

# Automated Data Extraction for Scanned Documents

Vijay Wagh and P.S. Deshpande

Department of Computer Science and Engineering

Shreeyash College of Engineering and Technology, Ch. Sambhajinagar-431001-India

---

**Abstract**— Manual entry of student report data into Excel sheets is a significant issue in higher education, leading to wasted faculty time and potential errors. To address these challenges, we have developed the Data Extraction System, a web application designed to streamline the process of storing, maintaining, and analyzing student mark sheet data as required by universities. This application automates the extraction of information from student report PDFs, such as semester grades, and compiles it into a comprehensive Excel spreadsheet. Users simply upload the PDFs and select the desired semester, and the app identifies key data, including the year, semester, enrollment number, student name, SGPA, CGPA, and individual grades. The organized data is then stored both in an Excel file and an SQL database. This system saves teachers' time by efficiently managing and consolidating student data from various reports, enhancing accuracy and accessibility.

**Keywords**— *Automation Web Application, Excel, Faculty Time Management, MySQL, PDF Processing Data Extraction, University Data Management.*

---

## 1. Introduction

Today, technology plays a significant role in making our lives easier, especially in education. Smartphones, with their multitude of apps, have become invaluable tools. One crucial way they contribute is by enhancing how schools manage student records and grades.

**The Problem:** Traditionally, recording student performance involves extensive paperwork and manual data entry, which is time-consuming and prone to errors. Extracting data from PDF documents and inputting it into Excel spreadsheets is particularly challenging and labor-intensive, diverting valuable time from teachers and administrators.

**Our Solution:** To address these issues, we've developed a web application that automates the extraction of data from PDF files and saves it into Excel files. This app streamlines the management of student academic records, reducing workload and minimizing errors.

**How It Works:** Users can upload multiple PDF files containing student records into the app. The app extracts crucial information, such as enrollment numbers, student names, and grades (SGPA and CGPA), based on the selected semester. It then organizes this data into an easily manageable and accessible Excel file.

**Organization of the paper:** This manuscript is organized as follows:

- **Section II:** Reviews related methods for managing academic records.
- **Section III:** Details the development of the PDF-to-Excel conversion app.

- **Section IV:** Discusses the results and benefits of using this app in schools.
- **Section V:** Concludes and suggests potential future improvements.

By leveraging this technology, schools can significantly enhance the efficiency and accuracy of their academic record-keeping processes.

## 2. Related Work

Many approaches have been proposed to solve the manual data storage of student reports. Here are some of the notable methods:

### Approach in [1]:

- **Description:** A data extraction system was proposed for automated PDF to Excel conversion.
- **Features:** This system includes local overall PDF extraction and paid features, and it is designed to handle high traffic.
- **Limitations:** Despite its advanced functionalities, this system often fails to extract accurate data from PDFs, making it unreliable for precise data management.

### Approach in [2]:

- **Description:** Similar to the first approach, this system also features local PDF extraction and paid functionalities.
- **Features:** It is equipped to manage high traffic and promises comprehensive data extraction.
- **Limitations:** The system struggles with high traffic and frequently extracts incorrect data from report PDFs. This inaccuracy undermines its effectiveness and reliability.

### Approach in [3]:

- **Description:** This system, while similar in its local PDF extraction and paid features, is another notable attempt to automate data extraction.
- **Features:** It includes features intended to handle significant data loads and promises efficient PDF extraction.
- **Limitations:** Under high traffic conditions, this system often hangs or crashes when report PDFs are uploaded. These performance issues lead to inefficiency and user frustration, hindering its practicality in real-world applications.

### Approach in [4]:

- **Description:** Another approach involves cloud-based data extraction systems that aim to mitigate local resource constraints.
- **Features:** These systems offer scalable solutions by leveraging cloud computing, which can handle larger volumes of data and traffic without compromising performance.
- **Limitations:** While cloud-based systems reduce the strain on local resources, they introduce concerns related to data security and privacy. Additionally, they often require a stable internet connection, which can be a limitation in some regions.

**Approach in [5]:**

- **Description:** Machine learning-based extraction methods are being explored to enhance accuracy.
- **Features:** These systems use algorithms to learn from existing data patterns, improving the precision of data extraction over time.
- **Limitations:** The implementation of machine learning models requires extensive training data and computational resources. Moreover, the initial setup and tuning of these systems can be complex and time-consuming.

