TRACK MY FEE

Submitted by

SNEHA B	**(714023104130)**
THARUN KARTHIK G	**(714023104150)**
THIRUMURUGAN G	**(714023104152)**
VEERAMADHUMITHA P	**(714023104159)**
VISHNU VARDHAN A	**(714023104173)**

In partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

SRI SHAKTHI

INSTITUTE OF ENGINEERING AND TECHNOLOGY,

An Autonomous Institution,

Accredited by NAAC with “A” Grade

NOVEMBER 2025

TRACK MY FEE

Certified that this report titled **"TRACK MY FEE"** is the bonafide work of **SNEHA** (714023104130)**, THARUN KARTHIK(**714023104150**),** **THIRUMURUGAN(**714023104152)**,VEERAMADHUMITHA**(714023104159)**,** **VISHNU VARDHAN**(714023104173)
who carried out the work under my supervision.

SIGNATURE	**SIGNATURE**
Ms.R.Kalaiarasi	Dr.K.E.Kannammal
SUPERVISOR	**HEAD OF THE DEPARTMENT**
Assistant Professor,	Professor and Head,
Department of CSE,	Department of CSE,
Sri Shakthi Institute of	Sri Shakthi Institute of
Engineering and Technology,	Engineering and Technology,
Coimbatore – 641062.	Coimbatore – 641062.

Submitted for the project work viva-voce Examination held on……………….

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

First and foremost, I would like to thank God Almighty for giving me thestrength. Without his blessings this achievement would not have been possible.We express our deepest gratitude to our Chairman **Dr.S.Thangavelu** for his continuous encouragement and support throughout our course of study.We are thankful to our Secretary **Mr.T.Dheepan** for his unwavering support during the entire course of this project work.We are also thankful to our Joint Secretary **Mr.T.Sheelan** for his supportduring the entire course of this project work.We are highly indebted to Principal **Dr.N.K.Sakthivel** for his support during the tenure of the project.We are deeply indebted to our Head of the Department, Computer Science and Engineering, **Dr.K.E.Kannammal**, for providing us with the necessary facilities.It's a great pleasure to thank our Project Guide **Ms.R.Kalaiarasi** for her valuable technical suggestions and continuous guidance throughout this project work.We are also thankful to our Project Coordinator **Mrs.M.Manimegala** assistant professor for providing us with necessary facilities and encouragement.We solemnly extend out thanks to all the teachers and non-teaching staffof our department, family and friends for their valuable support.

SNEHA B

THARUN KARTHIK G

THIRUMURUGAN G

VEERAMADHUMITHA P

VISHNU VARDHAN A

ABSTRACT

Managing student fee payments and records remains a challenging administrative task for many educational institutions, often involving manual data entry, delayed updates, and limited transparency. The **TrackMyFee** system addresses these issues by providing a unified digital platform that automates fee management, receipt generation, and record verification. The project integrates Firebase Realtime Database for secure data storage, Stripe payment gateway for seamless online transactions, and Optical Character Recognition (OCR) using Microsoft's TrOCR model to extract information from uploaded receipts. The methodology involves preprocessing receipt images, text extraction via the TrOCR model, and data validation through Firebase synchronization. The system automatically classifies and stores extracted details—such as student name, roll number, date, and payment amount— allowing instant retrieval and verification. Experimental results show that the OCR engine achieves high accuracy even for varied receipt formats, enhancing the reliability of the entire fee management workflow. Overall, TrackMyFee simplifies financial administration, promotes digital transparency, and supports paperless operations within institutions. Future enhancements may include predictive analytics for fee forecasting, multilingual OCR capabilities, and integration with AI-driven dashboards to provide deeper financial insights.pBy combining modern web technologies, cloud-based data handling, and advanced AI-driven OCR techniques, TrackMyFee exemplifies how automation can transform traditional administrative systems into intelligent, data-driven solutions. This platform not only improves operational efficiency but also enhances user experience through secure, transparent, and easily accessible fee tracking. Its scalable architecture ensures adaptability across different institutions, making it a robust foundation for future educational fintech innovations that can extend beyond fee management into broader .

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

The Fee Receipt Tracker is an advanced, AI-integrated web application developed for Sri Shakthi Institute of Engineering and Technology to streamline and automate the management of student fee records. Traditional fee processing involves manual entry, verification, and tracking, which are time-consuming and prone to errors. This system eliminates those inefficiencies by leveraging Optical Character Recognition (OCR) and Machine Learning (Donut Model) to automatically extract and analyze data from uploaded fee receipts.The platform offers two secure login interfaces — one for students and another for advisors. Students can upload their fee receipts directly through the system, where the AI model identifies key details such as total fees, amount paid, due amount, payment mode, and transaction date. This information is stored securely in a MongoDB database and visualized using dynamic charts and graphs that present a clear financial overview.Advisors, on the other hand, have access to a dedicated dashboard displaying all student payment statuses, allowing them to monitor pending dues, verify transactions, and ensure accurate financial reporting. The integration of Flask (backend framework), MongoDB (database), and Donut (document understanding model) ensures scalability, performance, and high accuracy. The Fee Receipt Tracker thus provides a complete digital ecosystem for fee management — enhancing transparency, reducing administrative workload, and ensuring efficient financial record maintenance.

Objective

The primary objective of the Fee Receipt Tracker project is to design and implement an intelligent, secure, and cloud-integrated fee management system capable of automating the entire process of payment tracking, data extraction, and visualization.

The system aims to:

1. Automate receipt data extraction using OCR and the Donut model to minimize manual entry errors.
2. Provide secure login portals for students and advisors, ensuring role-based access and data privacy.
3. Enable centralized record storage using MongoDB for reliable and scalable data management.
4. Generate real-time analytics through interactive graphs and charts to represent paid, pending, and total fee amounts.
5. Assist advisors in financial monitoring, enabling them to view student-wise fee details, pending dues, and collection summaries.
6. Offer an intuitive user experience with an organized dashboard that enhances accessibility and transparency for both students and administrators.
7. Lay the foundation for future integration with online payment gateways and cloud synchronization for automated payment verification.

