

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Volume 5, Issue 5, March 2025

Automated Diploma Result Evaluation System

Pranit Adhangle, Yogesh Kamankar, Aaryan Ahirrao, Darshan Aher, Mahesh Bhandakkar Department of Information Technology, Head of Department, Department of Information Technology

Matoshri Aasarabai Polytechnic, Eklahare, Nashik, Maharashtra, India

Abstract: The objective of this project is to develop an automated system that extracts student marks from PDF files, organizes the data into a structured format, and performs various analyses, including calculating the overall passing percentage, identifying toppers, and listing students who scored above 75aims to streamline the result analysis process for educational institutions by automating data extraction, analysis, and report generation. The system will be developed using a combination of modern web technologies. The frontend will be built using React.js to ensure a dynamic and user-friendly interface. The backend logic, responsible for handling API requests and processing the data, will be developed using Java or Node.js/Express. Student result data will be stored in a MySQL or PostgreSQL database, allowing efficient data storage and retrieval.

Keywords: Automated system, student result extraction, PDF parsing, data analysis, React.js, Node.js, MySQL, PostgreSQL, PDFKit

I. INTRODUCTION

This web application aims to streamline the process of managing student results by allowing users to upload PDF result files. The system will extract data from the PDF, store it in a database subject-wise, and provide analysis features like identifying top-performing students, those who achieved distinctions, and students who failed. Additionally, users can generate customized PDF reports based on these categories.

II. PROBLEM STATEMENT

- Educational institutions, particularly teachers, face a significant challenge when processing student results after they are declared. Traditionally, teachers are required to manually enter the marks of each student, perform time-consuming calculations to identify top- pers, determine students who scored above certain thresholds (e.g., 75%), and analyze overall class performance. This manual process is not only tedious and error-prone but also reduces the productive time that teachers could otherwise dedicate to educational activities.
- The primary objective of this project is to reduce the administrative burden on teachers by automating the result processing and analysis tasks. By allowing teachers to simply upload the result PDFs, the system will automatically extract and organize the data, perform the necessary calculations, and generate detailed reports. This automation minimizes errors, saves time, and allows teachers to focus on their core teaching responsibilities, thereby increasing their overall productivity.
- In essence, the need for an automated result management system stems from the inef- ficiency and stress associated with manual result processing, and the aim is to provide a reliable solution that simplifies this process while delivering accurate and insightful performance analytics.

III. Literature Survey

The literature survey highlights previous research and existing solutions related to result pro- cessing systems, PDF data extraction, educational databases, and automated reporting systems. Key contributions and limitations are summarized below:

3.1 Student Performance Management Systems

Existing Learning Management Systems (LMS) and School Management Software provide Ba-steresult management features. However, they lack advanced data analysis capabilities, PDF integration, and performance segregation. These Copyright to IJARSCT DOI: 10.48175/IJARSCT-24175 JARSCT 367 www.ijarsct.co.in



International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Volume 5, Issue 5, March 2025

systems are primarily focused on record keep- ing and do not offer in-depth analytical reports or customizable filters for student performance categories such as toppers or failed students.

3.2 PDF Data Extraction Techniques

Text mining and natural language processing (NLP) techniques have been widely explored for extracting data from PDF files with varying structures. Tools such as PyMuPDF and PDF.js have demonstrated efficiency in extracting tabular and textual data, enabling automated processing of unstructured result files.

3.3 Database Systems for Educational Data

Databases such as MySQL, PostgreSQL, and MongoDB are widely used to store student records. Research has explored efficient ways to store large volumes of educational data, focusing on the scalability and performance of relational and non-relational databases. The key challenge high-lighted in the literature is ensuring that the database schema is flexible enough to accommodate diverse grading structures, subject-wise marks, and performance categories

3.4 Automated Report Generation Systems

Automated report generation systems are increasingly being used in educational institutions for compiling student results into structured reports. These systems integrate data from various sources such as Learning Management Systems (LMS), grading systems, and student databases. While some tools provide basic report generation, they often lack advanced filtering options and fail to handle large datasets efficiently. Moreover, they typically offer limited customization for report formats, making them unsuitable for institutions that require specific analytical perspectives, such as highlighting trends, student performance metrics, and comparisons across different student cohorts.

