Appian AI Challenge

Project-1

Classification and processing of unstructured financial documents

Access links

Pradipta Sundar Sahoo
Indian Institute of Technology (IIT) Roorkee
pradipta_ss@ch.iitr.ac.in

Agenda

- About the project
- Features
- Proposed Solution
- Results
- Access links
- Future Advancements
- Setup Instructions
- Usage
- DB Architecture

Tip: Use links to go to a different page inside your presentation.

How: Highlight text, click on the link symbol on the toolbar, and select the page in your presentation you want to connect.

About

- Goal: To automate the **classification**, **association**, and **summarization** of **unstructured financial documents**, enabling faster processing and reducing manual effort while maintaining accuracy and scalability.
- Key Objectives:
  - Document Categorization
  - Summarization
  - Scalability and Efficiency, Ease of Use, Accuracy and Reliability

BACK TO AGENDA PAGE

Key **Features**

Caching

Speeds up access to processed data using Redis for frequently accessed information.

OCR Integrated Tool

Extracts text efficiently from PDFs and images using robust OCR tools.

Hierarchical Classification

Organizes documents by individual, type, and subcategories

Vector Search

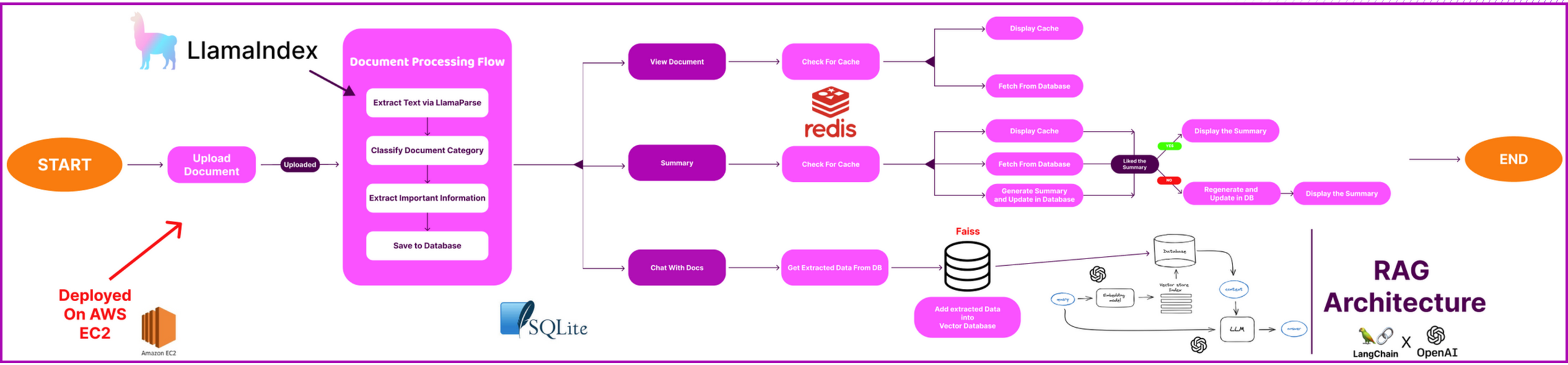
Handles large volumes of FAQs and multiple user queries.

Database Management

Stores structured data securely with SQLite for auditability and redundancy.

BACK TO AGENDA PAGE

Prosposed **Solution**



BACK TO AGENDA PAGE

Techniques **used**

- **Recursive Text Splitting:** Uses RecursiveCharacterTextSplitter for breaking down documents into manageable chunks while maintaining context
- **Retrieval Augmented Generation (RAG):** Implements a hybrid approach combining document retrieval with LLM generation for accurate document-based chat
- **Vector Similarity Search:** Employs FAISS for efficient document similarity matching and retrieval
- **Caching Strategy:** Uses a multi-level caching system combining Redis for distributed caching and local LRU cache for frequently accessed data
- **State Management:** Implements custom state handling for chat sessions and document processing workflows.

