**Meerasa Shaik**

Jacksonville, FL | +1(302)-568-9588 | [meerasa099@gmail.com](mailto:meerasa099@gmail.com) | [www.linkedin.com/in/meeru088](http://www.linkedin.com/in/meeru088)

**PROFESSIONAL SUMMARY**

Results-driven Python Developer with 5 years of experience building secure, high-performance backend applications using Django, Flask, and Fast API. Adept at designing RESTful APIs, integrating third-party services, and leveraging microservices architecture for maintainability and performance. Experienced in event-driven systems using Kafka, RabbitMQ, and AWS messaging services to support real-time, decoupled communication. Proficient in cloud-native development on AWS, including Lambda, DynamoDB, Cognito, and Fargate, with a focus on serverless design and automation. Demonstrated expertise in containerization with Docker and orchestration with Kubernetes, supported by robust CI/CD pipelines.

I recently built a **Personal AI Assistant on my local laptop**, capable of reading unread Gmail emails securely, summarizing them, answering deep technical and general questions, processing multiple document types for instant Q&A, and fetching real-time weather reports — showcasing expertise in Generative AI, Lang Chain, Streamlit, and OpenAI APIs. Reduced infrastructure costs by 30% through serverless migration, and decreased deployment time by 50% via CI/CD automation. Solid background in working with relational and NoSQL databases, including PostgreSQL, MySQL, MongoDB, and Redshift.

**SKILLS**

* **Programming Languages:** Python (Advanced) and Java
* **Python Frameworks:** Django, Flask, FastAPI, Django REST Framework.
* **API Development:** RESTful APIs, OAuth2, JWT, Asynchronous Programming
* **Data Analysis and Visualization**: NumPy, Pandas, Matplotlib, Seaborn, Plotly
* **Machine Learning:** Scikit-learn, XGBoost, LightGBM
* **Deep Learning:** PyTorch, TensorFlow, Keras
* **Large Language Models (LLMs):** GPT, LLaMA, Claude
* **Generative AI:** RAG, LangChain, LLM Fine-Tuning, Prompt Engineering, Agents
* **Natural Language Processing (NLP):** Transformers, BERT, Sentence Embeddings, Semantic Search
* **Vector Databases:** FAISS, Pinecone, ChromaDB
* **AI & Automation:** OpenAI GPT APIs, Streamlit, Gmail API, PyPDF2, python-docx
* **Front End Technologies:** HTML, CSS, JavaScript, React.js
* **Cloud & DevOps:** AWS, Azure, Docker, Kubernetes and GIT
* **Messaging:** Kafka, RabbitMQ
* **Monitoring & Logging Tools**: Grafana, Prometheus
* **Databases**: SQL, MySQL, PostgreSQL, Oracle, Amazon RDS
* **Issue & Project Tracking Tools**: Jira, ServiceNow, and Confluence
* **Python IDEs & Editors:** PyCharm, PyScripter, Spyder, PyDev, Sublime Text, VS Code

**PROFESSIONAL EXPERIENCE**

**TD Bank June 2024 - Present**

**Python Developer** Mount Laurel**,** NJ

* Built and maintained scalable RESTful APIs using Django REST Framework and FastAPI, enabling seamless integration between front-end apps and backend microservices.
* Developed custom FastAPI middleware for centralized logging, exception handling, and correlation ID tracking, integrated with AWS CloudWatch for observability.
* Designed and optimized PostgreSQL/MySQL schemas with efficient indexing and high-performance queries to support large-scale transaction data.
* Integrated Redis caching and Kafka event streaming to improve fraud detection pipelines, boosting performance and scalability of real-time processing.
* Built serverless components with AWS Lambda and API Gateway, reducing infrastructure costs and improving response scalability.
* Automated builds, testing, and deployments using Jenkins, GitHub Actions, and Docker, streamlining CI/CD workflows.
* Deployed containerized Python services on Kubernetes (EKS) with Helm charts, ensuring high availability and real-time monitoring via Prometheus & Grafana.
* Actively participated in Agile development processes including daily stand-ups, sprint planning, and retrospectives while using Jira and Confluence for task tracking and documentation.

**Tools & Technologies**: Python, Django, FastAPI, PostgreSQL, MySQL, Redis, Kafka, AWS (Lambda, CloudWatch, API Gateway, EKS), Docker, Kubernetes, Jenkins, GitHub, Prometheus, Grafana

**HCL Technologies March 2021 – Dec 2022**

**Python Developer** Pune India

* Developed RESTful APIs using Flask to handle user authentication, order management, and payment workflows, ensuring secure and seamless transactions.
* Integrated Flask-SQLAlchemy for ORM, optimized queries, and designed normalized database schemas to improve system performance and scalability.
* Built JWT-based authentication and role-based access control for secure access across internal dashboards and mobile apps.
* Containerized the application using Docker and deployed on AWS EC2 & ECS, ensuring high availability and scalable performance.
* Implemented CI/CD pipelines with GitHub Actions for automated testing, linting, and deployments.
* Used Redis caching & Celery task queues to improve API response times and support asynchronous background jobs.

