PARSHWANATH CHARITABLE TRUST'S

A.P. SHAH INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering

Data Science

ScanWise

24207010 – Aayush Balip
24207001 – Ashokkumar Bhati
24207014 – Sameer Saiyed
24207020 – Keval Shah

Project Guide
Ms. Sarala Mary

Outline

- Introduction
- Literature Survey of the existing systems
- Limitations of the existing systems
- Problem statement
- System Design
- Technologies and methodologies
- Implementation
- Conclusion
- References

Introduction

- In most universities, exam results are published as PDF documents or Excel sheets containing student details, paper codes, marks, and result status.

- These PDFs are unstructured, making it difficult for students and institutions to quickly analyze results.

- Current practice: Manual checking, downloading, and searching — a time-consuming and error-prone process.

- While some digital result portals exist, they often lack:
  - Structured data export
  - Visualization of each student's batch

Introduction

➢ **Objectives:**

- Automate the extraction of student results from unstructured PDFs using the PyPDF library and store the data in Excel using pandas.

- Reduce manual effort and minimize the chances of data entry errors.

- Extract and restructure data from Excel sheets using openpyxl, and apply formatting with openpyxl.fonts for better readability.

- Visualize the processed data from each Excel sheet using matplotlib for clearer insights.

Literature Survey of the existing system

Sr. No.	Title	Author(s)	Year	Outcomes	Methodology	Result
1	Extraction of User-Defined Information from PDF	R. Nadeem, T. Iqbal, N. Fatima, J. Altaf, A. Irshad and A. Farooq	2024	User-defined information extracted from PDFs using PyMuPDF library for automated data retrieval from unstructured documents.	PyMuPDF text extraction techniques applied to PDFs, utilizing page.get_text() methods for native text and OCR-based extraction.	Successful extraction of textual data from PDF documents with high accuracy using PyMuPDF's hybrid native and OCR approaches.
2	Python Data Analysis: Perform data collection, data processing, wrangling, visualization, and model building using Python	Avinash Navlani; Armando Fandango; Ivan Idris	2021	Comprehensive Python data analysis framework covering collection, manipulation, wrangling, visualization, and model building using Pandas and Matplotlib.	Pandas DataFrames for data structures, manipulation operations like head(), tail(), slicing, and integration with visualization libraries.	Pandas DataFrames for data structures, manipulation operations like head(), tail(), slicing, and integration with visualization libraries

Literature Survey of the existing system

Sr. No.	Title	Author(s)	Year	Outcomes	Methodology	Result
3	Automating Data Analysis with Python: A Comparative Study of Popular Libraries and their Application	P. Bhardwaj, C. Choudhury and P. Batra	2024	Comparative evaluation of Python libraries including Pandas, Matplotlib, Seaborn, Scikit-learn for automated data analysis task performance.	Performance testing and comparison of popular Python libraries across various data analysis tasks including manipulation, visualization, and modeling.	Performance testing and comparison of popular Python libraries across various data analysis tasks including manipulation, visualization, and modeling.
4	Programming language Python for data processing	Z. Dobesova	2021	Python established as effective programming language for data processing automation in research and academic contexts with scripting capabilities.	Used matplotlib and seaborn in Python to generate bar charts and histograms representing pass/fail courts and average performance.	Successful implementation of automated data processing workflows using Python, reducing manual intervention and improving processing speed..

Limitations of existing systems

- **Lack of Automation in PDF Extraction:** Raw university mark sheets in PDF format often need to be manually converted into Excel, as many systems lack reliable PDF parsing and data normalization features.

- **Limited Cross-Semester Integration:** Existing systems usually process one semester at a time without automatically merging and averaging results across multiple semesters.

- **Absence of Visual Analytics:** Many tools fail to provide visual insights such as pass/fail distribution charts or performance trends, which are valuable for faculty review and academic reports.

Limitations of existing systems

- **Inconsistent Data Formats:** Variation in result templates and file headers causes difficulties in data cleaning, standardization, and automation.

- **No Intelligent Highlighting or Error Detection:** Systems often miss features like automatic detection and color highlighting of failed students or invalid entries.

Problem statement

- **High Inefficiency and Error Rate:** The current process of manually checking, downloading, and entering data from result PDFs is extremely slow and prone to human error, leading to inaccurate records and wasted administrative hours.

