Divya
Full Stack Developer

Summary

● I am a Backend-focused Fullstack Developer with over 5+ years of experience, specializing in Python development and building scalable, production-ready backend systems and APIs using Flask and FastAPI. I also have hands-on experience in LLM integration using platforms like OpenAI and Gemini, and working with tools such as LangChain and vector search engines.
● Skilled in developing intelligent backend workflows, integrating LLMs into applications using RAG pipelines, and processing data with libraries like Pandas. I have experience working with RESTful APIs, data scraping, Google Search, and Google Lens API.
● Proficient in designing and managing databases including MySQL, PostgreSQL, and MongoDB, ensuring efficient data storage and retrieval for high-performance applications.
● Experienced in building interactive dashboards with Streamlit, and designing responsive front-end interfaces using React.js, HTML5, and CSS3 for UI/UX purposes.
● Continuously learning and implementing modern backend technologies to build efficient, maintainable, and secure systems. Strong understanding of OOP, software architecture, and deployment best practices.

Technical Skills

- **Languages**: Python (advanced),JavaScript, PHP,
- **Back-End Frameworks:** Flask, FastAPI, Node.js, Express, Laravel, Frappe
- **AI-ML:** Data Analytics, Business Intelligence, Machine Learning, Predictive Modelling, Data Visualization, Data Mining, Deep Learning, Computer Vision, NLP, GenAI, Hypothesis Testing, Prompt Engineering, Agentic AI, Explainable AI
- **Machine Learning:** Supervised-Unsupervised Learning, Linear & Logistic Regression, Random Forest, Clustering, Feature Engineering, Bagging & Boosting, Decision Trees, Naive Bayes, SVM, SVD/PCA, Anomaly Detection, Dimensionality Reduction Algorithms.
- **Statistical Analysis:** Statistical Inference, Hypothesis Testing, Regression Analysis, Statistical Models, Mathematics, Probability, ANOVA, Z-test, T-Test, F-Test, Chi-Square Test, Descriptive Statistics, Inference Statistics, Bayesian Inference
- **Deep Learning:** Neural Networks, MLP, RNN, LSTM, GRU, CNN, Back Propagation, Gesture Recognition, Transfer Learning, Regularization, Hyperparameter Tuning, Batch Normalization, Object Detection, Semantic Segmentation, GAN, Transformer, Autoencoder
- **Natural Language Processing (NLP):** Text Preprocessing, Tokenization, Stemming,

Lemmatization, Part-of-Speech (POS) Tagging, Named Entity Recognition (NER), Machine Translation, Speech Recognition-Generation, Text Summarization, Sentiment Analysis, Topic Modeling, Text Classification, Word Embeddings, Sequence to Sequence learning, Attention Mechanisms, Transformers, BERT, Language Modeling

- **LLM and GenAI Models:** Understanding of Large Language Models (LLM) And Other Generative AI (Genai) Models, Including Prompt Engineering, Model Evaluation, Stable Diffusion, Optimization And Deployment, LLMOps.
- **LLM Training Framework's/Deploying Tools:** LangChain, LangGraph, CrewAI, AutoGen, Swarm, Ollama, LangFlow, LlamaIndex, Chainlit, DeepSpeed, Colossal-AI, Langroid, LMDeploy, TensorRT, PhiData
- **LLM Tools & Concepts:** LangChain, LangGraph, OpenAI API, Gemini, RAG pipelines, Vector search
- **Front-End (Design Focused):** HTML, CSS, Bootstrap, jQuery, React.js, Next.js, Figma to HTML/CSS
- **Databases:** MySQL, MongoDB, PostgreSQL, ChromaDB, Faiss, QdrantDB
- **Data & APIs:** Pandas, Web Scraping, RESTful APIs, GraphQL, Google Search/Lens APIs
- **Tools**: Postman, Swagger
- **Version Control**:Github, Git lab
- **API:** RESTful APIs, GraphQL
- **Cloud**: AWS , GCP, Azure
- **Data Visualisation/Analytics:** Tableau, Seaborn, Ggplot, Matplotlib, Plotly, MicrosoftFabric, Google Analytics, Excel, PowerBI, STATA.
- **Data Wrangling & Preprocessing**: Data Cleaning, Feature Engineering, Dimensionality Reduction, Feature Tools.
- **Model Deployment Tools:** Continuous Integration and Continuous Deployment (CI/CD), GitHub Actions, Docker, Kubeflow, MLOps, MLFlow, DVC, TensorFlow-Serving, ZenML, Kedro, Kubeflow, Airflow, DataRobot, TFX, Streamlit, Heroku, Render (Git), Travis CI, CircleCI, Prefect, AWS ECR EC2, REST API, Flask, FastAPI, Git, GitHub, GitLab, Branching, Merging, Code Reviews.