3.5 Educational Data Mining in Student Result Analysis

Educational Data Mining (EDM) is increasingly used to predict student performance, identify at-risk students, and provide personalized learning recommendations. Machine learning algo- rithms, such as decision trees, SVMs (Support Vector Machines), and neural networks, have been employed in various studies to analyze historical data and predict future performance trends. These systems aim to improve the quality of education by proactively identifying students who may need extra help and rewarding high performers

Sr. No	Author(s)	Year	Paper Title	Summary	Our Additional
					Fea- tures
1	Singh, P., et al.	2016	Automated System for	Proposes an auto-	Our system will auto- mate
			Educational Resul	mated system to	result extraction from PDFs of
			Processing	process student re- sults	different structures using
				using prede- fined	advanced parsing techniques.
				templates.	
2	Patel, A., et al.	2018	Design and Devel- opmen	Focuses on an- alyzing	We will automate data
			of Student Performance	student performance	extraction directly from PDF
			Analytics System	but lacks real-time data	result files, ensuring real-time
				extraction from result	data analysis.
				files.	
2		2010			
3	Gupta, S., et al.	2019	Real-Time Reporting in	Emphasizes the	Our system will enable users
			Educational Result	generation of real- time	to filter and segre- gate
			Systems	reports but lacks	students by perfor- mance
				filtering capa- bilities	categories like top- per,
				for student categories.	distinction etc.

3.6 Comparison of Related Works







International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Volume 5, Issue 5, March 2025

IJARSCT

4	Khan, M., et al.	2020	Educational	D	ata	Explores t	the use	of AI	We	will	provide	detailed,
			Processing	and	Result	for result p	predicti	on and	custor	nizabl	e PDF	reports
			Management	Using	AI	pro- cessi	ing but	lacks	based	on us	er-define	d fil- ters
						customize	d repor	rt- ing	and an	nalysis	needs.	
						options.						
5	Sharma, A., et	2021	Automating		Result	Develops	a syste	em for	We w	ill de	sign an i	ntu- itive,
	al.		Compilation	for	Edu-	automating	g	result	user-fi	riendly	y inter-	face that
			cational Insti	tutions		compilatio	on but l	acks a	allows	s edu	icators 1	to easily
						user- frier	ndly in	terface	upload	a PD	Fs and	manage
						for educate	ors.		results	5.		

Table 1: Comparison of Related Works

This literature survey, therefore, helped me understand what already exists in the system of educational result processing. It emerges that most partially contend with the problems of result management and this is not comprehensive. Though automated result processing has brought down much manual effort, several limitations are still prevalent, such as support for diverse PDF formats, dependence on manual data entry, lack of filtering options, and no user-friendly interface. Furthermore, predictive capabilities have been introduced by AI-based approaches but lack customization for detailed reporting. Such findings bring forward the requirement of an advanced system that not only automates data extraction and analysis but also provides scalability, accuracy, and ease of use. Our system will fill these gaps by integrating robust technologies and features specifically tailored to the needs of educators and institutions.

IV. PREPOSED METHODOLOGY





The system is developed using the waterfall model, because all the requirements, resources process flow was already discussed. The waterfall model is termed as classical life cycle, as it suggests systematic, robust, ideal, sequential approach to software development. Here, any phase in the development process begins only if the previous phase is complete. Typically, the outcome of one phase is an input for the next phase. It is fast and easy to understand and implement.

