SARAVANAKUMAR U - Full Stack Developer

Highly-motivated backend developer with a Bachelor of Engineering in Electronics and Communication Engineering.

Skilled in Python, Django, FastAPI, and Java, with a strong foundation in HTML5, CSS3, and JavaScript. Known for strong analytical skills, adaptability, and the ability to work independently.

Passionate about continuous learning and solving complex problems, with a track record of delivering projects efficiently and effectively.

Areas of Expertise

- **Backend Development:** Extensive experience in developing and maintaining backend services using Python, Django, and FastAPI. Skilled in creating RESTful APIs and implementing robust, scalable server-side logic.
- **Frontend Development:** Proficient in HTML5, CSS3, and JavaScript, and react js with a solid understanding of building responsive and interactive user interfaces.
- **Database Management:** Strong knowledge of PostgreSQL and MySQL. Experienced in designing database schemas, writing complex queries, and optimizing database performance.
- **Data Processing:** Experienced in processing and managing data using technologies like pyiceberg, Trino, Apache Iceberg, Apache Hive, and object storage solutions like S3 and MinIO.
- **Real-time Data Analysis:** Expertise in creating solutions for real-time data monitoring and analysis, incorporating logs, metrics, traces, and

Professional Qualification.

Python Full Stack Development

Academic Qualifications

Bachelor of Engineering in Electronics and Communication Engineering

Community Contributions

https://github.com/opendatahub-io-contrib/data-mesh-pattern.git

Key Projects

Observability Platform

- **Developed Observability Solutions:** Implemented robust observability solutions to monitor and analyze system performance, leveraging Python and FastAPI to build efficient data pipelines and APIs.
- **API Development with FastAPI:** Utilized FastAPI to create high-performance APIs for data collection and monitoring, ensuring rapid response times and scalability to handle high traffic.
- **Storage Management with S3 MinIO:** Designed and managed scalable storage solutions using S3 MinIO, optimizing data storage and retrieval processes to support large-scale observability data.
- **Data Lake Architecture with Apache Iceberg:** Implemented Apache Iceberg for managing complex data lake architectures, enabling efficient querying and data versioning, which enhanced data reliability and consistency.

other telemetry data to provide insights and enhance application performance.

Skill Summary

- **Programming Languages**: Python, Java, JavaScript.
- **Web Development**: HTML5, CSS3,Recatjs.
- **Frameworks & Libraries**: Spring Boot, Django, FastAPI.
- **Databases**:PostgreSQL,MySQL, SQLAlchemy.
- **Tools & Technologies**: Git, PySpark (Basics).
- **Real-time Data Analysis**: Logs, Metrics, Traces, Telemetry Data.
- **Version Control:** GitHub
- **Storage** : S3 , Minio , Apache IceBerg
- **Data Processing:**Trino, Hive, pyiceberg.
- **Observability** : Open Telemetry

- **Performance Monitoring and Alerts:** Set up monitoring dashboards and alert systems to proactively identify and resolve performance bottlenecks and system issues.
- **Collaborative Cross-functional Efforts:** Worked closely with development, operations, and data engineering teams to ensure seamless integration of observability tools and practices across the organization.
- **Continuous Improvement:** Continuously optimized observability frameworks to enhance system visibility and performance, leading to improved system reliability and reduced downtime.

Brain Tumor Detection using MATLAB

Data Collection:

- Utilized publicly available MRI datasets, such as the **BraTS (Brain Tumor Segmentation)** dataset, which provides annotated images of brain tumors.
- Performed data preprocessing, including normalization and augmentation, to ensure robust model training.

Image Preprocessing:

- **Noise Reduction:** Applied filters like Gaussian, median, or anisotropic diffusion to reduce noise in MRI images.
- **Contrast Enhancement:** Utilized histogram equalization to improve the contrast of images, making tumor features more distinguishable.
- **Segmentation:** Implemented methods such as thresholding, region growing, or edge detection to segment brain regions and isolate potential tumor areas.

Water Pipeline Leakage Detection System

Developing a Water Pipeline Leakage Detection System involves integrating sensors, data analysis, and possibly machine learning to accurately identify and locate leaks in water pipelines

Designed and implemented a Water Pipeline Leakage Detection System using advanced sensors, data processing, and machine learning techniques to identify and locate leaks in real-time, reducing water waste and maintenance costs.

- **Objective:** The main goal is to detect leaks accurately and efficiently, minimizing water loss and damage.
- **Architecture:** Designed a modular system architecture that integrates various components such as sensors, data acquisition modules, and communication systems.
- **Deployment:** Set up the system to be scalable and adaptable to different pipeline configurations, including urban and rural networks.

Communication and Alert System:

- **Data Transmission:** Implemented wireless communication protocols (e.g., MQTT, LoRa) for transmitting sensor data to a central server.
- **Alert Mechanism:** Developed a notification system to alert maintenance teams via SMS, email, or mobile app when leaks are detected.
- **User Interface:** Created a dashboard for real-time monitoring of pipeline status and historical data visualization..