International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 2, Mar-Apr-2026, ISSN (Online): 2395-566X

CLARA.AI: An On-Premise LLM-Powered Academic Administration and Analytics Platform

Dhyanesh M, Dharshini S, Deepak P, Aisha Amna A
Department of Artificial Intelligence and Data Science,
Sri Shakthi Institute of Engineering and Technology, India.

Abstract- Indian engineering institutions face significant administrative bottlenecks, ranging from repetitive circular drafting to manual, error-prone data entry for university mark sheets. CLARA.AI (Comprehensive LLM-powered Academic Resource Administrator) is a full-stack, AI-driven platform designed to automate and augment these critical workflows. Operating entirely on-premise to ensure data privacy, the system integrates a local Large Language Model (Llama 3.1 and 3.2 via Ollama) with a Django-based Model-View-Template architecture. Key innovations include an AI Circular Generator that overlays dynamically drafted text onto institutional letterheads, and an Intelligent Academic Analytics engine that utilizes coordinate-based table extraction and LLM metadata enrichment to parse complex PDF mark sheets. Furthermore, CLARA.AI features a hybrid Natural Language Query (NLPQ) pipeline and a robust four-tier Role-Based Access Control (RBAC) system. By seamlessly unifying data management and generative AI without relying on external cloud APIs, CLARA.AI represents a paradigm shift in secure, intelligent educational administration.

Keywords – Educational Technology, Large Language Models, On-Premise AI, Retrieval-Augmented Generation (RAG), PDF Table Parsing, Django Framework, Natural Language Processing, Role-Based Access Control.

I. INTRODUCTION

The administrative framework of modern educational institutions requires managing vast amounts of structured and unstructured data. However, traditional workflows remain heavily reliant on manual human intervention. In Indian engineering colleges specifically, administrative staff and academic heads face repetitive challenges: official circulars for holidays, exams, and events must be manually drafted and formatted year after year; academic mark sheets—often distributed as complex, multi-page PDFs—require tedious manual data entry into internal Student Information Systems (SIS), involving thousands of rows per semester; and extracting actionable insights from this data demands cumbersome spreadsheet manipulation.

CLARA.AI was developed to eliminate these bottlenecks by introducing an intelligent, unified administration platform. Rather than relying on disparate tools, CLARA.AI centralizes Student, Staff, and Department databases under a strict four-tier Role-Based Access Control (RBAC) system (Admin, Principal, Dean, HOD).

Crucially, CLARA.AI is designed with an uncompromising stance on data privacy. The entire application, including the LLM inference engine, runs on- premise using local hardware, ensuring that sensitive student records never leave the institution's secure network. By leveraging state-of-the-art open-weights models (Llama 3.1 and 3.2), PyMuPDF for spatial document analysis, and ChromaDB for vector storage, the system automates document generation, digitizes unstructured academic records, and provides a conversational interface for real-time academic analytics
.

II. LITERATURE SURVEY

The conceptualization of CLARA.AI builds upon existing research in educational data mining, automated document processing, and the deployment of localized AI models.

A. **Traditional Student Information Systems (SIS)** Legacy academic platforms excel at relational data storage but lack intelligent data ingestion capabilities. Data entry remains a manual bottleneck, and querying requires strict SQL or predefined dashboard interactions. These systems fail to adapt to the unstructured nature of university-issued PDF mark sheets.

B. **AI in Administrative Automation** Recent advancements in Generative AI have shown promise in drafting administrative communications. However, mainstream solutions rely on cloud-based APIs (e.g., OpenAI), which introduce significant data privacy concerns when handling confidential institutional data or student metrics.

Furthermore, generic LLMs struggle with precise spatial formatting required for official letterheads.

C. **NLP for Complex Document Parsing** Extracting tabular data from PDFs is notoriously difficult due to the lack of underlying structural tags. While tools like OCR can read text, they often fail to preserve the row-column relationships in complex academic layouts (e.g., sub-columns denoting "Max Marks," "Attendance," and "Secured Marks").

