

SHUBHAM JAMDAR

Generative AI & Agentic AI Engineer | AI Automation | LLM | RAG | LangGraph | MCP | Python | UiPath

Pune, Maharashtra, India | shubhamjamdar2612@gmail.com | Portfolio: shubham-ai-lab.vercel.app
linkedin.com/in/shubham-jamdar-2571271b3 | github.com/Shubham261299

PROFESSIONAL SUMMARY

Generative AI Engineer with 2.8+ years building Python automation and GenAI / Agentic AI systems for enterprise workflows. Currently deployed at IDEaS Revenue Solutions, designing LangGraph multi-agent orchestration with human-in-the-loop approval, Model Context Protocol (MCP) servers that give AI agents secure, audited access to Salesforce CRM data, and local LLM pipelines for report generation. Hands-on with LLMs, RAG, AI agents, tool calling, prompt engineering, vector databases (FAISS, ChromaDB), embeddings, LangChain, FastAPI, Pydantic and Docker across OpenAI, Claude, Hugging Face and Ollama models. Previously at Searce, delivered 10+ production automation frameworks (UiPath RPA, Selenium/Playwright, OCR-based intelligent document processing, REST APIs, SQL) handling 3,000+ transactions/month and cutting manual effort by ~80%, plus a Gen AI RAG assistant and LLM-based exception handling.

TECHNICAL SKILLS

Agentic AI: Agentic AI, AI Agents, Multi-Agent Systems, Agent Orchestration, LangGraph (StateGraph, interrupt(), checkpointing), Tool Calling / Function Calling, ReAct, Human-in-the-Loop Approvals, Stateful Workflows, Execution Tracing, Model Context Protocol (MCP)

Generative AI & LLMs: Generative AI (GenAI), Large Language Models (LLM), LLM Applications, RAG (Retrieval Augmented Generation), LangChain, LiteLLM, Prompt Engineering, Context-only Prompting, Structured JSON / Schema-constrained Output, Guardrails, Microsoft 365 Copilot Agents grounded in SharePoint/OneDrive documents

Models & Retrieval: OpenAI API, Anthropic Claude, Google Gemini, Hugging Face, Ollama (Llama 3.1/3.3, Phi-4, Qwen2.5), Embeddings (all-MiniLM-L6-v2), Vector Databases, FAISS, ChromaDB, Semantic Search

Testing & Evaluation: pytest (unit, integration/API tests), UAT Support & Test Evidence, RAG Validation, LLM Evaluation (DeepEval, LLM-as-Judge)

Backend & Integration: Python, FastAPI, Starlette/Uvicorn, Pydantic v2, SQLAlchemy 2 (async), Alembic, Flask, httpx, REST APIs, JSON Schema, Salesforce Integration (SOQL), API Authentication, Rate Limiting, Audit Logging, Streamlit

Frontend: React, TypeScript, Vite, Tailwind CSS, shadcn/ui

RPA & Automation: UiPath, UiPath Orchestrator, REFramework, RPA, Power Automate, Python Automation, Playwright, Selenium, PyAutoGUI, Outlook / VBA Automation, Intelligent Automation, Workflow Automation

Document Intelligence: OCR, PyTesseract, EasyOCR, OpenCV, PyMuPDF, PDF Processing, Intelligent Document Processing (IDP)

Data & BI: SQL, SQL Server, PostgreSQL, SQLite, pandas, NumPy, matplotlib, openpyxl, Power BI, DAX, Excel, ETL, KPI Reporting

DevOps & Observability: Docker, Docker Compose, nginx, GitHub Actions (CI), Git, GitHub, Linux, Shell/PowerShell, Structured JSON Logging, OpenTelemetry, Postman, VS Code, Jupyter