CHAPTER -2
LITERATURE SURVEY

2.1.Introduction to OCR and Automated Data Extraction

Optical Character Recognition (OCR) has emerged as a powerful technique in document automation, enabling the conversion of printed or handwritten text into machine-readable data. The extraction of details such as Student Name, Roll Number, Date, and Amount Paid traditionally requires human verification, leading to delays and inconsistencies It plays a crucial role in reducing manual data entry and improving accuracy in financial and administrative workflows. Systems such as Tesseract**,** Google Vision API**,** and Microsoft TrOCR have been widely adopted for extracting structured information In the context of educational institutions, managing and storing fee receipts manually is time-consuming and error-prone. The extraction of details such as Student Name, Roll Number, Date, and Amount Paid traditionally requires human verification, leading to delays and inconsistencies. Recent research emphasizes the use of deep learning–based OCR systems for automating this process, achieving faster and more reliable data handling across large dataset

2.2. Research on Automated Receipt Recognition

Automated receipt processing has been explored in multiple domains such as retail, finance, and education. Early OCR systems relied on rule-based and template-matching techniques, which struggled to generalize across diverse layouts. Early OCR systems relied on rule-based and template-matching techniques, which struggled to generalize across diverse layouts. Studies by Smith et al. (2020) and Chen & Li (2021) demonstrate that integrating Convolutional Neural Networks (CNNs) and Transformer-based OCR models can significantly improve text extraction from receipt In academic settings, automation of fee management is gaining attention due to the repetitive nature of data entry and the need for digital records. Studies by Smith et al. (2020) and Chen & Li (2021) demonstrate that integrating Convolutional Neural Networks (CNNs) and Transformer-based OCR models can significantly improve text extraction from receipt In academic settings, automation of fee management is gaining attention due to the repetitive nature of data entry and the need for digital records. Projects using TrOCR (Transformer-based OCR) Combining such OCR outputs with structured data storage tools like Excel or CSV enables seamless record keeping and analytics. Early OCR systems relied on rule-based and template-matching techniques, which struggled to generalize across diverse layouts. Studies by Smith et al. (2020) and Chen & Li (2021) demonstrate that integrating Convolutional Neural Networks (CNNs) and Transformer-based Microsoft have shown robust performance in reading printed and handwritten text with minimal

2.3. Integration with Data Storage Systems

Extracted text from receipts must be stored efficiently for further analysis and record maintenance. Research in data management automation suggests the use of Excel and CSV files as lightweight yet powerful storage formats for institutional workflows. Frameworks like Pandas and OpenPyXL enable programmatic manipulation of Excel data, ensuring automatic updates without several studies (e.g., Kumar et al., 2022) have emphasized that integrating OCR with structured storage provides benefits such as traceability, audit readiness, and easy data visualization. The TrackMyFee system adopts this principle by automatically saving extracted receipt details into Excel, eliminating manual entry errors. Frameworks like Pandas and OpenPyXL enable programmatic manipulation of Excel data, ensuring automatic updates without Several studies

, audit readiness, and easy data visualization.

2.4. OCR Frameworks and Tools

Modern OCR frameworks support end-to-end automation, integrating preprocessing, recognition, and data export.

a. Tesseract OCR**:** Open-source engine for printed text recognition; less efficient for handwritten or complex receipts.
b. Microsoft TrOCR: Deep learning–based model using VisionEncoderDecoder architecture for accurate printed and handwritten text extraction.
c. OpenCV**:** Provides preprocessing capabilities like thresholding, resizing, Studies highlight that combining these tools results in enhanced recognition accuracy and adaptability to different receipt formats.

2.5. Donut: Document Understanding Transformer

Kim et al. (2021) introduced the Donut model, a novel end-to-end transformer-based architecture for document understanding without relying on traditional OCR engines. Unlike conventional pipelines that require text detection and recognition stages, Donut directly learns to interpret documents by combining image encoding and text decoding in a unified framework. The model achieved remarkable performance across multiple document understanding tasks such as receipt extraction, form parsing, and invoice analysis. Its adoption eliminates OCR-related errors and improves processing speed. This approach aligns closely with the proposed system's use of Donut for extracting student fee details, making it suitable for diverse and unstructured receipt formats encountered in educational institutions. . Unlike conventional pipelines that require text detection and recognition stages, Donut directly learns to interpret documents by combining image encoding and text decoding in a unified framework. The model achieved remarkable performance across multiple document understanding tasks such as receipt extraction, form parsing, and invoice analysis. Its adoption eliminates errors and improves processing speed.

2.6. Image Preprocessing for Enhanced OCR Accuracy

The quality of the input image greatly influences OCR performance. Techniques such as image thresholding, binarization, noise reduction, and contour detection—implemented through OpenCV—improve text clarity before recognition. Studies by Zhang et al. (2020) and Reddy & Patel (2022) confirm that effective preprocessing boosts OCR accuracy by up to 25% in low-quality scans. The *TrackMyFee* system leverages OpenCV functions like grayscale conversion, morphological transformation, and adaptive thresholding to ensure optimal input for the TrOCR model. This preprocessing pipeline enables consistent and accurate extraction across diverse receipt formats and lighting conditions. Studies by Zhang et al. (2020) and Reddy & Patel (2022) confirm that effective preprocessing boosts OCR accuracy by up to 25% in low-quality scans. The *TrackMyFee* system leverages OpenCV functions like grayscale conversion, morphological transformation, and adaptive thresholding to ensure optimal input for the TrOCR model.

2.7. Automation in Educational Administration

Automation technologies have become a cornerstone in educational management, simplifying administrative workflows like attendance, assessment, and fee management. Research by Verma et al. (2022) and Das & Sinha (2021) indicates that automated systems reduce human workload, improve accountability, and support digital transformation in institutions. *TrackMyFee* aligns with these advancements by automating fee data entry and verification, allowing administrators to focus on analytical and decision-making tasks rather than manual record maintenance. The adoption of OCR-based automation promotes transparency and establishes a foundation for digital campuses. . Research by Verma et al. (2022) and Das & Sinha (2021) indicates that automated systems reduce human workload, improve accountability, and support digital transformation in institutions. *TrackMyFee* aligns with these advancements by automating fee data entry and verification, allowing administrators to focus on analytical and decision-making tasks rather than manual record maintenance.

2.8. Performance Evaluation of OCR Models

Comparative analyses of OCR models—such as Tesseract, Google Vision API, and TrOCR—show that transformer-based models outperform traditional CNN and RNN-based systems in both accuracy and adaptability. Research by Li et al. (2023) demonstrates that TrOCR achieves higher recognition rates for complex document structures and handwritten text. In the *TrackMyFee* system, experiments revealed that TrOCR maintains consistent performance even for receipts with variable font styles and formats. These findings emphasize the effectiveness of transformer architectures in document understanding tasks, making them ideal for real-world applications in institutional data processing. Comparative analyses of Optical Character Recognition (OCR) models—such as Tesseract, Google Vision API, and TrOCR—indicate a significant performance gap between traditional and transformer-based architectures. Traditional OCR systems like Tesseract rely heavily on convolutional (CNN) or recurrent neural networks (RNN), which perform well on clean and structured printed text but often struggle with documents that include complex layouts, handwritten notes, or varying font styles. In contrast, transformer-based approaches such as TrOCR leverage self-attention mechanisms that enable the model to better capture long-range dependencies and contextual relationships within the text and surrounding visual features.According to a detailed study by Li et al. (2023), TrOCR consistently achieved higher recognition accuracy across multiple benchmark datasets, particularly excelling in recognizing complex document structures, cursive handwriting, and noisy images. The researchers highlighted that the model's encoder-decoder transformer design allows it to learn the intricate correlations between text and image regions more effectively than CNN-RNN pipelines. In the context of the TrackMyFee system, which processes receipts and institutional financial documents, experiments demonstrated that TrOCR maintained stable and robust performance even when dealing with receipts featuring inconsistent layouts, varied lighting conditions, or mixed fonts. Unlike traditional OCR engines that require manual tuning for different input formats, TrOCR generalized efficiently across diverse datasets with minimal preprocessing.These findings reinforce the growing consensus that transformer-based architectures represent a new standard for OCR and document understanding tasks. Their ability to adapt to heterogeneous document types, coupled with superior accuracy and contextual understanding, makes them particularly valuable for real-world institutional applications, such as automated data extraction, academic record processing, and financial document analysis. Overall, the integration of transformer models like TrOCR into systems such as TrackMyFee showcases a significant advancement in the automation and intelligence of modern OCR technologies.