Technology **used**

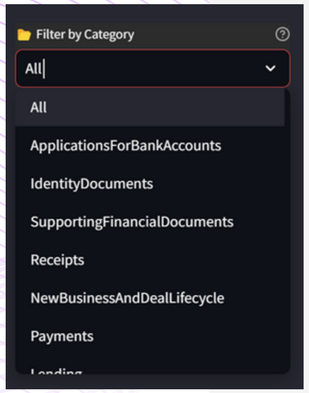
- **LlamaParse:** For robust PDF and document text extraction
- **OpenAI GPT-4:** Powers the document classification, summarization, and chat functionality
- **Streamlit:** Frontend framework for the web interface
- **Redis:** Distributed caching system for document data and chat history
- **SQLite:** Local database for document storage and metadata
- **FAISS:** Vector database for efficient similarity search
- **LangChain:** Framework for building the RAG pipeline
- **LangGraph:** For managing conversational workflows and state




BACK TO AGENDA PAGE

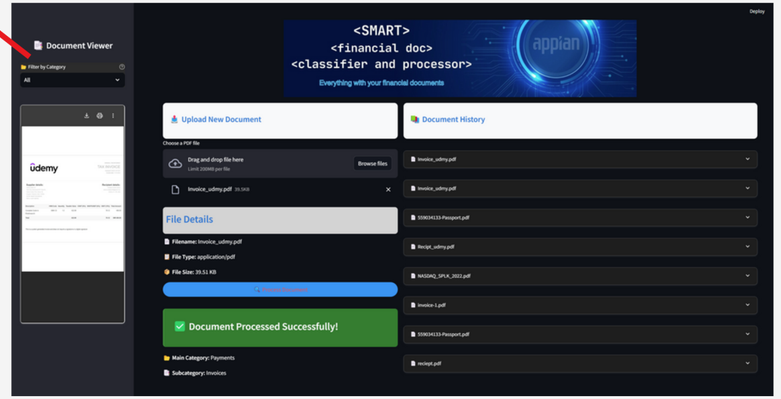
Results

Output :



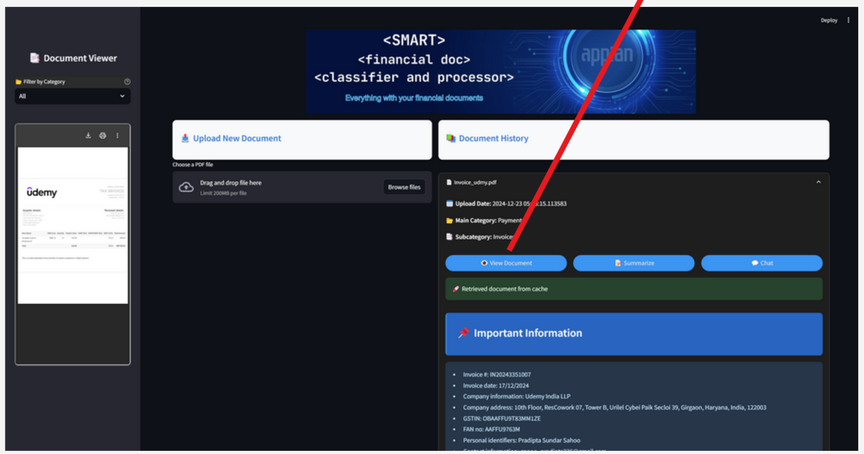
You can select the category of documents you have uploaded earlier