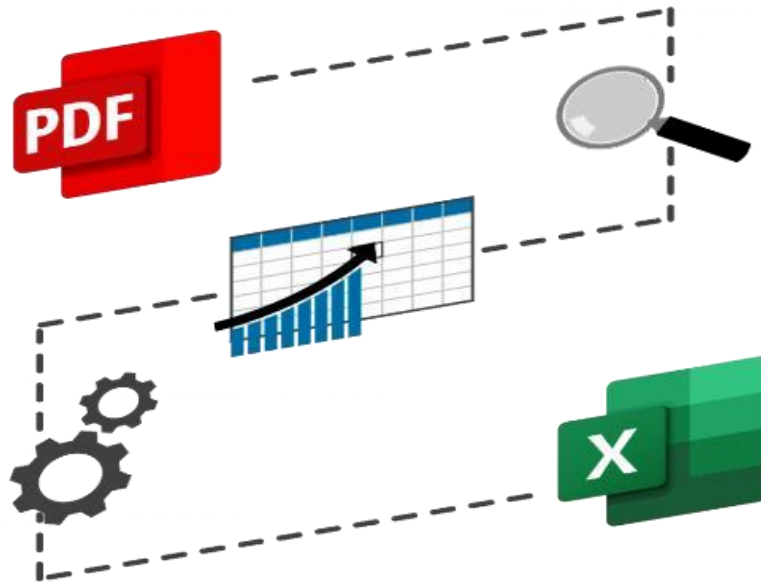


Fig. 1. Data extraction system proposed in [1].

**Our Solution**

Our web application was designed to overcome the limitations identified in previous approaches. It incorporates advanced data extraction algorithms to ensure accuracy and reliability. The app is built to handle high traffic without performance degradation and is available to users without hidden costs.

By integrating the strengths of previous approaches while addressing their weaknesses, our application offers a robust solution for managing student academic records efficiently and accurately.

**3. System Architecture and Functionality**

The proposed PDF to Excel-data extraction system comprises several components, as depicted in Fig. 2. This system is connected to a server and a file storage mechanism, allowing the administrator to oversee and manage the uploaded PDFs and the resulting Excel files.

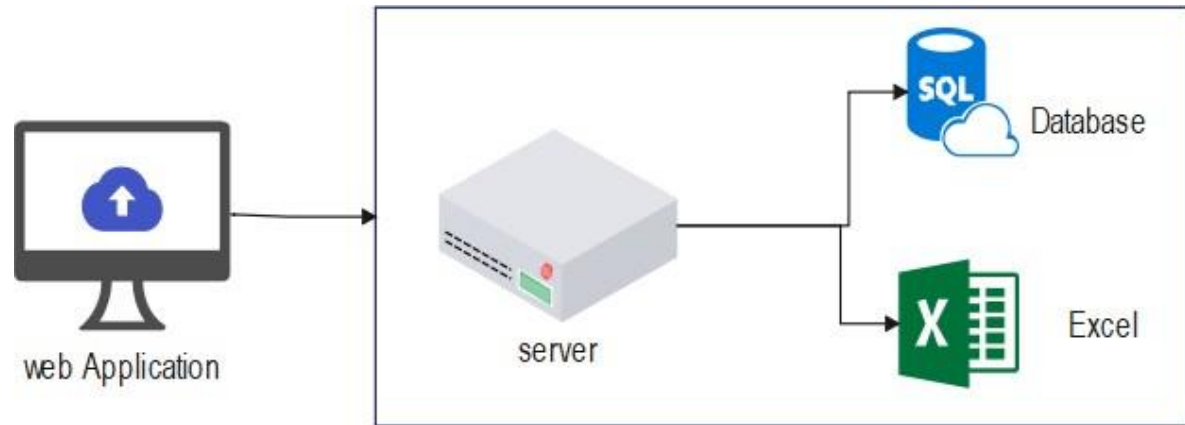


Fig. 2. Proposed Data Extraction System.

The application's front and back end are developed using Flask, a lightweight web framework for Python. This framework was chosen for its simplicity and efficiency in handling web requests and file uploads. Flask facilitates rapid development and provides the necessary tools to manage the application's core functionalities.

