

VARAMAL DEVRAJ KHERAJ

AI Engineer • Generative AI / LLM Engineer • Agentic AI & Automation Engineer • Full-Stack Developer • Software Engineer

Gujarat, India — PIN 361415 • +91 6354010494 • devrajgadhi726292@gmail.com
GitHub: github.com/dev4aibots • Portfolio: dev4aibots.com • Project Index: dev4aibots.github.io/PROJECTS • LinkedIn: linkedin.com/in/devraj-gadhvi-793338388

OPEN TO: AI Engineer • ML / Deep Learning Engineer • GenAI / LLM Engineer • RAG & Vector Search Engineer • AI Agent / Automation Engineer • Full-Stack Developer • Back-End / Python Developer • Front-End Developer • Software Engineer (Entry Level) • RPA / Automation Developer • AI QA & Evaluation Engineer • Technical Support Engineer • Prompt Engineer • AI Product / Implementation Engineer — Remote, hybrid, contract, freelance and project-based roles worldwide.

PROFESSIONAL SUMMARY

Self-directed AI and software engineer with **one year of intensive, full-time-equivalent professional training (2025-2026)** across twelve Coursera credentials from DeepLearning.AI, IBM, Vanderbilt University, UiPath and Edureka — covering generative AI, RAG, agentic systems, deep learning, computer vision, LLM fine-tuning, RPA, no-code automation and AI-assisted software engineering. Completed the full curriculum of every program, including **every course and every graded capstone, applied lab and portfolio project** — not a subset.

Independent builder of **10 public engineering repositories** implementing production-oriented systems: an enterprise RAG engine with evaluation, an autonomous DevOps/Kubernetes root-cause agent, a repository-scale semantic code search engine, a Model Context Protocol (MCP) tool registry, an agentic CLI coding workspace, a deep-research orchestrator, a low-latency realtime voice AI, a multimodal vision/document-understanding pipeline, and an end-to-end LLM fine-tuning MLOps framework. Comfortable across the stack — Python, Flask/FastAPI, REST APIs, TypeScript/Next.js, SQL/PostgreSQL, Docker, Git/GitHub — with mathematics and statistics foundations applied to model evaluation and error analysis.

Positioning. This is an *early-career / entry-level* profile: no formal paid employment is claimed. Evidence of capability comes from verified credentials, complete capstone deliverables, deployed repositories and independently engineered systems. Available immediately for global remote contract, freelance, project-based and full-time entry-level roles.

CORE SKILLS

Generative AI / LLM	Transformer and LLM architecture • language modelling • tokenization • positional encoding • self-attention and multi-head attention • encoder-only (BERT) and decoder-only (GPT) models • in-context learning • chain-of-thought, tree-of-thought and few-shot prompting • prompt templates and prompt management • structured JSON output • embeddings • multimodal prompting (text • image • audio • vision) • model comparison and evaluation • hallucination mitigation • responsible and trustworthy AI • AI security
RAG & Vector Search	Retrieval-Augmented Generation pipelines • chunking strategies • embedding generation • vector databases (ChromaDB, FAISS) • similarity search • metadata filtering • reranking • multimodal similarity fusion and score normalisation • CLIP and Sentence-Transformers embeddings • LlamaIndex • query optimisation • RAG evaluation (faithfulness, answer relevance, context precision)
AI Agents & Agentic AI	Agent architecture (Goals, Actions, Memory, Environment) • ReAct and Reflexion patterns • reflection and self-improving agents • tool/function calling • tool routing • multi-agent orchestration • stateful workflows • conditional logic and iteration • agentic RAG • memory and context management • MCP (Model Context Protocol) servers, clients and hosts • agent evaluation, safety and deployment • LangChain • LangGraph • CrewAI • AG2 / AutoGen • BeeAI • OpenAI GPTs
Machine Learning	Supervised and unsupervised learning • classification • regression • clustering • logistic regression • model training and validation • transfer learning • feature engineering • data augmentation • SciPy • Scikit-learn • XGBoost • evaluation metrics (accuracy, precision, recall, F1, ROC) • bias/variance and error analysis • model optimisation
Deep Learning & Vision	Artificial neural networks • CNNs • RNNs • autoencoders • GANs • Vision Transformers (ViT) • hybrid CNN-ViT architectures • computer vision and image processing • object recognition • geospatial land classification • NLP and document classification • sequence-to-sequence models • Word2Vec and n-gram models • PyTorch • Keras • TensorFlow • Hugging Face • tf.data and custom Dataset/DataLoader pipelines