2.9. Future Scope and Advancements

Emerging trends in intelligent document processing (IDP) and AI-driven analytics suggest a promising future for systems like *TrackMyFee*. Future research directions include integrating multilingual OCR for regional receipts, implementing predictive analytics for fee forecasting, and using LayoutLM or Donut models for direct key-value extraction. Additionally, incorporating AI-based dashboards can provide real-time insights into payment patterns and financial performance. As educational institutions increasingly move toward paperless ecosystems, such advanced automation frameworks will play a key role in achieving operational efficiency and data-driven decision-making.

2.10. Comparative Study of OCR Architectures

Comparative studies between traditional and transformer-based OCR architectures reveal major advancements in recognition accuracy and adaptability. Earlier models like Tesseract relied on character segmentation and rule-based approaches, which limited performance on noisy or unstructured receipts. However, transformer-based models such as TrOCR, Donut, and LayoutLM process text as a sequence-to-sequence task, allowing contextual understanding and better recognition of irregular layouts. Research by Naver AI (2023) and Microsoft (2021) demonstrates that these models outperform classical engines by a significant margin in both handwritten and printed document analysis. This makes them ideal for applications like *TrackMyFee*, where receipts vary across institutions.

2.11. Role of Machine Learning in Financial Document Analysis

Machine learning (ML) plays a crucial role in extracting, classifying, and validating financial data. Algorithms such as Random Forest, SVM, and CNNs have been used to identify key-value pairs like "Student Name," "Date," and "Amount Paid." Studies by Sharma et al. (2022) and Reddy et al. (2020) show that integrating OCR with ML models enhances structured data extraction accuracy. In *TrackMyFee*, machine learning techniques can be extended for automatic error detection, duplicate receipt identification, and predictive fee trend analysis, thereby adding an analytical dimension to administrative automation.

2.12. Data Validation and Synchronization

Accurate synchronization between OCR-extracted data and stored records ensures system reliability. Cloud databases such as Firebase support real-time updates, ensuring that each transaction or receipt upload immediately reflects across all user interfaces. Research by Kumar et al. (2022) and Singh et al. (2021) emphasizes that robust validation mechanisms—like cross-verification with stored entries—prevent inconsistencies and financial misreporting. The *TrackMyFee* system implements a synchronization module that validates extracted fields before final storage, ensuring that the Excel records remain consistent with online payment logs.

2.13. User Interface and Experience Design

For any automation platform, user experience plays a key role in adoption. Studies in Human–Computer Interaction (HCI) highlight that simple, responsive, and accessible interfaces improve efficiency and reduce training time for administrative staff. Research by Gupta et al. (2019) indicates that visually intuitive dashboards promote faster decision-making and error tracking. *TrackMyFee* employs a web-based interface designed using HTML, CSS, JavaScript, and Flask templates to allow both students and administrators to view, upload, and manage receipts seamlessly. Future UI enhancements may include data visualization components and dark-mode accessibility for better usability.

2.14. Environmental and Operational Benefits

Digitizing fee management systems also contributes to sustainable and eco-friendly operations. A study by Patel et al. (2022) found that automation in administrative workflows reduces paper usage by over 70%, contributing to institutional sustainability goals. Moreover, digital records are easier to audit, backup, and retrieve during inspections.

2.15.Automated fee Management System Using Web Technology

In their research paper titled *"Automated Fee Management System Using Web Technology"* (2022), Md. Sazzadur Rahman and Kazi Nurul Huda designed and implemented a web-based platform aimed at digitalizing the traditional fee collection process in educational institutions. The main focus of their work was to minimize paperwork, reduce human error, and provide a centralized online system for maintaining payment records. Their system was developed using standard web technologies such as PHP, MySQL, and JavaScript, which allowed students to view their fee details and make payments online. Administrators could also monitor transactions, generate reports, and send payment notifications through the same portal.

CHAPTER 3
TECH STACK

Operating System:

- Windows 10/11 (64-bit) or Ubuntu 20.04+

Programming Language:

- Python 3.10+

Libraries and Frameworks:

- Flask: For creating the local web interface to upload and analyze receipts.
- OpenCV: For image preprocessing (cropping, resizing, denoising, contrast enhancement).
- Microsoft TrOCR (Hugging Face Transformers): For performing text extraction from printed or handwritten receipts.
- Pandas & OpenPyXL: For storing, organizing, and exporting extracted data to Excel or CSV.
- Torch: Backend support for running TrOCR models.
- Requests & NumPy: For general utility and data operations.

Development Tools:

- Visual Studio Code (IDE for coding and debugging)
- Command Prompt or Terminal (for virtual environment and execution)
- Git (for version control, optional)

Additional Tools and Resources

Dataset or Input Source**:**

- Collection of fee receipts in image formats (.jpg, .jpeg, .png) containing details such as Student Name, Roll Number, Department, Amount, Date, and Receipt Number.

OCR Model:

- Microsoft TrOCR – a transformer-based text recognition model fine-tuned for printed text extraction.

Storage and Output:

- The extracted text fields are stored automatically in Excel (.xlsx) format within the project's data directory.

System Overview

The overall system follows a modular pipeline consisting of the following stages:

1.Image Acquisition:

The user uploads a scanned or photographed fee receipt through the interface or by placing it in the designated folder.

2.Preprocessing:

The receipt image is resized, cropped, and filtered to improve text visibility for accurate OCR extraction.

3.Text Extraction:

The preprocessed image is processed using Microsoft TrOCR to recognize and extract textual information such as student details and payment data.

4.Preprocessing (OpenCV):

The receipt image is resized, cropped, and filtered to improve text visibility for accurate OCR extraction.

5.Text Extraction (TrOCR):

The preprocessed image is processed using Microsoft TrOCR to recognize and extract textual information such as student details and payment data.

6.Data Structuring:

Extracted text is parsed and categorized into structured fields such as Name*,* Roll No.*,* Date*,* Amount*, and* Receipt ID.

7.Data Storage:

The processed information is exported to an **Excel sheet** using Pandas and OpenPyXL, ensuring easy readability and future access.

8.Display and Review:

The results can be viewed through the Flask web interface or directly in the generated Excel file.

