# A Retrieval-Augmented Multi-Agent Framework for Intelligent and Fair University Admissions

Archita Dasgupta[1*], Vasundhra Nahata[1], Shreyashee Das[1], Saikat Dutt[1]

[1*]Department of Computer Science and Engineering, Institute of Engineering and Management, Kolkata,India.

*Corresponding author(s). E-mail(s): archita.dasgupta2022@iem.edu.in;
Contributing authors: vasundhra.nahata2022@iem.edu.in;
shreyashee.das2022@iem.edu.in; saikat.dutt@iem.edu.in;

**Abstract**

The application process for most universities is still primarily dependent on human assessments; therefore, it may include inefficient processes, inconsistent judgments, and unclear processes for applicants. This paper outlines the development of an Agentic AI Framework utilizing a Retrieval Augmented Generation (RAG) system to transform the way universities use intelligent automation to assist in their admissions processes; specifically, by developing a combination of automated technologies that will enable document processing, semantic reasoning, query answering and multi-agent coordination to help resolve many of the issues associated with fairness and efficiency within university admissions. Within our proposed model are multiple AI Agents; each agent will be responsible for completing different tasks; e.g., document validation, applicant eligibility determination and applicant feedback. The agents will coordinate their actions based on the institutional data and policies used to ensure consistency and informed decision making through the use of data. We believe that integrating a semantic search and a RAG enables the concept of mass admissions to become a viable option. We are attempting to illustrate the manner in which AI can benefit Educational Administration through both task automation and the creation of scalable, equitable, and understandable decisions regarding university admissions.

**Keywords:** Agentic AI, Large Language Models(LLMs), Retrieval Augmented Generation(RAG),Student Admission System, CrewAI, PineconeDB

# 1 Introduction

Although rapid technological advancements are being experienced within the areas of finance, healthcare and transportation, traditional university admission processes in India continue to be largely dependent upon manual labor. The long-established processes used to manage university admissions which include long paper-forms that require time-consuming filling, slow updates and untransparency into the admissions process create considerable anxiety and uncertainty for students; and as such provide them with minimal insight into the status of their application[1]. University administrators also struggle to manage thousands of student applications with small numbers of administrative personnel leading to bottlenecks in verifying documents; inconsistencies in evaluating applicants; and limiting the ability of a university to scale during peak admission times. In order to alleviate these issues, we suggest a fully integrated, AI-based automated university admission process to enhance the level of transparency; speed up the processing time of applicant data; and enable fair and consistent evaluation of each applicant.

The system's modularity, explainability, and scalability represents a significant advancement in automating educational administration in a fair and efficient manner. Built on the principles of Responsible AI, our system uses Retrieval-Augmented Generation (RAG) to ensure that LLM responses are based entirely on the institutional policy, while Explainable AI (XAI) [2] techniques enhance transparency in eligibility assessment, shortlisting, and feedback generation.

# 2 Background Study

Automated administration of higher education has drawn attention to digital systems as well as AI for the decision-making process through several other researches on this topic. For example, L. Garg et al. [1] developed an online admission management system to allow the institutional workflow to run efficiently. Next, Md. T. Hassan [5] took a more advanced approach by developing a data-driven admissions platform using BERT and XGBoost for evaluating candidates fairly and at scale. Also, B. Sankarasubramanian et al. [6] developed a neural network-based admission model to reduce bias and improve efficiency. Additionally, Alothman et al. [7] used deterministic algorithms combined with machine learning classifiers (KNN, Decision Tree, Random Forest) to automate student categorization and develop an unbiased GPA-based ranking method. Finally, S. Baviskar et al. [3], created a multilingual RAG-enabled chatbot which responds continuously and personalized; therefore, the workload of staff could be minimized. Further to this, M. A. Kuhail et al. [4], developed MyAdvisor - a chatbot advising system that was found to possess good usability and functional clarity. Furthermore, in addition to admissions and advising, Dwivedi et al. [8] proposed an overall AI-driven school management system that includes both predictive analytics and a chatbot interface; to provide better administrative decisions as well as better individualized support for students.