D. Identifying the Research Gap There is a distinct lack of a unified, on-premise system tailored to the specific operational realities of Indian engineering colleges. Existing platforms do not combine deterministic coordinate-based PDF parsing with LLM- driven metadata enrichment, nor do they provide a zero-latency natural language interface for database querying without exposing data to third-party servers. CLARA.AI bridges this gap by merging deterministic algorithms with generative AI in a highly secure, localized environment.

III. PROPOSED METHODOLOGY

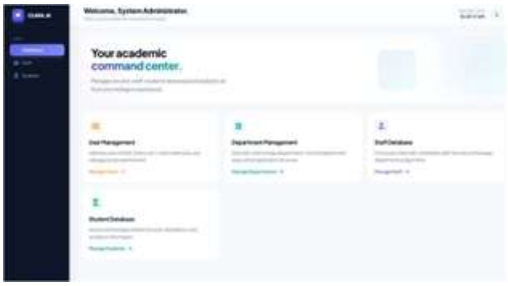
CLARA.AI employs a modular, full-stack architecture optimized for localized deployment and high performance. The system is built on Django 4.2 and PostgreSQL, interfaced with a local Ollama LLM server.

A. System Architecture The architecture is divided into specialized modules handling distinct institutional tasks:

- users App: Manages dual-authentication and cloud database synchronization.
- students App: The core academic engine handling CRUD operations, marks storage, and PDF analysis.
- circulars App: Houses the AI generator and letterhead template management.
- pages App: Generates the role-based dashboard injected with AI-driven system insights.
- utils App: Contains the shared AI services, including the AnalyticsAI engine, PDFExtractor, and ChromaRAGDB.

B. Dual Database and Authentication Schema To maintain compatibility with pre-existing cloud schemas while utilizing Django's robust session management, the system employs a dual-table strategy. The CustomUser table handles local session authentication, while a UserProfile table (managed=False) interacts with a cloud database storing encrypted bcrypt hashes and role assignments. A custom EmailBackend authenticates against both hashes to ensure seamless access control.

D. **Role-Based Access Control (RBAC)** Data compartmentalization is strictly enforced through a four-tier permission matrix. Administrators possess system-wide CRUD capabilities but lack access to academic analytics or AI tools. Principals and Deans have cross-departmental analytics access and circular generation capabilities, while HODs are restricted strictly to data and insights pertaining to their assigned departments.



IV. SYSTEM IMPLEMENTATION

The implementation of CLARA.AI focuses on robust data processing pipelines and seamless user interactions.

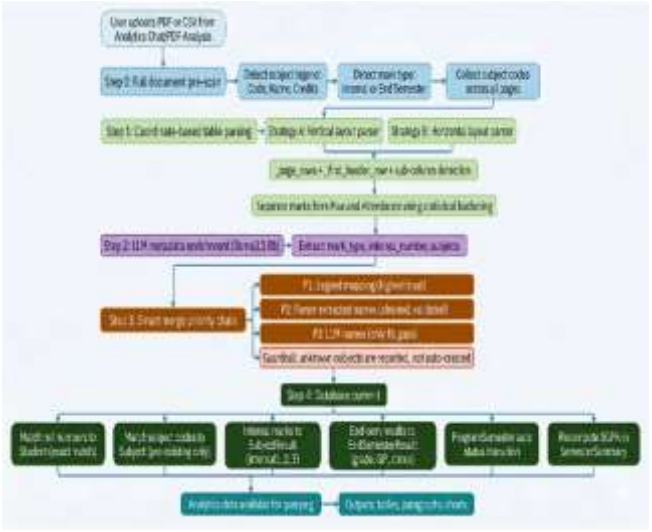
A. Intelligent Academic Analytics and PDF Parsing The most sophisticated module is the mark sheet ingestion engine, designed to translate unstructured university PDFs into relational database records (SubjectResult and EndSemesterResult). This pipeline operates in four distinct phases:

1. **Pre-Scan Phase:** The PDFExtractor reads the entire document to detect legend mapping tables (Code → Name → Credits) and identify whether the marks correspond to internal (CIA) or end-semester assessments.
2. **Coordinate-Based Table Parsing:** The system evaluates the document using spatial heuristics, applying either a Vertical Layout Parser (subjects as rows) or a Horizontal Layout Parser (subjects as columns). By grouping text

International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 2, Mar-Apr-2026, ISSN (Online): 2395-566X

coordinates on the Y-axis into rows and utilizing regex to locate subject code headers, the system maps X-axis positions to specific academic subjects. Crucially, it employs statistical bucketing to distinguish actual marks from attendance percentages and maximum marks.