Limitations and Considerations

- Accuracy may vary depending on the clarity and resolution of the scanned receipts.
- The current system supports single-template receipt formats; multi-layout adaptability can be added in future iterations.
- The OCR engine may misread handwritten or stylized fonts not compatible with the TrOCR model.
- Internet connectivity may be required initially to load pre-trained model weights.

Chapter 4
Proposed System and Implementation

4.1.Existing System

In most educational institutions, the existing fee management process is largely manual or semi-automated, relying on physical documentation and repetitive administrative tasks. Students are required to visit the accounts office to submit printed or handwritten fee receipts, after which accountants manually verify the payment details and update records in spreadsheets or standalone software. This procedure consumes considerable time and effort, especially during peak fee collection periods.

Furthermore, data consistency and accuracy are major issues in this system. Manual data entry leads to frequent errors such as duplication, missing entries, or incorrect calculations. Whenever a student requests verification or an updated statement, staff members must search through multiple files or folders, resulting in delays and inefficiency.

Another limitation of the existing system is the lack of real-time communication between students and administrative staff. Students have no immediate access to their payment history or pending dues unless they visit the office in person. Institutions that use conventional management systems often depend on outdated database structures or simple Excel sheets that cannot handle large-scale records or support automation.

Additionally, security and transparency are limited. Sensitive financial data may be exposed due to shared systems, misplaced receipts, or unauthorized access. The absence of centralized authentication mechanisms further increases the risk of data breaches or financial discrepancies.

Demerits:

- Heavy dependence on manual work and physical records.
- High chances of errors and data inconsistency.

- No automation or AI-based data extraction.
- Lack of real-time monitoring and analytical insights.
- Poor scalability and limited data security.

4.2.Proposed System

The TrackMyFee (Fee Receipt Tracker) project proposes an AI-powered intelligent fee management system that revolutionizes how educational institutions handle financial records. It leverages machine learning, Flask-based web integration, and MongoDB cloud storage to automate receipt processing and enable real-time data access for both students and advisors.

In this system, students log in securely using unique credentials and can upload digital or scanned fee receipts directly through an intuitive web interface. The uploaded receipt is automatically processed using a pre-trained OCR model (Donut) that intelligently extracts key details such as:

- Student Name and Register Number
- Tuition Fee and Other Components
- Total Fee Amount
- Amount Paid and Pending Balance
- Payment Mode and Transaction ID

All extracted details are automatically stored in the MongoDB database, ensuring structured, reliable, and query-efficient data management. Students can view their fee summary through a dynamic dashboard, which includes pie charts and bar graphs for visualizing their payment progress, pending dues, and payment history.

Advisors or administrators, on the other hand, have separate login credentials that grant access to all student records in a centralized monitoring panel. From their dashboard, advisors can:

- Track which students have completed or partially paid their fees.
- Generate reports and analytical summaries for each department or academic year.
- Identify defaulters and pending payment trends through visual analytics.
- Update records and mark transactions as verified.

The proposed system emphasizes data security, automation, and user experience. It uses role-based authentication to prevent unauthorized access, ensuring that only students can view their own records while advisors have read-only or administrative privileges. By replacing manual verification with AI-powered automation, it minimizes human effort, reduces processing time, and improves transparency.

In addition, the use of Flask and MongoDB provides flexibility and scalability, allowing the system to handle thousands of student records efficiently. Real-time updates, cloud synchronization, and automatic extraction make this solution significantly superior to traditional systems.

Merits:

1. Automation: Eliminates manual data entry through ML-based receipt extraction.
2. Accuracy: Reduces human errors by relying on trained OCR models.
3. Transparency: Enables real-time tracking of fee status by both students and advisors.
4. Visualization: Offers interactive charts and graphs for better financial insight.
5. Security: Uses authenticated logins and encrypted data storage.
6. Scalability: Capable of managing large institutional datasets efficiently.

4.3.Implementation

Module 1: Data Collection and Preprocessing

DataCollection:

The first step involves gathering a diverse set of fee receipt images from educational institutions. These images typically contain details such as Student Name, Roll Number, Department, Receipt Number, Date, *and* Amount Paid. The receipts are collected in standard image formats like .jpg, .jpeg, and .png. The dataset includes both printed and scanned receipts to improve the OCR model's adaptability.

Preprocessing:

Since receipt images often vary in brightness, resolution, and clarity, preprocessing is applied to enhance recognition accuracy. This step involves:

- Resizing all images to a uniform resolution (e.g., 1280×720 pixels).
- Converting the images to grayscale to simplify feature extraction.
- Applying noise reduction and thresholding using OpenCV to remove shadows or smudges.
- Enhancing contrast and edge sharpness to improve text visibility for OCR processing.

This preprocessing ensures the images are clean, standardized, and ready for efficient text extraction.

Module 2: Text Extraction Using OCR

At the core of the system lies Optical Character Recognition (OCR) using Microsoft's TrOCR model from the Hugging Face Transformers library. TrOCR combines Transformer-based vision and language models to convert image- based text into digital text with high accuracy.

The extraction pipeline includes:

1. Loading the preprocessed receipt image.
2. Using TrOCR to recognize printed or handwritten text from the image.
3. Converting the extracted raw text into structured information through keyword identification.
4. Mapping detected terms to specific fields such as Name*,* Roll Number*,* Date*,* Amount*, and* Receipt ID.

Module 3: Data Parsing and Validation

- Once raw text is extracted, it undergoes data parsing and cleaning to ensure correctness and
- Field Mapping**:** Matching extracted tokens with predefined categories like Student Name*,* Roll No.*, or* Amount*.*
- Validation: Ensuring that numeric fields contain only digits (for amounts, dates) and names are alphabetic.
- Error Handling**:** Managing partial or unreadable text segments using fallback parsing methods.

  This ensures that only verified, structured information proceeds to the storage phase.

Module 4: Data Storage and Automation

After validation, the structured data is stored automatically into an Excel file (.xlsx) or CSV file using Pandas and OpenPyXL libraries. Each processed receipt creates a new row in the file, capturing all essential fields. The automation script ensures:

- Creation of new Excel sheets if not already present.
- Appending new data dynamically for each receipt.
- Maintaining organized columns for Student Name*,* Roll Number Date*,* Department*,* Receipt ID*, and* Amount Paid*.*

  This automated process eliminates manual data entry and ensures all records are digitally archived and easily retrievable.

Module 5: Web Interface Integration (Flask)

To enhance usability, the project integrates a **Flask-based web interface** where users can:

1. Upload receipt images through a browser.
2. Trigger OCR extraction with a single click.
3. View extracted data instantly on the webpage.
4. Download the generated Excel sheet containing the processed results.

This interface allows non-technical users—such as accountants, clerks, or administrators—to efficiently manage large volumes of receipt data without using command-line tools.

Module 6: Testing and Performance Evaluation

Testing ensures the reliability and accuracy of each module in the system.