# 3 Methodology

## 3.1 Problem Statement

- Administrators manually check documents provided by candidates, resulting in errors and inefficiency.
- Handmade decisions for admitting students are time-consuming and biased.
- Candidate feedback relies on manual processes, which delays and lacks consistency.
- Prolonged waiting times for admission decisions cause concern and anxiety among students.
- each educational institution has their own set of eligibility criteria, admission methods and communication protocols our system combines organizational knowledge with external data like student information to take decisions

## 3.2 System Architecture

In Indian universities, admission eligibility depends on the program type. Most academic courses depend a student's 12th-grade marks or percentage, while engineering admissions require qualifying national, state, or institute-level entrance exams such as JEE Mains. At the interface level, applicants submit personal details (name, date of birth, gender, email, phone number), academic information (12th-grade marks and entrance exam scores), and supporting documents such as scorecards and address proof. The system supports both scanned and digitally accessible documents, providing flexibility in document submission. The data of the students will be read dynamically from

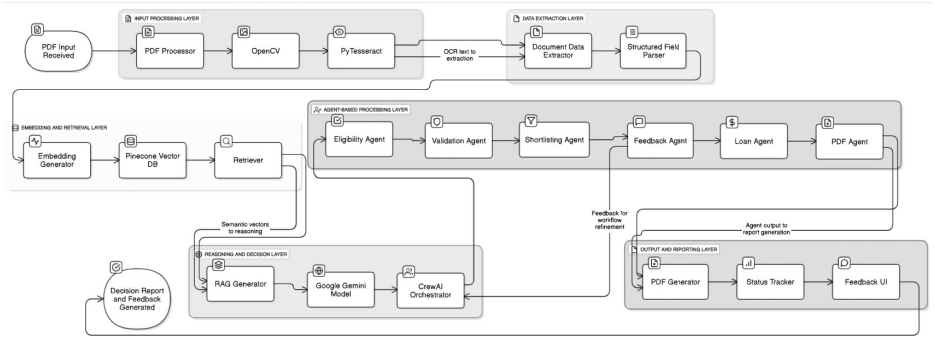


**Fig. 1** System Architecture

the submitted form data and then this data will be converted into vector embeddings with the help of Google Gemini-Embedding-001 model[9] [10] which will further be stored in a vector database, we used Pine cone DB in our work. PineconeDB is a fully managed vector database designed for high performance similarity search and retrieval, enabling efficient handling of vector embeddings in AI and machine learning workflows.[11] A multi-agent [12] architecture has been implemented in the backend by invoking the Gemini-1.5-pro model [10] to intelligently automate the process of manually verifying the documents and taking decision as to if a particular applicant

will be selected or not by comparing their data with the eligibility criteria of the university. Additionally, for the selected applicants,the system also evaluates their family income to determine loan eligibility, ensuring financial support is offered to those in need and also it generates a personalized feedback report for all applicants that will be sent via email or through downloadable in PDF files upon querying from the user interface. Additionally it has a dashboard accessible to only the director to review the applications and take the final decision about the admission process.The detailed breakdown of the architecture of our system is as below:

### 3.2.1 Document Processing

The applicant uploads a document(like class 12 mark sheets/JEE Mains rank card), if the document is digitally available PDF a simple text extraction with PyMuPDF is performed , else if it is a scanned PDF then it is converted back to an image and image processing is performed with OpenCV and the text is extracted by Optical Recognition Character(OCR) with Tesseract engine[13].The raw text is obtained and is processed with the Gemini Embeddings Model. The data from the document is validated with the data submitted by the applicant. If the data is matched successfully then application is updated else error is shown

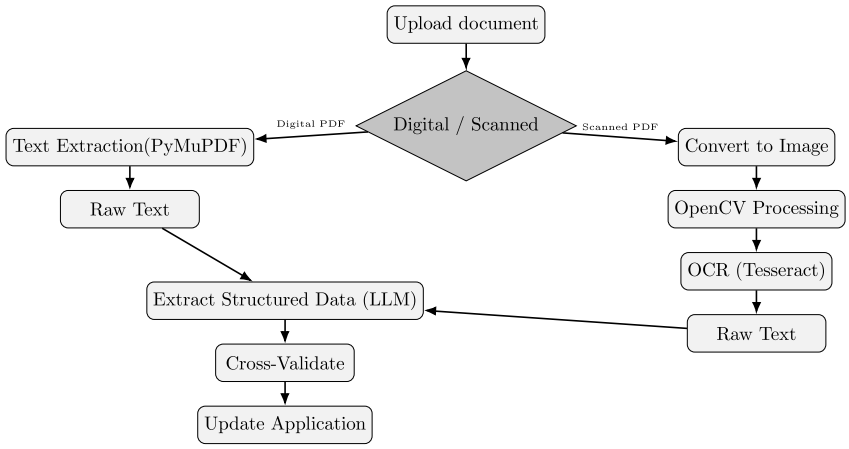


**Fig. 2** Document Processing Workflow

### 3.2.2 RAG and its use in our system

Incorporating Retrieval Augmented Generation(RAG) into the admission process escalates the efficient use of vector databases. This allows for precise matching, semantic search, and enriching context for AI-powered decision-making. RAG can work with structured databases but however integrating an LLM into a structured database like SQL can be tiring so the structured data is converted to flattened data and further stored it in a vector Db like PineconeDB[11] to ensure smooth working after integrating with an LLM by using vector search, the system can identify applicants with similar characteristics or past admission results ensuring a more personalized and contextually aware decision making process. As an applicant submits the application, the system