LLM Fine-Tuning & MLOps	Full-parameter fine-tuning · Parameter-Efficient Fine-Tuning (PEFT) · LoRA · QLoRA · instruction tuning · reward modelling · RLHF · Direct Preference Optimization (DPO) · PPO · model alignment · training pipelines for open-source LLMs
Software Engineering	Software Development Lifecycle (SDLC) · Agile and Scrum · configuration-driven development · Gang of Four design patterns (Singleton, Factory, Template Method, Strategy) · database design, CRUD and schema modelling · REST API design and integration · authentication and payload structuring · unit testing and test design · edge-case testing · debugging · static code analysis · code review · documentation · dependency management · DevSecOps and CI/CD concepts
Programming & Data	Python (primary) · JavaScript and TypeScript fundamentals · HTML5, CSS3, responsive design (Flexbox, Tailwind, Bootstrap) · SQL · JSON and Pickle serialisation · Pandas · NumPy · Matplotlib · Seaborn · Jupyter · SQLAlchemy · SQLite · PostgreSQL · Redis concepts · Supabase
RPA & Automation	UiPath Studio and Orchestrator · RPA workflow design · control flow · UI automation and selectors · screen scraping and PDF data extraction · data manipulation · Excel and email automation · exception handling · n8n · Make · Zapier · API/webhook workflows · no-code and low-code automation
AI-Assisted Engineering	Claude Code · Cursor AI · GitHub Copilot · OpenAI Codex · ChatGPT Advanced Data Analysis · GPT Vision · vibe-coding and agentic development workflows · CLAUDE.md project context files · reusable command libraries · parallel agent workflows across git branches · "Best of N" solution generation
Platforms & Tools	Git and GitHub · Docker · Flask · FastAPI · Gradio · Streamlit · Next.js · IBM watsonx · IBM Skills Network Labs · Hugging Face · Chroma · FAISS · Vercel · VS Code and PyCharm · Anaconda · Linux command line · REST clients and API tooling
Mathematics & Statistics	Probability and statistics · descriptive statistics · distributions · correlation · regression and classification reasoning · loss functions and optimisation concepts · vectors, matrices and linear algebra for embeddings and neural networks · cosine similarity and semantic distance · token economics · model comparison, trade-off analysis, validation and data-driven decision making

CERTIFICATIONS — 12 CREDENTIALS · COURSERA (2025–2026)

Each credential was completed in full — every course, lab, project and capstone. Click any title to open the verifiable certificate on Coursera.

1. Generative AI for Software Development Professional Certificate — DeepLearning.AI · Completed September 2026 · 3 courses

LLM-assisted software design, coding, testing, debugging and documentation · configuration-driven development · design patterns · database architecture · dependency management.

2. Build Anything with AI: No Code Required Specialization — Vanderbilt University · Completed September 2026 · 4 courses

Building working software, dashboards, automations and analyses with AI agents without traditional coding · gap and trade-off analysis · multimodal prompting · AI product strategy.

3. Generative AI Software Engineering Specialization — Vanderbilt University · Completed September 2026 · 4 courses

Code-writer to AI-orchestrator transition · Claude Code full-stack construction · Python agents across parallel git branches · custom GPTs for business workflows.

4. IBM AI Engineering Professional Certificate — IBM · Completed May 2026 · 6 courses

Machine learning, deep learning, CNNs, Vision Transformers, PyTorch, Keras and TensorFlow · geospatial image-classification capstone.

5. Robotic Process Automation (RPA) Specialization — UiPath · Completed March 2026 · 6 courses

UiPath automation design · control flow · UI automation and selectors · data manipulation · screen scraping · PDF extraction · Orchestrator and capstone.

6. IBM RAG and Agentic AI Professional Certificate — IBM · Completed February 2026 · 10 courses

Advanced RAG pipelines · multimodal AI · vector databases and retrievers · LangChain, LangGraph, CrewAI, AG2, BeeAI · MCP · end-to-end capstone.

7. Generative AI for Automating HR Workflows Specialization — Edureka · Completed February 2026 · 3 courses

Generative AI for recruitment, onboarding, engagement and performance management · agentic AI and no-code automation with Make, Zapier and n8n.

8. AI Agent Developer Specialization — Vanderbilt University · Completed February 2026 · 6 courses

Agent architectures, tool use, memory, custom GPTs, prompt engineering and responsible AI · applied automation and document-analysis agents.

9. AI Agents and Agentic AI in Python: Powered by Generative AI Specialization — Vanderbilt University · Completed January 2026

· 4 courses

Framework-independent agent engineering in Python · agent loops, tools, multi-agent coordination, self-prompting, token efficiency and resilient design.

10. Building AI Agents and Agentic Workflows Specialization — IBM · Completed January 2026 · 3 courses

LangGraph stateful agents · reflection and ReAct · CrewAI, AG2 and BeeAI multi-agent orchestration · 11 hands-on agent labs.

11. Generative AI Automation Specialization — Vanderbilt University · Completed January 2026 · 4 courses

Expert-level prompt engineering · advanced data analysis · trustworthy generative AI · GPT Vision applied to real-world tasks.

12. IBM AI Developer Professional Certificate — IBM · Completed December 2025 · 10 courses

Software engineering fundamentals · Python · HTML/CSS/JavaScript · Flask · generative AI application building · chatbots, voice assistants and RAG projects.