- **Lack of Immediate Analytics:** Students and institutions have no way to quickly generate valuable insights, such as identifying subject toppers, calculating pass/fail percentages across different papers, or tracking individual academic progress over time.

- **No Visualization of Data:** Institutions do not have any visual data to get to know about student's record and help in understanding the performance of each student batch.

System Design

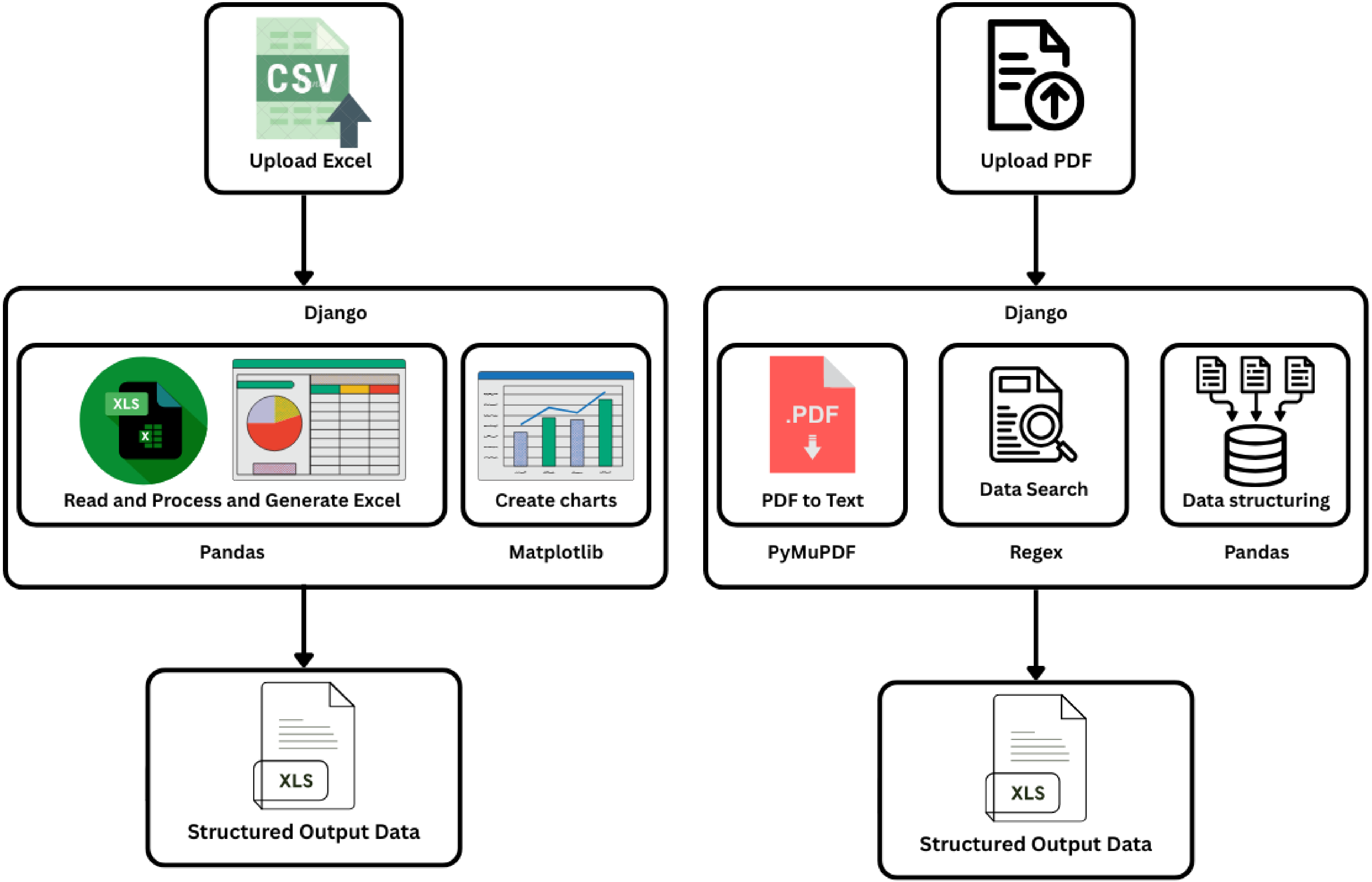


Fig 1: System Design for Hybrid Model

Technologies

- **PDF to Text:** PyPDF2
- **Reading and Generating Excel:** Pandas
- **Styling Excel:** openpyxl
- **Creating Charts:** Matplotlib
- **Frontend Development:**
  1. ReactJS
  2. Tailwind CSS
- **Backend Development:**
  1. Django 4.2.5
- **Hosting:**
  1. Vercel
  2. Render