Delivery: Agile/Scrum, Jira, Confluence, SharePoint

PROFESSIONAL EXPERIENCE

Automation Engineer | IDEaS Revenue Solutions (via staffing partner), Pune

Jul 2026 – Present

- Deployed at IDEaS Revenue Solutions (hotel revenue-management SaaS, a SAS company) with the Revenue Operations Analytics team, building Generative AI, Agentic AI and AI automation solutions using Python, LangGraph, FastAPI, LLMs and Model Context Protocol (MCP).
- Designed and built a multi-agent orchestration engine (LangGraph StateGraph, FastAPI, Pydantic v2) that routes hotel revenue-management support cases to pluggable AI agents (pricing, overbooking, Last Room Value, forecasting), with a critic node and bounded re-planning, full execution tracing and a human approval gate (LangGraph interrupt) before a response is finalised; paused cases are checkpointed to SQLite and survived a manual kill-and-restart with no state loss.
- Analysed a ~41,000-case Salesforce support export to validate the routing categories the agents cover.
- Hardened every LLM step: schema-constrained JSON output with one repair retry and a deterministic keyword-match fallback, plus agent failure isolation (20s timeouts; a failed agent never crashes the case) and structured logs correlated by case and agent.
- Built a standalone Salesforce MCP server (official MCP Python SDK, Streamable HTTP, Docker Compose) exposing 11 read-only CRM tools to AI agents, with fail-closed API-key authentication, rate limiting, table allow-listing, query guards and per-key audit logging; identified and fixed an authentication bypass in the reference codebase.
- Built and deployed internally a Smartsheet MCP server giving AI agents controlled read and write access: per-agent identity, a resource registry, resource- and operation-level permissions, per-key rate limits, audit logging and an admin panel, with row deletion disabled by default.
- Built an SFDC-G3 automation pipeline: Playwright-based Salesforce case monitoring, SQL Server state management, Outlook/VBA email intake queuing jobs for a Python worker, and automated G3 MR / FVR / Pricing report retrieval and consolidation, with an operational dashboard.
- Building a local LLM report-generation pipeline (pandas, matplotlib, Ollama) that replaces a Power Automate + AI Builder cloud flow for Forecast Review reports, passing only a deterministic facts packet to the LLM to keep numbers grounded.

- Built and maintained 10+ Python automation frameworks (RPA, browser automation, API integration and document processing) supporting 3,000+ transactions/month across logistics and accounts-payable workflows, reducing manual effort by ~80%.
- Built intelligent document processing (IDP) for invoices using Python, OCR (PyTesseract), OpenCV, PyMuPDF and REST APIs, cutting processing time from ~10 to ~2 minutes per invoice at ~95% accuracy.
- Developed RPA and web automation with UiPath, Selenium and Playwright for logistics operations and carrier-portal tracking, eliminating ~3 hours/day of manual shipment tracking.
- Built a Generative AI RAG assistant for SOP documents (LangChain, FAISS, Hugging Face embeddings, local LLM) with context-only prompting and page-level source citations.
- Integrated LLM-based decisioning into rule-based exception handling, reducing exception-handling effort by ~40%.
- Built 8+ Power BI / SQL dashboards (TAT, cost, carrier performance), reducing leadership reporting time by ~40%.
- Improved automation stability by ~30% through structured exception handling, retries and monitoring.
- Owned delivery end to end: requirements gathering and solution design with operations stakeholders, development, testing/UAT, deployment and L2/L3 production support.

KEY PROJECTS — PROFESSIONAL

Agentic Case-Routing Orchestrator (LangGraph Multi-Agent)

IDeaS | Jul 2026 – Present

Tech stack: Python 3.13 · LangGraph · FastAPI · Pydantic v2 · SQLite · httpx · Ollama

- Orchestration engine that routes hotel revenue-management support cases to independent, pluggable agent services (pricing, overbooking, Last Room Value, forecasting) behind a typed Pydantic v2 contract over httpx; adding an agent is a registry entry, not a graph change.
- LangGraph flow: understand → plan → dispatch agents concurrently → critic (PASS/FAIL) with up to 3 bounded re-plans → draft response → human approval via LangGraph interrupt(), with a bounded revise loop when changes are requested.
- Every step traced (input, output, duration); schema-constrained LLM output with a repair retry and keyword fallback; 20s agent timeouts isolate failures; structured JSON logs correlated by case, task and agent.
- State checkpointed with AsyncSqliteSaver: a case paused for approval survived a manual kill-and-restart of the orchestrator with no state loss. A local LLM (Ollama) handles understanding, planning, critique and drafting. Working prototype; domain agents are mocked pending the domain teams.