PROFESSIONAL TRAINING & PROJECT EXPERIENCE — 1 YEAR (2025–2026)

Nature of this experience: one continuous year of structured, full-time-equivalent professional training delivered by Coursera and its university and industry partners (DeepLearning.AI, IBM, Vanderbilt University, UiPath, Edureka). Every program below was completed **in full — all courses, all labs, all graded projects and all capstones**. This is documented training and independent engineering work, not paid employment; the resume states this plainly so hiring teams can weigh the evidence accurately.

IBM AI Engineering Professional Certificate — IBM · Issued May 2026 · 6 courses

Machine learning, deep learning and generative AI engineering with Python; capstone delivered in both Keras and PyTorch.

Courses completed (all 6):

- Machine Learning with Python
- Introduction to Deep Learning & Neural Networks with Keras
- Deep Learning with Keras and TensorFlow
- Introduction to Neural Networks and PyTorch
- Deep Learning with PyTorch
- AI Capstone Project with Deep Learning

Capstone — AI Capstone Project with Deep Learning (geospatial land classification):

- Built an end-to-end geospatial image-classification system that identifies terrain types from satellite imagery, trained and evaluated in **both Keras and PyTorch** for framework-level comparison.
- Implemented **Vision Transformers (ViT)** in Keras and in PyTorch, then engineered **hybrid CNN–ViT architectures** combining convolutional spatial feature extraction with global self-attention; hybrid models proved strongest on complex geospatial data.
- Ran comparative evaluation across frameworks and architectures — accuracy, precision, F1-score, ROC curves, training speed and inference speed — and documented the trade-offs.
- Engineered high-performance data pipelines: benchmarked in-memory versus generator-based loading, then built augmentation and preprocessing pipelines using the **tf.data API** (caching, prefetching) in Keras and custom **Dataset / DataLoader** classes for parallel processing in PyTorch.
- Applied transfer learning to raise model accuracy and reduce training cost.

Additional graded labs and projects from this program:

- Implemented supervised and unsupervised machine learning models with **SciPy** and **Scikit-learn** — classification, regression and clustering workflows end to end.
- Implemented **positional encoding, masking and attention mechanisms** and applied them to document classification.
- Created LLM architectures such as **GPT and BERT**; explored generative architectures (GPT, RNNs, GANs) and built the NLP data pipeline with Hugging Face and PyTorch, including advanced tokenization strategies and custom data loaders for training LLMs.
- Implemented text-to-feature techniques (**Word2Vec** embeddings, Bag-of-Words, one-hot encoding) and built and evaluated sequence-to-sequence and n-gram models for document classification and language generation.
- Developed transfer-learning applications in NLP using LangChain, Hugging Face and PyTorch.
- Set up a **Gradio** interface for model interaction and constructed a **QA bot with LangChain and an LLM** that answers questions from loaded documents.
- Completed computer vision and image processing coursework within the programme (image and video processing, object recognition).

Stack: Python · TensorFlow · Keras · PyTorch · Scikit-learn · SciPy · Hugging Face · LangChain · Pandas · NumPy · Matplotlib · Seaborn · Jupyter

IBM RAG and Agentic AI Professional Certificate — IBM · Issued February 2026 · 10 courses (~99 hours)*Advanced program in retrieval-augmented generation, multimodal AI, autonomous multi-agent systems and Model Context Protocol.***Courses completed (all 10, with course hours):**

- Develop Generative AI Applications: Get Started — 9 h (generative AI fundamentals, prompt engineering, LangChain, Flask)
- Build RAG Applications: Get Started — 7 h (RAG pipelines, LlamaIndex, Gradio interfaces)
- Vector Databases for RAG: An Introduction — 9 h (vector stores, similarity search, Chroma)
- Advanced RAG with Vector Databases and Retrievers — 8 h (advanced retrieval, reranking, metadata filtering)
- Build Multimodal Generative AI Applications — 8 h (text, image and audio processing; multimodal RAG)
- Fundamentals of Building AI Agents — 11 h (agent architectures, function calling, tool use)
- Agentic AI with LangChain and LangGraph — 10 h (LangGraph state machines, agent loops)
- Agentic AI with LangGraph, CrewAI, AutoGen and BeeAI — 13 h (multi-agent frameworks and orchestration)
- Build AI Agents using MCP — 10 h (Model Context Protocol servers, clients and hosts)
- RAG and Agentic AI Capstone Project — 14 h (end-to-end integration of the full program)

Capstone — RAG and Agentic AI Capstone Project:

- Architected an end-to-end, production-style agentic RAG application: data ingestion, chunking and embedding, vector indexing, retrieval, agentic reasoning, tool use and a deployed user interface.
- Designed the agent so it maintains conversational memory across multi-turn sessions, routes dynamically between tools, and returns grounded, context-aware answers rather than model recall.
- Delivered the solution behind a working Gradio/Flask interface with structured JSON contracts between the LLM, tools and data layer.