Methodology

1. **Extraction and Reading the Data:**

   1. PDF: Using PyPDF, the PDF files are extracted, and text data is retrieved.

   2. Excel: Using openpyxl, Excel sheets are opened and read.

2. Processing the data:

   1. PDF: After extracting the text from PDFs, regex operations are applied to process and extract specific data.

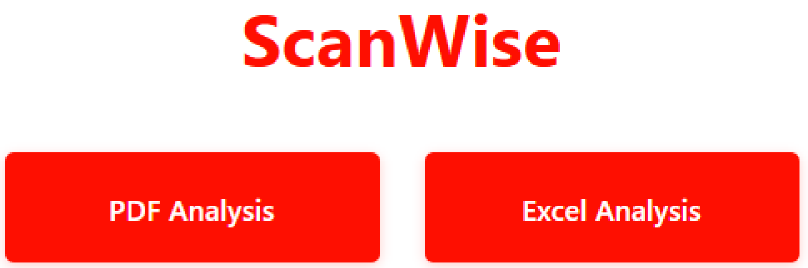
   2. Excel: After reading the Excel sheet, pandas is used to access specific columns and perform the desired operations.

3. Result:

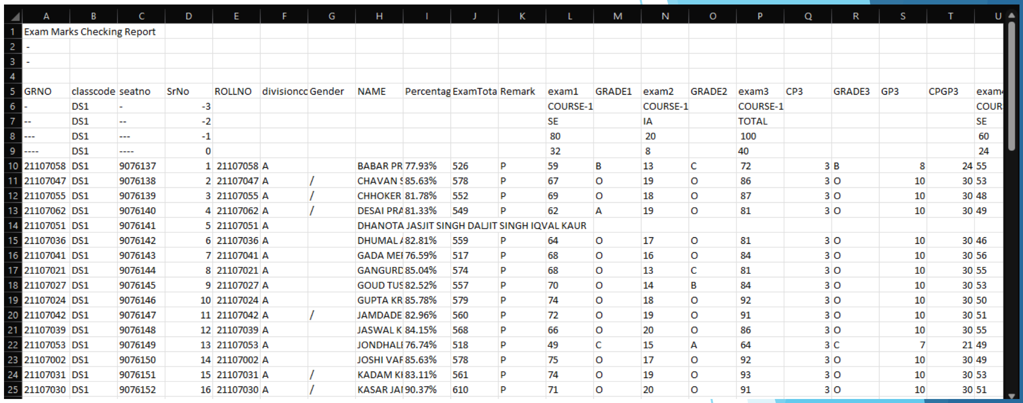
   1. PDF: The processed data is stored in Excel sheets using pandas.

   2, Excel: The final results are saved to Excel sheets, and visualizations are created using matplotlib.