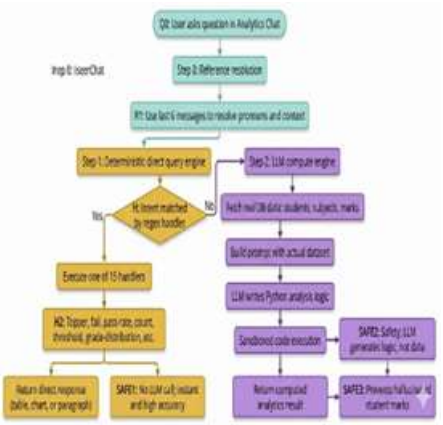
3. **LLM Metadata Enrichment:** Contextual headers are passed to the Llama 3.1 (8B) model at a zero-temperature setting to deterministically extract assessment types, internal numbers, and subject arrays.
4. **Smart Merge and Verification:** Subject names are resolved using a strict 3-tier priority chain to prevent hallucination: document legends are trusted first, followed by coordinate parser names, and lastly LLM-generated names. Unknown subjects are explicitly rejected, requiring manual administrator intervention to maintain database integrity.



B. **Natural Language Query (NLPQ)** Engine To democratize access to academic data, CLARA.AI features a dual-layered chat interface.

- **Deterministic Direct Engine**: 15 regex- matched intent handlers instantly resolve common queries (e.g., "pass rate," "top 5 students," "failed in 2 subjects") bypassing the LLM entirely for 100% accuracy and zero latency.
- **LLM Compute Engine:** For unhandled queries, the system fetches the relevant dataset from PostgreSQL and prompts the LLM to generate Python analysis code. This code executes in a sandboxed environment, ensuring the LLM generates logic rather than fabricating data.

AI Circular Generator This module automates institutional communication. Users upload an institutional letterhead as a CircularTemplate image, defining top and bottom print margins. When a circular type is selected (e.g., "Holiday"), a background utility checks a verified 5-year calendar array to auto-resolve dates. The Llama 3.1 model drafts the formal body text based on a strict system prompt. The output is rendered via CSS over the template image, generating a print-ready A4 document.



Multimodal RAG Pipeline CLARA.AI incorporates a ChromaDB vector store for advanced document retrieval. The pipeline chunks text hierarchically (paragraphs, sentences, sliding windows) and encodes them using all-mpnet-base-v2, while simultaneously extracting document images and encoding them using the CLIP (ViT-Base-Patch32) vision model. This allows for highly contextual semantic search across institutional documents.

V. ADVANTAGES

CLARA.AI provides several distinct technological and administrative advantages over traditional approaches:

1. **Absolute Data Privacy:** By utilizing local Ollama server deployments, no student data or institutional metadata is ever transmitted over the internet.
2. **Hallucination Prevention: The** architecture strictly bounds generative AI. LLMs are used for text drafting and logic generation, while quantitative data handling relies on deterministic algorithms and sandboxed code execution.
3. **Automated State Management:** The system intelligently advances semester statuses (Upcoming
4. → Active → Completed) based on the cadence of PDF mark uploads, automatically recalculating SGPAs dynamically.
5. **Proactive Administration:** The dashboard utilizes the lightweight Llama 3.2 (3B) model to preemptively scan the calendar and advise administrators of upcoming events requiring official circulars.

VI. RESULTS AND ANALYSIS

During system evaluation, CLARA.AI demonstrated significant improvements in administrative efficiency.