Framework Activities:

- **Requirements:** This phase involves analyzing and documenting the application's poten- tial requirements in a specification document, defining what the application should do but not how.
- Analysis: The system is analyzed to generate models and business logic for the applica- tion.
- Design: Covers technical requirements like programming language, data layers, and ser-vices, creating a design specification for implementing the analyzed business logic.
- **Coding:** The source code is written, implementing models, logic, and service integrations outlined earlier.

Copyright to IJARSCT www.ijarsct.co.in

DOI: 10.48175/IJARSCT-24175

369

IJARSCT



International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Volume 5, Issue 5, March 2025

- **Testing:** Testers identify and report issues to be resolved, often requiring coding adjust- ments to address bugs.
- **Deployment:** The application is deployed to a live environment, including ongoing sup- port and maintenance to ensure functionality

V. MODULES

User Authentication:

- Admin login to manage result uploads and report generation.
- Teacher login for viewing and analyzing class-wise results.

PDF Upload and Parsing:

- Functionality to upload student result PDF files.
- Extract text from PDFs and map it to respective subjects and students.

Data Segregation and Filtering:

- Identify and list topper students (highest marks).
- List students who got distinctions in subjects.
- List failed students.
- Provide filters to view data class-wise, subject-wise, or overall.

Database Management:

- Store subject-wise student marks, grades, and overall performance.
- Support for student profiles linked with their academic records.

Report Generation:

Allow users to generate PDF reports based on various performance filters (e.g., top- pers, failed students).



Figure 2: Architecture Diagram

This architecture outlines an automated system for managing and analyzing student results efficiently. The workflow begins with Admin and Teacher Users, who interact with the system via a Web Application secured by an Authentication Service. Users upload student records as PDFs through the PDF Upload Service, which are then processed using a PDF Parsing Service. Key processes include OCR (Optical Character Recognition) and Text Parsing, which ex- tract data from unstructured documents. The extracted information is validated and structured before being stored in a Subject-wise Data Storage module. This ensures data integrity and facilitates easy retrieval for further operations.

Copyright to IJARSCT www.ijarsct.co.in DOI: 10.48175/IJARSCT-24175



370

IJARSCT



International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Volume 5, Issue 5, March 2025

The system integrates advanced analysis features, such as the Analysis Report Service for generating performance insights, and filters like the Toppers Filter and Fail Students Filter to identify exceptional and underperforming students, respectively. Data can be queried and aggregated through a Query Service and Aggregation Service, with results displayed on a user- friendly Frontend Interface or exported as final PDF outputs.

VI. ADVANTAGES

• Automation and Efficiency: The system automates the entire process from uploading student records to generating reports, saving time and reducing manual effort.

• Customizable Reporting: The system offers flexible reporting options, enabling reports based on specific criteria like toppers, failed students, or subject-wise performance.

• Data Integrity and Security: With secure data storage, validation, and regular back- ups, the system ensures the integrity and safety of student data.

• User-Friendly Interface: The web-based system has an intuitive interface that makes it easy for educators and administrators to navigate and use effectively.

• Performance Segregation and Filtering: It can segregate and filter data based on performance, enabling detailed analysis of student achievements and areas needing im- provement.

VII ALGORITHM AND PSEUDO CODE

START

// Step 1: Upload PDF and Extract Data DISPLAY "Upload Student Result PDF" INPUT pdfFile data <- PARSE pdfFile</pre>

// Step 2: Validate and Clean Data IF data is INVALID THEN DISPLAY "Error: Invalid or Missing Data" EXIT **ENDIF** // Step 3: Store Data in Database FOR each studentRecord IN data DO STORE studentRecord in StudentsTable STORE subjectMarks in ResultsTable END FOR // Step 4: Analyze Data // Topper List students <- GET all students with total marks from Database SORT students by totalMarks DESCENDING topperList <- SELECT top N students // Distinction Achievers distinctionList <- GET students where marks >= distinctionThreshold for all subjects // Failed Students failedList <- GET students where marks < passThreshold in any subject // Step 5: Generate Reports GENERATE report for topperList GENERATE report for distinctionList GENERATE report for failedList // Step 6: Display or Download Results DISPLAY options to VIEW or DOWNLOAD reports END

Key Highlights in Pseudo Code

- 1. PDF Parsing: PARSE pdfFile assumes a function to extract structured data from the PDF.
- 2. Analysis Logic: Uses simple conditions to generate results like toppers, distinctions, and failures.
- 3. Report Generation: Separate functions are used to create reports in the desired format.
- 4. User Interaction: Allows the user to upload, view, or download results.