extracts and generates embeddings for the submitted documents. These embeddings and metadata are stored in the db for reference and querying.Additionally, we provide a guardrail mechanism by feeding institutional admission policies into the RAG pipeline, ensuring that every LLM call yields consistent, policy-aligned predictions and upholds the standards of responsible and transparent AI usage.

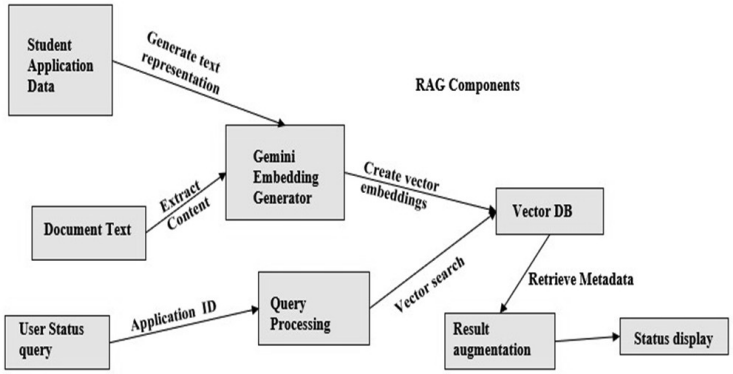


**Fig. 3** RAG components and workflow

### 3.2.3 Multi-Agent Architecture

Automated AI systems or Agentic AI, complete autonomous tasks to achieve complex goals without human intervention. They have characteristics which include autonomy, decision making ability, and adaptability to be able to function well in dynamic and rapidly changing environment [14]. In order to create this Multi-Agent System we utilized the Crew-AI Framework [15] to provide an easier way of developing agentic workflows, by automating processes, defining agent roles, and by providing seamless integration of tools. The simplicity of the design of Crew-AI provides it with a competitive advantage over other agentic frameworks through established control and modularity. We developed six separate agents to provide the correct division of the workflow to prevent overlapping, maintain proper levels of specialization, and ensure full automation of all critical phases of the application process from document receipt to the provision of decision feedback. The six agents, each having their own unique characteristics, were designed to produce modularity, efficiency, and explainability throughout the entire admission process. These six agents work independently and utilize already verified data that has been passed down to enable consistent and quality decisions.

## 4 Results and Discussions

Our system efficiently proves that smart automation can transform and automate the university admission process. Our solution uses Google Gemini Embeddings[10] for semantic document representation, PineconeDB for efficient storage and retreival of data and Gemini 1.5 pro model for document processing and decission making.Our

