**Summary**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

* 7 months of experience in AI-powered applications.
* Python & Generative AI Developer with 1 year and 5 months of experience as a Python Full-Stack Developer.
* Specializes in LLMs, chatbot development, and AI agents.
* Proficient in Python, Django, and FastAPI.
* Experienced in AI frameworks like LangChain and Langgraph.
* Passionate about LLM fine-tuning, quantization, and GPU-based optimizations.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Education Summary:**

* BSC in Computer Science  
  *Rayalaseema University, Andhra Pradesh*  
  2018 - 2021

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Experience Summary:**

| **Company Name** | **Tenure** | **Designation** |
| --- | --- | --- |
| DTskill | 2024-Sep-Till Date | Senior Python & Generative AI Developer |
| Techfnatic | 2023 - May - 2024 - Sep | Python Full-Stack Developer |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Technical Proficiency:**

| Programming Language | Python, JavaScript, SQL |
| --- | --- |
| Frameworks & Libraries | Django,FastAPI, React Js, Node Js |
| Web Services Technologies | Websockets, JSON, RESTful Web Services |
| Technical Tools & IDEs | Microsoft Visual Studio Code, Postman, Pycharm, Pgadmin, Mysql workbench |
| Database & Cloud | MySQL, MongoDB, PostgreSQL & Docker, Redis |
| Code Management tools | GIT & GitHub. |
| Project Management tools | JIRA |
| Operating Systems | Ubuntu (Linux), Windows. |
| Machine Learning | Pandas, Numpy, Scikit-Learn, Feature Engineering |
| Deep Learning | ANN, CNN, Simple RNN, KERAS |
| Data Analysis | Nltk, Spacy, Matplotlib, Seaborn |
| Generative AI | LLMs (Large Language Models), Retrieval-Augmented Generation (RAG)  Langchain, Langraph, Langsmith, OpenAI, Groq, Hugging Face, Llama Models, Mistral Models |
| Vector DBs | Pinecone, Zilliz, Qudrant, Weaviate |

**Projects Summary:**

| **#1 GenE** | **AI Agents** |
| --- | --- |
| My Role | AI Agent Backend Developer, Developing, Bug Fixing, Testing (Development Testing or Unit Testing) |
| Technologies | Django, LLMs, Langchian, Langraph, PostgreSql, Redis, Vector DBs |
| Description | Our AI Agents feature allows users to create intelligent, customizable personas with predefined attributes like name, role, and goals. These agents use LLMs with RAG for dynamic, context-aware responses and support real-time data fetching. Designed for both personal and enterprise use, they integrate with chat systems, enable multi-agent collaboration, and offer authentication and role-based access. Future enhancements include voice integration and adaptive learning. |
| My Contributions | * Designed and developed the AI agent backend using Django and LangChain and Langgraph. * Integrated LLMs with RAG for dynamic, context-aware responses. * Implemented multi-agent collaboration and real-time data fetching. * Developed authentication and role-based access control (RBAC). * Optimized Redis caching for faster response times. * Conducted unit testing and debugging to ensure system reliability. |

| **#2 GenE** | **Persona Builder** |
| --- | --- |
| My Role | Developing, Bug Fixing, Testing (Development Testing or Unit Testing) |
| Technologies | Django, LLMs, Langchian, PostgreSql, Redis |
| Project Description | The Persona Builder allows users to create and customize AI personas using predefined prompt templates. It enables fine-tuned control over AI behavior, ensuring responses align with specific roles, goals, and contexts. Users can modify attributes, experiment with different personas, and integrate them seamlessly into chat systems for enhanced interaction. |
| My Contributions | * Developed the core backend for persona creation and customization. * Integrated predefined prompt templates for controlled AI behavior. * Implemented database models and APIs using Django and PostgreSQL. * Ensured seamless integration with chat systems. * Performed bug fixes and unit testing to improve system stability. |

