INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD)
© 2026 IJNRD | Volume 11, Issue 3, March 2026 | ISSN: 2456-4184 | IJNRD.ORG

AI-Enabled GST Invoice Processor Using Automated OCR-Based Data Extraction: A Web Application Approach

Author 1 - Dr. Dhivya. K
Assistant Professor, Department of Computer Science
Dr. N. G. P. Arts and Science College, Coimbatore, Tamil Nadu, India

Author 2 - Sanjay D
B.Com Computer Applications (Final Year) | Reg. No.: 231CM047
Department of Commerce CA, Dr. N. G. P. Arts and Science College
Coimbatore, Tamil Nadu, India

ABSTRACT

The growing volume of Goods and Services Tax (GST) invoices in Indian commerce has created a pressing need for automated tools that reduce human effort and error in financial data processing. This paper presents an AI-enabled, web-based GST Invoice Processor that automates the extraction, classification, and storage of invoice data using Optical Character Recognition (OCR) technology. The system accepts invoice files in image (JPG, PNG) or PDF format, extracts relevant textual information using Tesseract OCR and pypdf libraries, identifies applicable GST rates through keyword-based product classification, and automatically computes CGST, SGST, and IGST components. All processed data is stored in a structured SQLite database and presented through an interactive analytics dashboard with real-time charts. The platform also supports dynamic updates to the GST classification dataset via CSV file uploads, making it adaptable for diverse business requirements. The system is developed using Python (FastAPI), HTML5, CSS3, JavaScript with Jinja2 templating, and deployed on a Render cloud server. Testing conducted on four sample invoices demonstrated a 100% processing success rate with accurate tax computation across all test cases. The proposed system significantly reduces manual effort, improves accuracy, and provides centralized financial record management for small businesses and accounting professionals.

Keywords: *GST Invoice Processing, Optical Character Recognition (OCR), Tesseract, FastAPI, Automated Tax Calculation, SQLite, Python, Web Application, CGST, SGST, Financial Automation.*

I. INTRODUCTION

The introduction of Goods and Services Tax (GST) in India in July 2017 unified the country's indirect taxation system, resulting in a significant increase in GST-compliant invoices generated across businesses of all sizes. Managing and extracting information from these invoices manually is time-consuming, error-prone, and inefficient, particularly for small businesses and accounting professionals handling large transaction volumes. To address these limitations, this paper proposes an AI-Enabled GST Invoice Processor — a web-based application that uses Optical Character Recognition (OCR) technology to automatically extract text from uploaded invoice images and PDF files, classify products against a configurable GST rate dataset, compute CGST, SGST, and IGST components accurately, and present results through an interactive analytics dashboard. The system is developed using Python (FastAPI), Tesseract OCR, pypdf, HTML5, CSS3, and JavaScript, and deployed on the Render cloud platform.

II. LITERATURE REVIEW

Several studies in the areas of document automation, OCR-based data extraction, and web-based financial applications provide the foundation for the proposed system. The following review covers key contributions relevant to automated invoice processing and GST-related digital tools.

Manjunath et al. (2023) developed a web-based automated invoice data extraction system using OpenCV image preprocessing combined with a fine-tuned Tesseract OCR engine. Their system was tested on over 25 invoice types and achieved extraction accuracy scores ranging between 85% and 95%. The study confirmed that Tesseract OCR, particularly when pre-processing techniques such as noise reduction and contrast enhancement are applied, is a practical and cost-effective tool for invoice automation in small and medium business environments [1].

Hegghammer (2022) conducted a benchmarking experiment comparing the OCR performance of Tesseract, Amazon Textract, and Google Document AI across multiple document types. The study found that while cloud-based OCR services outperformed Tesseract in noisy environments, Tesseract delivered reliable results for clean, digitally-generated documents and remained the preferred open-source option due to its zero-cost deployment, extensive documentation, and active developer community [2].

Patel (2025) presented the design and implementation of an OCR-powered pipeline for table extraction from invoices using Tesseract OCR combined with OpenCV preprocessing and custom post-processing logic. The study demonstrated that hybrid OCR pipelines significantly improve data extraction accuracy for non-standard invoice layouts and highlighted the importance of modular pipeline architecture for scalability in financial automation applications [3].

A GST Tracking System paper published in IJCRT (2023) described a web-based solution for capturing and storing GST amounts paid by customers using barcode scanning and Excel integration. The study identified that automating GST data capture through digital interfaces substantially reduces data entry errors and improves real-time visibility of tax records for small retailers — directly motivating the need for OCR-based extraction systems targeting invoice documents rather than only point-of-sale transactions [4].