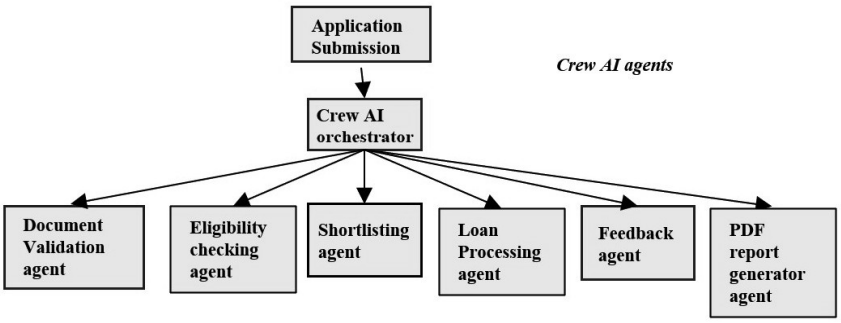


**Fig. 4** CrewAI agents

**Table 1** Agents, Their Roles, and Descriptions

| Agent | Role | Description |
|---|---|---|
| Validation Agent | Validator | Ensures submitted application data is complete, correctly formatted, and logically consistent. Checks missing entries, verifies formats and solves basic inconsistencies before forwarding cleaned data. |
| Eligibility Agent | Eligibility Checker | Evaluates academic criteria, validates required qualifications, filters ineligible cases and labels each application as Eligible or Ineligible with justification. |
| Shortlisting Agent | Shortlister | Ranks eligible candidates based on performance metrics and reservation rules to generate a shortlist based on priority. |
| Loan Processing Agent | Loan Processor | Assesses financial-aid applicants using financial indicators and scoring rules, producing decisions such as Approved, Rejected, or Review Needed. |
| Feedback Agent | Feedback Generator | Converts decision outcomes into personalized explanations, informing applicants of their admission/loan status and suggesting improvements where relevant. |
| PDF Generation Agent | Report Generator | Compiles final decisions and feedback into a structured downloadable PDF report for communication and record-keeping. |

multi agent system can read a vast range of documents- both digitally created PDFs or scanned documents through an adaptive pipeline that uses PyMuPDF for direct text reading and Tesseract OCR with OpenCV for recognizing scanned documents. One of the primary benifits of our system is its ability to reformat raw,unstructured data into clean structured data representations. With the use of proper prompts and fallback mechanisms, the AI agents were successful in correctly extracting the applicant's details, all in JSON format. Automatic validation against user-submitted form data was carried out, mismatches being flagged for inspection. This guarantees efficiency and integrity in the data verification process. The pipeline has been experimented with different university policies(eligibily criterias), and the system has shown highly precise results in both text extraction and information parsing. Gemini embeddings

also enabled contextual understanding of documents, making the system resistant to small layout changes and discrepancies in document formatting.

**Table 2** Evaluation Summary Across 15 University Policies

| Univ | Tested | Matched | Mis |
|---|---|---|---|
| 01 | 8 | 8 | 0 |
| 02 | 7 | 6 | 1 |
| 03 | 6 | 5 | 1 |
| 04 | 5 | 5 | 0 |
| 05 | 8 | 6 | 2 |
| 06 | 7 | 7 | 0 |
| 07 | 6 | 6 | 0 |
| 08 | 6 | 4 | 2 |
| 09 | 7 | 6 | 1 |
| 10 | 5 | 5 | 0 |
| 11 | 7 | 6 | 1 |
| 12 | 6 | 5 | 1 |
| 13 | 5 | 4 | 1 |
| 14 | 6 | 5 | 1 |
| 15 | 11 | 10 | 1 |
| **Total** | **100** | **88** | **12** |

**Table 3** Case Study: Manual vs System Decisions (Univ-05)

| ID | Manual | System | Remarks |
|---|---|---|---|
| A027 | Acc. | Rej. | 83%, below 85% cutoff. |
| A014 | Rej. | Acc. | Reserved quota applied. |
| A009 | Acc. | Acc. | Strong academics. |
| A011 | Acc. | Acc. | Fully eligible. |
| A018 | Rej. | Rej. | Below cutoff, missing docs. |
| A020 | Wait | Wait | Borderline case. |

As demonstrated in Table 2, the system achieved an 88% accuracy with manual reviewers across 15 diverse university policies, highlighting its ability to generalize and adapt to varying institutional rules. Additionally, Table 3 presents a focused case study under Univ-05, where out of six sample students, four decisions matched perfectly with human judgment. In the remaining two cases, the system either strictly enforced policy cutoffs or correctly applied quota logic that was overlooked manually. These outcomes further reinforce the system's consistency, fairness, and explainability in real-world admission scenarios. In conclusion,our system not only makes the task easy and efficient, but also makes the process more transparent, consistent and scalable when large number of applications are considered. We have verified this by setting the 'verbose' parameter in CrewAI as True —this allowed us to trace agent interactions, observe decision paths, and validate that the system performed as expected at each stage. It also helped us detect and minimize potential hallucinations[16] by ensuring that each agent's output remained grounded in the provided institutional context and not some arbitrary generation. It lays a strong foundation for future potential for 100% automated decision-making, upon which the Director's role can be recast to reviewing and approving AI-led recommendations.This strikes the proper balance between automation and accountability, particularly in the sensitive area of student admissions.

# 5 Future Scope

The present version of our system is capable of ranking applicants by their qualifications, however the ultimate decision of admitting a student will remain with the Director. While having a safeguard layer to protect the organization from potential mismanagement or liability issues could be very beneficial, the creation of a bottleneck

in the director's ability to efficiently manage the admissions process has become a major issue. The system has greatly reduced the director's workload by providing a user friendly interface allowing the director to search for candidates based upon their rankings as well as validating documentation. The admissions process for students is a very sensitive area and the current system has achieved the perfect balance of automation and accountability.