Applied labs and projects completed (every lab in the program):

- **Extract structured JSON from unstructured text** — used an LLM with schema-constrained prompting to parse raw restaurant descriptions from a culinary dataset into clean structured JSON.
- **Process multimodal data with LLMs** — combined recipe text with images to generate JSON-augmented multimodal outputs.
- **Command-line data management with LLMs** — built an interactive CLI tool where LangChain chains and agents manage and query structured data.
- **Construct a multimodal vector index** — built parallel vector indexes over articles with Sentence-Transformers and over food images with CLIP, stored in Chroma.
- **Similarity retrieval with metadata filtering** — implemented the retrieval layer, executing similarity searches with optional metadata constraints for precision-oriented retrieval.
- **Multimodal similarity fusion and retrieval ranking** — implemented cross-modal score normalisation and weighted fusion to merge text and image results into a single ranked list, mirroring production multimodal RAG.
- **Build an MCP server** — implemented a FastMCP server exposing a dataset as an MCP resource with three search and retrieval tools.
- **Build an MCP client** — connected to the server over stdio with ClientSession, implemented roots and sampling callbacks, and invoked every server tool through the protocol.
- **Design a full LLM MCP host** — assembled a complete host application: a Gradio chat interface backed by IBM watsonx running a **ReAct agent loop** that discovers MCP tools at runtime, invokes them iteratively and renders natural-language responses.
- **Prompt engineering and LangChain prompt templates** — built and managed dynamic prompt templates, in-context learning and chain-of-thought patterns.
- **Empowering LLMs with LangChain** — built smart AI applications using chains, tools and agents.
- **Function calling and external tool integration** — connected LLMs to external APIs and tools for context-aware responses.
- **Similarity search on text with Chroma Vector DB and a custom data-visualisation agent** that answers analytical questions and produces charts.

*Stack: Python 3.10+ · LangChain · LangGraph · LlamaIndex · FastMCP · ChromaDB · Sentence-Transformers · CLIP · IBM watsonx · Flask · Gradio · Jupyter · CrewAI · AG2 (AutoGen) · BeeAI · Milvus/Pinecone concepts***Generative AI Software Engineering Specialization** — Vanderbilt University · Issued September 2026 · 4 courses*Transformation from code writer to AI orchestrator: directing AI agents that build, test and integrate software.***Courses completed (all 4):**

- Claude Code: Software Engineering with Generative AI Agents
- AI Agents and Agentic AI with Python & Generative AI
- Prompt Engineering for ChatGPT

- OpenAI GPTs: Creating Your Own Custom AI Assistants

Applied project — integrated AI-powered business automation system (built progressively across all four courses):

- Used **Claude Code** to rapidly prototype and construct full-stack applications — user interfaces, databases, business logic and APIs — while focusing on architecture and strategy.
- Orchestrated Claude Code **concurrently across multiple git branches** with parallel AI subagents developing different features, then had the agent perform merging and integration.
- Applied the "**Best of N**" pattern — generating three to five versions of each feature and cherry-picking the strongest implementation — plus self-critique rubrics that let the agent catch bugs before human review.
- Authored **CLAUDE.md** project-context files and reusable command libraries that compress code review, feature build and test-suite workflows into single prompts.
- Built **Python agents** that autonomously handle complex workflows such as document analysis and data processing.
- Engineered sophisticated, cross-domain prompts that enable precise AI reasoning, and used multimodal prompting to turn whiteboard sketches into complete UI components and architectures.
- Deployed **custom GPTs** as intelligent end-user interfaces — including assistants that apply company policy, process travel and expense reports, and analyse legal documents.

Stack: Claude Code · OpenAI GPTs and API · Python · Git and git worktrees · multi-agent orchestration · prompt engineering

Build Anything with AI: No Code Required Specialization — Vanderbilt University · Issued September 2026 · 4 courses

Turning plain-language intent into working software, automations and analyses using AI agents — no traditional coding required.

Courses completed (all 4):

- Build Anything with AI — No Code Required (9 h)
- Build Apps with AI — From Idea to Working Product (6 h)
- Prompt Engineering for ChatGPT
- AI Agent Skills for Leaders

Applied projects — every project was hands-on and personal, built on real files, data and problems:

- Built working software, dashboards, automated workflows and custom tools by describing requirements in plain English to AI agents (**Claude Code**, **OpenAI Codex**, **MAJK**) and iterating the agent through to a functioning product.
- Automated real business and personal workflows end to end, and converted messy, unstructured information into usable tools and insights.
- Applied the underlying patterns and mental models that transfer across any AI agent tool, so the workflow holds regardless of which tool is current.
- Performed gap and trade-off analysis to decide when an AI-built solution is the right answer and where it fails, and defined multimodal prompting strategies for text, image and document inputs.
- Designed AI integrations and applied AI product strategy — scoping capabilities, risks and delivery paths for AI-built products.

Stack: Claude Code · OpenAI Codex · MAJK · multimodal prompting · no-code and low-code automation · AI product strategy

Generative AI for Software Development Professional Certificate — DeepLearning.AI · Issued September 2026 · 3 courses

LLM pair-programming across the full development lifecycle: design, coding, testing, debugging, documentation and dependency management.

Instructor: Laurence Moroney, former AI Lead at Google.