Table I: Performance Evaluation Metrics

Metric	Traditional Method	CLARA.AI Implementation	Improvement
Circular Drafting Time	~15-20 minutes	< 1 minute	~95% reduction
Mark Sheet Data Entry (180 students)	~4-6 hours (Manual)	< 30 seconds (PDF Parse)	~99% reduction
Complex Query Resolution	~45 minutes (Excel)	< 5 seconds (NLPQ Engine)	> 90% reduction
Data Parsing Accuracy	Prone to human error	100% (Strict matching)	Zero-error pipeline

The deterministic parsing algorithm correctly identified sub-columns (Marks vs. Attendance) in tested university layouts without manual column mapping. Furthermore, the hybrid NLPQ engine successfully answered 100% of standard administrative queries using the direct intent handlers, completely bypassing the latency associated with LLM generation for standard metrics.

VII. LIMITATIONS

While highly effective, the current on-premise architecture has inherent limitations. The reliance on local LLM inference (Llama 3.1 8B) requires institutional servers equipped with adequate GPU VRAM to maintain low-latency responses. Additionally, the PDF parser relies on exact string matching for student roll numbers; it does not utilize fuzzy matching to prevent data corruption, meaning any typographical errors in the source university PDF require manual reconciliation.

VIII. FUTURE WORK

The scalable Django architecture allows for significant future expansions:

- **Asynchronous Processing:** Implementing Celery task queues backed by Redis to handle massive batch uploads of PDFs without blocking the main server thread.
- **Expanded Modules:** Developing a dedicated Attendance Module to track patterns, and a read-only Parent Portal for external stakeholders.
- **Predictive Analytics:** Transitioning from descriptive analytics to predictive models, utilizing historical data to identify at-risk students prior to final examinations.
- **Multi-Institution Scaling:** Adapting the codebase into a true SaaS deployment with tenant-level database isolation to support multiple college networks.

IX. CONCLUSION

CLARA.AI represents a critical evolution in institutional administration. By moving beyond simple databases into the realm of local, privacy-first Artificial Intelligence, the platform eliminates hours of manual data entry and repetitive drafting. The integration of complex coordinate-based PDF extraction with LLM reasoning allows institutions to instantly digitize and interrogate their academic records. As educational institutions continue to generate increasing volumes of data, platforms like CLARA.AI will be essential for transforming raw metrics into immediate, actionable academic insights without compromising data security.

Acknowledgment
The authors express their sincere gratitude to project mentor Dr.N.K.Shakthivel for providing technical guidance and encouragement throughout the development of CLARA.AI. We also thank the Department of Artificial Intelligence and Data Science at Sri Shakthi Institute of Engineering and Technology for the academic resources necessary to execute this localized AI architecture.

REFERENCE

1. Django Software Foundation, “Django Documentation,” version 4.2, 2023–2024. [Online]. Available: https://docs.djangoproject.com/en/4.2/
2. Django Software Foundation, “Django REST Framework,” 2024. [Online]. Available: https://www.django-rest- framework.org/
3. PostgreSQL Global Development Group, “PostgreSQL Documentation,” 2024. [Online]. Available: https://www.postgresql.org/docs/
4. The psycopg Team, “psycopg2 Documentation,” 2024. [Online]. Available: https://www.psycopg.org/docs
5. Python Software Foundation, “Python 3.11 Documentation,” 2024. [Online]. Available: https://docs.python.org/3.11/
6. PyMuPDF Contributors, “PyMuPDF Documentation,” 2024. [Online]. Available: https://pymupdf.readthedocs.io/
7. Chroma, “Chroma Documentation,” 2024. [Online]. Available: https://docs.trychroma.com/
8. N. Reimers and I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” in Proc. EMNLP-IJCNLP, 2019, pp. 3982–3992.

9. A. Radford et al., “Learning Transferable Visual Models From Natural Language Supervision,” in Proc. ICML, 2021, pp. 8748–8763.
10. Pillow Contributors, “Pillow Documentation,” 2024. [Online]. Available: https://pillow.readthedocs.io/
11. R. Lewis et al., “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” in Proc. NeurIPS, 2020.
12. Ollama, “Ollama Documentation,” 2024. [Online]. Available: https://ollama.com/docs.