| **#3 GenE** | **Authentications & Authorizations** |
| --- | --- |
| My Role | Developing, Bug Fixing, Testing (Development Testing or Unit Testing) |
| Technologies | Django, PostgreSql, Redis |
| Project Description | The system supports both JWT-based authentication and session-based authentication, where a session key is securely stored within the JWT token. For authorization, role-based access control (RBAC) is implemented, allowing different levels of permissions for users, such as Super Admin and regular users, ensuring secure and controlled access to system features. |
| My Contributions | * Implemented JWT-based and session-based authentication systems. * Developed secure session key storage within JWT tokens. * Designed and enforced RBAC (Role-Based Access Control) for different user levels. * Optimized authentication and authorization workflows for security and efficiency. * Conducted debugging and testing to ensure secure access control. |

| **#4 Cosmology AI Assistant** | **Cosmology AI Assistant (Proxion)** |
| --- | --- |
| My Role | Developing, Bug Fixing, Testing (Development Testing or Unit Testing) |
| Technologies | Django, LLMs, Langchian, Langraph, PostgreSql |
| Project Description | An AI-powered assistant specialized in cosmology, designed to provide accurate and engaging responses across different modes—Kids Mode, Casual Mode, Scientific Mode, and Story Mode. It features automatic note-taking for each conversation and upcoming enhancements like Voice Mode and Vector DB integration for improved knowledge retrieval. |
| My Contributions | * Developed AI assistant logic using LLMs, LangChain, and LangGraph. * Implemented different conversation modes (Kids, Casual, Scientific, Story). * Designed automatic note-taking functionality for user interactions. * Planned and optimized Vector DB integration for knowledge retrieval. * Conducted unit testing and debugging to refine AI responses. |

| **#5 Machine Learning** | **House Price Prediction** |
| --- | --- |
| My Role | Cleaning Data, Training Model, Bug Fixing, Testing (Development Testing or Unit Testing) |
| Technologies | Machine Learning, Matplotlib |
| Project Description | Developed a predictive model using regression techniques to estimate house prices based on features like location, size, number of rooms, and market trends. |
| My Contributions | * Collected and cleaned real estate data for training. * Built a regression model for predicting house prices. * Visualized market trends using Matplotlib. * Performed model evaluation and debugging for accuracy improvement. |

| **#6 Machine Learning** | **Movie Recommendation System** |
| --- | --- |
| My Role | Cleaning Data, Vectorising Data, Bug Fixing, Testing (Development Testing or Unit Testing) |
| Technologies | Machine Learning, Matplotlib |
| Project Description | Implemented a recommendation engine using collaborative filtering and content-based filtering to suggest movies based on user preferences and viewing history. |
| My Contributions | * Processed and vectorized movie dataset. * Implemented content-based filtering techniques. * Developed an algorithm for personalized recommendations. * Debugged and optimized recommendation accuracy. |

| **#7 Machine Learning** | **WhatsApp Chat Data Analysis** |
| --- | --- |
| My Role | Analysis |
| Technologies | Pandas, Streamlit, Machine Learning, Matplotlib |
| Project Description | Built an analytics tool to process WhatsApp chat logs and generate insights such as active users, message frequency trends, most-used words, and sentiment analysis. |
| My Contributions | * Extracted and cleaned WhatsApp chat data using Pandas. * Built an analytics dashboard with Streamlit. * Implemented message frequency trends and user activity analysis. * Integrated sentiment analysis for message classification. |

| **#7 Deep Learning** | **Predicting Temperature on a Specific Date** |
| --- | --- |
| My Role | Cleaning Data, Training Model, Bug Fixing, Testing (Development Testing or Unit Testing) |
| Technologies | Deep Learning, Matplotlib |
| Project Description | Developed a time series forecasting model using deep learning techniques like LSTMs to predict temperature trends for a given date based on historical weather data. |
| My Contributions | * Collected and cleaned historical weather data. * Built and trained an LSTM model for temperature forecasting. * Optimized the model for better prediction accuracy. * Debugged and tested the deep learning pipeline. |

**Interests & Exploration:**

* Fine-tuning LLM models
* Quantization of LLM models
* Running LLM models on GPU and multi-GPU setups
* Techniques like Flash Attention, Offloading
* DeepSpeed, vLLM, TGI
* Voice AI tools like DeepFilterNet, NVIDIA Riva, OpenVINO