Project Experience

1) Realtime Product Matching

Description: This intelligent product-matching platform allows clients to seamlessly compare their own products with competitors through an automated, data-driven workflow. Clients initiate the process by selecting products and competitors through a web interface. Upon submission, a backend system—powered by Python—automatically triggers smart search routines leveraging Google Search, Google Lens, and custom web scraping lgorithms to identify and extract relevant competitor product data from across the web.Using advanced data matching and similarity-checking logic, the system efficiently maps competitor listings to the client's products, ensuring accurate comparisons. This enables clients to gain real-time insights into competitor pricing, specifications, availability, and

positioning—helping them make informed business decisions and refine market strategies.

Responsibilities:

- Developed automated Python workflows to extract competitor product data using **Google Search**, **Google Lens**, and web crawling techniques.
- Engineered data-matching algorithms to intelligently pair client products with similar competitor items using metadata, textual similarity, and image-based cues.
- Designed scalable modules to handle high-volume product matching requests in real-time.
- Integrated Python services with the Laravel-based web system to manage form submissions, request queues, and result delivery.
- Monitored and optimized data retrieval pipelines to improve accuracy, performance, and resilience of the product-matching engine.
- Contributed to backend architecture to ensure smooth orchestration between client-side actions and Python-based processing logic.

Technologies Used: Python, Flask API, Mysql, MongoDB, AWS s3

2) Sales Automation Process

Description : This project focuses on automating the end-to-end sales outreach process by leveraging Python, AI/ML concepts, and automation tools. The system identifies potential leads by extracting data from various platforms—such as Google, LinkedIn, Upwork, Freelancer, government portals, and partner websites—based on a defined Ideal Customer Profile (ICP).The collected data is structured and exported to Excel for further processing. An automated outreach workflow then generates personalized email templates and launch email campaigns targeting the collected leads. The system also gathers and analyzes campaign performance metrics, such as total leads found, engagement levels, and identification of high-potential (“hot”) leads.

Responsibilities:

- Developed Python scripts to automate data extraction from multiple online platforms based on ICP criteria.
- Processed and cleaned extracted data, organizing it into Excel files for easy management and outreach use.
- Created dynamic and personalized email templates to support automated campaign outreach.
- Implemented automation for launching and managing email campaigns using the extracted lead data.
- Built analytics dashboards to track campaign effectiveness, including lead generation, open rates, and hot lead identification.
- Applied AI/ML techniques to improve lead scoring and enhance targeting precision.
- Collaborated with cross-functional teams to integrate the automated sales pipeline with existing CRM and marketing tools.

Technologies Used: Python, Flask API, Mysql, pandas, stremlit, React js for UI

3) Automating Project Development & Planning using Python and AI/ML

Description: This project focuses on automating critical aspects of project management and software development workflows. It includes automatically generating Jira tickets from project requirement documents, creating and structuring GitHub repositories aligned with these tickets, and producing linked test cases based on the Jira tasks. The automation ensures streamlined coordination between planning, coding, and testing phases, significantly enhancing efficiency and traceability.