Khanchandani et al. (2026) proposed an automated invoice data extraction pipeline combining Large Language Models (LLMs) with OCR techniques. Their research highlighted that conventional OCR systems face challenges with variable invoice layouts and low-quality scans, and that keyword-based pattern matching combined with regular expressions remains a robust and computationally lightweight alternative for structured invoice formats, particularly in resource-constrained educational and small-business deployments [5].

Collectively, the reviewed literature affirms that OCR-based invoice automation using lightweight Python frameworks and relational databases represents a practical, accessible, and scalable approach to GST invoice management. Existing systems, however, are often proprietary, limited to specific invoice formats, or lack integrated analytics dashboards. The proposed system addresses these gaps by combining open-source OCR extraction, automated keyword-based GST classification, dynamic dataset management, and real-time web dashboard visualization within a single, freely accessible web application.

III. METHODOLOGY

The methodology of the proposed system follows a structured, end-to-end pipeline from invoice ingestion to output visualization. The development approach is modular and iterative, enabling independent testing and enhancement of each processing stage.

A. System Architecture

The application is built on a client-server architecture. The frontend, developed with HTML5, CSS3, and JavaScript using Jinja2 templates, communicates with a Python FastAPI backend that handles all server-side logic. The complete technology stack is presented in Table 1.

Table 1: System Technology Stack

Component	Technology Used	Purpose
Frontend	HTML5, CSS3, JavaScript (Jinja2)	Web interface, form handling, dynamic rendering
Backend	Python (FastAPI Framework)	API routing, OCR logic, business processing

OCR Engine	Tesseract OCR (pytesseract), pypdf	Text extraction from images and PDF invoices
Database	SQLite	Structured storage of invoice and GST records
Server / Hosting	Render Cloud Platform	Online deployment and remote accessibility
Hardware	Intel i5, 8GB RAM, 512GB SSD, Nvidia GTX 3050	Development and testing environment

B. Invoice Input and File Handling

Users upload invoice files through a web-based form interface that accepts JPG, PNG, and PDF formats. Uploaded files are read as binary content in memory and passed directly to the extraction module without permanent server-side storage, reducing overhead and improving data privacy. Input validation ensures that only permitted file types are accepted, and error handling captures any extraction failures gracefully with informative user feedback.

C. OCR-Based Text Extraction

Text extraction is handled by the extract_text() function in the extractor.py module. For PDF invoices, the pypdf library extracts embedded text directly from the document structure — an efficient approach for digitally generated invoices. For image files (JPG, PNG), the Tesseract OCR engine is invoked via the pytesseract Python wrapper to perform character recognition. The extracted raw text string is passed to the classification and amount extraction functions for subsequent processing.

D. GST Classification via Keyword Matching

The find_gst_rate() function implements keyword-based GST classification. It queries the product_gst database table — which stores product keyword-to-GST-rate mappings — and performs a case-insensitive substring search across the extracted invoice text. The first matched keyword's GST rate and category label are returned. If no match is found, the system defaults to 0% GST and flags the record as 'Unknown', ensuring graceful degradation without errors. The dataset is updatable by administrators through CSV uploads without modifying application code.

E. Amount Extraction and Tax Computation

The extract_total_amount() function uses regular expression pattern matching to identify the invoice total from extracted text. It searches for common label patterns such as 'Total', 'Grand Total', 'Amount', and 'Net Payable' followed by numeric values, returning the last matched occurrence (totals typically appear at the bottom of invoices). A fallback mechanism identifies the largest decimal value in the text when no labeled total is detected. Tax computation applies the formula: Tax = Total − (Total ÷ (1 + Rate/100)), back-calculating the GST-exclusive base and deriving the tax portion from the GST-inclusive total. CGST and SGST are each set to half the computed tax for intrastate transactions.

F. Database Design

All processed invoice data is stored in a SQLite relational database comprising six tables. Table 2 summarizes the schema. Foreign key relationships between the invoices, invoice_items, and gst_calculation tables ensure referential integrity and support efficient aggregation queries for dashboard analytics.

Table 2: Database Schema Summary

Table Name	Primary Fields	Key Type	Description
users	user_id, username, email, password	PK, AUTO	Stores user login and account information
admins	admin_id, username, email, password	PK, AUTO	Administrator credentials and access control

invoices	invoice_id, filename, total_amount, tax_amount, gst_rate	PK, AUTO	Processed invoice records with extracted values
invoice_items	item_id, invoice_id, item_name, gst_rate	PK, FK	Item-level details linked to parent invoice
gst_calculation	gst_id, invoice_id, cgst, sgst, igst, total_tax	PK, FK	Computed CGST, SGST, IGST per invoice
product_gst	keyword, gst_rate	PK	Product keyword to GST rate mapping (CSV-updatable)
upload_log	upload_id, file_name, upload_date, status	PK, AUTO	File upload history and processing status tracking

G. Dashboard and Reporting

The dashboard module queries all processed invoice records and computes aggregate statistics including total invoices processed, cumulative extracted amounts, and total GST collected. GST rate distributions are computed server-side and passed to the frontend for visualization using bar chart and line graph components. Users can monitor invoice processing history, review individual extracted results, and analyze tax trends in real time through the web interface.