Salesforce MCP Server for AI Agents

IDeaS | Jul 2026 – Present

Tech stack: Python · MCP SDK · Starlette/Uvicorn · PostgreSQL · SQLite · Docker Compose

- Standalone Model Context Protocol server exposing 11 read-only Salesforce tools over Streamable HTTP so any AI agent can connect through a connector.
- Fail-closed API-key system with admin key management, rate limiting and per-key audit logs; table-level allow-listing and query guards after finding 100+ out-of-scope tables; live-SOQL fallback gated behind a per-key opt-in flag.
- Found and fixed an authentication bypass in the reference codebase during architecture review; v1.1 deployed internally for team testing.

Smartsheet MCP Server for AI Agents

IDeaS | 2026 – Present

Tech stack: Python · MCP SDK · Streamable HTTP · SQLite · Docker

- Standalone MCP server giving internal AI agents and MCP clients (Claude Code, Cursor, LangGraph agents) one controlled Smartsheet integration: 8 read tools plus controlled add_rows / update_rows, with row deletion disabled by default.
- Per-agent identity on every API key, a resource registry with stable IDs instead of raw sheet IDs, and authorization at resource and operation level before any call reaches Smartsheet.
- Per-key rate limits, audit logging of every tool call, structured error codes and an admin panel for keys, permissions, usage and audit; deployed internally with Docker.

SFDC-G3 Report Automation Pipeline

IDeaS | Jul 2026 – Present

Tech stack: Python · Playwright · SQL Server · Outlook VBA · REST APIs

- Playwright monitoring of new Salesforce cases via a custom report, tracked in SQL Server, triggering downstream G3 MR, FVR and Pricing report downloads.
- Outlook/VBA intake processes report emails and queues jobs for a Python Playwright worker, with a health check, broadened G3 domain handling and a property/date/case folder convention for consolidation.

Local LLM Forecast Review Report Generator

IDeaS | 2026 – Present

Tech stack: Python · pandas · matplotlib · Ollama

- Replacing a Power Automate + AI Builder cloud flow with a fully local pipeline: pandas reads FVR exports, matplotlib charts, deterministic analysis builds a compact facts packet, and a local LLM writes ROA-style prose into V1/V2 Word reports.

AI-Powered SOP Assistant (GenAI + RAG)

Searce | Dec 2025 – Apr 2026

Tech stack: Python · LangChain · FAISS · Hugging Face · Qwen2.5 · PyMuPDF | github.com/Shubham261299/sop-assistant

- Built a local-first RAG pipeline ingesting SOP PDFs via PyMuPDF, chunked with RecursiveCharacterTextSplitter (800/150) and indexed in FAISS with all-MiniLM-L6-v2 embeddings.

- Applied strict context-only prompting with explicit refusal instructions to prevent hallucination in compliance-sensitive retrieval.
- Surfaced deduplicated page-level source citations for human verification; projected ~96% reduction in lookup time for 500-page SOP corpora.

End-to-End Invoice Automation (IDP)

Source | Dec 2023 – Jul 2026

Tech stack: Python · OCR · OpenCV · PyMuPDF · REST APIs · UiPath

- Extracted and validated invoice data from PDFs using PyTesseract, OpenCV and PyMuPDF.
- Integrated carrier REST APIs for real-time validation; applied rule-based and LLM-assisted validation logic.
- Processed 3,000+ transactions/month at ~95% accuracy, cutting processing time from ~10 to ~2 minutes per invoice.