Responsibilities:

- Developed automation to parse project requirement documents (PDF, Word, text) and create corresponding Jira tickets automatically.
- Built scripts to create GitHub repositories structured by Jira tickets, including branch creation per ticket and committing initial files.
- Designed a system to generate test cases automatically from Jira tickets, linking them to user stories and tasks for better traceability.
- Integrated the automation tools to maintain seamless synchronization between project management (Jira), codebase (GitHub), and testing workflows.

Technologies Used: Python, Flask API,React js for UI, Gemini LLM , Groq API,docx2txt, PDF/text handling

4) RAG Based AI Intelligent LLM Chatbot

Description: RAG Chatbot is an AI-powered document assistant that enables instant question-answering from uploaded PDF files using Retrieval-Augmented Generation (RAG). It integrates LangChain, Groq's LLaMA model, and FAISS vector search to deliver fast, accurate, and context-aware responses. Built with Streamlit for an intuitive and interactive UI & ideal for answering & handling queries in real time for students, professionals, and researchers seeking quick insights from large documents without reading them entirely.

Responsibilities:

- Developed the core backend logic using Python and LangChain for orchestrating LLM interactions.
- Integrated Groq’s LLaMA model for generating context-aware responses from document content.
- Implemented FAISS vector search to enable efficient document indexing and semantic retrieval.
- Designed and deployed the user interface using Streamlit for seamless user interaction.

- Enabled PDF upload, parsing, and chunking mechanisms to process large documents in real time.
- Ensured scalability and performance optimization for handling multiple documents and queries.
- Tested and validated RAG responses for accuracy and relevance to improve user trust and experience.

Technologies Used: Groq API, LangChain, LLaMA3, HuggingFace Embeddings, FAISS, Streamlit, Python

5) Intelligent Employee Prospecting & Engagement Platform

Description: AI-powered HR recruitment automation platform that streamlines the end-to-end hiring process. It automatically extracts candidate data from public professional profiles and ICP-based sources, presents it through an intuitive UI with downloadable Excel reports, and automates outreach by sending personalized connection or recruitment messages. The platform further reduces manual effort by automating Google Meet link creation for both candidates and interviewers, managing interview schedules via integrated calendar APIs, and using LLMs to generate interview reviews by analyzing feedback collected during or after the interview process. This ensures an efficient, data-driven, and scalable hiring workflow.

Responsibilities:

- **Developed backend services** in Python using Flask to automate data extraction, message sending, and calendar integration.
- **Built frontend UI** using React.js for visualizing candidate data, tracking outreach status, and managing interview pipelines.
- **Automated candidate data extraction** from public directories and ICP-aligned websites.
- **Implemented outreach workflows** to send connection or recruitment messages automatically.
- **Integrated Google Calendar API** to:
    - **Automatically create and send Google Meet links** for candidates and interviewers.
- **LLM Integration:** Used LLMs (using LLama) to:
    - Analyze interview notes or chat transcripts.
    - Generate structured interview summaries and candidate review reports.

Technologies Used: Python **,** LLaMA 3.2, Flask, Mysql, HTML, CSS, JS

6) Restaurant booking agent

Description: This restaurant booking agent uses the Google Maps API to help users find restaurants based on their input prompts and preferred area locations. It checks table availability across different time slots, ensuring users can easily

select the best option for their needs. Once a booking is made, the system automates sending booking confirmations and details directly to customers via WhatsApp messages. This seamless communication streamlines the reservation process, making it efficient for both the booking agent and the customer.

Responsibilities:

- Developed backend services to integrate Google Maps API for restaurant search and availability checks.
- Designed and implemented booking workflows that manage table reservations and time slot validations.
- Automated WhatsApp message notifications to customers with booking confirmations and details.
- Built user interfaces to facilitate prompt-based restaurant searches and real-time booking status.
- Ensured smooth data flow between frontend, backend, and third-party APIs for a seamless user experience.

Technologies Used: Python **,** Flask API, Mysql, Gemini flask 2.0, React js for UI