Courses completed (all 3):

- Introduction to Generative AI for Software Development
- Team Software Engineering with AI
- AI-Powered Software and System Design

Projects completed (each is a graded, buildable deliverable):

- **Configurable image-generation application** built with the **DALL-E API** and implemented strictly on **configuration-driven development (CDD)** principles, so behaviour changes through configuration rather than code edits.
- **E-commerce database system** — designed the schema and implemented full **CRUD** functionality with performance optimisation.
- **Financial services application** demonstrating applied **Gang of Four design patterns** — Singleton, Factory, Template Method and Strategy.
- **Social network database** created and queried as a graded exercise, covering schema design, relationships and query construction.
- **Pair-coding with an LLM** to modify data structures efficiently for production use and big-data scale, including implementing linked lists and trees and optimising them for performance and security.

- **Working with an LLM as a software tester** — identified bugs, generated edge-case tests and updated code to correct the errors found.
- **Implemented a functioning local database from scratch** and partnered with an LLM to reason through software design problems and optimise for efficient, secure data access.
- Applied file serialisation techniques with **JSON and Pickle** and worked with external APIs as part of application design.

Stack: Python · SQLAlchemy · SQLite · JSON · Pickle · DALL·E API · design patterns · LLM pair programming

Robotic Process Automation (RPA) Specialization — UiPath · Issued March 2026 · 6 courses

Business-process automation design and delivery on the UiPath platform, from workflow construction to Orchestrator deployment.

Courses completed (all 6):

- RPA Basics and Introduction to UiPath
- Data Manipulation in RPA
- UI Automation and Selectors
- Control Flow in RPA
- Automation Techniques in RPA
- UiPath Orchestrator and Capstone Projects

Projects and capstone work:

- Designed and implemented end-to-end RPA workflows with robust control flow — sequences, decision branching, loops, switch logic and state machines — plus structured exception handling.
- Automated user interfaces with reliable **selectors**, anchors and UI synchronisation to make bots resilient to interface change.
- Performed data manipulation and extraction — Excel and CSV processing, data tables, string manipulation, **screen scraping and PDF data extraction** — to feed downstream automation.
- Built automation techniques for real business processes, including email and document handling, and applied reusable workflow libraries.
- Worked with **UiPath Orchestrator** concepts — publishing, deploying, scheduling, queue management, asset and credential handling, and monitoring job execution.
- Completed capstone automation projects that integrate the full toolkit into deployable business-process bots.

Stack: UiPath Studio · UiPath Orchestrator · selectors and UI automation · data tables · screen scraping · PDF/Excel/email automation

Generative AI for Automating HR Workflows Specialization — Edureka · Issued February 2026 · 3 courses

Applied generative AI and agentic automation across the HR lifecycle, built on authentic HR scenarios with an ethical, data-driven emphasis.

Courses completed (all 3):

- Generative AI Foundation for HR Professionals
- Generative AI for People Management
- Agentic AI & No-Code Automation for Strategic HR

Projects completed:

- Designed **AI-powered recruitment workflows** — job-description generation, candidate screening criteria, structured interview question sets and shortlisting logic.
- Built **personalised onboarding experiences** that adapt content, sequencing and guidance to the role and the individual.
- Created **employee-engagement surveys** with AI-generated question design, response analysis and insight summarisation.
- Optimised **performance-management workflows** — review templates, feedback synthesis and goal-tracking automation.
- Implemented **strategic no-code automation** with **Make, Zapier and n8n**, wiring AI models into multi-step business workflows through webhooks and API calls.
- Applied ethical, privacy-aware and data-driven implementation practices — bias-aware evaluation of AI output and responsible handling of employee data.

Stack: generative AI tools · agentic AI · Make · Zapier · n8n · API and webhook workflows · no-code automation

AI Agent Developer Specialization — Vanderbilt University · Issued February 2026 · 6 courses

Designing, building and deploying intelligent software agents in Python with tools, memory, reasoning and responsible-AI practice.

Courses completed (all 6):

- AI Agents and Agentic AI with Python & Generative AI (11 h)
- AI Agents and Agentic AI Architecture in Python (7 h)
- OpenAI GPTs: Creating Your Own Custom AI Assistants
- Prompt Engineering for ChatGPT
- ChatGPT Advanced Data Analysis

- Trustworthy Generative AI

Applied projects — built functional AI agents against real-world problems:

- Designed complete **agent architectures** — goal definition, action and tool selection, memory implementation and environment interaction — and implemented them from first principles in Python.
- Built agents that **automate multi-step workflows**, analysing documents and returning structured results instead of free text.
- Built agents that generate **domain-specific insights** from specialised datasets, simulating authentic industry scenarios across multiple sectors.
- Implemented **tool use and function calling**, agent memory and reasoning loops; developed high-level design before writing code, then translated the design into working Python.
- Created and deployed **custom GPTs** tailored to a domain, including testing scenarios for accuracy, reliability and tone control.
- Applied prompt-engineering patterns to guide agent behaviour and **trustworthy-AI techniques** — problem framing to reduce risk, output verification and validation, and designing for human oversight rather than replacement.
- Used ChatGPT Advanced Data Analysis for autonomous analysis workflows: turning spreadsheets into visualisations and slide decks, extracting and querying PDF data, and automating document transformation.