Shipment Tracking Bot

Source

Tech stack: Python · Selenium · Outlook VBA

- Web automation monitoring AWB statuses across carrier portals, with automated Outlook/VBA email alerts for delays; eliminated ~3 hours/day of manual tracking.

Logistics Dashboard Suite

Source

Tech stack: Power BI · SQL · Python · APIs

- Built 8+ interactive dashboards tracking TAT, costs and carrier performance on automated SQL and API data pipelines; reduced leadership reporting time by ~40%.

PERSONAL PROJECTS

OpsPilot – AI Back Office for a Small Distributor

Personal Project | Sep 2026 – Present

Tech stack: Python · FastAPI · LangGraph · LiteLLM · Ollama · PostgreSQL · SQLAlchemy 2 · Alembic · React · TypeScript · Docker · GitHub Actions | github.com/Shubham261299/opsilot

- Building in public (week 3 of 7, v0.2 done): turns WhatsApp orders, an Excel stock sheet and supplier invoices into clean data; agents propose reorders, payment reminders and order holds, and the owner approves every action.
- Cleaners report 15 of 15 planted data problems with no false alarms; policy rules hit 8 of 8 reorders and 10 of 10 payment actions; the local-LLM WhatsApp parser gets 44 of 45 lines exact with 0 wrong products or quantities (answer-key tests).
- LangGraph approval agents pause at interrupt() with a Postgres checkpoint; every number in LLM-written explanations is checked by code (33 of 34 passed on the first run).
- React + TypeScript web app, Docker Compose + nginx, and GitHub Actions CI (lint, tests against real Postgres, frontend build).

Agentic Tool Orchestrator

Personal Project

Tech stack: Python · LangGraph · LangChain · Anthropic Claude · FastAPI · Pydantic | github.com/Shubham261299/agentic-tool-orchestrator

- Built a LangGraph ReAct agent with 12 callable tools for multi-step natural-language tasks.
- Exposed it through FastAPI with Pydantic-validated inputs/outputs and structured safety guardrails.
- Implemented LLM-as-Judge evaluation scoring correctness, completeness, clarity and tool-use appropriateness.

Daybook – Local AI Engineering Journal

Personal Project | Jul 2026 – Present

Tech stack: Python · FastAPI · SQLite · GitPython · Ollama · Docker | github.com/Shubham261299/daybook

- Local-first app that collects Git commit activity and notes and uses a local LLM (Ollama) with Pydantic schema-constrained output to write daily, weekly and monthly summaries; ~24x faster incremental collection, Docker Compose, in daily use.

Self-Healing Invoice Automation

Personal Project

Tech stack: Python · Playwright · Claude API · OCR · Streamlit | github.com/Shubham261299/self-healing-invoice-automation

- Engineered LLM-powered browser automation that recovers from broken DOM selectors at runtime: on Playwright failure it captures live HTML, prompts Claude to identify the intended element and retries.
- Demonstrated 5 of 5 simulated portal failures recovered with zero manual intervention in the public demo.

YouTube Researcher – ReAct AI Agent with RAG

Personal Project

Tech stack: Python · Ollama · ChromaDB · Embeddings

- Built a ReAct agent on a local LLM (Ollama) with YouTube search, transcript fetching and RAG over persistent ChromaDB embeddings; fixed a grounding bug that caused hallucinated answers.
- Standardized tool/LLM contracts, added trace logging and multi-session memory across 3 versioned releases.

CERTIFICATIONS

- UiPath Automation Developer Professional Training — UiPath Academy
- Full Stack Generative & Agentic AI with Python — Udemy | Python — Kaggle
- Forage Virtual Job Simulations: BI Developer (Apr 2023), Data Analyst (Dec 2022)

EDUCATION

B.Tech in Information Technology (Specialization: Data Science)

2019 – 2023

Ajeenkya DY Patil University, Pune | CGPA: 7.85 / 10

Languages: English, Hindi, Marathi