Paperframe: Framing Question Paper From Input Source Questions

Nithin Kumar B
Dept. of Computer Science and Engineering
Bangalore Institute of Technology
Bengaluru, India

Maya B S
Dept. of Computer Science and Engineering
Bangalore Institute of Technology
Bengaluru, India

***Abstract*—The development of a Flask- based web application designed to automate the generation of question papers using Optical Character Recognition (OCR). The system enables users to upload three PDF documents containing exam questions. It is capable of extracting content from both text-based PDFs and scanned images embedded within PDF files. To accomplish this, the application integrates text extraction tools such as PyMuPDF for retrieving textual content from PDFs and Tesseract OCR for processing image-based content. After extracting the text, a question identification module employs regular expressions or pattern-matching techniques to isolate valid questions from the extracted data. The system then randomly selects a specified number of questions, ensuring a balanced and diverse selection. These questions are compiled into a professionally formatted question paper, which is made available for download as a PDF file. The web application features an intuitive user interface that allows users to upload the required PDFs and receive the final question paper with minimal effort. This approach streamlines the process of question paper creation, reducing manual workload and offering customization options such as selecting the number of questions to include. Additionally, the system holds potential for future enhancements, such as classifying questions by difficulty level or subject area, making it a flexible and valuable tool for academic institutions.**

Keywords—paper frame, question paper, python, paper generation, OCR.

I. Introduction

The increasing demand for streamlined and efficient educational tools has led to the development of automated systems that simplify routine academic processes. One such process is the generation of question papers, which traditionally involves extensive manual effort from educators. This project introduces Paper Frame, a Flask-based web application designed to automate the creation of question papers from user-uploaded PDF files using Optical Character Recognition (OCR) and text extraction techniques.

The application enables users to upload up to three PDF files that may contain either standard text-based questions or scanned image-based content. To extract meaningful information, the system employs the PyMuPDF library for handling text-based PDFs and integrates Tesseract OCR (via the pytesseract wrapper) for recognizing text in scanned documents. This dual capability allows the system to handle a wide range of document types with high accuracy and flexibility. After extracting the text, the system uses pattern recognition methods, such as regular expressions, to identify valid questions from the document content. It then applies randomization logic to select a user-defined number of questions, ensuring a diverse and balanced mix. The selected questions are formatted and compiled into a structured, professional-quality PDF document using the FPDF or Report Lab library. The application also includes a secure login system, a user-friendly interface, and customizable options such as topic-based filtering and difficulty-level selection, making it highly adaptable for various academic needs. Through its intuitive workflow from uploading files to downloading the final question paper the system significantly reduces the time and effort required to prepare examination materials.

This project is particularly useful for teachers, academic institutions, examination boards, and private tutors who frequently need to create customized test papers. By automating the question extraction and formatting process, Paper Frame not only improves productivity but also ensures consistency and accuracy in assessments. Future enhancements could include advanced analytics, integration with learning management systems (LMS), and machine learning models for intelligent question classification.

II. Related Work

Improved systems in Optical Character Recognition (OCR), text pull-out, and document generation automation have significantly impacted the creation of smart systems for educational content processing.

An extensive survey of OCR methods evokes preprocessing and segmentation as key components to efficient text identification, which is core to the OCR module in this project for content extraction from scanned PDFs [1]. The suggested system incorporates such principles using Tesseract OCR to recognize machine-readable text from scanned question papers, including noisy or low-resolution ones. Computer-generated question paper methods that integrate OCR with Natural Language Processing (NLP) offer methods of identifying question patterns and arranging them into formal structures [2]. This is in line with the pattern-based extraction and formatting logic this project uses to identify and arrange questions using question numbers and marks. In order to enhance the performance of OCR, numerous studies have explored increasing accuracy in learning environments using noise reduction and image preprocessing techniques [3]. This project utilizes a similar approach, including image scaling and resolution changes, to enhance text extraction from intricate document structures. Text extraction from scanned PDFs, particularly those with inconsistent layouts and alternating formatting, has been researched thoroughly [4]. The application of PyMuPDF and PDF Miner in this project accommodates both text and image-based PDFs, enhancing versatility in diverse document structures. Intelligent question generation via OCR and AI focuses on transforming raw learning text into formatted sets of questions [5]. The current system replicates this objective by auto-extracting questions, randomly selecting them, and formatting them into a set of PDF finals.