1. Upload Document

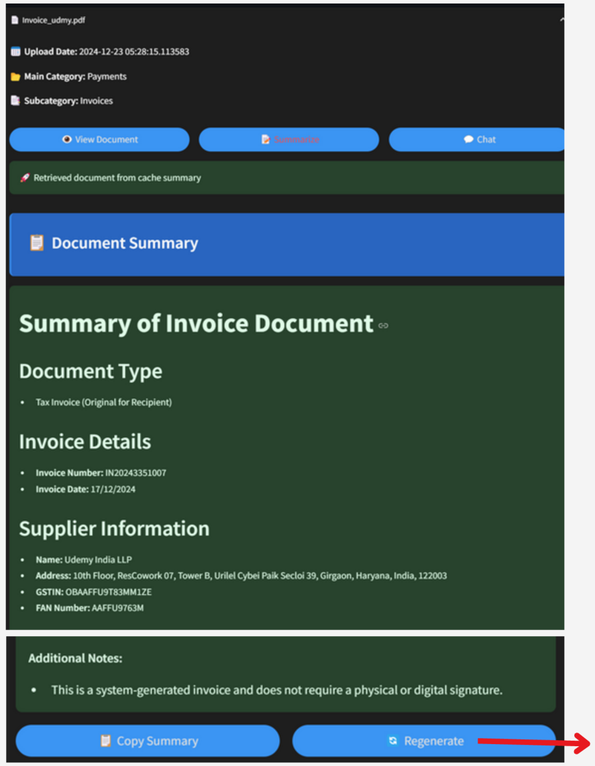


2. View Document

Tap to view the document and important info

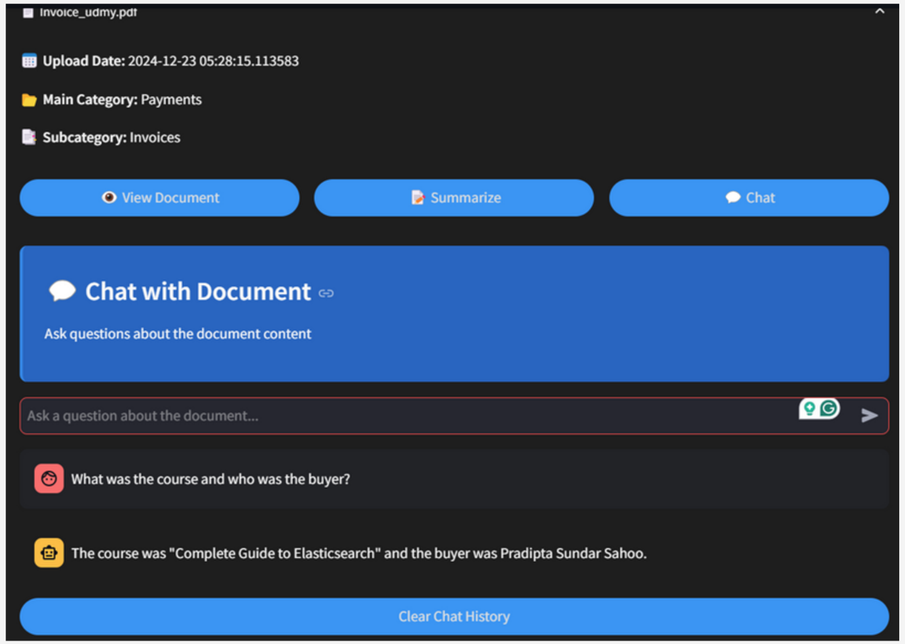


3. Summarise



You can regenerate the summary

4. Chat With the Doc



Bonus Video:

Bonus Video Link

I was working on a finance project myself,which is not published yet anywhere(till now).

Have A Look!!!
120 page Bank statement,with lots of numbers

Access **Links**

Live Access Link:

Appian AI Fin Document

Demo Video Link:

Demo Video Link

Github Link:

Github Repo Link

BACK TO AGENDA PAGE

Evaluation?

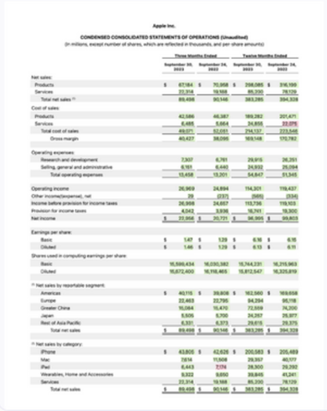
The most important thing.

Have updated evaluation.py

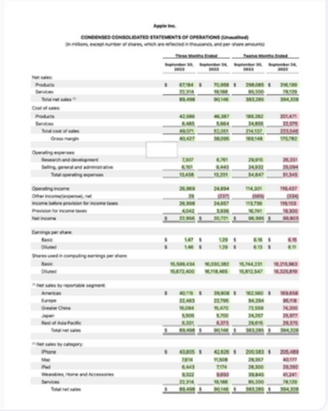
But need to create a test dataset manually.

Which I will do on moving to next round.

LlamaParse



PyPDF



LlamaParse Wins !!!

Metrics	Baseline (PyPDF + Naive RAG)	LlamaParse + Recursive Retrieval
mean_correctness_score	3.874	4.270
mean_relevancy_score	0.844	0.926
mean_faithfulness_score	0.667	0.915

Results over the Uber 10K Dataset. For more information on our evaluation metrics check out our evaluation page here.

