp. 1–32, 2023, doi: 10.1007/s13748-023-00295-9.

[12] B. J. L. Narayana, V. N. S. Harsha, K. Prudhvi, B. G. H. Vardhan, and A. Sravika, "Automated Question Paper Generation using Natural Language Processing," in 2024 International Conference on Computational Intelligence for Green and Sustainable Technologies (ICCIGST), 2024, doi: 10.1109/ICCIGST60741.2024.10717510.

[13] S. Mishra et al., "Towards Automatic Generation of Questions from Long Answers," 2020.

[14] R. Bhirangi and S. Bhoir, "Automated Question Paper Generation System," International Journal of Emerging Research in Management & Technology, vol. 5, no. 4, pp. 28–34, Apr. 2016.

[15] R. T. Sairaj and S. R. Balasundaram, "Ensemble learning with RAG model to reduce redundant question topics in auto-generated exam questions," Discover Computing, vol. 28, no. 1, Art. no. 161, 2025.