Some of the OCR-based question extraction challenges, including mixed and inconsistent layouts, have been mentioned in previous work [6]. These have been overcome in this project by implementing regular expressions and strong pattern-matching to identify valid questions while excluding irrelevant information. Template alignment and question formatting methods have also been reported to maximize OCR outputs in computerized exam systems [7]. Guided by this, the system incorporates uniform formatting and renumbering for presentation and clearness consistency in the final question paper. Improved recognition accuracy through post-OCR correction methods, including deep learning enhancements, has been proposed in recent studies [8]. While this project currently uses rule-based methods, such enhancements are considered for future integration to further refine question quality. Randomized selection processes for producing varied sets of questions are well established within the field of educational automation [9]. This is utilized in the system for ensuring diversity of produced papers and minimizing repetition for customized generation according to predetermined criteria. Also, machine learning and topic modelling-based classification of educational content has been suggested to facilitate topic-wise filtering as well as difficulty-level classification [10]. These features are being considered for future releases of this system to provide better control to the user for the question selection process.

Building on top of these research works, this project presents an integrated, OCR-supported question paper generation system that simplifies and automates exam paper generation from various PDF sources.

III. Proposed work

The proposed system aims to mechanize the process of creating question papers from input PDF sources via Optical Character Recognition (OCR) and text extraction methods. The system overcomes the inefficiencies and labour involved in the conventional question paper generation, especially in the case of scanned or unstructured documents.

The central concept is to enable users mainly educators to upload one to three PDF files containing examination questions. The PDFs can be either text-based (i.e., containing embedded content) or image-based (i.e., scanned papers). To support both, the system utilizes a dual pipeline extraction: for text-based PDFs, PyMuPDF and PDF Miner are applied to parse the embedded content directly; for image-based PDFs, the system utilizes the Tesseract OCR engine to extract textual data from page images. After text is mined, a question identification module processes the content with pattern-matching mechanisms like regular expressions to identify valid questions. The identified questions are kept in structured data formats and processed further to remove unwanted text and categorize them on the basis of question numbers, marks, or patterns like "1.a", "2.b", etc.

A question selection module next supports randomized or rule-driven picking of questions on the basis of user-specified criteria like the number of questions in total, certain topics (if supported), or level of difficulty (in future implementations). The picked questions are organized and presented in an ordered question paper with the help of the FPDF library, including proper labelling, section titles (e.g., Part A, Part B), and add-on metadata like the institution name, department, and exam name. For ensuring security and individual access, the system is designed using a Flask web framework with user authentication and session control. It offers an easy-to-use web interface through which teachers can register, login, upload files, and download the final generated question paper as PDF.

This suggested system reduces workload drastically, wipes out redundancy, and adds consistency and flexibility while making exam papers. It will be followed up with semantic categorization of questions via NLP, difficulty-level marking, and topic-wise screening for more precise customization..

A. *Workflow*

The figure 1 illustrates the end-to-end workflow of the proposed automated question paper generation system. The process begins with Login/Registration, where new users can register and their credentials are securely stored in the system database. Upon successful registration, users proceed through authentication to access the platform. After logging in, users are provided with the interface to upload scanned or digital question papers in PDF format. The system then initiates the question extraction phase, where content is parsed using OCR and text extraction methods to identify valid questions. Once extracted, a random selection algorithm is applied to choose

