

# Aayushmaan Hooda

📍 Sydney, NSW 📩 aayushmaan162@gmail.com ☎ 0412222762  
👤 aayushmaanhooda 💬 aayushmaanhooda 💾 aayushmaanhooda

## Skills

---

**Languages:** Python, JavaScript, Shell Scripting, SQL, NoSQL, CypherQuery

**Frameworks:** React, Flask, FastAPI, Streamlit, LangChain, LangGraph, CrewAI, Mem0

**Databases:** PostgreSQL, MongoDB, Pinecone, PgVector, Neo4j

**Libraries:** FastMCP, Numpy, Pandas, Pydantic, SQLAlchemy, SQLModel, Scikit-learn, boto3, Pytorch, Ragas

**AWS Cloud:** EC2, API Gateway, Bedrock, S3, Lambda, PartyRock

**Tools:** Docker, Git, GitHub Actions, Jira, Postman, LangSmith, Figma, Flowise, Vapi, K6, Nginx

**Concepts:** Agentic AI, Multi-Agent Systems, Knowledge Graph, RAG, Corrective Rag (CRAG), Transformers, Fine Tuning, Context/Prompt Engineering, MCP, VectorStore, UI/UX Principles

**Talk To My Bot:**

[Aayushmaan Chatbot](#)

## Education

---

**University of New South Wales**  
*Master's in Information Technology*

Sydney, NSW, Australia  
*Sept 2023 – Sept 2025*

**Maharshi Dayanand University**  
*Bachelor of Computer Science and Engineering*

Rohtak, Haryana, India  
*Sept 2017 – Sept 2021*

## Experience

---

**Backend Developer Intern**  
*Stoik*

Sydney, Australia  
*Jan 2025 – Jan 2025*

- Developed a deep understanding of Docker architecture and FastAPI orchestration to enhance application performance.
- Analyzed and interpreted FastAPI code alongside MongoDB schema to ensure seamless application functionality.

**Associate Software Engineer**  
*Annalect*

Gurugram, India  
*Oct 2021 – Sept 2023*

- Authored and maintained YAML files for OpenAPI specifications to generate APIs in Apigee.
- Collaborated with the backend team to resolve and improve the YAML file.
- Developed and maintained microservice APIs using Flask, enhancing application scalability and performance.
- Delivered progress updates through regular demo presentations to the US-based team, ensuring alignment and transparency.

## Projects

---

**LangGraph ChatbotApp**

[github/chatbot](#)

- Built a conversational chatbot using LangGraph and LangChain, with a Streamlit frontend for real-time interaction
- Managed conversation state via a ChatState TypedDict and InMemorySaver for checkpointing across ses-

sions.

- **Tools Used:** Python, LangChain, LangGraph, Streamlit

### Resume Selector Agent

[github/resume-selector ↗](#)

- Developed a multi-agent AI system using the crewAI framework to orchestrate automated resume selection.
- Built configurable agent definitions (in agents.yaml) and task configurations (in tasks.yaml) to allow flexible extension of agent roles, tools, and objectives.
- **Tools Used:** Python, CrewAI, Pydantic, Streamlit

### RAG Evaluation using RAGAS

[github/rag-evaluation ↗](#)

- Developed a full-pipeline evaluation framework for retrieval-augmented generation (RAG) systems using LangChain, Pinecone, and RAGAS metrics.
- Integrated RAGAS to compute key metrics such as Faithfulness, Context Precision/Recall, and Response Relevancy for quantitative model evaluation.
- **Tools Used:** Python, Langchain, Ragas, Pinecone

### Voice RAG

[github/voice-rag ↗](#)

- Built a voice-enabled RAG application, integrating faster-whisper for speech-to-text and edge-tts for text-to-speech synthesis.
- Implemented a retrieval pipeline with LangChain, FAISS, and OpenAI embeddings to enable real-time conversational document querying via a browser-based interface with live waveform visualization using FastAPI.
- **Tools Used:** Python, Langchain, edge-tts, faster-whisper, faiss

### Leave Manager MCP Server

[github/MCP ↗](#)

- Designed and developed a robust leave management system using FastMCP and MongoDB, supporting both persistent and in-memory data storage.
- Implemented REST-style MCP tools for applying leaves (with balance checks and date history), employee CRUD, multi-employee batch onboarding, and real-time leave balance and audit queries.
- **Tools Used:** Python, UV, FastMCP, MongoDB, Claude Desktop

### Travel History RESTful API

[github/travel-history ↗](#)

- Developed a Flask-RESTX service to store, retrieve, and visualise countries visited, backed by SQLite.
- Integrated the external Countries' GraphQL API to import country metadata (flags, capitals, languages, currencies).
- Created a /countries/visited endpoint that streams a Matplotlib-generated PNG summarising travel history.
- **Tools Used:** Python, Flask-RESTX, SQLite, GraphQL client, Matplotlib, Docker

## Certificates

---

- [AWS Cloud Practitioner ↗](#)
- [LangChain Essentials - Python ↗](#)
- [RAG Apps Using MongoDB ↗](#)
- [Introduction to Model Context Protocol ↗](#)

## Professional Summary

---

Aspiring AI Engineer with hands-on experience in building AI-powered applications, intelligent workflows, and scalable backend systems. Skilled in LangChain, LangGraph, RAG pipelines, MCP Servers, and Python frameworks like FastAPI and Flask. A quick learner who's always excited to explore new technologies, frameworks, and ideas, passionate about transforming innovative concepts into real-world Generative AI solutions with clean, high-performance code.