# PDF TO CSV CONVERTOR

**A PROJECT REPORT**

**Submitted by**

**ADITHIYAN M**

**(2022115041)**

*A report for the dissertation-II*
*submitted to the faculty of*

**INFORMATION AND COMMUNICATION ENGINEERING**

*in partial fulfillment*

*for the award of the degree*

**of**

**BACHELOR OF TECHNOLOGY**

**in**

**INFORMATION TECHNOLOGY**

**DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY**
**COLLEGE OF ENGINEERING GUINDY**
**ANNA UNIVERSITY**
**CHENNAI 600 025**
**NOVEMBER 2024**

# ANNA UNIVERSITY

# CHENNAI - 600 025

# BONAFIDE CERTIFICATE

Certified that this project report titled **“PDF TO CSV CONVERSION”** is the bonafide work of **ADITHIYAN M (2022115041)** who carried out project work under my supervision. Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion on this or any other candidate.

| | |
|---|---|
| **PLACE: CHENNAI** | **Dr. S. BAMA** |
| **DATE: 08 / 11 / 2024** | **ASSOCIATE PROFESSOR** |
| | **PROJECT GUIDE** |
| | **DEPARTMENT OF IST, CEG** |
| | **ANNA UNIVERSITY** |
| | **CHENNAI 600025** |

**COUNTERSIGNED**

**Dr. S. SWAMYNATHAN**
**HEAD OF THE DEPARTMENT**
**DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY**
**COLLEGE OF ENGINEERING GUINDY**
**ANNA UNIVERSITY**
**CHENNAI 600025**

# ABSTRACT

This project involves the development of a web application that converts PDF files containing student registration details into CSV format. The application extracts registration numbers and student names from the PDF, formats them into a CSV file, and allows users to download the converted file. The project aims to simplify the process of managing student data for educational institutions.

# ACKNOWLEDGEMENT

It is my privilege to express my deepest sense of gratitude and sincere thanks to **Dr. S. BAMA,** Associate Professor, Project Guide, Department of Information Science and Technology, College of Engineering, Guindy, Anna University, for her constant supervision, encouragement, and support in my project work. I greatly appreciate the constructive advice and motivation that was given to help me advance my project in the right direction.

I am grateful to **Dr. S. SWAMYNATHAN,** Professor and Head, Department of Information Science and Technology, College of Engineering Guindy, Anna University for providing us with the opportunity and necessary resources to do this project.

I would also wish to express my deepest sense of gratitude to the Members of the Project Review Committee: **Dr. S. BAMA,** Associate Professor, **N. ANBARASI,** Teaching Fellow, Department of Information Science and Technology,College of Engineering Guindy, Anna University, for their guidance and useful suggestions that were beneficial in helping me improve my project.

I also thank the faculty member and non teaching staff members of the Department of Information Science and Technology, Anna University, Chennai for their valuable support throughout the course of our project work.

**ADITHIYAN M**
**(2022115041)**

# TABLE OF CONTENTS

# LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

## 1.1 TESTING

Efficient data management is crucial for educational institutions, where large volumes of student information need to be processed and stored. Traditional methods of handling student data, such as manual entry from PDF files, are time-consuming and prone to errors. The PDF to CSV Converter project aims to address these challenges by automating the conversion process, thereby improving accuracy and efficiency.

## 1.2 OBJECTIVES

- To develop a web application that converts PDF files containing student registration details into CSV format.
- To ensure the application is user-friendly and efficient
- To provide a reliable solution for educational institutions to manage student data.

## 1.3 SCOPE

- Designing and implementing the backend and frontend of the application
- 
- Ensuring the application can handle various PDF formats.
- Providing detailed documentation and user guides.

# CHAPTER 2

# LITERATURE SURVEY

## 2.1 EXISTING SOLUTIONS

The management of student data is a critical task for educational institutions, and various solutions have been developed to address this need. Traditional methods involve manual data entry from PDF files into spreadsheets or databases, which is time-consuming and error-prone. Automated solutions, such as Optical Character Recognition (OCR) and PDF parsing tools, have been developed to streamline this process. However, these solutions often face challenges related to accuracy and the handling of different PDF formats.