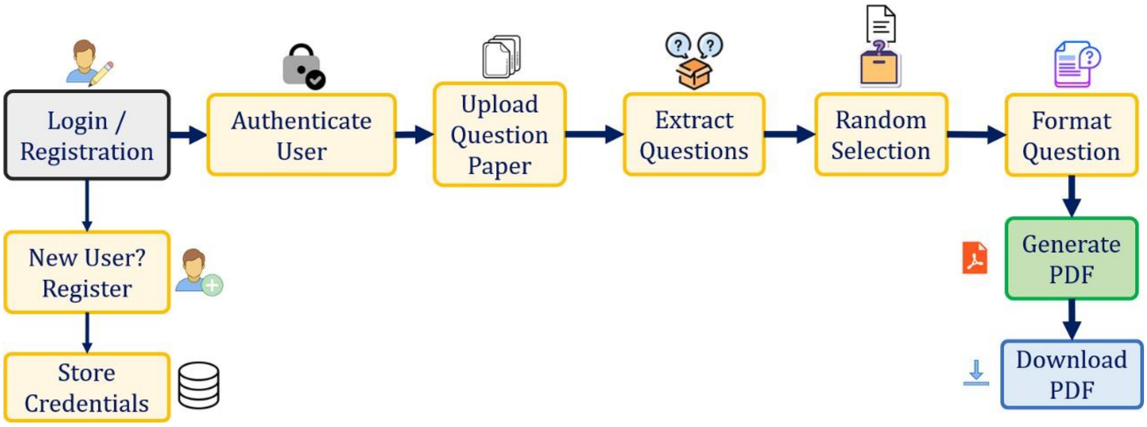


Figure 1: Block Diagram of Propose System

a diverse and balanced set of questions. These selected questions are passed through the formatting module, which organizes them into a structured layout (e.g., section-wise like Part A and Part B). The formatted questions are then sent to the PDF generation module, where a final question paper is created. Users are finally given the option to download the generated PDF, which is ready for printing or digital distribution. This pipeline automates the traditionally manual process of question paper creation, improving speed, accuracy, and ease of use.

B. *Optical Character Recognition (OCR)*

Optical Character Recognition (OCR) is a core feature that enables the extraction of text from scanned question papers stored as image-based PDFs. This is particularly useful for educators working with older, printed documents that are not digitally editable.

The system uses the Tesseract OCR engine, a reliable open-source tool, to convert scanned pages into machine-readable text. Each page of the uploaded PDF is first rendered as an image using PyMuPDF (fitz) before being processed by tesseract. Since OCR output may contain noise or formatting issues, the system applies preprocessing techniques such as resolution scaling, matrix transformations, and noise filtering to improve accuracy. The extracted text is then parsed using regular expressions to detect valid question formats, such as numbered or multiple-choice questions, while ignoring irrelevant content like headings or footers.

By integrating OCR, the system supports both text-based and image-based PDFs, making it significantly more flexible and practical for real-world academic use. It eliminates the need for manual transcription, enabling fully automated and accessible question paper generation.

C. *System Test Cases*

Each module is tested for correctness whether it meets all the expected results. Condition loops in the code are properly terminated so that they don't enter into an infinite loop. Proper validations are done so as to avoid any errors related to data entry from the user.

Table 1. System Test Cases

Test Case Number	**Testing Scenario**	**Expected result**	**Result**
TC-01	Upload PDF Files	PDF file uploaded successfully	Pass
TC-02	Upload PDF Files (Invalid)	Error message: "File format not supported"	Pass
TC-03	OCR Text Extraction	Extracted text matches the content of the question paper	Pass
TC-04	Extracting Questions	Extracted questions: "1. What is Python?", "2. Define OOP"	Pass
TC-05	Random Question Selection	Randomly selected questions (e.g., Q2, Q4, Q5)	Pass
TC-06	File Download (Generated Question Paper)	PDF file available for download	Pass
TC-07	User Logout	User logged out and redirected to login page	Pass

D. *Sequence Diagrams*

The sequence diagram depicts the communication among the user, system, and server during the life cycle of the question paper generation process in the figure 2. The user first logs in to the system, which passes the login credentials to the server for verification. Upon successful verification, the server confirms the login, and the user uploads PDF files with questions. The system saves this data and allows the user to verify uploaded details. If the user chooses to manage papers, the system invokes the process of question extraction from uploaded documents and fetches structured question details from the server. These details are then displayed to the user for confirmation or correction. On review, the user makes a download request for the formatted question paper, and the system accordingly updates records and prepares the final