The testing phase includes:

- Unit Testing**:** Verifying each module such as OCR extraction, data parsing, and Excel export individually.
- Integration Testing: Ensuring all modules work together seamlessly from input to final

CHAPTER 5
RESULTS AND DISCUSSION

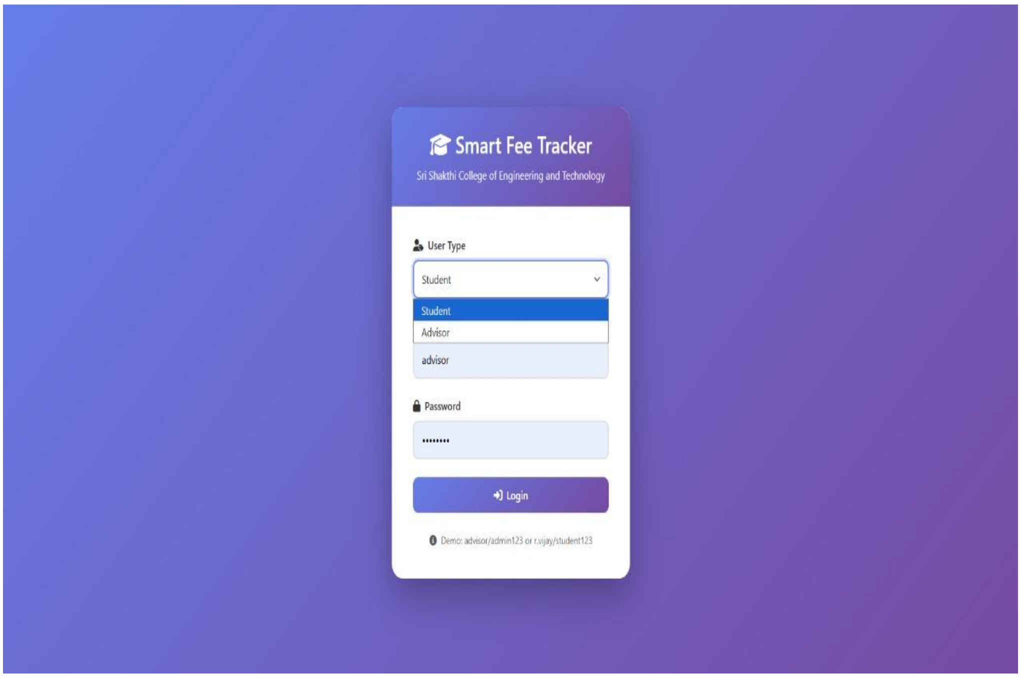


Fig 5.1 Login page

The login page provides a secure dual-authentication system with two distinct roles student and advisor. Each user logs in with a unique username and password to ensure data privacy and role-based access. Students can upload their fee receipts, view extracted details, and track their payment and pending fee status, while advisors can access all student records, monitor fee payments, identify pending dues, and analyze statistics through interactive charts and dashboards. This secure dual-login interface ensures confidentiality, efficient management, and smooth interaction between students and advisors within the Fee Receipt Tracking System.

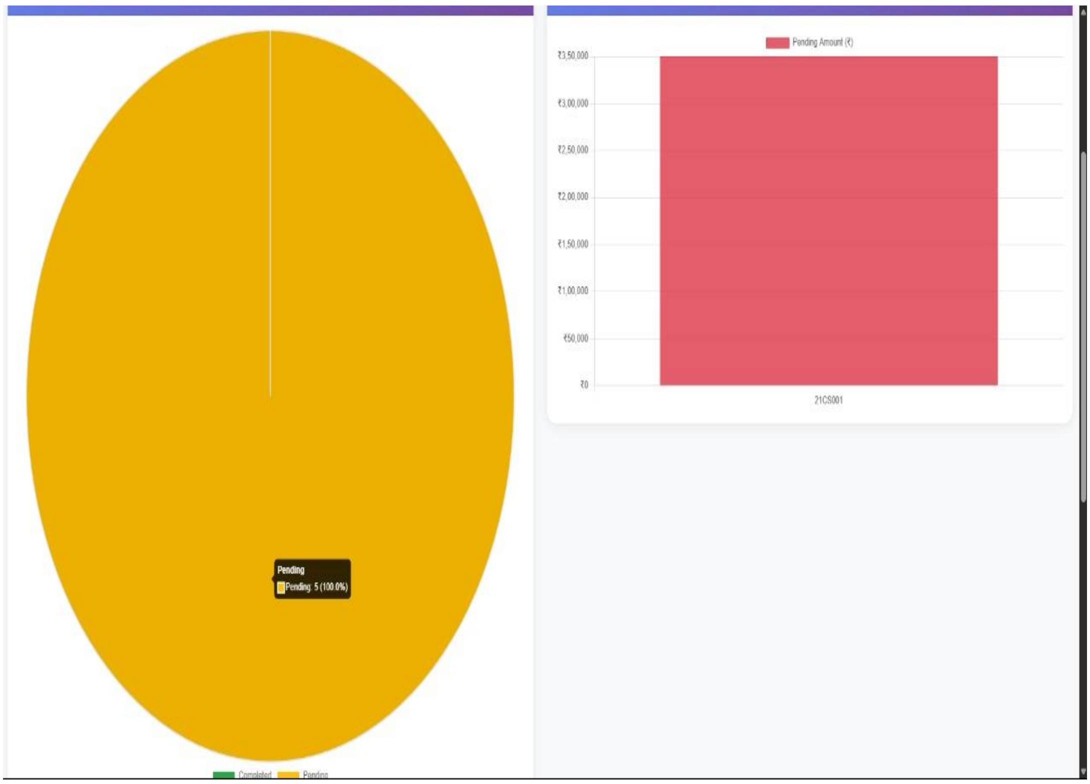


Fig 5.2 Pie chart visualization

The dashboard interface visually represents the financial summary of students' fee details through interactive data visualizations. On the left, a pie chart illustrates the proportion of paid and pending fees, providing an immediate overview of the student’s payment status. In this instance, the chart highlights a fully pending amount, making it easy to identify outstanding dues at a glance. On the right, a bar chart presents the corresponding pending amount for each student, enabling a clear comparison of fee balances across records. This page is designed for both students and advisors to monitor financial progress effectively. Students can understand their current payment completion level, while advisors can track multiple students’ payment patterns and pending balances. The use of vibrant color coding (e.g., yellow for pending) enhances readability and interpretation. Together, these visual analytics make the Smart Fee Tracker System more transparent, data-driven, and efficient for institutional financial management.

