Shahmeer Hussain

Python Engineer

shahmeerhusain20@gmail.com
+92 322 5457183 Lahore, Pakistan LinkedIn
github.com/ShahmeerHussainK

Education

University of Central Punjab, *BSCS* 08/2012 – 08/2016
I have graduated from UCP with a CGPA of 3.87 Lahore, Pakistan

Skills

Python | Django | FastAPI | Django REST framework | SQL | RESTful APIs | Microservices | Stripe Integration | ORM | JWT | Swagger | Azure Data Factory | Azure Active Directory | Azure Translator | Azure Services | Azure Content Moderator | Cosmos DB

Professional Experience

Python Team Lead, *Argon Teq* 07/2021 – present

- Led the development and deployment of advanced AI models for various applications, ensuring high performance and scalability.
- Oversaw a team of data scientists and machine learning engineers, driving innovation and efficiency in AI projects.
- Implemented best practices for model training, validation, and deployment, ensuring robust and reliable AI solutions.

Software Engineer, *Devsinc* 05/2019– 06/2021
Lahore, Pakistan
I have worked almost 2 years at Devsinc as a Software Engineer and I have worked on different python projects including Different frame-works. I have also extensively worked on AI projects which includes training models, fine tuning them, data visualization etc

1Associate Software Engineer, *Systems Limited* 04/2019– 02/2017
Lahore, Pakistan
As an Associate Software Engineer at Systems Limited, I worked as a Python developer, focusing on building and optimizing software solutions. I was involved in designing, developing, and maintaining scalable applications, ensuring code quality, and implementing best practices for efficient performance. My role also included collaborating with cross-functional teams to integrate new features and enhance existing functionalities.

Projects

prospectx

- **Developed** comprehensive real estate management system using Python, providing users with tools to schedule and manage meetings with agents.
- **Integrated Twilio API** for real-time communication, enabling users to seamlessly schedule and conduct virtual meetings with agents.

- **Built a website creation feature** allowing users to design and publish real estate websites directly from the platform, utilizing Namecheap and GoDaddy APIs for domain registration and management.
- **zEnabled campaign management** for users to create, launch, and market real estate listings, driving engagement through targeted campaigns.
- **Implemented** various user actions, including lead generation, property management, and marketing tools, enhancing the overall functionality and usability of the platform.

LAZAZA

- Utilized FastAPI framework to build a high-performance backend for the application, ensuring scalability and efficiency.
- Implemented Azure Active Directory for secure authentication and access control, enhancing the application's security posture.
- Integrated Cosmos DB as the database backend, leveraging its globally distributed, multi-model database service for seamless data storage and retrieval.
- Integrated Stripe payment gateway to enable seamless and secure payment processing within the application, providing users with a frictionless payment experience.
- Implemented meta tagging functionality to integrate with LinkedIn and Google Ads, enabling targeted advertising campaigns and improving user engagement.
- Utilized Azure Queues for building asynchronous messaging between application components, ensuring reliable and scalable communication.
- Leveraged OpenAI to generate dynamic titles and descriptions based on images uploaded by users, enhancing content relevance and engagement.
- Leveraged Azure Key Vault for secure storage and management of sensitive information such as API keys, credentials, and cryptographic keys, ensuring data confidentiality and integrity.
- Implemented testing to ensure the reliability and stability of the application, reducing the risk of regressions.

Maths Transformation

- Leveraged Django framework to build a scalable and robust backend for the virtual classroom application, providing a stable foundation for its features and functionalities.
- Implemented user authentication and authorization mechanisms to ensure secure access to the platform, allowing only authenticated users (teachers and students) to interact with the system.
- Developed features for teachers to schedule classes, manage course materials, and conduct live online sessions, enhancing the teaching experience and enabling remote learning.
- Integrated Azure Translator service to enable automatic translation of voice communication into multiple languages, promoting inclusivity and accessibility for users from diverse linguistic backgrounds. Implemented features for assignment creation, submission, and grading, enabling teachers to assess student progress and provide timely feedback.
- Conducted thorough testing, including unit tests, mocks.

Octagon

- Utilized Django framework to build a robust backend for the job-hunting platform, providing a stable and scalable foundation for its features and functionalities.
- Implemented web scraping functionality using libraries BeautifulSoup to gather job listings from various sources.
- Developed advanced filtering options allowing users to refine job search results based on criteria such as job title, salary rates, and geographic location.
- Integrated OpenAI's language model capabilities to generate personalized cover letters dynamically, leveraging job descriptions to tailor the content to specific job listings.

- Implemented user authentication and authorization mechanisms to secure access to the platform, allowing registered users to save preferences, track application statuses, and receive personalized recommendations. Conducted thorough testing, including unit tests, integration tests, and end-to-end tests, to ensure the reliability, functionality, and security of the job-hunting platform.

Task Manger

- Developed a web-based project management application similar to Jira using Django
- Implemented user authentication and authorization functionalities for secure access control
- Designed modules for creating, assigning, and tracking tasks within projects
- Created a flexible project hierarchy and workflow system for customization
- Developed reporting and analytics features for project progress insights.