The system connects to a local file system to store and retrieve uploaded PDFs and generated Excel files, as illustrated in Fig. 3. This local storage approach ensures quick access and efficient management of files, supporting the seamless operation of the data extraction process.

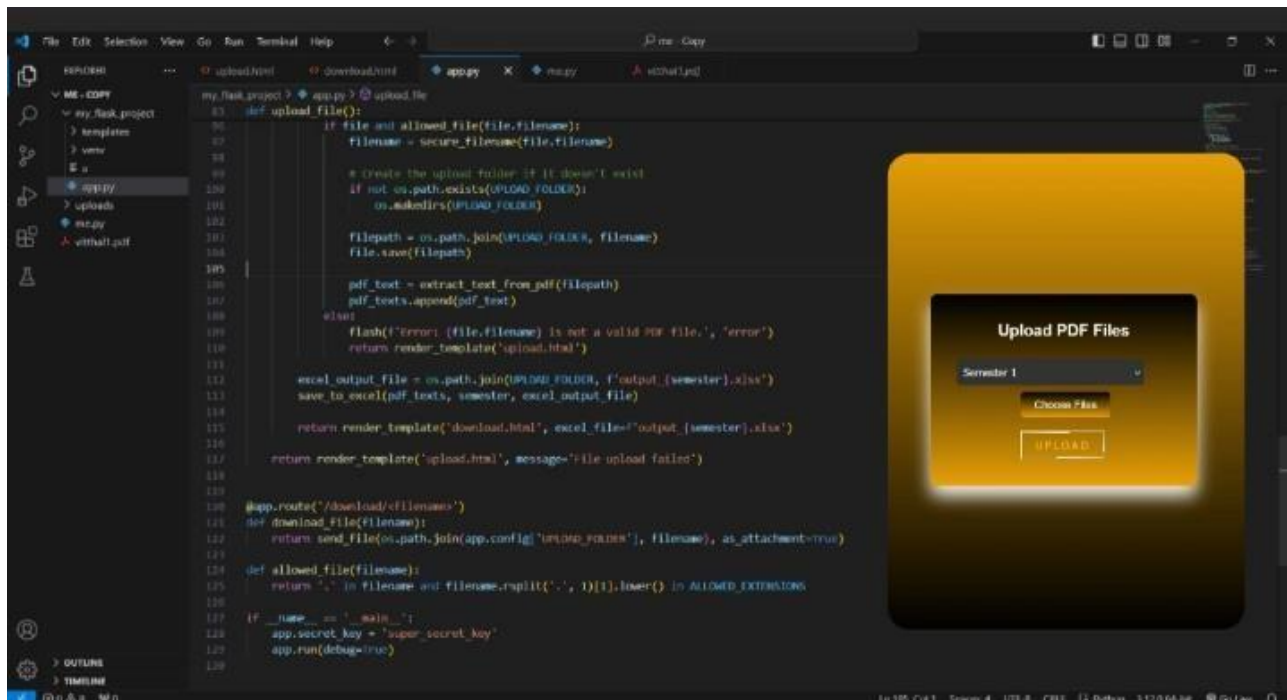


Fig. 3. User Interface Development using Flask.

The proposed PDF to Excel data extraction system is composed of several interconnected components, as illustrated in Fig. 5. This system is designed to efficiently manage the extraction and organization of student data from PDF reports, converting it into structured Excel spreadsheets.

### *Key Components*

#### **1. Front-End and Back-End Development:**

The application's front and back end are developed using Flask, a lightweight web framework for Python. Flask was chosen for its simplicity and efficiency in handling web requests, file uploads, and its compatibility with other essential libraries.

#### **2. PDF Text Extraction:**

The application uses the PyPDF2 library to read PDF files and extract their textual content. PyPDF2 is a robust tool for parsing and extracting text from PDFs, which serves as the foundation for data processing.

#### **3. Data Processing and Excel Generation:**

Once the text is extracted from the PDFs, it is processed and structured into an Excel spreadsheet using the Openpyxl library. Openpyxl enables the creation, manipulation, and customization of Excel files in Python, allowing the application to organize extracted data into a coherent and accessible format.