GPT-4o , undoubtedly one of the best

Future **Advancements**

Multimodal input and output:
have already been implemented.(**in repo**). But do we actually need it?

Multilingual:
knows how to implement. Could be useful !!!.

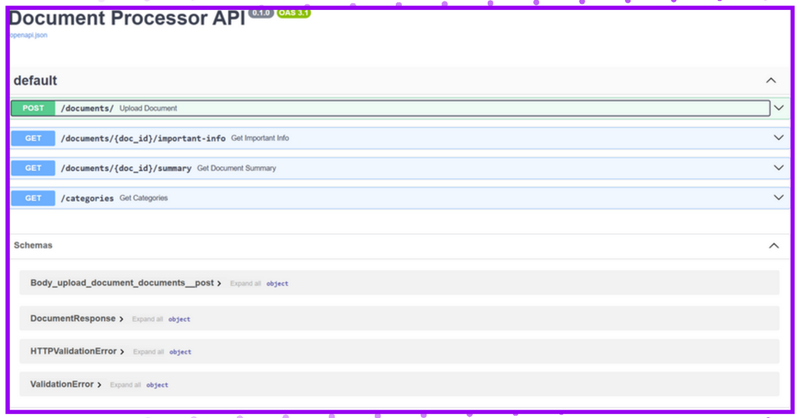
Faster Inference :
could use small llms.

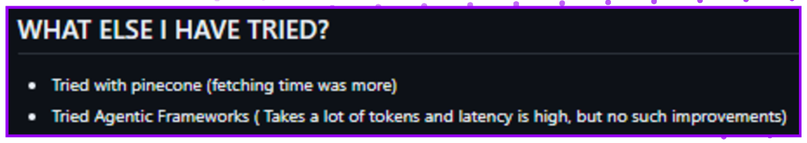
Scalability:
have already implemented a **FastAPI** code. It could be used for production. (**in repo**).

Relevance:
Reranked all results according to relevance. May a more robust way be done.

More robust:
I have already some extra info about different categories that APPIAN could track in future.
But more info can be fed.

Cost-effective:
Instead of EC2, can be deployed in serverless instances.
More intermediate stuffs can be removed to lower the tokens count.





Setup **Instructions**

[Github Repo Link]

BACK TO AGENDA PAGE

📋 Prerequisites

- Python 3.8+
- Redis Server
- OpenAI API Key
- LlamaParse API Key

🔧 Installation

1. Clone the repository:

```
git clone <repository-url>
cd document-categorizer
```

2. Install required packages:

```
pip install -r requirements.txt
```

3. Set up environment variables:

```
OPENAI_API_KEY=your_openai_api_key
LLAMA_CLOUD_API_KEY=your_llama_parse_api_key
```

4. Start Redis server:

```
redis-server
```

5. Run the application:

```
streamlit run app.py
```

Usage

Github Repo Link

💻 Usage

1. **Upload Document**:
   - Click on the file uploader
   - Select a PDF document
   - Click "Process Document"
2. **View Documents**:
   - Browse through uploaded documents in the Document History section
   - Filter documents by category
   - View document details and content
3. **Document Analysis**:
   - View automatic categorization results
   - See extracted important information
   - Generate and read document summaries
4. **Interactive Features**:
   - Chat with documents using AI
   - Download processed documents
   - View document summaries

DB **Architecture**

Github Repo Link

- **Database Architecture**:

```sql
-- Main Document Table
CREATE TABLE documents_new (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    filename TEXT NOT NULL,
    category TEXT NOT NULL,
    subcategory TEXT NOT NULL,
    important_info TEXT,
    pdf_data BLOB,
    upload_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    all_extracted TEXT,
    summary TEXT
);

-- Document Versions Table
CREATE TABLE document_versions (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    doc_id INTEGER,
    version_number INTEGER,
    changes TEXT,
    modified_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (doc_id) REFERENCES documents_new(id)
);

-- Chat History Table
CREATE TABLE chat_history (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    doc_id INTEGER,
    user_message TEXT,
    ai_response TEXT,
    timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (doc_id) REFERENCES documents_new(id)
);
```