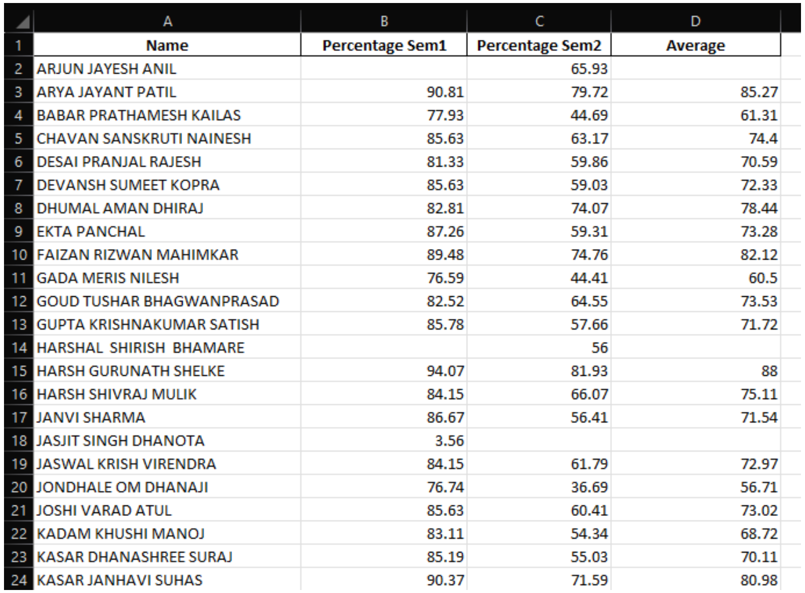
Implementation



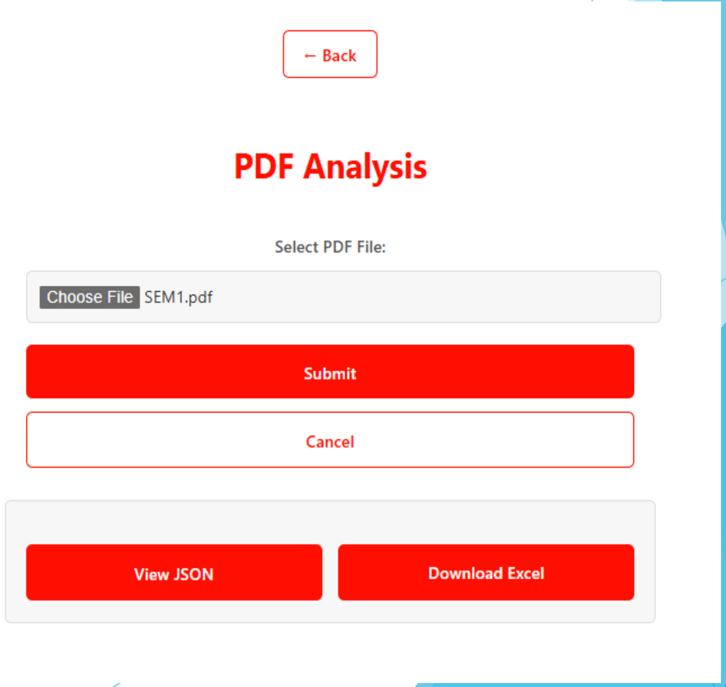
ScanWise



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Exam Marks Checking Report																				
2	-																				
3	-																				
4																					
5	GRNO	classcode	seatno	SrNo	ROLLNO	divisioncc	Gender	NAME	Percentag	ExamTota	Remark	exam1	GRADE1	exam2	GRADE2	exam3	CP3	GRADE3	GP3	CPGP3	exam
6	-	DS1	-	-3								COURSE-1		COURSE-1		COURSE-1					COUR
7	--	DS1	--	-2								SE		IA		TOTAL					SE
8	---	DS1	---	-1								80		20		100					60
9	----	DS1	----	0								32		8		40					24
10	21107058	DS1	9076137	1	21107058	A		BABAR PR	77.93%	526	P	59	B	13	C	72	3	B	8	24	55
11	21107047	DS1	9076138	2	21107047	A	/	CHAVAN S	85.63%	578	P	67	O	19	O	86	3	O	10	30	53
12	21107055	DS1	9076139	3	21107055	A	/	CHHOKER	81.78%	552	P	69	O	18	O	87	3	O	10	30	48
13	21107062	DS1	9076140	4	21107062	A	/	DESAI PRA	81.33%	549	P	62	A	19	O	81	3	O	10	30	49
14	21107051	DS1	9076141	5	21107051	A		DHANOTA JASJIT SINGH DALJIT SINGH IQVAL KAUR													
15	21107036	DS1	9076142	6	21107036	A		DHUMAL A	82.81%	559	P	64	O	17	O	81	3	O	10	30	46
16	21107041	DS1	9076143	7	21107041	A		GADA MEI	76.59%	517	P	68	O	16	O	84	3	O	10	30	56
17	21107021	DS1	9076144	8	21107021	A		GANGURD	85.04%	574	P	68	O	13	C	81	3	O	10	30	55
18	21107027	DS1	9076145	9	21107027	A		GOUD TUS	82.52%	557	P	70	O	14	B	84	3	O	10	30	53
19	21107024	DS1	9076146	10	21107024	A		GUPTA KR	85.78%	579	P	74	O	18	O	92	3	O	10	30	50
20	21107042	DS1	9076147	11	21107042	A	/	JAMDADE	82.96%	560	P	72	O	19	O	91	3	O	10	30	51
21	21107039	DS1	9076148	12	21107039	A		JASWAL K	84.15%	568	P	66	O	20	O	86	3	O	10	30	55
22	21107053	DS1	9076149	13	21107053	A		JONDHALI	76.74%	518	P	49	C	15	A	64	3	C	7	21	49
23	21107002	DS1	9076150	14	21107002	A		JOSHI VAF	85.63%	578	P	75	O	17	O	92	3	O	10	30	49
24	21107031	DS1	9076151	15	21107031	A	/	KADAM KI	83.11%	561	P	74	O	19	O	93	3	O	10	30	53
25	21107030	DS1	9076152	16	21107030	A	/	KASAR JAI	90.37%	610	P	71	O	20	O	91	3	O	10	30	51



