Muhammad Mohsin Rashid

0332 8450678
mohsinrashid64@gmail.com
64 D1 Nespak Society,
Lahore, 54770

SKILLS

- Python
- Machine Learning/Deep Learning
- Data Science
- LLMs
- Generative AI
- Django
- Flask
- FastAPI
- React.js
- Node.js
- JavaScript
- Web3
- Blockchain
- C
- C++
- Linux

EXPERIENCE

Oct 2022 – April 2024 **Software Projects Developer, Educative | Lahore**

- Developed projects for computer science and software engineering courses.
- Covered diverse topics such as web development, blockchain, machine learning, and data science.
- Facilitated hands-on learning experiences for users.
- Enabled users to build websites, comprehend intelligent algorithms, and analyze data.
- Conducted meetings with external clients to understand their needs and requirements
- Provided technical assistance to resolve user issues and ensure smooth operations.

April 2024 – Present **Machine Learning Engineer, DigiMark Developers | Lahore**

- Developed and implemented advanced AI models using Generative AI techniques to improve the efficiency of existing systems.
- Designed and built Retrieval-Augmented Generators (RAGs) to enhance the performance of information retrieval systems.
- Engaged in Prompt Engineering, optimizing the interaction between machine learning models and input prompts to improve model performance.
- Made APIs using both FastAPI and Flask frameworks, enhancing the accessibility and usability of machine learning models.
- Worked extensively with Vector Stores.

EDUCATION

Oct 2018 - July 2022 **Bachelors in Computer Science**, University of Central Punjab, Lahore

Oct 2016 - Jun 2018 **Advanced Level**, The Message School System, Lahore

Oct 2013 - Jun 2016 **Ordinary Level**, Beaconhouse Johar Town, Lahore

PROJECTS

Video Chat App in React
Developed a real-time video chat application using MERN stack and Jitsi (Open-source video conferencing platform with screen sharing, encryption, and flexibility for remote collaboration), providing seamless communication.

Chat App Using React and Hardhat
Developed a chat application employing React and Hardhat. Utilizing Hardhat, smart contracts were scripted and deployed onto the local Ethereum blockchain. React served as the front-end framework seamlessly integrating with the smart contract. Each chat message was treated as a transaction, securely stored within the blockchain.

Breast Cancer Detection Using Python TensorFlow
Developed a breast cancer detection system utilizing Python TensorFlow. Multiple convolutional neural networks were developed to classify breast tissue images as malignant or benign. The process commenced with employing U-Net to locate breast cells and subsequently counting them, followed by the creation of a CSV file to store pertinent information, i.e., the number of cells in each image alongside image names. Several artificial neural networks were then crafted and evaluated to ascertain the model with the highest accuracy.

Image Recognition in Python Using OpenCV
Developed a live image recognition tool using Python's OpenCV library. This application compares real-time images or videos with stored images in a database, providing straightforward similarity assessments.

Full Stack Web Application on the Arweave Blockchain
Developed a complete web app on the Arweave blockchain for storing data securely. Used React to create a simple blogging tool with basic functions performing CRUD (Create, Read, Update, Delete) operations. Integrated these functions directly into the smart contract for backend operations. Engineered the smart contract to manage interactions securely within the blockchain. Deployed both the frontend and smart contract onto Arweave, ensuring a strong, decentralized setup for a lasting and reliable web app.

Address Resolution Protocol Spoofer
Created an ARP spoofer using Python, experimenting with network security and packet manipulation. Tested the spoofer across multiple virtual machines using Oracle VirtualBox with Kali Linux. Essentially, this project involved rerouting network traffic from a victim to the attacker, showcasing insights into network security.

Binary Classification Using Convolutional Neural Network for Cats and Dogs
Built a binary classification system using Python and TensorFlow, distinguishing between images of cats and dogs. Employed Convolutional Neural Networks (CNN) for this task, achieving an impressive 90% accuracy rate. Trained and tested the model on a significant dataset of cats and dogs, demonstrating the effectiveness of the approach.

Chemical Distillation Using Self Organizing Maps
Utilized Self Organizing Maps for chemical distillation, improving efficiency in the separation process**.** This project was done using python and a library named minisom used for implementing self organizing maps. It was trained and tested on a dataset containing name of different chemical compounds along with their composition. So self organizing maps were utilized to see the similarities between different elements. The obersevations were seen via heatmap.

CV Generator Using Django
Implemented Self-Organizing Maps (SOM) using Python and the 'minisom' library to optimize the chemical separation process. Employed a dataset comprising chemical compound names and compositions for training and testing. Utilized SOM to identify similarities between different elements, visualizing observations through heatmaps. This project aimed to improve distillation efficiency through enhanced understanding of chemical composition relationships.

Cryptocurrency Visualization Dashboard Using Django
Created a dynamic visualization dashboard with Django, dedicated to monitoring and analyzing cryptocurrency data. Utilizing Django as the web application framework and integrated the CoinGecko API to retrieve real-time data for display on the frontend. The frontend showcased line and bar charts, depicting cryptocurrency price trends against various fiat currencies. Additionally, implemented a feature to calculate exchange rates among different cryptocurrencies, enhancing the dashboard's analytical capabilities.

Chat with Document
This project uses smart tools like RAGs, LLAMA, and OpenAI to make chatting with uploaded documents possible. It has a simple interface made with Streamlit for easy document upload and chat. This turns static documents into interactive ones, making it easier to find information and engage with the content. It shows how AI can change the way we interact with text data.

COURSES AND CERTIFICATIONS

1. Deep Learning in C
2. Deep Learning A-Z™ 2023: Neural Networks, AI
3. Deep Learning with TensorFlow 2
4. Intro to NLP for AI
5. Intro to ChatGPT and Generative AI
6. Intro LLMs
7. Data Science & Machine Learning
8. (Theory+Projects)A-Z 90 HOURS
9. Web Scraping and API Fundamentals
10. Linux Mastery: Master the Linux Command Line in 11.5 Hours
11. Blockchain A-Z™: Learn How To Build Your First Blockchain
12. The Complete 2023 Web Development Bootcamp
13. Python Django - The Practical Guide

OTHER RELEVANT SKILLS:

- Fluent English speaker
- Proficient in prompt engineering practices in the context of generative AI
- Skilled in analyzing various types of problems