# Professional Summary

* Proficient Python Developer: Over 4 years of hands-on experience in developing robust and scalable applications using Python, with a deep understanding of language features, libraries, and best practices. Expertise in backend development, capable of building end-to-end solutions using Python frameworks like Flask and Fast APIs.
* Expertise in LLM (Machine Learning and Artificial Intelligence): Extensive background in leveraging Python for implementing machine learning algorithms and AI solutions to solve complex business problems and enhance decision-making processes.
* Strong SQL Skills: Demonstrated proficiency in working SQL databases, writing complex queries, and managing database systems effectively to ensure data integrity and performance.
* Cloud Services Proficiency: Solid understanding and practical experience with cloud platforms such as AWS, Azure. Skilled in deploying, managing, and optimizing cloud-based applications and services to achieve scalability, reliability, and cost-efficiency.
* Collaborative Team Player: Proven track record of collaborating effectively with cross-functional teams, including data scientists, engineers, and business stakeholders, to deliver high-quality solutions that meet both technical requirements and business objectives.

# Technical Skills:

* Areas of expertise : APIs, Rest APIs
* Programming Languages : Python, SQL
* Libraries : Pandas, NumPy, matplotlib, Flask, Fastapi, langchain,

openAI, hugging face

* Database : My SQL, PostgreSQL
* Cloud platforms : Azure, AWS
* IDE’s : Anaconda3, Jupyter Notebook, VSCode, Pycharm,

Google colab

# Experience Summary

# Project: #1

**Role:** Backend Developer

**Industry:** Manufacturing

**Project Description:** Developed a robust and efficient RAG Chatbot application for enhancing customer service using Azure AI Search and OpenAI technologies. This project aimed to streamline customer interactions, automate responses to common queries, and provide personalized assistance.

**Environment:** Python3.10, Azure, Pandas, GenAI, Enterprise search.

**Responsibilities:**

* Integrated data sources from Zendesk, Riversand, and FTP server to gather customer service-related information, product data, and additional resources for the chatbot.
* Utilized LLM embedding models to transform the data from disparate sources into Azure AI search format suitable for vector search.
* Implemented Azure AI Search for natural language understanding capabilities, enabling the chatbot to interpret user queries and extract relevant information.
* Leveraged Azure AI Search's indexing and querying functionalities to efficiently retrieve relevant responses from the integrated data sources.
* Incorporated GPT-4 model to generate appropriate responses to user queries based on the interpreted intent and context.
* Incorporated dynamic content generation techniques to provide up-to-date information sourced from Zendesk, Riversand, and FTP server.
* Deployed the chatbot application on Azure cloud function apps and web apps for scalability and reliability.
* Implemented monitoring and logging mechanisms to track user interactions, system performance, and errors for continuous improvement.

# Project: #2

**Role:** Python backend developer

**Industry:** Legal

**Description:** An Enterprise Application that uses Artificial Intelligence to expedite and automate the process of documents’ review through content analysis based on user input(s).

**Environment:** Python, Azure, Git, Docker, Flask, GenAI

**Responsibilities:**

* Developed and deployed a document review tool utilizing OpenAI's embedding model and GPT-4 for enhanced efficiency.
* Designed and executed a data preprocessing pipeline to refine, tokenize, and encode input documents for model training and inference.
* Employed OpenAI's APIs alongside tailored prompts to generate a QA model utilizing "text-embedding-ada".
* Conducted fine-tuning of the embedding model and GPT-4 using proprietary datasets to bolster accuracy and efficacy on domain-specific documents.
* Created an intuitive web interface allowing users to upload documents for review, with highlighted sections and pertinent answers provided by the model.
* Seamlessly integrated the tool with a backend database for streamlined storage and management of user documents and review records.
* Deployed the application on Azure, configuring the environment and deploying Docker containers to ensure scalability, fault tolerance, and simplified administration.