# 6 Conclusion

Using AI to manage processes (such as verifying documents, checking applicant qualifications, and recommending individual applicant outcomes) ensures that all aspects of an applicant's admissions process are fair, accessible and easy-to-understand. The combination of Semantic Vector Storage and Contextual Reasoning allow for a very high level of accuracy and quality when it comes to making decisions regarding applicant acceptance; further, the rapid roll-out of the first user-friendly interface will help alleviate the administrative burden on users and will allow applicants to immediately gain access to the insights needed. Overall, the Student Admission AI System was created to develop an even more efficient, transparent, and data-driven method of how educational institutions evaluate applicants. In the near future, our intention is to deploy the Student Admission AI System at numerous educational institutions and expand the functionality of the system to support multiple languages and adaptively learn and improve the decision-making process for students. Further, we plan to design a more intuitive user-friendly platform that will allow educators and administrators to more easily navigate through the student application process..

# References


[1] Garg, L., Agarwal, D., Gupta, D., Goel, P. and Jain, P., 2024, August. An Intelligent Approach to Admissions using Blockchain and Artificial Intelligence. In 2024 1st International Conference on Advanced Computing and Emerging Technologies (ACET) (pp. 1-5). IEEE.

[2] Dwivedi, R., Dave, D., Naik, H., Singhal, S., Omer, R., Patel, P., Qian, B., Wen, Z., Shah, T., Morgan, G. and Ranjan, R., 2023. Explainable AI (XAI): Core ideas, techniques, and solutions. ACM computing surveys, 55(9), pp.1-33.

[3] Baviskar, S. (2024). EduAssist: Chatbot for Student Admission Queries. International Journal for Research in Applied Science and Engineering Technology.

[4] Kuhail, M.A., Al Katheeri, H., Negreiros, J., Seffah, A. and Alfandi, O., 2023. Engaging students with a chatbot-based academic advising system. International Journal of Human–Computer Interaction, 39(10), pp.2115-2141.

[5] Hasan, M.T., 2024. A DATA-DRIVEN APPROACH FOR AUTO-MATING THE UNIVERSITY ADMISSION PROCESS.

[6] Sankarasubramanian, B., Rathinakumar, M., Balakrishnan, B. and Bharathwajan, R., 2014. An automated implementation of indian university admission system using artificial neural networks. Ictact Journal on Soft Computing, 4 (3), pp.767-771.

[7] Alothman, B., Alazmi, H., Ali, M.B., Alqallaf, N. and Khan, M., 2022, July. Accelerating university admission system using machine learning techniques. In

2022 Thirteenth International Conference on Ubiquitous and Future Networks (ICUFN) (pp. 439-443). IEEE.

[8] Dwivedi, A., Dsouza, M. and Aggarwal, A., 2024. AI-Driven School Management System: A React-Based Web Application Enhancing Educational Administration and Student Performance Analytics. International Journal of Innovative Science and Research Technology, pp.3042-3054.

[9] Maryamah, M., Irfani, M.M., Raharjo, E.B.T., Rahmi, N.A., Ghani, M. and Raharjana, I.K., 2024, February. Chatbots in academia: a retrieval-augmented generation approach for improved efficient information access. In 2024 16th International Conference on Knowledge and Smart Technology (KST) (pp. 259-264). IEEE.

[10] Google, "Gemini Embeddings API Documentation." [Online]. Available: https://ai.google.dev/docs/embeddings

[11] Pinecone Systems Inc., "Pinecone Vector Database," 2024. [Online]. Available: https://www.pinecone.io

[12] Weiss, G. ed., 1999. Multiagent systems: a modern approach to distributed artificial intelligence. MIT press.

[13] Kumar Garai, S., Paul, O., Dey, U., Ghoshal, S., Biswas, N. and Mondal, S., 2022. A novel method for image to text extraction using tesseract-OCR. American Journal of Electronics & Communication, 3(2), pp.8-11.

[14] Acharya, D.B., Kuppan, K. and Divya, B., 2025. Agentic AI: Autonomous Intelligence for Complex Goals–A Comprehensive Survey. IEEE Access.

[15] CrewAI, "Introduction," 2025. [Online]. Available: https://docs.crewai.com/introduction

[16] Bai, Z., Wang, P., Xiao, T., He, T., Han, Z., Zhang, Z. and Shou, M.Z., 2024. Hallucination of multimodal large language models: A survey. arXiv preprint arXiv:2404.18930.