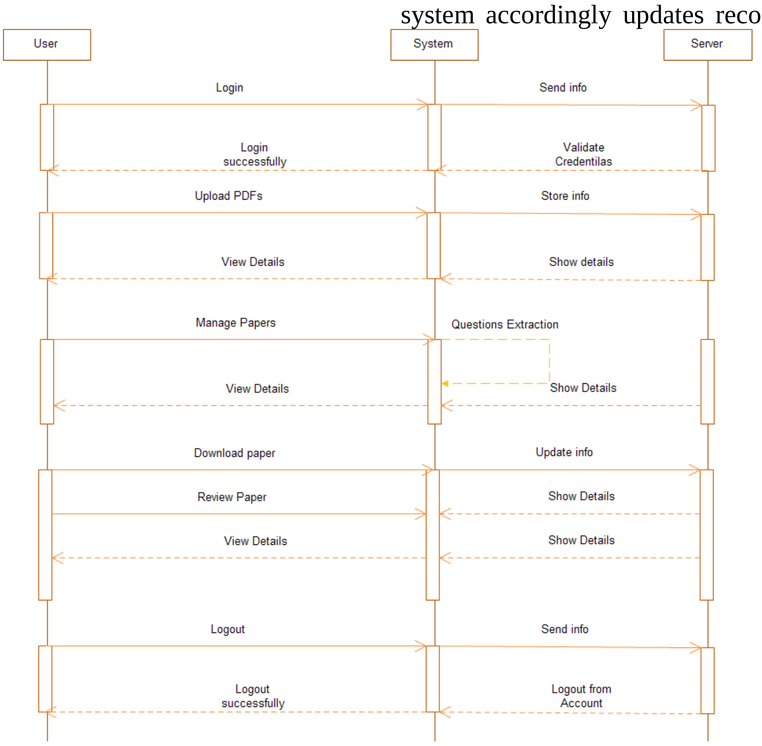


Figure 2: Sequence Diagrams

paper. The server returns the compiled paper details, and the user downloads and examines it. The user logs out after completing all tasks, and the system sends a logout request to the server to end the session securely. This series guarantees an efficient, secure, and automatic process for handling question papers from upload to download.

IV. Result and Discussions

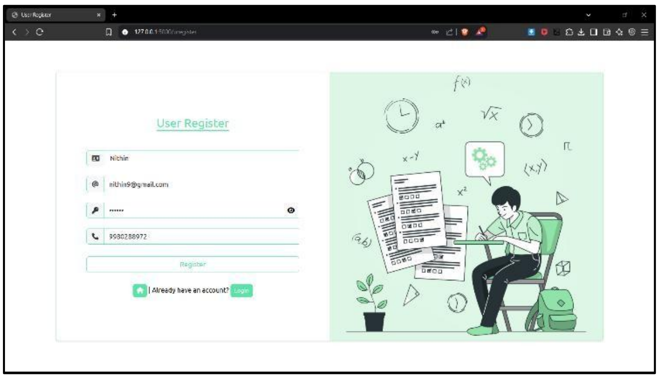
The system implemented successfully as a web application based on Flask that could create structured question papers from uploaded PDFs. The system proved extremely effective in processing both text-based and image-based PDFs through the use of PyMuPDF for parsing PDFs and Tesseract OCR for recognizing scanned images. The OCR module was found to correctly extract information from low-resolution scanned documents with little error after using preprocessing methods such as resolution scaling during testing. The question extraction rationale, implemented with regular expressions, accurately detected different question types such as numbered and multiple-choice questions.

Up to three PDFs were uploaded by the users from which the system pooled together and randomly picked questions to create a question paper of an appropriate balance. The output was finally displayed in a nicely formatted PDF layout with suitable sectioning (e.g., Part A, Part B) and could be viewed and downloaded immediately. The random selection mechanism of the system guaranteed randomness in generated papers between different runs and hence improved usability in educational settings. All unit and system test cases that were defined passed successfully, validating robustness, correctness, and ease of use. Overall, the system significantly minimized time and effort in creating question papers manually and provided a flexible and user-friendly interface for instructors.

A. Snapshots of results

1. User Registration Page

The registration page enables new users to sign up by entering their name, email address, phone number, and a safe password. Upon submission, the system saves these credentials in the backend database, providing secure entry and personalized interaction with the application. Error messages assist the user in the event of empty or incorrect input.