Stack: Python · OpenAI API and GPTs · LangChain · prompt engineering · document analysis agents · data-analysis automation

AI Agents and Agentic AI in Python: Powered by Generative AI Specialization — Vanderbilt University · Issued January 2026 · 4 courses

Framework-independent agent engineering: understanding and building the core machinery of AI agents rather than depending on a single library.

Core courses in the series:

- AI Agents and Agentic AI with Python & Generative AI (11 h) — agent loops, the GAME model (Goals, Actions, Memory, Environment), Python metaprogramming for flexible agent frameworks, connecting agents to file systems and external APIs, and building developer-productivity agents for code, tests and documentation.
- AI Agents and Agentic AI Architecture in Python (7 h) — architectural design of resilient, modular, maintainable agents optimised for token efficiency, speed and predictable outcomes.
- Companion courses in prompt engineering and applied agent development integrated into the same specialization series.

Final project and applied work:

- Designed and implemented a **complete AI agent system in Python** with reasoning, memory and tool access, integrating LangChain and the OpenAI API.
- Built agents that **operate autonomously**, interact with APIs, retrieve external data and recover from errors instead of failing silently.
- Modelled **agent communication protocols**, task delegation and coordinated multi-agent workflows with collaboration logic written in Python.
- Implemented **self-prompting** behaviour for greater autonomy and designed agents independently of any single framework or programming language.
- Evaluated agents with **structured metrics**, mitigated hallucination and error paths, applied ethical deployment practice, and deployed agents through a web app and API.

Stack: Python · LangChain · OpenAI API · vector stores · tool calling · multi-agent orchestration · agent evaluation and deployment

Building AI Agents and Agentic Workflows Specialization — IBM · Issued January 2026 · 3 courses

Modern agentic engineering with LangGraph, CrewAI, AG2 (AutoGen) and BeeAI across OpenAI GPT, Meta Llama and IBM Granite models.

Courses completed (all 3):

- Fundamentals of Building AI Agents
- Agentic AI with LangChain and LangGraph
- Agentic AI with LangGraph, CrewAI, AutoGen and BeeAI

Lab and project work completed (every lab in the specialization):

- Built an **AI Math Assistant with LangChain** — tool calling with calculators and code execution to solve problems LLMs fail at unaided.
- Performed **AI-powered data analysis with LCEL** (LangChain Expression Language), composing declarative chains over data.
- Built **interactive LLM agents with tools**, using LCEL tool binding and manual tool calling for greater control and accuracy.
- Used built-in agents that **analyse data, create visualisations and run SQL queries** from natural language.
- Built a **Reflection Agent with LangGraph** and a **Reflexion Agent with external knowledge integration**, so agents critique and improve their own output.

- Implemented **ReAct agents that reason before acting**, and designed stateful LangGraph workflows with memory, iteration and conditional logic.
- Built **DocChat — a multi-agent RAG system** — and applied agentic RAG over document collections.
- Implemented **CrewBase** and structured a crew; built a **structured meal and grocery planner with CrewAI** using YAML and Pydantic outputs and custom functions.
- Built agentic AI systems with **BeeAI**, and a **multi-agent chatbot with AG2 (AutoGen) for healthcare** using role-based collaboration.
- Studied and applied agentic design patterns — **sequential, routing and parallelisation** — plus orchestration, query routing and governance strategies for scalable, reliable multi-agent systems.

Stack: Python · LangChain · LangGraph · CrewAI · AG2 (AutoGen) · BeeAI · LCEL · IBM Granite · Meta Llama · OpenAI GPT · cloud lab environments

Generative AI Automation Specialization — Vanderbilt University · Issued January 2026 · 4 courses

Expert-level prompt engineering and applied LLM automation for professional and personal productivity.

Courses completed (all 4):

- Prompt Engineering for ChatGPT
- ChatGPT Advanced Data Analysis
- Trustworthy Generative AI
- GPT Vision: Seeing the World through Generative AI

Projects and applied exercises:

- Turned an **Excel file into visualisations and then into a PowerPoint presentation** entirely through conversational prompting.
- Built **PDF data-extraction pipelines** — pulled data from a series of PDFs, answered questions about their content and visualised the results.
- Automated **receipt and document compliance checking** — determined whether a receipt complies with a travel policy captured in a PDF.
- Transformed a document into a **training presentation with associated quizzes**, and reorganised document sets based on their content.
- Produced **social media and marketing content** from document sets and video transcripts.
- Automated **resizing and editing of videos and images** while cataloguing assets in CSV.
- Applied **GPT Vision** to convert photographs into recipes, shopping lists, DIY plans, reorganisation plans, social posts, organised notes, emails and expense entries — and used photo-versus-plan comparison for progress tracking.
- Practised **trustworthy generative AI** techniques: prompt designs that produce reliable outputs, output verification and validation, problem reframing to reduce risk, and human-in-the-loop engagement.