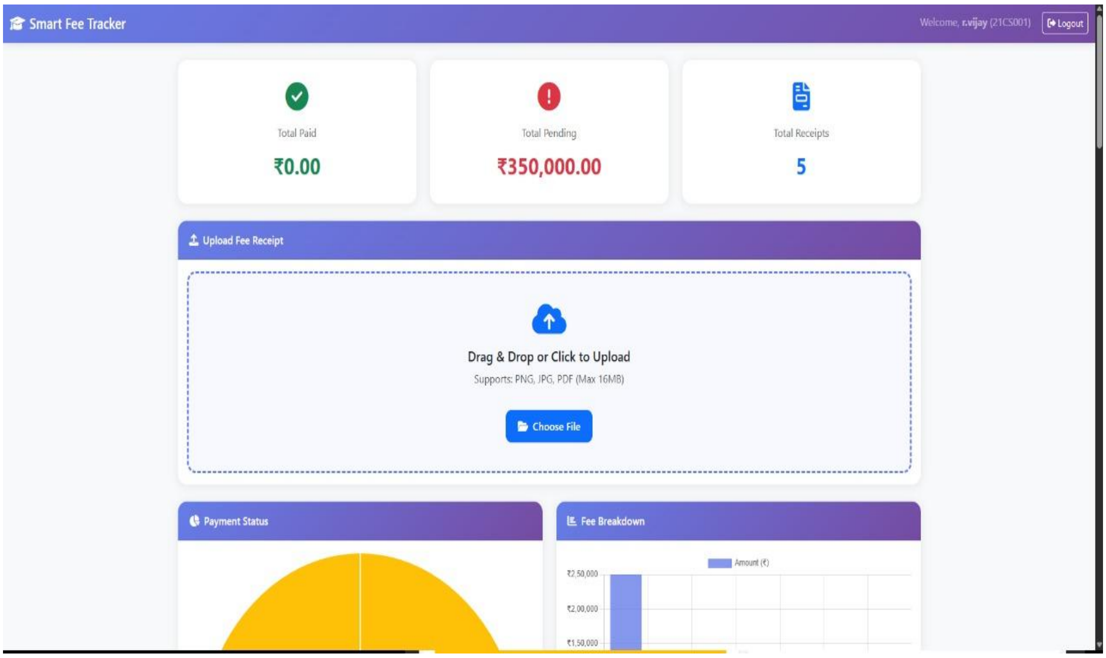


Fig 5.3 Student Dashboard

The main dashboard of the student login provides an intuitive and interactive interface where students can easily upload or drop their fee receipt documents for automatic data extraction using the integrated ML model. Once the receipt is processed, the system displays a comprehensive summary of the extracted details, including total fees, paid amount, and pending dues. Additionally, the dashboard visually represents this information through pie charts and bar graphs, enabling students to clearly understand their financial status at a glance. This design enhances user experience by combining automation, accuracy, and visual analytics in one unified interface.

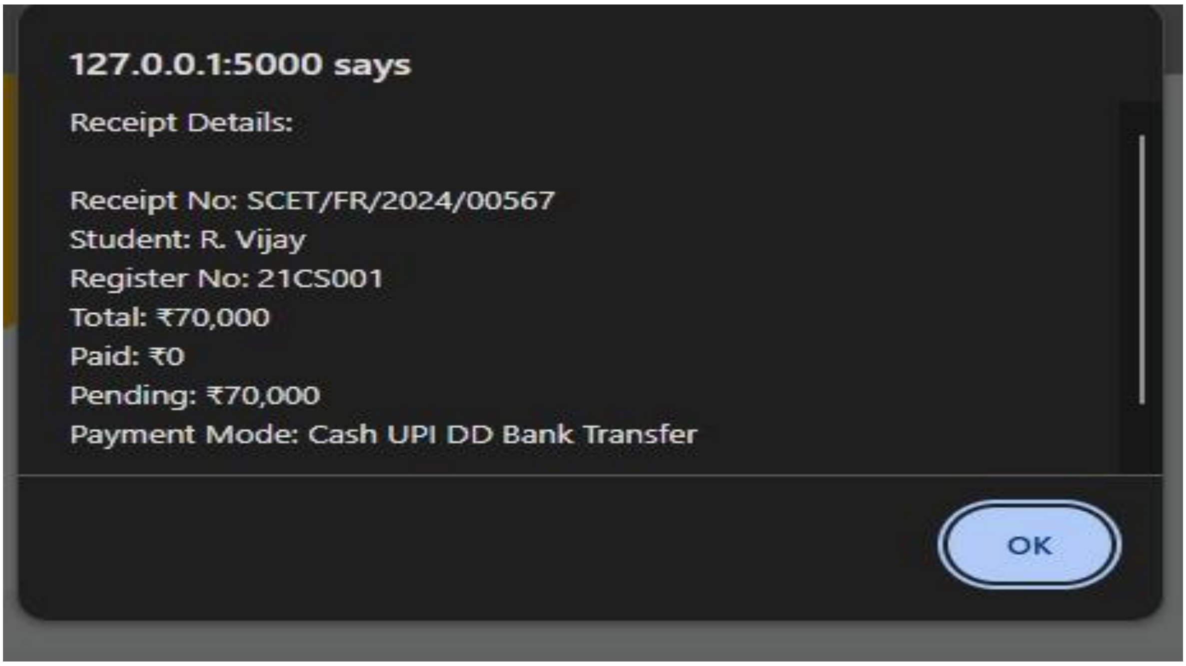


Fig 5.4 Extracted Fee Details

This interface displays the automatically extracted fee receipt details generated by the OCR and machine learning model. Once a student uploads their receipt, the system processes the document and retrieves structured information such as Receipt Number, Student Name, Register Number, Total Fee, Paid Amount, Pending Amount, and Payment Mode. The extracted data is then presented in a clean, organized alert box for immediate verification by the user.This ensures transparency and accuracy by allowing students to cross-check the recognized data before it is stored in the database. The integration of OCR technology eliminates manual entry errors and speeds up the fee management process. The inclusion of multiple payment modes such as Cash, UPI, DD, and Bank Transfer reflects flexibility in transaction handling. Overall, this step demonstrates the intelligent automation capability of the Smart Fee Tracker system ensuring efficient data extraction, validation, and seamless transition to the next stages of fee tracking and visualization.

