Rishi pandey, Python Developer

Ahmedabad, 9510310510, riship4611@gmail.com

LINKS

Linkdin, github, Portfolio

PROFILE

Dynamic Python developer with **3.5+ years of experience** in building scalable, production-ready applications, with a strong focus on backend systems, AI integration, and cloud deployment. Proficient in **FastAPI**, **Docker**, and **PostgreSQL**, with hands-on expertise in deploying services on **AWS EC2** and managing containerized environments.

Experienced in developing complex applications using **LLMs**, **LangChain**, and **LangGraph**, including **RAG pipelines**, **vector databases** (pgvector, Redis), and tool-augmented agents. Skilled in **data ingestion**, **PDF parsing**, and **web scraping**, with a focus on clean architecture, scalability, and performance optimization. Passionate about delivering reliable, maintainable code and driving innovation in fast-paced, cross-functional teams.

SKILLS

Python

Docker

Fastapi

LangChain

LangGraph

AWS (EC2, S3, Bedrock)

Linux

Git

Django

Pandas

Web scraping

Postgresql

EMPLOYMENT HISTORY

Jun 2022 — Present

Python Developer, Brainerhub solutions

Currently employed as a Python developer at Brainerhub solutions, where my role involves developing and maintaining Python-based applications. I utilize my skills in Python, FastAPI, PostgreSQL, Git, Lang-chain, SQL, Docker, web scraping, and Django to deliver high-quality software solutions.

Dec 2021 — Jun 2022

Python developer, Coodeit solutions

At Coodeit Solutions, I commenced my tenure as a Backend Developer, engaging in a pivotal project amalgamating Artificial Intelligence with backend development. This multifaceted endeavor entailed leveraging Django Rest framework in conjunction with SQL database technologies. My role encompassed implementing robust backend solutions that seamlessly integrated AI functionalities, thereby contributing to the project's overarching objectives with professionalism and expertise.

PROJECTS

Ai-Agent

Tools: LangChain, LangGraph, Docling, FastAPI, AWS Bedrock, Neo4j, Redis, Docker, AWS EC2

Built an AI-powered agent that generates **risk matrices for industrial activities** based on **user queries or uploaded images**.

- Developed two custom **knowledge bases**: One from **structured Excel data**, converted into vector embeddings with detailed metadata. Another from **unstructured sources**, enabling broader context-aware reasoning.
- Integrated **LangChain agents with LangGraph**, enabling tool-augmented decision-making and response generation.
- Employed **Redis** to maintain chat context across sessions for more coherent conversations.
- Leveraged **AWS Bedrock** for scalable LLM inference and **Neo4j** for querying relational data structures.
- Containerized the application using **Docker** and deployed it on **AWS EC2** for production use.
- Built a **lightweight frontend interface** to allow easy interaction with the agent.

RAG Pipeline for Medical Reports

Tools: FastAPI, LangChain, Docling, pgvector, Docker

Developed a **Retrieval-Augmented Generation (RAG)** pipeline to enable patients to **ask natural language questions** about their medical reports uploaded by doctors.

- Built a FastAPI-based backend to manage report uploads, query handling, and LLM responses.
- Used **Docling's text extraction** to accurately parse tabular data from complex PDF medical reports.
- Stored vector embeddings in **pgvector**, running in a Docker container, to enable efficient and scalable similarity search.
- Integrated **client-hosted embedding and LLM models** for data privacy and custom performance.
- Containerized the application using **Docker** for consistent local development and deployment environments.

Digital Twin

Tools: FastAPI, Docker, SQLAlchemy, PostgreSQL, Alembic, AWS S3, Databricks, Pytest

Led the development of a **Digital Twin platform** at Intemic, designed to simulate real-world scenarios digitally and enable data-driven decision-making.

- Developed **role-based user and organization management** to ensure secure, structured access to data and services.
- Created APIs for **uploading CSV/Excel files** and dynamically generating database tables to support flexible data ingestion.
- Wrote comprehensive **unit and integration tests using Pytest** to validate core functionalities and ensure stability.
- Built **Docker images and Docker Compose configurations** to support efficient local development and scalable deployment.
- Integrated **AWS S3** for file storage and **Databricks** for advanced data processing and analytics.
- Collaborated with cross-functional teams across backend, data engineering, and DevOps to deliver a production-ready system.

Discord Bot

Tools: Python, PostgreSQL, Docker, Docker Compose, Alembic, Shell Scripting, AWS EC2, Pillow

Designed and developed a **feature-rich Discord bot** from scratch, acting as a multifunctional assistant for community interaction and automation.

- Built entirely in **Python**, with modular design to support extensible features and commands.
- Used **PostgreSQL** for persistent data management and **Alembic** for handling database schema migrations.
- Containerized the application using **Docker** and orchestrated services with **Docker Compose** for smooth development and deployment.
- Automated key operations through **shell scripts**, improving management and deployment efficiency.
- Deployed the bot on **AWS EC2**, ensuring high availability and scalability.
- Integrated **third-party APIs** for enhanced functionalities and **Pillow (PIL)** for dynamic image generation and manipulation.

OTHER SMALL PROJECTS

Face detection using Python

Developed a Python project for similar face detection using the face detection library. This project aimed to identify and compare similarities between faces in images, enabling applications such as facial recognition and similarity analysis.

Technologies Used:

- **Programming Language:** Python
- **Libraries:** Face detection library

Assert Gateway

Developed a robust inventory management system using FastAPI for backend API services and Jinja templates for frontend rendering. Leveraging PostgreSQL and SQLAlchemy 2.0, the system provided efficient data storage and retrieval capabilities, ensuring seamless inventory tracking and management.

Technologies Used:

- **Backend Framework:** FastAPI
- **Frontend Templating:** Jinja
- **Database:** PostgreSQL

- **ORM:** SQLAlchemy 2.0

Web scrapping Project

Developed a custom Django API to scrape data from sports news articles based on client requirements. Leveraging Selenium and BeautifulSoup, the API provided dynamic web scraping capabilities, allowing clients to retrieve relevant sports news data programmatically.

- **Web Scraping:** Selenium, BeautifulSoup

EDUCATION

Aug 2018 — May 2021

BTech Electrical Engineering , Gyanmanjari institute of technology

Bhavnagar

Aug 2015 — May 2018

Diploma Electrical engineering, Sir Bhavsinhji Polytechnic Institute

Bhavnagar