Stack: ChatGPT Advanced Data Analysis · GPT Vision · prompt engineering · document and image automation

IBM AI Developer Professional Certificate — IBM · Issued December 2025 · 10 courses · IBM digital badge

Software engineering and AI-application development from first principles: programming, web development, Flask, generative AI applications and career preparation.

Courses completed (all 10):

- Introduction to Software Engineering (15 h) — SDLC, Agile and Scrum, programming foundations, design, architecture and deployment; hands-on IDE and Python labs.
- Introduction to Artificial Intelligence (AI) (14 h) — machine learning, deep learning, neural networks, generative AI models, NLP, computer vision, AI ethics and governance; lab and project work.
- Generative AI: Introduction and Applications (9 h) — generative AI across text, image, audio, video, code and data; GPT, DALL-E, Stable Diffusion, Synthesia; labs with IBM Generative AI Classroom and ChatGPT.
- Generative AI: Prompt Engineering Basics (11 h) — zero-shot and few-shot prompting, the Interview Pattern, chain-of-thought and tree-of-thought, multimodal prompting, image generation and output evaluation.
- Introduction to HTML, CSS, & JavaScript (15 h) — HTML5 structure and forms, DOM and scripting, CSS selectors and layout with Flexbox, responsive design with Tailwind CSS and Bootstrap, JavaScript variables, control structures, functions, prototypes and DOM interaction; final project building a web page portfolio piece.
- Python for Data Science, AI & Development (25 h) — Python from zero: data types, data structures, branching, loops, functions, exception handling and OOP; file handling; NumPy and Pandas; REST APIs with requests and web scraping with BeautifulSoup.
- Developing AI Applications with Python and Flask (12 h) — Flask web applications, routing, GET/POST handling, dynamic routes and error management; clean modular Python; static code analysis; writing and running unit tests; a peer-reviewed AI-enabled Flask application project.

- **Building Generative AI-Powered Applications with Python (15 h)** — guided builds with Python, Flask, Gradio and LangChain: voice assistant, meeting summariser, language translator and personalised career coach, using GPT-3, Llama 2 and Flan-UL2 on IBM watsonx and Hugging Face; RAG for external knowledge; speech-to-text and text-to-speech with IBM Watson Speech Libraries and OpenAI Whisper.
- **Project: Generative AI Applications with RAG and LangChain** — production-ready QA bot that retrieves and reasons over private documents (PDF, CSV and URLs) with text splitting, vector database indexing and a Gradio interface.
- **Software engineering career preparation** — job-search strategy, resume and portfolio building, cover letters, elevator pitch, networking, interview process preparation from screening through final rounds, and post-interview follow-up.

Projects completed in this program:

- **Portfolio website** built with HTML, CSS and JavaScript.
- **Sentiment analysis application** using Python, Flask and embedded AI libraries.
- **AI photo captioning** — generated meaningful captions for photographs with generative AI.
- **ChatGPT-like website** built on open-source LLMs.
- **Voice assistant** created with OpenAI GPT APIs and IBM Watson libraries.
- **AI meeting companion** built with Llama.
- **Private-data summarisation** with LLMs and generative AI.
- **Babel Fish** — a voice-enabled universal language translator built with Flan and Gradio.
- **Document Q&A bot** — complete RAG pipeline over private PDFs, CSVs and URLs with vector indexing and Gradio UI.

Stack: Python · Flask · Gradio · LangChain · HTML5 · CSS3 · JavaScript · IBM watsonx · Hugging Face · OpenAI Whisper · IBM Watson Speech Libraries · Pandas · NumPy

INDEPENDENT ENGINEERING PROJECTS — GITHUB.COM/DEV4AIBOTS

Ten public repositories — nine independent, production-oriented AI system builds plus a portfolio index. Each project is engineered end to end (architecture, implementation, evaluation, documentation) and deployed on Vercel where a live demo is available.