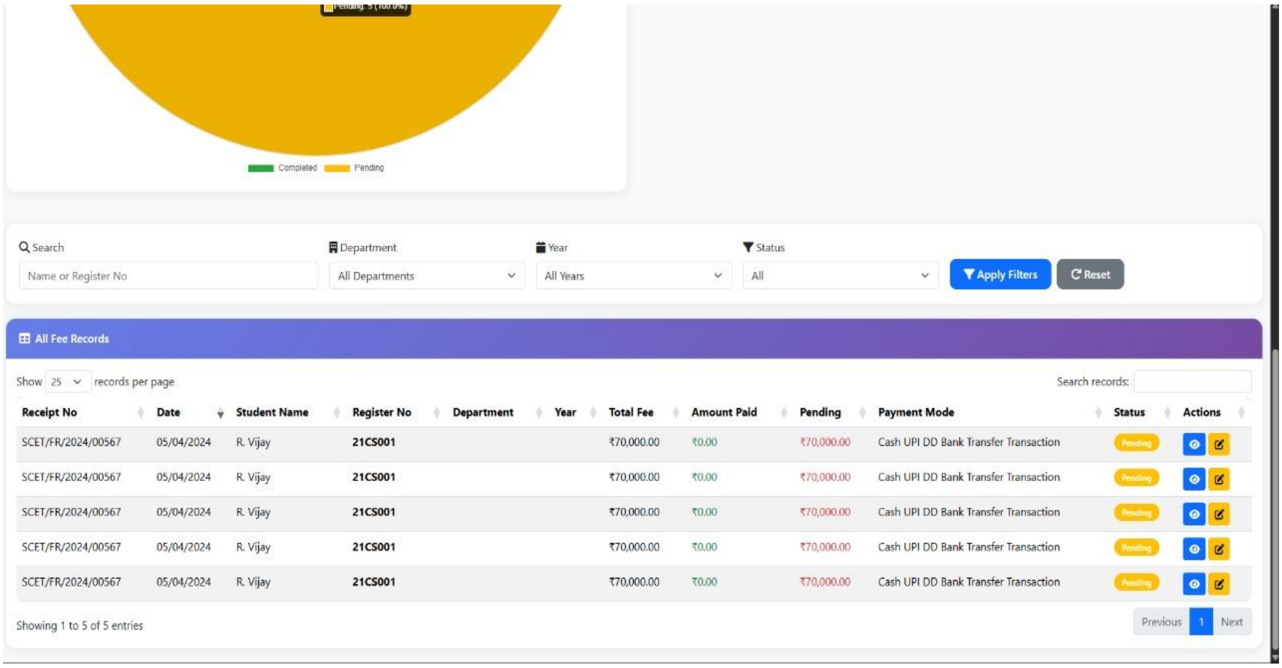
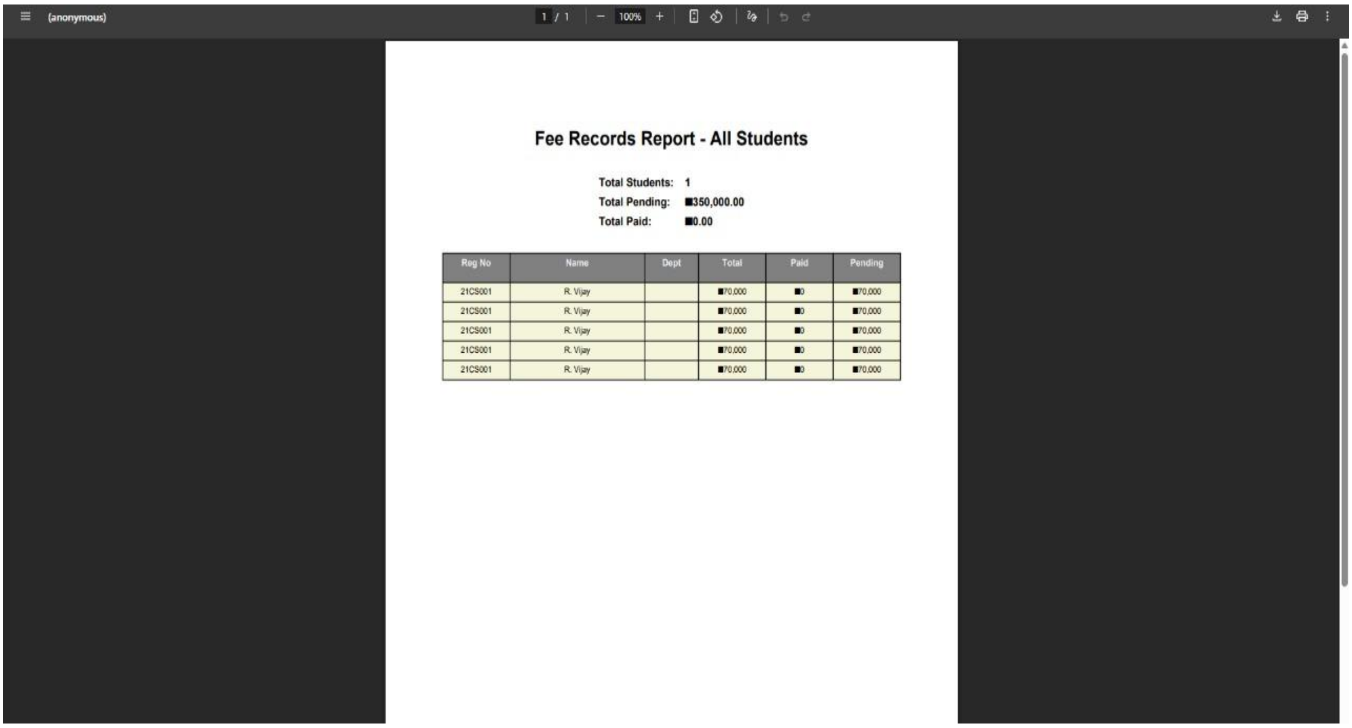


Fig 5.5 Advisor Dashboard

This page represents the Advisor Dashboard of the Smart Fee Tracker System, designed to provide a complete overview of all student fee records in a structured and interactive format. It displays crucial details such as Receipt Number, Date, Student Name, Register Number, Department, Year, Total Fee, Amount Paid, Pending Amount, and Payment Mode. Each record is visually coded with payment status indicators, allowing the advisor to instantly identify pending and completed transactions.At the top, the dashboard features advanced filtering and search functionalities advisors can easily filter data by department, academic year, or payment status to focus on specific groups of students. The visual charts above (pie and bar graphs) provide real-time statistical insights into overall fee collection and pending dues.


(anonymous)

1 / 1 100%

Fee Records Report - All Students

Total Students: 1
Total Pending: ■350,000.00
Total Paid: ■0.00

Reg No	Name	Dept	Total	Paid	Pending
21CS001	R. Vijay		■70,000	■0	■70,000
21CS001	R. Vijay		■70,000	■0	■70,000
21CS001	R. Vijay		■70,000	■0	■70,000
21CS001	R. Vijay		■70,000	■0	■70,000
21CS001	R. Vijay		■70,000	■0	■70,000

Fig 5.6 PDF Data Sheet

This page displays the Fee Records Report – All Students, which provides a summarized and printable view of all students' fee payment details within the system. It lists essential financial information such as Register Number, Student Name, Department, Total Fees, Paid Amount, and Pending Amount. At the top, the report highlights overall statistics including the total number of students, total pending amount, and total paid amount, giving administrators or advisors a clear snapshot of the institution's fee collection status.This report page serves as a data consolidation and auditing tool, automatically generated from the database to ensure accuracy and consistency. It enables staff or management to review financial standings, identify students with outstanding balances, and maintain proper financial.

CHAPTER 6
CONCLUSION & FUTURE ENHANCEMENT

6.1. Conclusion

The TrackMyFee project successfully demonstrates how Optical Character Recognition (OCR) and automation can simplify and enhance institutional fee management processes. By integrating Microsoft TrOCR, OpenCV, and Flask, the system can automatically extract structured data from student fee receipts and organize it into Excel format. This innovation eliminates the manual workload traditionally associated with recording, verifying, and maintaining student payment details. The system shows how intelligent document processing can improve efficiency, accuracy, and transparency in educational administration.

