Job Description:

Role: Full Stack Developer with a Focus on Data Processing
Company: Mercedes Benz R&D

Overview:
During my internship at Mercedes Benz R&D, I played a pivotal role in developing a comprehensive platform aimed at optimizing data processing for Apache Flink users. My work encompassed frontend development, backend engineering, containerization, orchestration, cloud integration, and CI/CD, with a primary focus on enhancing data processing workflows and performance.

Key Responsibilities:

Data Processing Platform Development:

- Unified Flink Platform: Developed a centralized Apache Flink platform, consolidating multiple data processing tasks into a single, scalable cluster. This approach streamlined operations and reduced complexity.
- Real-Time Processing: Enabled real-time stream processing, allowing for low-latency analysis and immediate insights from live data streams.
- Advanced Data Optimization: Leveraged Flink's advanced features, such as stateful computations and event time processing, to efficiently handle diverse data processing needs.
- Collaborative Data Access: Improved data sharing across teams within a unified network, fostering collaboration and enhancing data integration.
- Integrated Data Workflows: Facilitated the integration of various data sources and sinks, optimizing data workflows and processing efficiency.
- Fault Tolerance and Monitoring: Implemented Flink's fault tolerance capabilities and provided real-time monitoring for proactive management and system optimization.

Frontend Development:

- Vue.js Interface: Created a user-friendly interface for data input and management using Vue.js, incorporating components such as text inputs, file uploads, sliders, and AG Grid tables.

Backend Development:

- RESTful APIs: Designed and implemented RESTful APIs with Spring Boot Java for handling data processing tasks, including CRUD operations and file storage.

Containerization and Orchestration:

- Docker and Kubernetes: Containerized services with Docker and orchestrated deployments using Kubernetes, transitioning from Minikube to Azure Kubernetes Service (AKS) for scalable management.

Cloud Integration:

- Azure Services: Integrated Azure Blob Storage, Azure Container Registry (ACR), and AKS for cloud-based data management and orchestration.
- Terraform Automation: Automated infrastructure management with Terraform, provisioning and managing resources like AKS clusters and ACR instances.

CI/CD Pipelines:

- Automation: Developed CI/CD pipelines for the build, testing, and deployment of Docker images, including pipelines for node pool management and deployment automation.