AI Image Generator

- Designed and trained a deep learning model tailored for high-quality image generation using frameworks such as TensorFlow and PyTorch.
- Utilized Generative Adversarial Networks (GANs) for realistic image creation and implemented advanced techniques for model training, including data augmentation and transfer learning.
- Developed robust APIs to handle user requests and facilitate interactions with the AI model.
- Integrated the AI model into the mobile application backend, ensuring seamless communication between the frontend and the model.
- Optimized the model for performance and efficiency to ensure quick image generation and minimal latency. Deployed the model and backend services in a cloud environment using AWS/GCP for scalability and reliability.
- Used Docker for containerization to simplify deployment and ensure consistency across different environments.
- Implemented user authentication and secure data handling within the backend system.
- Conducted extensive backend testing using PyTest to ensure stability, performance, and security of the application.
- Monitored and maintained backend services to ensure high availability and quick response times.

VidYou

- Developed a backend system for AI-based video generation, transforming user prompts or stories into short videos.
- Designed and implemented a deep learning pipeline using advanced AI frameworks such as TensorFlow and PyTorch for video generation.
- Utilized a combination of Recurrent Neural Networks (RNNs) for handling sequential data and Convolutional Neural Networks (CNNs) for frame generation.
- Employed Generative Adversarial Networks (GANs) to enhance the realism and coherence of generated video frames.
- Created and exposed RESTful APIs to receive user prompts, process inputs through the AI model, and return generated video content.
- Implemented sequence-to-sequence models and attention mechanisms to effectively capture and translate narrative elements into video sequences.
- Integrated the video generation model with backend microservices to ensure efficient data flow and processing of user inputs.
- Optimized AI models for performance and efficiency using techniques such as model pruning, quantization, and hardware acceleration (e.g., GPU/TPU).
- Deployed the AI model and backend services on cloud platforms like AWS/GCP, leveraging services like AWS Lambda and Google Cloud Functions for scalability.

- Employed Docker for containerization, ensuring consistent and reproducible environments across development, testing, and production.
- Implemented user authentication and secure data handling using JWT (JSON Web Tokens) and encryption protocols.

LLM Word

- Developed an LLM (Large Language Model) for a Word plugin, enabling Q&A capabilities with ingested Word documents, primarily research papers.
- Utilized state-of-the-art transformer-based architectures such as GPT-3 or GPT-4 for natural language understanding and generation.
- Implemented a robust data ingestion pipeline to parse and preprocess Word documents, converting them into a format suitable for the LLM.
- Employed Natural Language Processing (NLP) techniques for document parsing, entity extraction, and context understanding.
- Created and exposed RESTful APIs to facilitate the interaction between the Word plugin and the backend LLM model.
- Integrated the LLM model with the Word plugin, ensuring seamless interaction and real-time response to user queries.
- Optimized the LLM for performance, utilizing techniques such as model distillation, fine-tuning, and hardware acceleration (e.g., GPU/TPU) to handle large documents efficiently.
- Deployed the LLM and backend services on cloud platforms like AWS/GCP for scalability and reliability.
- Used Docker for containerization to maintain consistent development and deployment environments.
- Implemented secure data handling and user authentication mechanisms to protect sensitive research data. Conducted comprehensive testing of the LLM and backend services using frameworks like PyTest to ensure robustness, accuracy, and performance.

YouMe, *dating app*

- **Developed** a dating application in Python, providing users with a seamless platform to connect and interact. **Implemented real-time chat functionality** using sockets, enabling smooth and instant messaging between users. **Designed and optimized user matching algorithms** , enhancing user experience by providing more accurate matches.
- **Integrated secure user authentication** and profile management features, ensuring data privacy and a userfriendly interface.
- **Collaborated with cross-functional teams** to enhance app features and improve overall performance.

AI-Powered Medical Diagnosis System

- This project involves developing an AI system capable of analyzing medical images (e.g., X-rays, MRIs, CT scans) to detect various diseases such as tumors, fractures, or pneumonia. The AI model is trained on a large dataset of medical images annotated by experts, learning to identify patterns associated with specific conditions.
- The system can be integrated with Electronic Health Records (EHR) to access patient history and provide context-aware diagnosis and treatment recommendations. It can also include a decision-support system that assists doctors in making more informed decisions.

Smart Traffic Management System

- This project focuses on creating an AI-driven traffic management system that optimizes traffic flow in real-time. The system gathers data from various sources, including traffic cameras, IoT sensors embedded in roads, GPS data from vehicles, and even social media feeds.
- The AI model predicts traffic patterns, identifies potential congestion points, and dynamically adjusts traffic signals to improve flow. It can also provide optimal route suggestions for emergency vehicles or public transportation.

-

AI-Driven E-commerce Recommendation Engine

- In this project, you'll develop a recommendation engine for an e-commerce platform that provides personalized product suggestions based on user behavior, purchase history, and real-time interactions. The AI model uses collaborative filtering, content-based filtering, or a hybrid approach to make recommendations.
- The engine can also include a real-time bidding system where advertisers bid for ad placements, optimizing ad delivery based on user preferences and engagement.

-

-