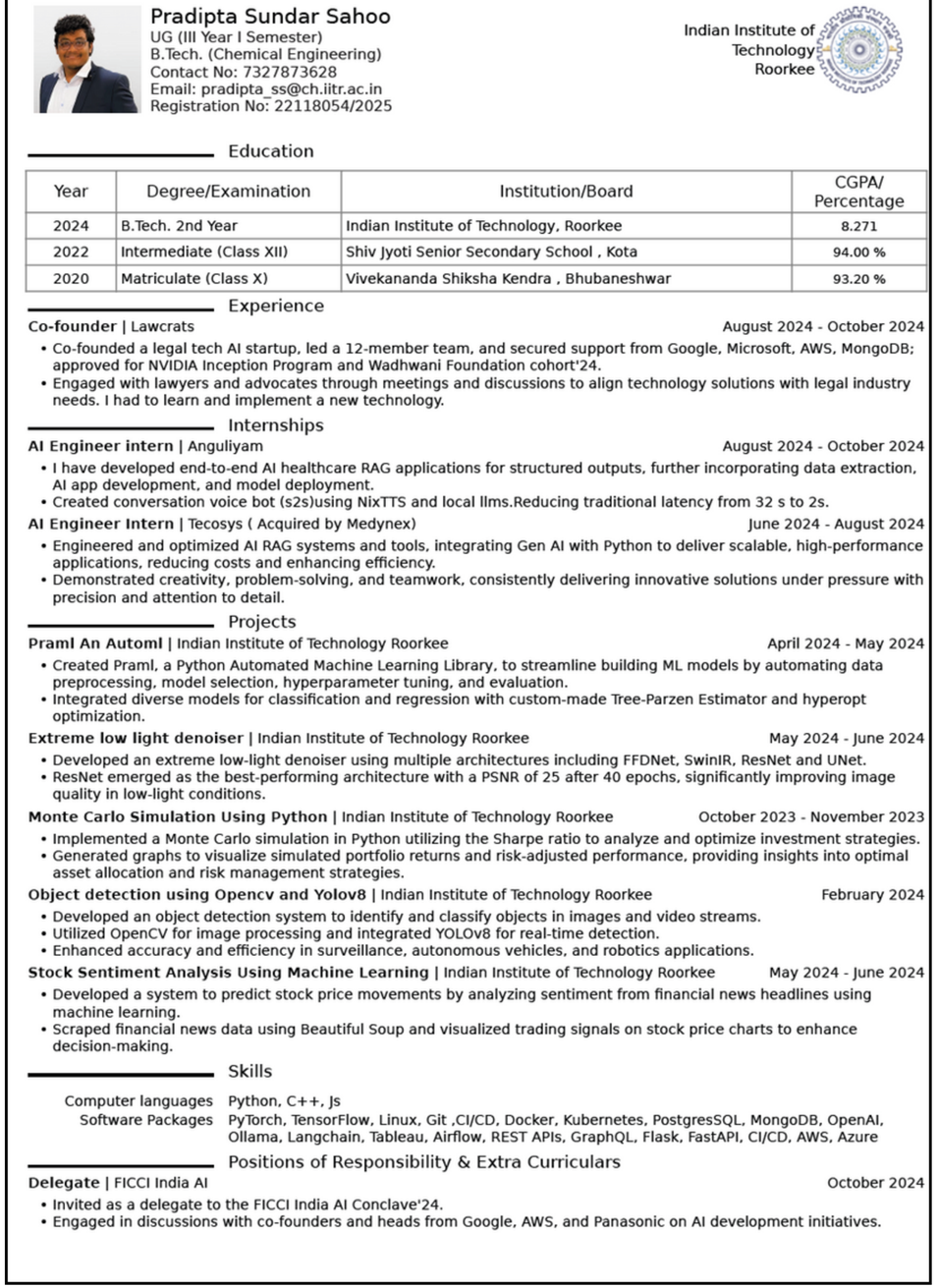
BACK TO AGENDA PAGE

BACK TO AGENDA PAGE

Thank **You**

About **Me**

- **Avid Learner with strong problem solving skills.**
- **Keep tracks of recent GenAI Hacks**
- **Worked with multiple startups (healthtech,legaltech,fintech) build their MVP.**
- **Collaborated with cross-domain teams, showcasing good leadership and team management skills.**
- **Currently, focussing on AI Agents Architecture.**