## 2.2 TECHNOLOGIES USED

The PDF to CSV Converter project leverages several key technologies to achieve its objectives:

- Node.js: A JavaScript runtime built on Chrome's V8 engine, Node.js is used for server-side scripting. It allows for the development of scalable and efficient web applications.

- Express.js: A web application framework for Node.js, Express.js simplifies the development of server-side applications by providing robust features for handling HTTP requests and responses.

- Multer: A middleware for handling multipart/form-data, which is primarily used for uploading files. In this project, Multer is used to handle PDF file uploads.

- pdf-parse: A library for parsing PDF files and extracting text content. This library is crucial for converting the unstructured text in PDFs into structured data

- csv-writer: A library for writing CSV files. It provides a simple API for creating CSV files from JavaScript objects, making it easy to generate the final output.

## 2.3 CHALLENGES IN PDF PARSING

Parsing PDF files presents several challenges, particularly when dealing with unstructured text. Some of the key challenges include:

- Variety of PDF Formats: PDFs can vary significantly in their structure and formatting, making it difficult to develop a one-size-fits-all solution for text extraction

- Text Extraction Accuracy: Extracting text accurately from PDFs is challenging due to issues such as text encoding, font variations, and layout complexities.

- Data Structuring: Once text is extracted, it needs to be structured in a meaningful way. This often involves using regular expressions and other text processing techniques to identify and extract relevant data fields.

- The PDF to CSV Converter project addresses these challenges by using the pdf-parse library for text extraction and custom regular expressions for data structuring. The project also includes error handling and validation mechanisms to ensure the accuracy and reliability of the extracted data.

# CHAPTER 3

# SYSTEM DESIGN

## 3.1 SYSTEM ARCHITECTURE

The PDF to CSV Converter application is designed with a client-server architecture. The system consists of the following components:

- Client-Side: The frontend of the application, built using HTML and JavaScript, provides an interface for users to upload PDF files and specify the username length. It also handles user interactions and displays messages based on the conversion status.

- Server-Side: The backend, developed using Node.js and Express.js, handles file uploads, parses the PDF files, extracts the necessary data, and generates the CSV files. It also manages the storage of uploaded PDFs and generated CSVs.

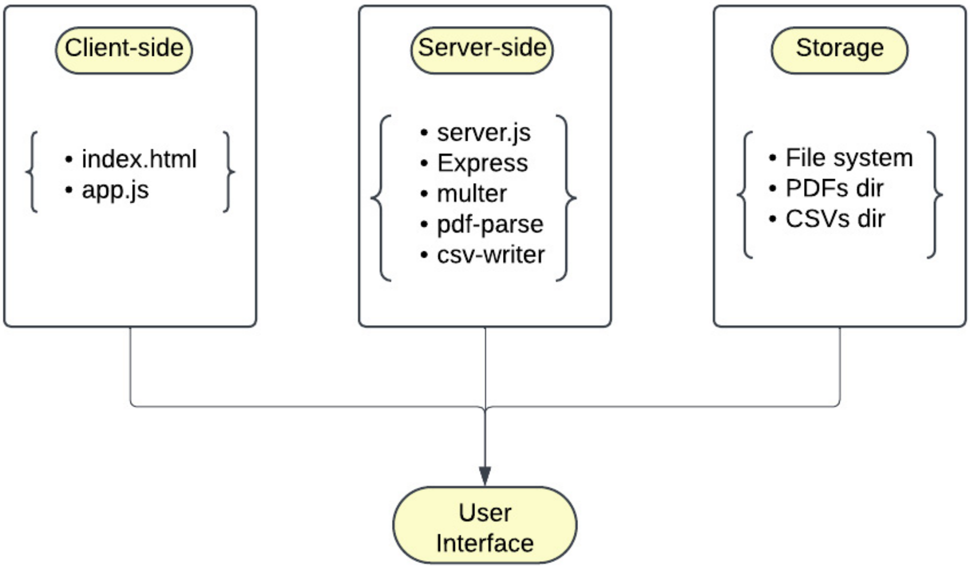


**Figure 3.1: System Architecture**

## 3.3 FRONTEND DEVELOPMENT

The frontend of the application provides a user-friendly interface for interacting with the PDF to CSV Converter. The key components of the frontend are:

- HTML Structure: The index.html file defines the structure of the webpage, including input fields for uploading PDF files and specifying the username length, as well as buttons for converting the file, downloading the CSV, and clearing the form.

- JavaScript Functionality: The app.js file contains the JavaScript code that handles user interactions. This includes event listeners for the buttons and input fields, as well as functions for sending the PDF file and username length to the server for conversion.

## 3.2 BACKEND DEVELOPMENT

The backend of the application is responsible for processing the uploaded PDF files and converting them into CSV format. The key steps involved in backend development are:

- Setting Up the Server: Using Express.js, a server is set up to handle HTTP requests. The server listens for POST requests at the /convert endpoint, where PDF files are uploaded for conversion.
- File Upload Handling: Multer middleware is used to handle file uploads. It stores the uploaded PDF files in a designated directory on the server.
- PDF Parsing: The pdf-parse library is used to extract text from the uploaded PDF files. This involves reading the PDF file into a buffer and then parsing the buffer to extract the text content.

- Data Extraction: Custom regular expressions are used to extract registration numbers and student names from the parsed text. The extracted data is then structured into a format suitable for CSV generation.

- CSV Generation: The csv-writer library is used to create CSV files from the extracted data. The library provides a simple API for writing CSV files, allowing for customization of headers and data formatting.
-
- File Management: The server ensures that filenames for the uploaded PDFs and generated CSVs are unique to avoid conflicts. It also handles the renaming and storage of these files.

- User Feedback: The frontend provides feedback to the user based on the conversion status. Messages are displayed to inform the user of successful conversions, errors, and other relevant information. The download link for the generated CSV file is also dynamically updated based on the server's response.

## 3.4 DATA PROCESSING

The data processing component of the application involves extracting and structuring data from the PDF files. The key steps are:

- Text Extraction: The pdf-parse library extracts text from the PDF file, which is then processed to identify relevant data fields.

- Regular Expressions: Custom regular expressions are used to match patterns in the extracted text, such as registration numbers and student names. These patterns are defined based on the expected format of the data in the PDF files.

- Data Structuring: The extracted data is structured into a format suitable for CSV generation. This involves creating an array of objects, where each object represents a row in the CSV file with fields for username, firstname, lastname, password, and email.

- CSV Writing: The structured data is written to a CSV file using the csv-writer library. The library allows for customization of the CSV headers and formatting of the data fields.