User Register

Nithin

nithin9@gmail.com

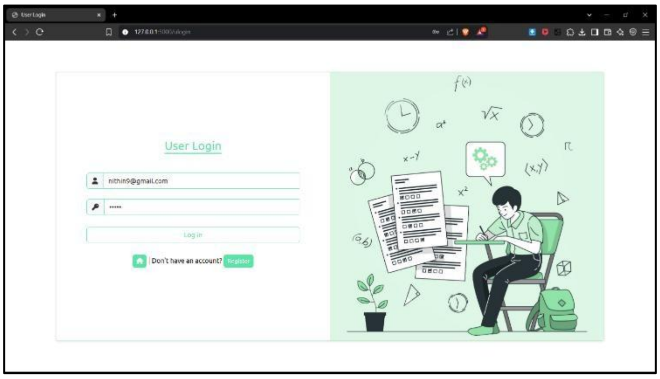
••••••

9980288972

Figure 3. Registration Page

2. Login Page

The login page gives access to registered users by authenticating their email and password. When the credentials are correct, the system verifies the user and redirects to the dashboard. Invalid inputs invoke proper error notifications, securing user authentication and session integrity.



User Login

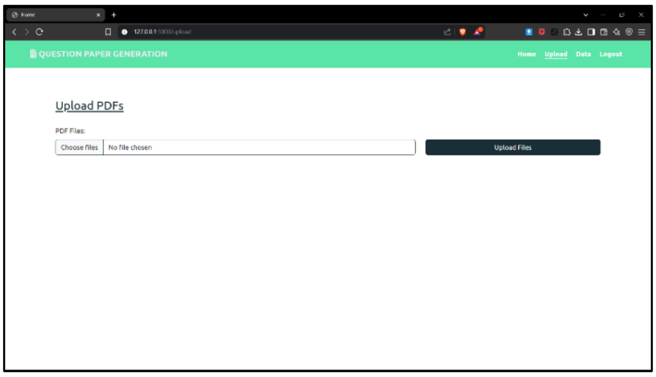
nithin9@gmail.com

•••••

Figure 4. Login Page

3. Upload Page

After logging in, the users are prompted to the upload page where they have the option to upload a maximum of three PDF files. The PDFs may be text-based or scanned question sheets. The system reads the files, performs OCR if necessary, and commences question extraction. The interface offers file validation and provides status messages to ensure that the uploading has been successful.



Upload PDFs

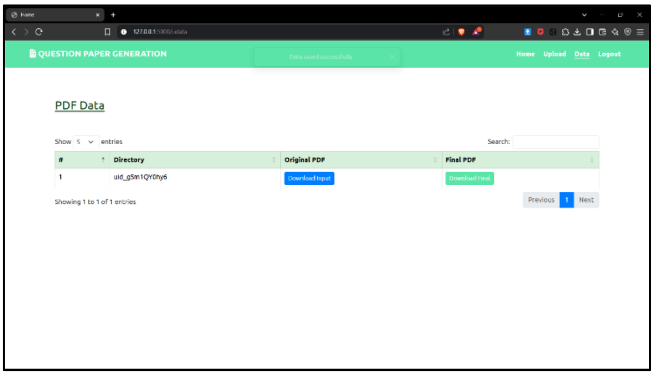
PDF Files:

No file chosen

Figure 5. Upload Page

4. Download Page

Once the question paper is produced, the download page displays the user with a button to download the final formatted question paper in PDF format. This makes it convenient for access and easy availability of the document for viewing or printing purposes.



Data saved successfully

PDF Data

Show 5 entries Search:

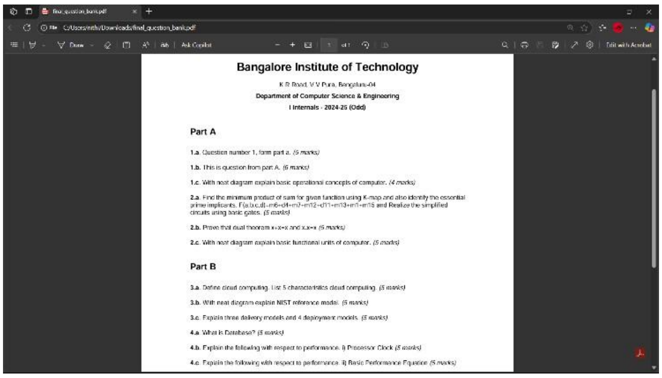
#	Directory	Original PDF	Final PDF
1	uid_g5m1QY0hy6	Download Input	Download Final

Showing 1 to 1 of 1 entries

Figure 6. Question Paper Downloading Page

5. Final Generated PDF (Output Page)

The output page displays a preview of the final question paper, organized and formatted into sections such as Part A and Part B. Questions are numbered distinctly, optionally labelled with their respective weightage, and categorized according to standard academic formatting. This verifies the success of question extraction, randomization, and layout generation.