Pradipta Sundar Sahoo
UG (III Year I Semester)
B.Tech. (Chemical Engineering)
Contact No: 7327873628
Email: pradipta_ss@ch.iitr.ac.in
Registration No: 22118054/2025

Indian Institute of Technology Roorkee

Education

Year	Degree/Examination	Institution/Board	CGPA/ Percentage
2024	B.Tech. 2nd Year	Indian Institute of Technology, Roorkee	8.271
2022	Intermediate (Class XII)	Shiv Jyoti Senior Secondary School , Kota	94.00 %
2020	Matriculate (Class X)	Vivekananda Shiksha Kendra , Bhubaneshwar	93.20 %

Experience

Co-founder | Lawcrats — August 2024 - October 2024
- Co-founded a legal tech AI startup, led a 12-member team, and secured support from Google, Microsoft, AWS, MongoDB; approved for NVIDIA Inception Program and Wadhwani Foundation cohort'24.
- Engaged with lawyers and advocates through meetings and discussions to align technology solutions with legal industry needs. I had to learn and implement a new technology.

Internships

AI Engineer intern | Anguliyam — August 2024 - October 2024
- I have developed end-to-end AI healthcare RAG applications for structured outputs, further incorporating data extraction, AI app development, and model deployment.
- Created conversation voice bot (s2s)using NixTTS and local llms.Reducing traditional latency from 32 s to 2s.

AI Engineer Intern | Tecosys (Acquired by Medynex) — June 2024 - August 2024
- Engineered and optimized AI RAG systems and tools, integrating Gen AI with Python to deliver scalable, high-performance applications, reducing costs and enhancing efficiency.
- Demonstrated creativity, problem-solving, and teamwork, consistently delivering innovative solutions under pressure with precision and attention to detail.

Projects

Praml An Automl | Indian Institute of Technology Roorkee — April 2024 - May 2024
- Created Praml, a Python Automated Machine Learning Library, to streamline building ML models by automating data preprocessing, model selection, hyperparameter tuning, and evaluation.
- Integrated diverse models for classification and regression with custom-made Tree-Parzen Estimator and hyperopt optimization.

Extreme low light denoiser | Indian Institute of Technology Roorkee — May 2024 - June 2024
- Developed an extreme low-light denoiser using multiple architectures including FFDNet, SwinIR, ResNet and UNet.
- ResNet emerged as the best-performing architecture with a PSNR of 25 after 40 epochs, significantly improving image quality in low-light conditions.

Monte Carlo Simulation Using Python | Indian Institute of Technology Roorkee — October 2023 - November 2023
- Implemented a Monte Carlo simulation in Python utilizing the Sharpe ratio to analyze and optimize investment strategies.
- Generated graphs to visualize simulated portfolio returns and risk-adjusted performance, providing insights into optimal asset allocation and risk management strategies.

Object detection using Opencv and Yolov8 | Indian Institute of Technology Roorkee — February 2024
- Developed an object detection system to identify and classify objects in images and video streams.
- Utilized OpenCV for image processing and integrated YOLOv8 for real-time detection.
- Enhanced accuracy and efficiency in surveillance, autonomous vehicles, and robotics applications.

Stock Sentiment Analysis Using Machine Learning | Indian Institute of Technology Roorkee — May 2024 - June 2024
- Developed a system to predict stock price movements by analyzing sentiment from financial news headlines using machine learning.
- Scraped financial news data using Beautiful Soup and visualized trading signals on stock price charts to enhance decision-making.

Skills

Computer languages: Python, C++, Js
Software Packages: PyTorch, TensorFlow, Linux, Git ,CI/CD, Docker, Kubernetes, PostgresSQL, MongoDB, OpenAI, Ollama, Langchain, Tableau, Airflow, REST APIs, GraphQL, Flask, FastAPI, CI/CD, AWS, Azure

Positions of Responsibility & Extra Curriculars

Delegate | FICCI India AI — October 2024
- Invited as a delegate to the FICCI India AI Conclave'24.
- Engaged in discussions with co-founders and heads from Google, AWS, and Panasonic on AI development initiatives.