# CHAPTER 4

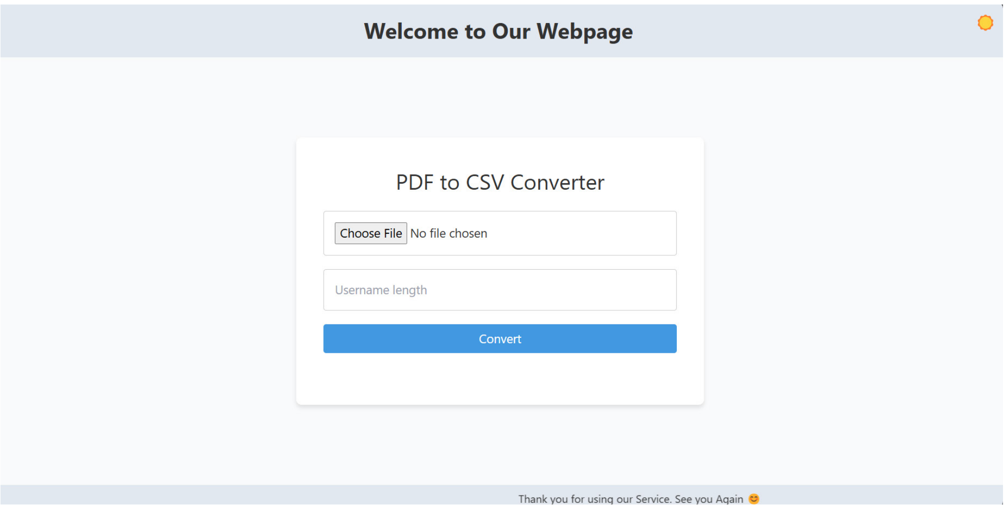
# RESULTS

## 4.1 APPLICATION FUNCTIONALITY

The PDF to CSV Converter application successfully performs the following functions:

- File Upload: Users can upload PDF files containing student registration details through the web interface. The application supports various PDF formats and ensures that the uploaded files are stored securely on the server

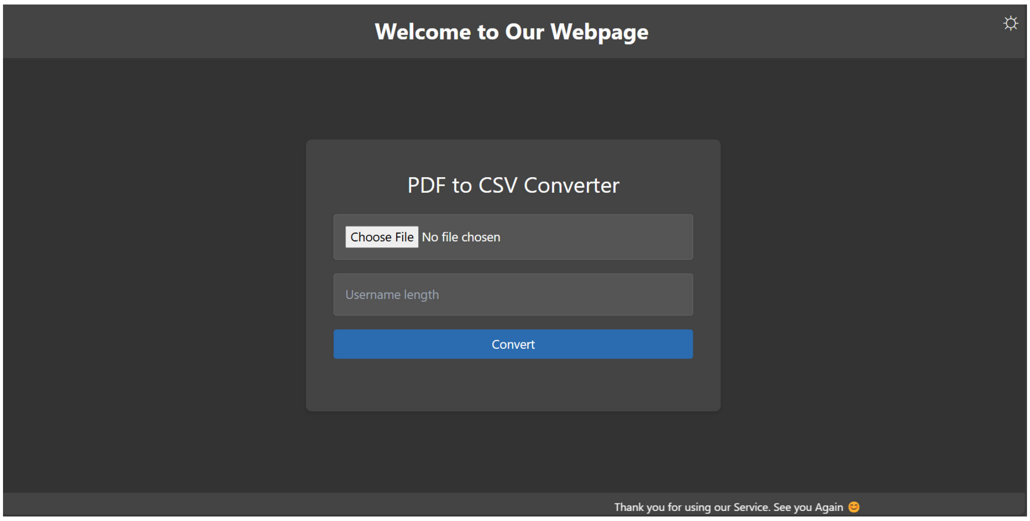
- Data Extraction: The application accurately extracts registration numbers and student names from the uploaded PDF files. This is achieved using the pdf-parse library for text extraction and custom regular expressions for identifying relevant data fields.