PROJECTS — Portfolio Index	Master portfolio index for the production-oriented AI systems built around agentic AI, RAG, LLMs, evaluation and reliability. Live · Source (HTML)
enterprise-rag-engine	Production-oriented Retrieval-Augmented Generation system with a built-in evaluation harness: document ingestion, chunking, embedding, vector search and answer-quality measurement. Live · Source (TypeScript · Python · FastAPI · vector search)
infra-diagnostics-agent	Autonomous DevOps agent performing root-cause analysis over Kubernetes and Prometheus telemetry — investigates signals, correlates evidence and reports the probable cause and remediation. Live · Source (TypeScript · Python · Kubernetes · observability · agents)
semantic-code-search	Repository-scale search engine that maps natural-language questions onto AST nodes, combining embeddings with abstract-syntax-tree structure for precise, semantics-aware code retrieval. Live · Source (TypeScript · Python · AST · vector search · ML)
agentic-cli-workspace	Production-oriented coding agent that performs repository analysis, implementation and testing from the command line — plan, edit, run tests and verify its own changes. Live · Source (TypeScript · LangChain · LLM · agents)
mcp-tool-registry	Model Context Protocol implementation that standardises tool execution — a registry and runtime for discovering, exposing and invoking tools safely across agents and hosts. Live · Source (TypeScript · Python · MCP · agents)
deep-research-orchestrator	Agentic workflow for long-horizon planning and deep web research: decomposes a question, plans investigation steps, gathers and reconciles sources, and synthesises a cited report. Live · Source (Python · agents · LLM · research orchestration)
realtime-voice-ai	Low-latency conversational voice AI built on WebSockets — streaming speech interaction with barge-in handling for natural, responsive dialogue. Live · Source (Python · WebSockets · LLM · voice)
multimodal-vision-qa	Vision-first AI pipeline that extracts structure from PDFs and charts and answers questions grounded in the visual content, rather than relying on text layers alone. Live · Source (Python · vision · LLM · ML)
llm-finetuning-framework	End-to-end MLOps pipeline for fine-tuning open-source LLMs — dataset preparation, parameter-efficient training runs, evaluation and packaging for downstream use. Live · Source (Python · PyTorch · MLOps · fine-tuning)

ai-engineer-portfolio

Centralised portfolio hub that aggregates and presents the AI engineering project set with architecture notes and deployment links. [Source \(HTML\)](#)

ADDITIONAL INDEPENDENT PROJECT EXPERIENCE

- **AI customer-support and automation platform architecture** — conversation persistence, lead capture and qualification, appointment booking, follow-up sequencing, CRM integration, usage tracking and workflow orchestration.
- **Document AI and RAG systems** — OCR, embeddings, vector search and LLM-based question answering over unstructured business information.
- **AI-agent and automation architecture** — tool routing, workflow orchestration, memory and context handling, API and database persistence, validation, logging, testing and deployment design.
- **AI-assisted software engineering workflow** — requirements to architecture to database design to implementation to testing and debugging, executed with modern coding agents and development tooling.

EDUCATION, CREDENTIALS AND MATHEMATICS PREPARATION

Professional education	Coursera — 12 verified credentials (2025-2026) from DeepLearning.AI, IBM, Vanderbilt University, UiPath and Edureka, covering generative AI, RAG and agentic AI, deep learning and computer vision, LLM fine-tuning, software engineering, RPA and AI automation. These are professional learning credentials, not university degrees, and no university degree is claimed.
Coursework volume	Approximately 45 individual courses completed across the twelve programs — every course listed on this resume was completed and passed, together with all labs, graded assignments and capstone projects.
Certificate verification	Every credential links to its Coursera accomplishment page (see Certifications, page 1). Credential identifiers: O7DG9ZERU8H5 · O74711W8MCDW · DZBT1TGK7UHO · LHXC3NICCSN4 · US2X5C7BJSA1 · Z0OKDI2ZR9FU · FES503EY9D5T · RXMXCXEXK0X0Z · PAXQSQ3YEBW6 · OZ5YDDG7DBF7 · 0DGZLL2L2MAI · VZGP0VOJ4RZA
Statistics and probability	Statistical methods, descriptive analysis, probability concepts, distributions, correlation, and interpretation of model results.
ML mathematics	Regression and classification reasoning, loss and error concepts, optimisation and model optimisation, evaluation metrics, transfer-learning concepts.
Linear algebra and vectors	Vectors and matrices and their role in embeddings, neural networks and transformations; cosine similarity and semantic distance in high-dimensional vector space; token economics and context-window constraints.
Analytical reasoning	Model comparison, framework trade-off analysis, retrieval-quality evaluation, validation, error analysis, data-driven decision making and structured problem solving.
Scope note	This resume positions mathematical and analytical foundations as supporting evidence for AI and ML engineering work. It does not claim a university-level mathematics degree or advanced theoretical mathematics specialisation.

PROFESSIONAL STATUS, AVAILABILITY AND HOW THIS RESUME WAS VERIFIED

Employment status	No formal paid employment experience is claimed. The experience on this resume is one year of documented professional training and independent engineering work, presented transparently so hiring teams can evaluate the evidence directly.
Education status	No completed university degree is claimed. Credentials are professional certificates and specializations from Coursera and its partners.
Proof of work	Twelve verifiable Coursera credentials · every program course completed · all capstone, lab and applied projects · ten public GitHub repositories with live deployments on Vercel · portfolio at dev4aibots.com.
Availability	Immediate start. Global remote contract, freelance, project-based and full-time entry-level roles — including internships and graduate programmes — subject to each employer's and platform's eligibility rules.
Working preferences	Remote or hybrid · async-friendly teams · AI engineering, software engineering, automation and developer-tooling roles where shipping working systems matters more than years served.

Verification basis. Program names, course lists, course hours, capstone descriptions and applied-project details on this resume were cross-checked against the current official Coursera, DeepLearning.AI, IBM and Vanderbilt program pages and their published syllabi. Descriptions are deliberately limited to what those sources support. Project entries under "Independent Engineering Projects" reflect the author's own public repositories at github.com/dev4aibots.