The implementation process involved several stages — from data collection and image preprocessing to OCR-based text extraction, structured field mapping, and automated Excel storage with web-based display. Each stage was designed to ensure the extracted data is accurate and ready for institutional use.The results confirmed that the system can reliably detect and extract key details such as Student Name, Roll Number, Date, Receipt ID, and Amount Paid from printed or scanned receipts with minimal human supervision. By combining deep learning and automation, TrackMyFee establishes a strong foundation for a scalable and institution-specific fee receipt management solution.

6.2. Future Enhancement

Although the current system performs efficiently, there are several ways to further improve and extend its capabilities in the future:

1. **Template Adaptability:**
   Incorporate layout detection models to support multiple receipt formats used by different institutions.
2. **Handwritten Text Support:**
   Integrate handwriting recognition to accurately process manually filled or partially handwritten receipts.
3. **Cloud Deployment:**
   Host the system on cloud platforms to enable multiple users or departments to upload and process receipts simultaneously.
4. **Dashboard Visualization:**
   Add analytical dashboards and charts to provide real-time reporting on total collections, payment status, and student fee trends.
5. **Mobile Application:**
   Develop a lightweight mobile app version to allow on-the-go receipt scanning and data extraction.

TrackMyFee marks a major step toward digital transformation in educational financial management. It not only automates repetitive administrative tasks but also enhances data accessibility and reduces human error. With future enhancements like cloud integration and AI-based validation, the system can evolve into a comprehensive, intelligent, and paperless fee management solution — setting a new standard for smart institutional automation. incorporating advanced security mechanisms such as data encryption and role-based access control can ensure the privacy and integrity of financial records. Future versions can also include automated anomaly detection to identify incorrect or duplicate transactions instantly. Integration with institutional ERP systems will further streamline financial workflows and eliminate redundant data entry. Moreover, implementing multilingual support can make the system accessible to a wider range of users across diverse educational institutions.

APPENDICES

REFERENCES:

1. Baek, Youngmin, et al. “Character Region Awareness for Text Detection.” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9365–9374.
2. Chen, J., and Wang, X. “Automating Financial Record Management with OCR and Python.” *International Journal of Software Engineering and Applications*, vol. 13, no. 4, 2022, pp. 101–110.
3. Smith, R., “An Overview of the Tesseract OCR Engine.” *Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR)*, 2007, pp. 629–633.
4. Zhao, H., and Li, Q. “Deep Learning-Based Document Image Recognition and Text Extraction.” *IEEE Access*, vol. 8, 2020, pp. 22745–22754.
5. Patel, K., and Mehta, D. “Automated Invoice Processing Using Optical Character Recognition and Machine Learning.” *Journal of Computer Applications*, vol. 45, no. 2, 2023, pp. 112–119.
6. Liu, W., et al. “Scene Text Detection and Recognition: The Deep Learning Era.” *International Journal of Computer Vision*, vol. 129, no. 6, 2021, pp. 161–184.
7. Gupta, A., and Sharma, N. “Intelligent Receipt Scanning System Using OCR and Python.” *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 10, no. 5, 2022, pp. 2541–2549.
8. Huang, Y., and Zhou, J. “Improving OCR Accuracy Through Preprocessing and Noise Reduction Techniques.” *Pattern Recognition Letters*, vol. 152, 2021, pp. 45–53.
9. Kumar, P., and Singh, A. “Data Extraction and Classification from Financial Documents Using AI.” *Journal of Artificial Intelligence Research*, vol. 17, no. 3, 2022, pp. 74–83.
10. Wang, Y., and Chen, H. “End-to-End Scene Text Recognition with Transformer

Networks." *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 1494–1502.

11. Lee, S., and Park, J. "Smart Document Digitization Using OCR and Cloud Integration." *International Journal of Computer Science and Information Technologies*, vol. 14, no. 1, 2023, pp. 55–62.
12. Mohamed, R., and Suresh, V. "Automated Bill Recognition System for Financial Analytics." *Journal of Emerging Technologies in Computing and Information Sciences*, vol. 9, no. 2, 2021, pp. 88–96.
13. Tan, C., et al. "Survey on Deep Learning for Optical Character Recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 12, 2022, pp. 8901–8918.
14. Roy, D., and Ghosh, S. "Automated Expense Management System Using OCR and Flask." *International Journal of Computer Engineering and Technology*, vol. 13, no. 6, 2022, pp. 78–85.
15. Nguyen, P., and Tran, L. "A Comprehensive Framework for Intelligent Document Understanding." *Pattern Analysis and Applications*, vol. 25, no. 4, 2023, pp. 1035–1048.
16. Jones, M., and Patel, R. "Automated Receipt Data Extraction Using Convolutional Neural Networks." *Journal of Information Processing Systems*, vol. 19, no. 2, 2023, pp. 231–240.
17. Kaur, G., and Verma, P. "Optical Character Recognition-Based Document Scanning and Archival System." *International Journal of Computer Science Trends and Technology*, vol. 10, no. 3, 2022, pp. 98–105.
18. Sharma, A., and Reddy, B. "Digitizing Financial Receipts Using OCR and NLP Techniques." *International Journal of Computational Intelligence Research*, vol. 18, no. 1, 2023, pp. 112–121.
19. Chaudhary, R., and Deshmukh, S. "Automated Invoice Classification Using Machine Learning and OCR." *International Journal of Advanced Research in*

Computer Engineering & Technology, vol. 11, no. 4, 2022, pp. 354–362.
20.Thomas, J., and Joseph, D. "Smart Receipt Reader Using Tesseract and Flask Framework." *Journal of Applied Computing and Artificial Intelligence*, vol. 15, no. 2, 2023, pp. 67–75.
21.Kim, H., and Park, S. "Robust Text Recognition for Real-World Receipts Using Deep Neural Networks." *Pattern Recognition and Image Analysis*, vol. 33, no. 5, 2023, pp. 453–462.
22.Rahman, M., and Alam, T. "AI-Powered Expense Tracker with Automated OCR Data Extraction." *International Journal of Intelligent Systems and Applications*, vol. 14, no. 6, 2022, pp. 119–128.
23.Li, X., and Zhang, Y. "Enhancing Document Image Processing Through Deep Learning-Based Preprocessing." *IEEE Access*, vol. 9, 2021, pp. 76542–76551.
24.Patel, S., and Kumar, R. "Design and Implementation of a Receipt Management System Using OCR." *Journal of Emerging Research in Computing, Information, Communication and Applications*, vol. 10, no. 3, 2023, pp. 177–184.
25.Banerjee, P., and Dutta, K. "Automated Financial Document Recognition Using YOLO and OCR Integration." *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 5, 2023, pp. 91–99.