IV. SYSTEM MODULES

The GST Invoice Processor is organized into four primary functional modules, each serving a distinct role within the overall system.

A. User Module

The User Module provides the primary interface through which end users interact with the system. It includes secure login authentication to restrict access to authorized personnel. Users can upload invoice files in image or PDF format and immediately receive processed results on the dashboard. The module also allows users to review extracted invoice details such as vendor information, invoice amounts, GST classification, and computed tax values for quick verification of accuracy.

B. Processing Module

The Processing Module forms the computational core of the system. It receives uploaded invoice files and routes them through the OCR extraction pipeline, GST keyword classification, regex-based amount extraction, and tax computation using the back-calculation formula. The module supports both digitally-generated PDFs (via pypdf) and scanned image invoices (via Tesseract), applying the appropriate extraction method automatically based on file type. All processing is performed server-side via FastAPI route handlers.

C. Admin Module

The Admin Module provides elevated administrative controls for system management. Administrators can log in with dedicated credentials, view all processed invoices across all user accounts, and monitor system performance. The module supports GST dataset management, allowing administrators to upload updated product-to-GST-rate CSV files that immediately update the classification logic without application redeployment. Summary usage reports are accessible through the admin interface.

D. Database and Reporting Module

The Database and Reporting Module manages all data persistence and retrieval operations, interfacing with the SQLite database to store, query, and update invoice records. The module supports report generation by aggregating invoice data into summary views rendered on the dashboard. Historical invoice records can be retrieved and reviewed at any time, ensuring traceability of all processed financial documents.

V. SYSTEM TESTING

Comprehensive testing was conducted across five levels to validate the functionality, accuracy, security, and usability of the system. Table 3 summarizes the test cases executed and their outcomes.

Unit testing confirmed that individual components — including login authentication, file upload handling, OCR text extraction, GST keyword classification, and tax computation — performed correctly in isolation. Integration testing verified that the complete data pipeline from invoice upload through OCR extraction to database storage and dashboard rendering operated without errors. Sequential processing of all four sample invoices during system testing confirmed stable multi-invoice handling. Security testing validated that invalid file formats were rejected and that access controls functioned correctly. User acceptance testing confirmed that the interface was intuitive and that results were clearly presented to end users without requiring technical knowledge.

Table 3: Test Cases and Outcomes

Test Type	Module Tested	Test Case / Condition	Outcome
Unit Testing	User Login	Valid credentials entered	✓ Passed
Unit Testing	File Upload	JPG invoice uploaded — OCR extraction triggered	✓ Passed
Unit Testing	GST Classification	Product keyword matched from dataset	✓ Passed
Unit Testing	Tax Computation	Back-calculation of CGST and SGST from total	✓ Passed
Integration Testing	Full Pipeline	Upload → OCR → Classification → DB Storage	✓ Passed
Integration Testing	Dataset Update	CSV uploaded → new keywords active in classification	✓ Passed
System Testing	Multi-Invoice	Four invoices processed sequentially without errors	✓ Passed
Security Testing	Input Validation	Non-PDF/image file upload rejected with error message	✓ Passed
UAT	End-to-End Flow	User completed upload-to-dashboard flow independently	✓ Passed

VI. RESULTS AND DISCUSSION

The system was evaluated using four sample GST-applicable PDF invoices. Table 4 presents the processing results, and Table 5 provides a comparison between the proposed system and traditional manual invoice processing methods.

Table 4: Invoice Processing Results

Invoice File	File Type	Ext. Amount (₹)	GST Rate (%)	Tax (₹)	Status
Invoice_1.pdf	PDF	64,900.00	5%	3,090.48	Processed
Invoice_2.pdf	PDF	91,000.00	12%	9,750.00	Processed
Invoice_3.pdf	PDF	94,000.00	18%	14,338.98	Processed
Invoice_4.pdf	PDF	95,000.00	28%	20,781.25	Processed
Overall	—	3,44,900.00	—	47,960.71	100% Success

All four invoices were processed successfully with a 100% completion rate and no manual intervention at any stage. The system correctly classified and applied different GST rates across all four invoices — 5% for Invoice 1 (essential goods category), 12% for Invoice 2 (standard goods category), 18% for Invoice 3 (services category), and 28% for Invoice 4 (luxury goods category) — demonstrating the system's ability to handle the full range of GST rate slabs. Tax computation results were verified manually and confirmed to be accurate using the back-calculation formula. Total GST computed across all four invoices amounted to ₹47,960.71 on a combined extracted invoice value of ₹3,44,900.00. The dashboard displayed all results in real time immediately after each upload, with bar charts and line graphs providing clear visual summaries of tax and total amount trends across processing dates.