#### **4. File Management and Security:**

The system connects to a local file system for storing and retrieving uploaded PDFs and generated Excel files, as depicted in Fig. 3. Uploaded PDFs are securely stored with filenames sanitized using the secure filename utility from Werkzeug to prevent directory traversal attacks.

#### **5. User Interaction:**

Users interact with the system through a user-friendly web interface. They can upload multiple PDF files simultaneously, select the relevant semester, and initiate the data extraction process. The application extracts key data such as enrollment numbers, student names, SGPA, and CGPA from the PDFs, and saves this data into an organized Excel file.

#### **6. Error Handling and Feedback:**

The application includes robust error-handling mechanisms to provide feedback to users in case of invalid file uploads or other issues. This ensures a smooth user experience and helps in quickly identifying and resolving any problems.

#### **7. File Download:**

After processing, users can download the resulting Excel file directly from the application. This end-to-end process automates the otherwise manual task of extracting and organizing data from PDF documents, significantly improving efficiency and accuracy.

## Technologies Used

- **Flask:** Simplifies web interactions and handles the core functionalities of the application.
- **PyPDF2:** Extracts text from PDF files efficiently.
- **Openpyxl:** Creates and manipulates Excel files, allowing for structured data representation.
- **Werkzeug:** Provides utilities for secure filename handling to enhance security.

The application leverages the simplicity and flexibility of Flask for web interactions, the robustness of PyPDF2 for PDF text extraction, and the versatility of Openpyxl for Excel file creation. This combination of technologies makes the system a powerful tool for managing and processing academic data, automating the extraction and organization tasks, and significantly enhancing efficiency and accuracy in educational institutions.

## 4. Results and Discussions



Fig. 4. Upload and Download page.

The Data Extraction Web App has been tested at Shreeyash College of Engineering and Technology, Ch. Sambhajinagar, India. This web-based tool is designed to efficiently extract text from PDF files and organize it into a structured Excel spreadsheet. Built using Flask, a Python web framework, the application provides a straightforward and user-friendly interface.

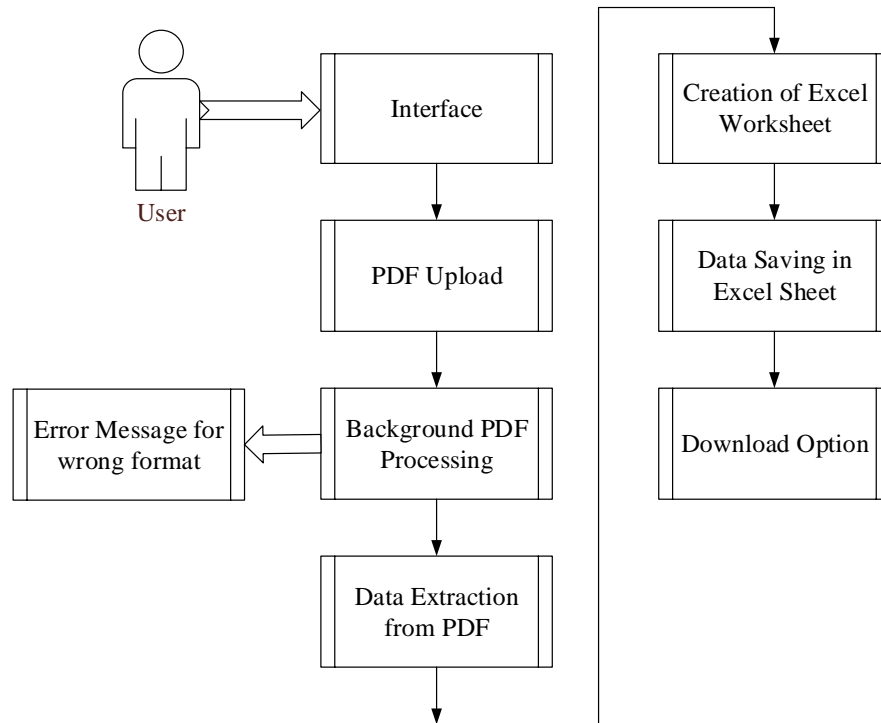


Fig. 5. Graphical abstract of the proposed system

### Functionality

**1. User Interface:** When users visit the application's main page, they encounter a form for uploading PDF files. The interface supports the simultaneous upload of multiple PDF files, streamlining the data extraction process. Users also select the relevant semester they wish to extract data before submitting the form.