final_question_bank.pdf

C:/Users/mithi/Downloads/final_question_bank.pdf

Bangalore Institute of Technology

K R Road, V V Pura, Bengaluru-04

Department of Computer Science & Engineering

I Internals - 2024-25 (Odd)

Part A

1.a. Question number 1, form part a. *(5 marks)*

1.b. This is question from part A. *(5 marks)*

1.c. With neat diagram explain basic operational concepts of computer. *(4 marks)*

2.a. Find the minimum product of sum for given function using K-map and also identify the essential prime implicants. F(a,b,c,d)=m6+d4+m7+m12+d11+m13+m1+m15 and Realize the simplified circuits using basic gates. *(5 marks)*

2.b. Prove that dual theorem x+x=x and x.x=x *(5 marks)*

2.c. With neat diagram explain basic functional units of computer. *(5 marks)*

Part B

3.a. Define cloud computing. List 5 characteristics cloud computing. *(5 marks)*

3.b. With neat diagram explain NIST reference model. *(5 marks)*

3.c. Explain three delivery models and 4 deployment models. *(5 marks)*

4.a. What is Database? *(5 marks)*

4.b. Explain the following with respect to performance. i) Processor Clock *(5 marks)*

4.c. Explain the following with respect to performance. ii) Basic Performance Equation *(5 marks)*

Figure 7. Final Output

V. Conclusion

The OCR-based question paper generation system delivers an effective and automated method for teachers and schools to create question papers quickly and precisely. Utilizing Optical Character Recognition (OCR) technology, the system can read and capture questions from scanned PDF documents, disallowing manual inputting of data. It delivers the ability to process text-based as well as image-based PDFs, providing a broad compatibility with available resources. The use of Flask as the web framework guarantees that the user experience is smooth, with features like user authentication, file upload, extraction, randomization, and generation of final question paper in PDF form. Also, libraries such as Tesseract for OCR, PyPDF2 for handling PDFs, and Report Lab for PDF creation make the system strong and adaptable. The functionality to randomly choose questions on pre-defined parameters like difficulty level or subject strengthens the usability of the system for multiple educational contexts. Additionally, the security features of the application, like password hashing and secure user login, maintain the integrity of the user's data. Overall, the project showcases the strength of automation in educational functions. Mitigating the level of manual effort required in question paper preparation, it enables teachers to spend more time on teaching and less on bureaucracies. With time, more features like content-based filtering of questions, analytics, and learning management system integration could make it an even more powerful tool in the new educational environment.

References

[1] Smith J, & Wang L, “OCR for question paper generation: A review”. Journal of Educational Technology, Vol 15(3), pp 25–35, 2020.

[2] Lee. R, & Chan A, “Automated question paper generation using OCR and natural language processing”. Proceedings of the International Conference on Educational Technology (pp. 112–120), Springer, 2019.

[3] Patel K., & Jain M, “Improving OCR accuracy for educational document processing”, International Journal of Machine Learning, Volume 8(1), 58–72, 2021.

[4] Gupta S., & Kumar P, “Text extraction from scanned PDFs for educational applications”, Journal of Educational Research, Volume 24(4), pp. 103–112, 2018.

[5] Zhang, T., & Lin X, “Automated question generation using OCR and AI”, International Conference on Artificial Intelligence in Education, pp. 130–140, 2020.

[6] Zhang, Y., & Sun C, “Challenges in OCR for scanned question paper extraction”, Journal of Artificial Intelligence in Education, Volume 35(7), 15–25, 2019.

[7] Kumar A., & Mehta R, “Optimizing OCR-based question paper creation”, Proceedings of the IEEE International Conference on Document Analysis and Recognition, pp. 205–210. IEEE. 2021.

[8] Fernandez, J., & Clark R, “Improved text recognition in educational documents with OCR”, Journal of Computational Linguistics, Volume 18(2), pp 65–78, 2022.

[9] Thomas, D., & Edwards P, “Leveraging AI for question paper automation”, IEEE Conference on Educational Technology pp. 150–160. IEEE. 2020.

[10] Johnson L., & Lee K, “OCR and machine learning for educational document processing”, Journal of Educational Innovation, Volume 23(5), pp. 50–60, 2021.