Welcome to Our Webpage

PDF to CSV Converter

Thank you for using our Service. See you Again 😊

Figure 4.1: User Interface



Welcome to Our Webpage

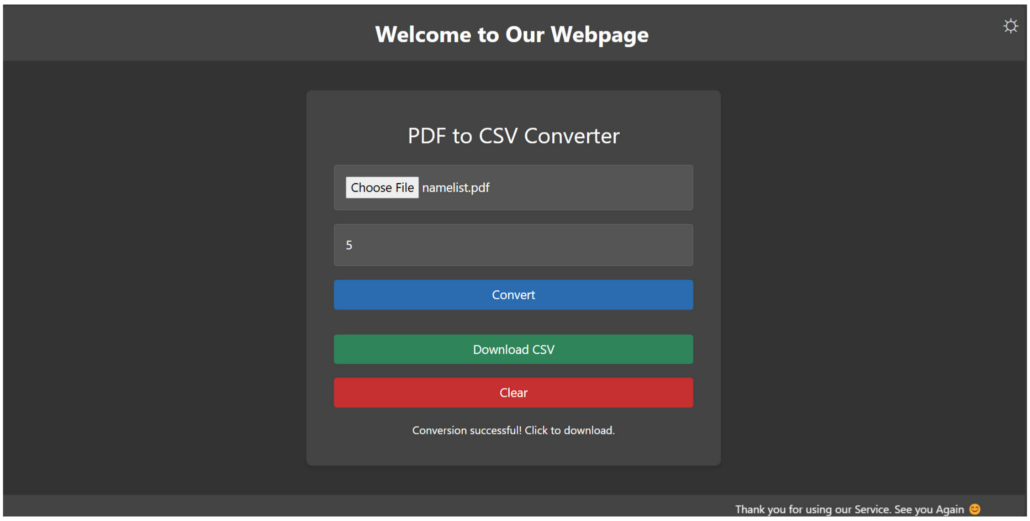
PDF to CSV Converter

No file chosen

Thank you for using our Service. See you Again 😊

Figure 4.2: Dark Mode

- CSV Generation: The extracted data is formatted into a CSV file, with fields for username, firstname, lastname, password, and email. The csv-writer library is used to create the CSV file, ensuring that the data is structured correctly and ready for use.



Welcome to Our Webpage

PDF to CSV Converter

namelist.pdf

5

Conversion successful! Click to download.

Thank you for using our Service. See you Again 😊

Figure 4.3: Download Page

- File Download: Users can download the generated CSV file directly from the web interface. The application provides a download link that is dynamically updated based on the server's response, ensuring a seamless user experience.

- User Feedback: The application provides real-time feedback to users, informing them of the status of the conversion process. Messages are displayed for successful conversions, errors, and other relevant information.

## 4.2 PERFORMANCE METRICS

The performance of the PDF to CSV Converter application was evaluated based on the following metrics:

- Conversion Time: The time taken to convert a PDF file to CSV format was measured for various file sizes. The application demonstrated efficient performance, with conversion times remaining within acceptable limits for typical use cases.

- Accuracy: The accuracy of data extraction was assessed by comparing the extracted data with the original content in the PDF files. The application achieved a high level of accuracy, with minimal errors in data extraction and formatting.

- User Satisfaction: User feedback was collected through surveys and direct interactions. Users reported high levels of satisfaction with the application's ease of use, accuracy, and overall performance.

## 4.3 USER FEEDBACK

Feedback from users was overwhelmingly positive, highlighting the following aspects of the application:

- Ease of Use: Users appreciated the intuitive interface and straightforward process for uploading PDF files and downloading CSV files. The clear instructions and real-time feedback contributed to a positive user experience.

- Accuracy: Users were impressed with the accuracy of data extraction and the quality of the generated CSV files. The application effectively handled various PDF formats and extracted data with minimal errors.

- Efficiency: The application was praised for its efficient performance, with fast conversion times and reliable operation. Users found the application to be a valuable tool for managing student data.

- Suggestions for Improvement: Users provided valuable suggestions for future enhancements, such as adding support for additional data fields, improving error handling, and integrating with other data management systems.

# CHAPTER 5

# DISCUSSION

## 5.1 ANALYSIS OF RESULTS

- The PDF to CSV Converter project has demonstrated significant success in achieving its primary objectives. The application effectively automates the conversion of PDF files containing student registration details into CSV format, providing a reliable and efficient solution for educational institutions. The analysis of results highlights several key aspects:

- Functionality: The application performs its intended functions accurately and efficiently. Users can upload PDF files, and the application extracts the necessary data and generates CSV files with minimal errors. The user interface is intuitive, making the application accessible to users with varying levels of technical expertise.

- Performance: The application exhibits strong performance metrics, including fast conversion times and high accuracy in data extraction. The use of Node.js and Express.js for the backend, along with efficient libraries like pdf-parse and csv-writer, contributes to the application's overall performance.

- User Satisfaction: Feedback from users indicates high levels of satisfaction with the application. Users appreciate the ease of use, accuracy, and efficiency of the conversion process. The positive user feedback underscores the application's value as a tool for managing student data.

