

MADHESH VIVEKANANDAN

Software Engineer

✉ madheshvivekanandan@gmail.com ☎ 6374051514 📍 Chennai 🌐 madhesh-vivekanandan

PROFESSIONAL SUMMARY

GenAI-focused Software Engineer with a strong foundation in Python backend development and intelligent LLM-driven systems. Experienced in building scalable APIs, agent-like workflows, and retrieval-based AI applications that transform natural language inputs into accurate, actionable outputs. Comfortable working across the full lifecycle of GenAI solutions, from problem understanding and system design to implementation and validation in real-world environments.

EXPERIENCE

Tarka Labs, *Software Developer*

06/2025 – Present | Chennai, India

- Working on Python-based backend systems with a focus on scalable API design, modular architecture, and maintainability.
- Contributing to LLM-powered workflows, including prompt engineering, API orchestration, tool integration, and response handling.
- Involved in designing multi-step agentic reasoning flows where the system dynamically determines when to retrieve data, invoke tools, or generate responses.
- Collaborating on frontend workflow design and user experience implementation for AI-powered web applications using modern React-based architectures.
- Integrating AI capabilities into enterprise applications and participating in deployment, containerization, and environment setup for production-ready systems.
- Contributing to testing and validation workflows to ensure reliability, correctness, and maintainability of backend services and AI-driven features.

Infotel India Pvt Ltd, *Trainee Software Developer*

11/2024 – 05/2025 | Chennai, India

- Assigned as a shadow resource on an enterprise Nissan Supply Chain Management project.
- Worked closely with senior mainframe developers to understand batch job processing, background job execution, and enterprise workflows.
- Assisted in writing and validating background jobs under guidance.

EDUCATION

Sri Ramakrishna Institute of Technology,
BE Computer Science and Engineering

09/2020 – 05/2024 | Coimbatore

TECHNICAL SKILLS

Frontend Technologies (React.js • React Native • Javascript • HTML • CSS) | **Backend Technologies** (Python • FastAPI • REST APIs • GraphQL • SQLAlchemy) | **DevOps & Tooling** (Docker • Github • n8n • Linux • Shell Scripting • Postman) | **Databases** (Postgres • MySQL • Qdrant • Adabas) | **GenAI & AI Technologies** (RAG • LLM APIs (OpenAI, Hugging Face, Ollama))

PROJECTS

Agentic Order-to-Cash (O2C) Automation Platform,

N8N, Python, FastAPI, React Js, PostgreSQL, Azure OpenAI, Docker

- Built an agentic enterprise Order-to-Cash (O2C) automation platform that processes customer inquiries and purchase order PDFs into validated ERP-ready order entries. The system orchestrates multi-step workflows using n8n, Azure OpenAI, IBM Docling, and Microsoft Graph API to extract, validate, and reconcile unstructured email and document data against Odoo ERP systems. Designed with a modular FastAPI backend, PostgreSQL-based validation workflows, and a React-based dashboard enabling human-in-the-loop verification, real-time product matching, and scalable enterprise workflow automation.

Agentic Data Analyst, Python, React Js, Docker, Large Language Models (OpenAI)

- Agentic Data Analyst is an intelligent GenAI-powered analytics platform that enables users to query data warehouses using natural language and receive results as executable SQL, tables, charts, and summaries. The system is built using a multi-step agentic workflow inspired by DSPy, incorporating query interpretation, entity extraction, ambiguity detection, workflow planning, LLM-driven SQL generation, execution, and result summarization. A dedicated ThinkingSignature layer reformulates user intent to improve SQL accuracy, while agent decision logic dynamically determines whether to generate raw queries, visualizations, or analytical summaries. The platform is exposed via FastAPI, designed with a modular and extensible architecture, and integrated with MLFlow to track LLM calls, agent decisions, SQL executions, and artifacts for observability and reproducibility, with automated validation using pytest-based natural language test cases.

RAG-Based Document Question Answering System, Python, Qdrant Vector DB, React Js

Github link [🔗](#)

- Built a Retrieval-Augmented Generation (RAG) system that allows users to ask natural language questions over custom documents and receive grounded, context-aware answers. The system ingests documents, generates embeddings, and stores them in Qdrant vector database, retrieving the most relevant chunks at query time and injecting them into the LLM for answer generation. The application is exposed via a FastAPI backend and demonstrates practical use of vector search, semantic retrieval, and LLM integration for enterprise knowledge systems.

LLM-Powered Conversational Assistant, Python, React Js

Github link [🔗](#)

- Developed a full-stack LLM-powered conversational assistant that enables users to interact with an AI system through a web interface for tasks such as question answering and text summarization. The application uses prompt engineering techniques (zero-shot, few-shot, and role-based prompting) to ensure consistent and context-aware responses, with a FastAPI backend orchestrating LLM calls and managing request/response flows. The system demonstrates end-to-end integration of frontend, backend, and LLM APIs, laying the foundation for scalable GenAI applications.

Singapore Resale Flat Price Prediction System, Python, Pandas, Scikit-learn, Decision Tree Regressor

Github link [🔗](#)

- Developed a machine learning-based web application to predict the resale prices of HDB flats in Singapore using historical transaction data. The project involved consolidating multiple datasets spanning different time periods, performing data cleaning and feature engineering, and training a regression model to estimate resale prices on factors such as location, flat type, floor area, storey range, and lease details. The trained model was deployed as an interactive Streamlit application, allowing users to input flat attributes and instantly receive an estimated resale price, helping buyers and sellers make informed decisions.