**Tools & Technologies**: Python, Flask, Flask-SQLAlchemy, PostgreSQL/MySQL, JWT, Redis, Celery, Docker, AWS (EC2, ECS, RDS), GitHub Actions, REST APIs

**Briq Feb 2020 – Mar 2021 Junior Python Developer** Hyderabad, India

* Designed and implemented a RESTful API using Django and Django REST Framework to manage product inventory, order processing, and supplier integration across multiple retail location.
* Designed and implemented a RESTful API for product inventory, order processing, and supplier integration across retail locations.
* Built backend logic for stock alerts, order tracking, and automated restocking using Python scripting.
* Created CRUD endpoints, input validation, and pagination for internal staff tools and mobile applications.
* Developed relational database schemas, implemented stored procedures, and wrote optimized SQL queries for consistency.
* Used Git/GitHub with Git Flow to collaborate with other developers and manage version control.

**Briq Sep 2019 – Feb 2020 Junior Python Developer** **(Internship)** Hyderabad, India

* Worked on building and testing backend APIs using Django and Django REST Framework.
* Assisted in designing database schemas and writing SQL queries for data validation and reporting.
* Supported data ingestion automation from Excel/CSV files using Pandas for vendor product data.

**PROJECTS**

**Multimodal Retrieval-Augmented Generation (RAG) System**

* Built an AI system that answers questions directly from documents and images instead of manually searching through them.
* Developed a content extraction pipeline to read text from PDFs, Word docs, HTML, and scanned images using OCR.
* Preprocessed text with NLTK (cleaning, stopword removal, lemmatization, tokenization) to improve text quality before embeddings.
* Converted text into dense embeddings using Hugging Face sentence-transformer models for semantic understanding.
* Stored embeddings in FAISS (vector database) to enable fast and accurate similarity search.
* Implemented document retrieval with LangChain Retriever to fetch the most relevant chunks for a given query.
* Passed retrieved results to LLMs (OpenAI) to generate context-aware answers with citations.
* Delivered a Google Colab notebook that is fully self-contained, with step-by-step explanations, code, and outputs for each stage.
* Designed the system to be scalable and modular, supporting both text and image data for real-world enterprise use cases.

**Technologies Used:** Python, Hugging Face Transformers, FAISS, ChromaDB, LangChain, OCR (Tesseract/EasyOCR), NLTK, OpenAI, PyPDF2, python-docx, BeautifulSoup, Google Colab

**Personal AI Assistant | Python, OpenAI GPT, Lang Chain, Streamlit, Gmail API**

* Built a local AI assistant to securely log into Gmail, read unread emails from real people, and summarize them, ignoring automated notifications.
* Integrated Generative AI to answer coding, technical, and general questions with context-rich outputs.
* Enabled document QA by processing multiple PDFs and DOCX files for instant question answering using retrieval-augmented generation.
* Designed an interactive Streamlit UI to support multi-functional AI workflows, including real-time weather queries.

**Technologies Used:** Python, OpenAI GPT APIs, Lang Chain, Streamlit, PyPDF2, python-docx, Gmail API.

**Upload Multiple PDFs Chatbot | Python, OpenAI GPT, Lang Chain, FAISS, Streamlit**

* Developed an AI-powered chatbot that processes multiple PDF documents simultaneously, enabling users to ask natural language questions and receive accurate, context-aware answers instantly.
* Implemented retrieval-augmented generation (RAG) by chunking documents, creating vector embeddings with Lang Chain, and performing similarity search using FAISS.
* Designed a user-friendly Streamlit interface for seamless PDF uploads, chat-based Q&A, and real-time answer rendering.
* Enhanced productivity for document-heavy workflows by transforming static PDFs into interactive knowledge bases accessible via chat.

**Technologies Used:** Python, OpenAI GPT APIs, Lang Chain, FAISS, PyPDF2, Streamlit.

**CERTIFICATION**

* Introduction to Generative AI(Google)
* Google AI Essentials (Google)
* Academy Accreditation: Generative AI Fundamentals (Databricks)
* Generative AI Masterminds INTL (Out skill)
* AWS Certified Cloud Practitioner CLF-C02

**EDUCATION**

**Wilmington University Apr 2024**

Master of Science in Information Technology Wilmington, DE