Copyright to IJARSCT www.ijarsct.co.in DOI: 10.48175/IJARSCT-24175



IJARSCT



International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Volume 5, Issue 5, March 2025

VIII. FUTURE SCOPE

• Integration with Machine Learning Models: Future versions can integrate machine learning models for predictive analytics, identifying trends and patterns in student per- formance, such as forecasting grades or identifying at-risk students earlier.

• Support for Multiple File Formats: Expanding the system's capability to handle other file formats like Excel or CSV would increase its versatility.

• Mobile Application Support: A mobile version of the system could be developed, allowing teachers and admins to manage reports and access results on-the-go.

• Advanced Data Analytics: Incorporating data visualization tools, trend analysis, and real-time performance tracking could further empower educators with actionable insights.

• Collaboration and Communication Tools: Adding features for collaboration among teachers and real-time communication can enhance the system's value.

• Integration with Other Educational Systems: Future versions could integrate with Learning Management Systems (LMS) or student management platforms to offer a com- prehensive solution for educational institutions.

• Improved OCR Accuracy: As OCR technology evolves, its accuracy in extracting text from scanned PDFs can further enhance the system's performance.

IX. CONCLUSION

The Student Result Management System will simplify The Automated Diploma Result Eval- uation System will revolutionize the way educational institutions handle and analyze student results. This system, integrating automation with data analysis, will minimize human error and significantly reduce the time required for processing student performance data. It will auto- matically scan the PDF and retrieve results; then it organizes the data and executes multiple analysis jobs like topper identification, failing student tracking, and detailed performance re- ports. Educators and administrators will be able to make better decisions since it allows them to easily track students' progress and produce reports that are consistent and accurate. This tool would be user-friendly, thus enhance operational efficiency while providing real-time insights and assuring the correctness of results that would simplify the management of academies.

REFERENCES

[1] React.js Official Documentation. React: A JavaScript library for building user interfaces.

https://reactjs.org/ : React JS offers detailed guidance for implementing dynamic UIs.

[2] Node.js Official Documentation. Node.js: JavaScript runtime built on Chrome's V8 JavaScript engine. https://nodejs.org/en/docs/ : Node.js is used for the backend of the project for API and logic.

[3] PyMuPDF (MuPDF). Python bindings for MuPDF, a PDF rendering library. https://pymupdf.readthedocs.io/en/latest/ : PyMuPDF is used for PDF data extraction in the result analysis system.

[4] PDFKit Documentation. Creating PDFs in Node.js. https://pdfkit.org/ : PDFKit is em- ployed to generate PDF reports for the system.

[5] PostgreSQL Official Documentation. PostgreSQL: The world's most advanced open-source database. https://www.postgresql.org/docs/: PostgreSQL is used to store student result data securely.

[6] MySQL Official Documentation. MySQL: The world's most popular open-source database. https://dev.mysql.com/doc/ : MySQL is used for efficient data management of academic results.

[7] Waterfall Model in Software Development. A traditional project management approach to software engineering. https://www.tutorialspoint.com/sdlc/sdlcwaterfallmodel.htm : The Waterfall model is applied for structured development of this system.

[8] Data Flow Diagrams in Software Engineering. Techniques for modeling the flow of information in a system. https://www.tutorialspoint.com/sdlc/sdlcdataf lowdiagram.htm : DFDs are essential for depicting the flow and processing of student result data.

DOI: 10.48175/IJARSCT-24175