	A	B	C	D
1	Name	Percentage Sem1	Percentage Sem2	Average
2	ARJUN JAYESH ANIL		65.93	
3	ARYA JAYANT PATIL	90.81	79.72	85.27
4	BABAR PRATHAMESH KAILAS	77.93	44.69	61.31
5	CHAVAN SANSKRUTI NAINESH	85.63	63.17	74.4
6	DESAI PRANJAL RAJESH	81.33	59.86	70.59
7	DEVANSH SUMEET KOPRA	85.63	59.03	72.33
8	DHUMAL AMAN DHIRAJ	82.81	74.07	78.44
9	EKTA PANCHAL	87.26	59.31	73.28
10	FAIZAN RIZWAN MAHIMKAR	89.48	74.76	82.12
11	GADA MERIS NILESH	76.59	44.41	60.5
12	GOUD TUSHAR BHAGWANPRASAD	82.52	64.55	73.53
13	GUPTA KRISHNAKUMAR SATISH	85.78	57.66	71.72
14	HARSHAL SHIRISH BHAMARE		56	
15	HARSH GURUNATH SHELKE	94.07	81.93	88
16	HARSH SHIVRAJ MULIK	84.15	66.07	75.11
17	JANVI SHARMA	86.67	56.41	71.54
18	JASJIT SINGH DHANOTA	3.56		
19	JASWAL KRISH VIRENDRA	84.15	61.79	72.97
20	JONDHALE OM DHANAJI	76.74	36.69	56.71
21	JOSHI VARAD ATUL	85.63	60.41	73.02
22	KADAM KHUSHI MANOJ	83.11	54.34	68.72
23	KASAR DHANASHREE SURAJ	85.19	55.03	70.11
24	KASAR JANHAVI SUHAS	90.37	71.59	80.98



PDF Analysis

Select PDF File:

Conclusion

- Developed an automated system to extract, process, and visualize student exam results from PDFs and Excel files.
- Utilized **PyPDF2**, **pandas**, **openpyxl**, and **matplotlib** to eliminate manual errors and inefficiencies.
- Implemented a **ReactJS + Tailwind CSS frontend** with a **Django backend**, hosted on **Vercel** and **Render** for scalability.
- Provides quick access to structured data, performance metrics, and visual analytics.
- Significantly reduces administrative workload, enhances data accuracy, and supports informed academic decision-making.

References

[1] R. Nadeem, T. Iqbal, N. Fatima, J. Altaf, A. Irshad and A. Farooq, "Extraction of User-Defined Information from PDF," doi: 10.1109/DASA63652.2024.10836169.

[2] Avinash Navlani; Armando Fandango; Ivan Idris, Python Data Analysis: Perform data collection, data processing, wrangling, visualization, and model building using Python , Packt Publishing, 2021. http://ieeexplore.ieee.org/document/10163464

[3] P. Bhardwaj, C. Choudhury and P. Batra, "Automating Data Analysis with Python: A Comparative Study of Popular Libraries and their Application," doi: 10.1109/ICTACS59847.2023.10390032.

[4] Z. Dobesova, "Programming language Python for data processing," doi: 10.1109/ICECENG.2011.6057428.

Thank You...!!