**2. Text Extraction:** The application utilizes the PyPDF2 library to extract text from the PDF files. PyPDF2 reads each PDF file and extracts the text content from every page. The extracted text is then processed to identify and capture essential information, such as the academic year, enrollment number, student name, SGPA, and CGPA.

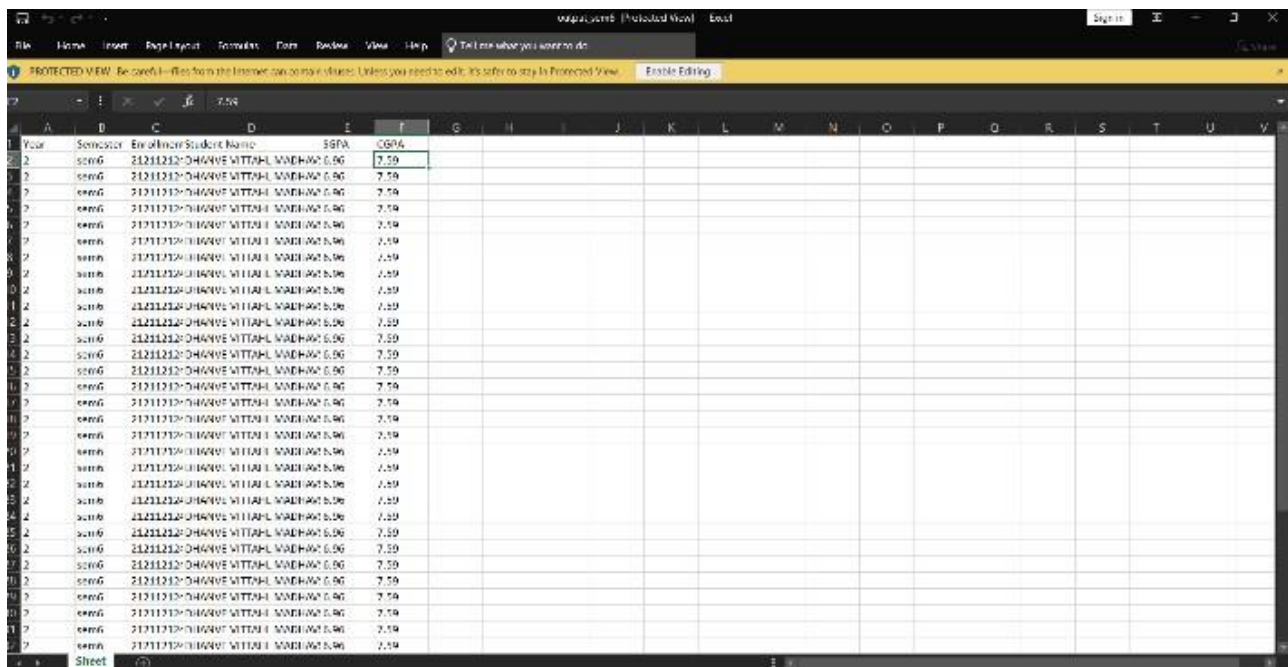
**3. Data Structuring:** Once the relevant data is extracted, it is organized into a structured format using the Openpyxl library. This library is responsible for creating and managing the Excel file. The application generates a new workbook and writes the extracted data into the appropriate cells, ensuring clarity and accuracy.

**4. File Management:** After processing, the application saves the generated Excel file in a designated folder on the server. Users can then download the file through a link on the results page.

### Advantages

- **Automation:** Significantly reduces the manual effort required for extracting and organizing data from PDF documents.
- **Efficiency:** Handles multiple PDF files simultaneously, saving users time and effort.
- **Accuracy:** Ensures precise data extraction and organization into an easily accessible Excel format.
- **User-Friendly:** Offers a simple and intuitive interface for uploading files and retrieving processed data.

In summary, the Data Extraction Web App automates the labor-intensive task of converting data from PDF documents into an organized Excel spreadsheet. It enhances efficiency, accuracy, and user convenience by providing a secure and reliable solution for managing academic records.