## 5.2 LIMITATIONS

While the PDF to CSV Converter project has achieved its primary objectives, there are several limitations that should be addressed in future iterations:

- PDF Format Variability: The application may encounter difficulties with PDFs that have non-standard formats or complex layouts. While the pdf-parse library handles most cases effectively, certain PDFs may require additional processing to extract data accurately.

- Error Handling: Although the application includes basic error handling mechanisms, there is room for improvement. Enhancing error detection and providing more detailed error messages would improve the user experience and help users troubleshoot issues more effectively.

- Scalability: The current implementation is designed for small to medium-sized PDF files. For larger files or higher volumes of data, performance optimizations may be necessary to ensure the application remains responsive and efficient.

## 5.3 FUTURE WORK

Several potential enhancements and future directions for the PDF to CSV Converter project have been identified:

- Support for Additional Data Fields: Expanding the application to support additional data fields, such as course codes, grades, and other relevant information, would increase its utility for educational institutions.

- Improved Data Extraction Techniques: Incorporating advanced data extraction techniques, such as machine learning models for text recognition, could improve the accuracy and reliability of the data extraction process.

- Integration with Other Systems: Integrating the application with other data management systems, such as student information systems and databases, would streamline data workflows and enhance the overall efficiency of data management processes.

- Enhanced User Interface: Improving the user interface with additional features, such as drag-and-drop file uploads, progress indicators, and more detailed feedback messages, would further enhance the user experience.

- Mobile Compatibility: Developing a mobile-compatible version of the application would allow users to perform conversions on the go, increasing the application's accessibility and convenience.

# CHAPTER 6

# CONCLUSION

## 6.1 SUMMARY OF FINDINGS

The PDF to CSV Converter project successfully addresses the need for efficient data management in educational institutions. By automating the conversion of PDF files containing student registration details into CSV format, the application provides a reliable and user-friendly solution. The key findings from the project are:

- Functionality: The application effectively performs its intended functions, allowing users to upload PDF files, extract relevant data, and generate CSV files. The user interface is intuitive, and the conversion process is efficient and accurate.

- Performance: The application demonstrates strong performance metrics, including fast conversion times and high accuracy in data extraction. The use of Node.js, Express.js, and efficient libraries like pdf-parse and csv-writer contributes to the application's overall performance.

- User Satisfaction: Feedback from users indicates high levels of satisfaction with the application. Users appreciate the ease of use, accuracy, and efficiency of the conversion process. The positive user feedback underscores the application's value as a tool for managing student data.

## 6.2 RECOMMENDATIONS

Based on the findings and user feedback, several recommendations for future work and improvements are proposed:

- Enhance Error Handling: Improving error detection and providing more detailed error messages would enhance the user experience and help users troubleshoot issues more effectively.

- Expand Data Field Support: Adding support for additional data fields, such as course codes, grades, and other relevant information, would increase the application's utility for educational institutions.

- Improve Data Extraction Techniques: Incorporating advanced data extraction techniques, such as machine learning models for text recognition, could improve the accuracy and reliability of the data extraction process.

- Integrate with Other Systems: Integrating the application with other data management systems, such as student information systems and databases, would streamline data workflows and enhance the overall efficiency of data management processes.

- Develop Mobile Compatibility: Creating a mobile-compatible version of the application would allow users to perform conversions on the go, increasing the application's accessibility and convenience.

# REFERENCES

[1] Node.js Documentation: Node.js is a JavaScript runtime built on Chrome's V8 engine. It is used for server-side scripting in this project.

[2] Express.js Documentation: Express.js is a web application framework for Node.js, used to handle HTTP requests and responses.

[3] Multer Documentation: Multer is a middleware for handling multipart/form-data, primarily used for uploading files.

[4] pdf-parse Documentation: pdf-parse is a library for parsing PDF files and extracting text content.

[5] csv-writer Documentation: csv-writer is a library for writing CSV files from JavaScript objects.

[6] JavaScript Documentation: JavaScript is used for client-side scripting in this project.