Table 5: Proposed System vs. Manual / Existing Approach

Feature	Manual / Existing Approach	Proposed System
Data Entry	Manual — time-consuming, error-prone	Automated via OCR — no manual typing
GST Classification	Requires accountant judgment per invoice	Keyword-based auto-matching from dataset
Tax Calculation	Calculator / spreadsheet required	Automatic CGST / SGST / IGST computation
Data Storage	Physical files or unstructured folders	Structured SQLite relational database
Analytics	Not available	Interactive dashboard with real-time charts
Processing Speed	5–15 minutes per invoice (manual)	Under 10 seconds per invoice (automated)
Scalability	Limited — increases workload with volume	Scalable — CSV-updatable, cloud-deployable
Error Rate	High — human calculation errors common	Low — automated formula-based computation

The results demonstrate that the proposed system substantially reduces the time and effort required for GST invoice processing. OCR-based extraction eliminates manual data transcription, and keyword-based classification eliminates the need for accountant-level judgment for routine invoice categories. Cloud deployment on the Render platform ensures system accessibility across devices and geographic locations, extending practical applicability beyond single-workstation environments. The CSV-updatable dataset mechanism allows the system to remain current with evolving GST rate schedules without application code changes.

VII. CONCLUSION

This paper presented an AI-enabled GST Invoice Processor — a web-based application that automates the extraction, classification, and storage of GST invoice data using Optical Character Recognition technology. The system was developed using Python (FastAPI), Tesseract OCR, pypdf, SQLite, and HTML5/CSS3/JavaScript with Jinja2 templating, and deployed on the Render cloud platform. The application successfully automates the complete invoice processing pipeline from file upload to tax computation and real-time dashboard visualization, demonstrating the practical feasibility of applying lightweight automation and AI-assisted extraction techniques to financial document management challenges faced by small businesses and accounting professionals in India.

Future work may include integration of advanced OCR models such as EasyOCR or PaddleOCR to improve extraction accuracy for handwritten and low-quality scanned invoices, incorporation of multi-language invoice support, direct API integration with the GST e-Invoice portal for IRN-based compliance, anomaly detection for identifying potentially erroneous or duplicate invoices, and role-based multi-user access management for enterprise deployments.

REFERENCES

1. Manjunath, A. A., Nayak, M. S., Nishith, S., Pandit, S. N., Sunkad, S., Deenadhayalan, P., and Gangadhara, S. (2023). Automated Invoice Data Extraction Using Image Processing. IAES International Journal of Artificial Intelligence, Vol. 12, No. 2, pp. 514–521. ISSN: 2252-8938. DOI: 10.11591/ijai.v12.i2.pp514-521.

2. Hegghammer, T. (2022). OCR with Tesseract, Amazon Textract, and Google Document AI: A Benchmarking Experiment. Springer — Quality & Quantity, Vol. 56, pp. 4275–4289. DOI: 10.1007/s11135-021-01268-4.

3. Patel, P. D. (2025). Design and Implementation of an OCR-Powered Pipeline for Table Extraction from Invoices. arXiv preprint arXiv:2507.07029. Available at: https://arxiv.org/abs/2507.07029.

4. Students of VCETP, Puttur. (2023). GST Tracking System Using Barcode Scanning and Web Interface. International Journal of Creative Research Thoughts (IJCRT), Vol. 11, Issue 6, June 2023. ISSN: 2320-2882. Available at: www.ijcrt.org/papers/IJCRTX020016.pdf.

5. Khanchandani, K., Thakur, A., Shetty, A., Reddy, C., and Behera, R. (2026). Automated Invoice Data Extraction: Using LLM and OCR. arXiv preprint arXiv:2511.05547. Available at: https://arxiv.org/abs/2511.05547.

6. Government of India — GSTN. (2020). GST e-Invoice System — Electronic Invoice Reporting to IRP. Available at: https://einvoice1.gst.gov.in. [Introduced October 2020, phased rollout through 2023].

7. FastAPI Official Documentation. (2024). Building Fast and Modern APIs with Python. Available at: https://fastapi.tiangolo.com.

8. Tesseract OCR — Official Documentation. (2024). Open Source OCR Engine. Available at: https://tesseract-ocr.github.io.

9. Python Software Foundation. (2024). Python Programming Language — Official Documentation. Available at: https://docs.python.org/3.

10. SQLite. (2024). SQLite Database Engine — Official Documentation. Available at: https://www.sqlite.org/docs.html.