Year	Semester	Enrollment Number	Student Name	GPA	CGPA
2	sem6	212112121	CHANDRA MITTAL	6.86	7.59
2	sem6	212112122	CHANDRA MITTAL	6.86	7.59
2	sem6	212112123	CHANDRA MITTAL	6.86	7.59
2	sem6	212112124	CHANDRA MITTAL	6.86	7.59
2	sem6	212112125	CHANDRA MITTAL	6.86	7.59
2	sem6	212112126	CHANDRA MITTAL	6.86	7.59
2	sem6	212112127	CHANDRA MITTAL	6.86	7.59
2	sem6	212112128	CHANDRA MITTAL	6.86	7.59
2	sem6	212112129	CHANDRA MITTAL	6.86	7.59
2	sem6	212112130	CHANDRA MITTAL	6.86	7.59
2	sem6	212112131	CHANDRA MITTAL	6.86	7.59
2	sem6	212112132	CHANDRA MITTAL	6.86	7.59
2	sem6	212112133	CHANDRA MITTAL	6.86	7.59
2	sem6	212112134	CHANDRA MITTAL	6.86	7.59
2	sem6	212112135	CHANDRA MITTAL	6.86	7.59
2	sem6	212112136	CHANDRA MITTAL	6.86	7.59
2	sem6	212112137	CHANDRA MITTAL	6.86	7.59
2	sem6	212112138	CHANDRA MITTAL	6.86	7.59
2	sem6	212112139	CHANDRA MITTAL	6.86	7.59
2	sem6	212112140	CHANDRA MITTAL	6.86	7.59
2	sem6	212112141	CHANDRA MITTAL	6.86	7.59
2	sem6	212112142	CHANDRA MITTAL	6.86	7.59
2	sem6	212112143	CHANDRA MITTAL	6.86	7.59
2	sem6	212112144	CHANDRA MITTAL	6.86	7.59
2	sem6	212112145	CHANDRA MITTAL	6.86	7.59
2	sem6	212112146	CHANDRA MITTAL	6.86	7.59
2	sem6	212112147	CHANDRA MITTAL	6.86	7.59
2	sem6	212112148	CHANDRA MITTAL	6.86	7.59
2	sem6	212112149	CHANDRA MITTAL	6.86	7.59
2	sem6	212112150	CHANDRA MITTAL	6.86	7.59
2	sem6	212112151	CHANDRA MITTAL	6.86	7.59
2	sem6	212112152	CHANDRA MITTAL	6.86	7.59
2	sem6	212112153	CHANDRA MITTAL	6.86	7.59
2	sem6	212112154	CHANDRA MITTAL	6.86	7.59
2	sem6	212112155	CHANDRA MITTAL	6.86	7.59
2	sem6	212112156	CHANDRA MITTAL	6.86	7.59
2	sem6	212112157	CHANDRA MITTAL	6.86	7.59
2	sem6	212112158	CHANDRA MITTAL	6.86	7.59
2	sem6	212112159	CHANDRA MITTAL	6.86	7.59
2	sem6	212112160	CHANDRA MITTAL	6.86	7.59
2	sem6	212112161	CHANDRA MITTAL	6.86	7.59
2	sem6	212112162	CHANDRA MITTAL	6.86	7.59
2	sem6	212112163	CHANDRA MITTAL	6.86	7.59
2	sem6	212112164	CHANDRA MITTAL	6.86	7.59
2	sem6	212112165	CHANDRA MITTAL	6.86	7.59
2	sem6	212112166	CHANDRA MITTAL	6.86	7.59
2	sem6	212112167	CHANDRA MITTAL	6.86	7.59
2	sem6	212112168	CHANDRA MITTAL	6.86	7.59
2	sem6	212112169	CHANDRA MITTAL	6.86	7.59
2	sem6	212112170	CHANDRA MITTAL	6.86	7.59
2	sem6	212112171	CHANDRA MITTAL	6.86	7.59
2	sem6	212112172	CHANDRA MITTAL	6.86	7.59
2	sem6	212112173	CHANDRA MITTAL	6.86	7.59
2	sem6	212112174	CHANDRA MITTAL	6.86	7.59
2	sem6	212112175	CHANDRA MITTAL	6.86	7.59
2	sem6	212112176	CHANDRA MITTAL	6.86	7.59
2	sem6	212112177	CHANDRA MITTAL	6.86	7.59
2	sem6	212112178	CHANDRA MITTAL	6.86	7.59
2	sem6	212112179	CHANDRA MITTAL	6.86	7.59
2	sem6	212112180	CHANDRA MITTAL	6.86	7.59
2	sem6	212112181	CHANDRA MITTAL	6.86	7.59
2	sem6	212112182	CHANDRA MITTAL	6.86	7.59
2	sem6	212112183	CHANDRA MITTAL	6.86	7.59
2	sem6	212112184	CHANDRA MITTAL	6.86	7.59
2	sem6	212112185	CHANDRA MITTAL	6.86	7.59
2	sem6	212112186	CHANDRA MITTAL	6.86	7.59
2	sem6	212112187	CHANDRA MITTAL	6.86	7.59
2	sem6	212112188	CHANDRA MITTAL	6.86	7.59
2	sem6	212112189	CHANDRA MITTAL	6.86	7.59
2	sem6	212112190	CHANDRA MITTAL	6.86	7.59
2	sem6	212112191	CHANDRA MITTAL	6.86	7.59
2	sem6	212112192	CHANDRA MITTAL	6.86	7.59
2	sem6	212112193	CHANDRA MITTAL	6.86	7.59
2	sem6	212112194	CHANDRA MITTAL	6.86	7.59
2	sem6	212112195	CHANDRA MITTAL	6.86	7.59
2	sem6	212112196	CHANDRA MITTAL	6.86	7.59
2	sem6	212112197	CHANDRA MITTAL	6.86	7.59
2	sem6	212112198	CHANDRA MITTAL	6.86	7.59
2	sem6	212112199	CHANDRA MITTAL	6.86	7.59
2	sem6	212112200	CHANDRA MITTAL	6.86	7.59