# Project: #3

**Role:** Backend Developer

**Industry:** HR

**Project Description:** This project is aimed to develop a resume scoring model for scoring the resumes based on the required job description for shortlisting the eligible candidates. Due to a very high volume of applications, it is a very tedious process for HR to shortlist the profiles manually. NLP techniques are used in determining the candidate's eligibility based on the required skillset.

**Environment:** Python3.10, NumPy, Word2Vec, Spacy, Flask, Docker, Azure, Pandas

**Responsibilities:**

* Spearheaded entire backend development lifecycle, including containerization, deployment, and application management.
* Implemented code pipelines, data pipelines, web applications, triggers, and alerts utilizing a toolkit comprising Docker, Python, Azure CI/CD pipelines, blob storage, WebApps, Azure monitoring alerts, Azure blob storage, and table services.
* Employed Python scripting to encrypt data and passwords at both rest and transit levels.
* Engineered a profiling algorithm to rank profiles from best to minimum fit.
* Led end-to-end backend development using Flask framework for web applications.
* Created RESTful APIs with Flask to facilitate seamless communication between frontend and backend systems.
* Leveraged Docker for containerizing applications, ensuring smooth deployment and scalability.

**Project: #4**

**Role:** Python Developer

**Industry:** Private Sector

**Packages:** Python3.9, PostgreSQL, GitHub, Azure

**Description** This Project aims to Predict the results based on trained or pre-trained models on invoices, PO, contracts, etc.

**Responsibilities:**

* Implemented a flow to extract data from various business documents like invoices, PoS, etc., received in a shared mailbox.
* Utilized Azure logic apps to create a flow for separating attachments from emails, employing dedicated extraction logic to retrieve required fields.
* Integrated Microsoft Azure's form recognizer services, leveraging prebuilt models to extract data from diverse file formats including .xlsx, .png, .jpeg, and .pdf. Custom code was developed to ensure effective data extraction.
* Employed blob storage efficiently for storing both historical and latest results.
* Enhanced data security by utilizing PostgreSQL features such as symmetric encryption and decryption. This was complemented with Python logic to handle various security scenarios effectively.
* Executed basic table operations including dropping and creating tables, columns, and updating their features such as length and datatype.
* Adapted to changes resulting from the deprecation of certain form recognizer versions, updating data extraction functions accordingly to ensure compatibility with the new output format.
* Developed helper functions and files to streamline tasks such as data extraction, generating responses, managing datetime functions, and more.

# Project: #5

**Role:** Python developer

**Industry:** Private Sector

**Packages:** Python3.9, PostgreSQL, GitHub, AWS services, Flask Restx

**Description:** The aim of the project is to predict and extract data from the invoice pdfs and other form of invoices. The requirement was to build logic for multiple features like block user on exceeding failed attempts and restricting uploading double extension or malicious files etc,.

**Responsibilities:**

* Worked on end-to-end implementation for a tool aimed at extracting various fields from diverse file formats.
* Integrated Microsoft Azure's form recognizer services, utilizing prebuilt models to extract data from formats like .png, .jpeg, and .pdf.
* Navigated changes in output data post-deprecation of certain form recognizer versions, updating extraction functions for compatibility with new output formats.
* Developed helper functions and files to streamline tasks such as data extraction, response generation, and datetime management.
* Employed JWT tokens for user authentication and session management during user login.
* Implemented an Audit Log API and corresponding database table to record and display user activity, bolstering security measures.
* Incorporated measures to disable users after repeated failed login attempts, using Redis for temporary storage to monitor and block such users.
* Enforced restrictions on unwanted file extensions based on content, utilizing Azure libraries like ContentSettings for setting appropriate content types.
* Conducted research and development on base64 data format to accurately extract file types such as PDFs, JPEGs, PNGs, and Word documents.
* Optimized the codebase by introducing new functions and consolidating constant values into a single file, enhancing maintainability and efficiency.