Fig. 6. Excel data.

### 5. Conclusions

The manuscript describes the development of a Data Extraction web application designed to convert student report data into both a database and Excel format. Traditionally, data extraction involved cumbersome manual processes, but the proposed application automates this task effectively. By utilizing the PyPDF2 library for text extraction and Openpyxl for organizing data into Excel sheets, the application streamlines the conversion process.

The automated nature of the application significantly reduces the time and effort required for manual data extraction and entry. It ensures secure handling of files and supports the batch processing of multiple documents simultaneously. This makes the application a valuable tool for efficiently converting PDF content into well-structured Excel data, enhancing productivity and accuracy for users managing academic records.



## References

- [1] F. T. Schreiber, S. Burkhardt, and T. Kämpke, "Information extraction from PDF documents," International Conference on Information Technology: Coding and Computing, Las Vegas, NV, USA, 2004, pp. 140-145, doi: 10.1109/ITCC.2004.1286621.
- [2] A. N. Langville and C. D. Meyer, "Extracting information from PDF files," Information Retrieval, vol. 8, no. 2, pp. 235-250, 2005, doi: 10.1007/s10791-005-2247-6.
- [3] H. Nakagawa and T. Mori, "A study on extraction of text information from PDF files," Fifth International Conference on Document Analysis and Recognition, ICDAR 1999, Bangalore, India, 1999, pp. 237-240, doi: 10.1109/ICDAR.1999.791796.
- [4] S. Jiang and Y. Li, "Research and Implementation of PDF Specific Element Fast Extraction," 2023 4th International Conference on Big Data & Artificial Intelligence and Software Engineering (ICBASE), Nanjing, China, 2023, pp. 77-83, doi: 10.1109/ICBASE59196.2023.10303081.
- [5] F. Grijalva, E. Santos, B. Acuña, J. C. Rodríguez and J. C. Larco, "Deep Learning in Time-Frequency Domain for Document Layout Analysis," IEEE Access, vol. 9, pp. 151254-151265, 2021, doi: 10.1109/ACCESS.2021.3125913.
- [6] M. R. L. Smyth and D. P. O'Donoghue, "A tool for converting PDF to XML," Proceedings of the 2004 ACM Symposium on Document Engineering, Milwaukee, Wisconsin, USA, 2004, pp. 123-125, doi: 10.1145/1030397.1030430.
- [7] P. Deshpande and B. Iyer, "Research directions in the Internet of Every Things(IoET)," 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 2017, pp. 1353-1357, doi: 10.1109/CCAA.2017.8230008.