Arogya Dhakal

Linkedin: https://www.linkedin.com/in/arogyad/
Github: https://github.com/arogyadhakal

Email : arogyadhakal@gmail.com
Mobile : +1-919-600-1777

Education

- **University of North Carolina at Chapel Hill** — Chapel Hill, NC
  Bachelors in Computer Science, Minors in Statistics and Neuroscience — *Aug 2020 - Dec 2023*
  - *Courses: Data Structures and Analysis, Discrete Math, Systems Fundamentals, Foundations of Programming, Models of Languages and Computations, Files and Databases, Algorithms and Analysis, Introduction to Probability, Introduction to Machine Learning, Software Engineering Lab, Data Science in the Business World*

Skills

- **Languages**: Java, C++, Python, JavaScript, C, SQL
- **Tools**: Angular, NodeJS, React, Django, Docker, Spring, GIT, Postman, JIRA, Matlab, JUNIT, Mockito, Cypress, Kafka, Snowflake, AWS, Kubernetes, Terraform, Datadog, Jenkins, EKS, FastAPI, Karma

Experience

- **Fidelity Investments** — Durham, NC
  Software Engineer — *Jan 2025 - Present*
  - Maintained Spring-based microservices for tier zero APIs, facilitating requests for positions, trade, and dividend calls for 10,000,000+ monthly users across 2,000 financial institutions using Jenkins, Terraform, Datadog, and Splunk with AWS EKS for container orchestration and proactive health monitoring allowing for a 99.9999999% uptime.
  - Created Java-based microservices to transform frontend SOAP requests into backend-compatible REST API formats, streamlining workflows across six teams and reducing trade execution time by 30%.
  - Modernized tier one legacy applications built in .NET to use Angular and Java/Spring, migrated all CI/CD pipelines to multi cluster Jenkins pipelines utilizing Terraform and Datadog for IaC and real-time analytics respectively
- **Fidelity Investments** — Durham, NC
  Associate Software Engineer — *Jan 2024 - Jan 2025*
  - Produced a microservices-driven, two-way, dynamic family investments platform using Angular, Java, Kafka, and PostgreSQL to stream real-time data, integrating with RESTful APIs and deployed on AWS's Elastic Kubernetes Service to efficiently handle 10,000+ record changes per minute
  - Created a modern, responsive, and accessible analytics dashboard enabling 2,000 internal users to concurrently track 100,000+ user interactions with a 80% reduction in latency, enabling a 60% increase in user-engagement
  - Built and maintained CI/CD pipelines for external-client facing APIs using Jenkins, Terraform, Datadog, and AWS to keep application's uptime at 99.9999999%
- **Fidelity Investments** — Durham, NC
  Software Engineering Intern — *Jun 2023 - Aug 2023*
  - Implemented a real-time data streaming pipeline between Microsoft Dataverse into OracleDB using Kafka eliminating the need for slow bulk processing pipelines and reducing latency by 60% for batch processing of records within 100,000 files daily
  - Created producer and consumer Python scripts to fetch, produce, and consume data between Kafka topics for multiple databases spread between Oracle, Dataverse, and on-premise RDBMS PostgreSQL databases
  - Deployed the pipeline onto the Cloud by creating containerized images using Docker and deploying onto Kubernetes on AWS
  - Collaborated directly with Microsoft to create integrated solutions for the PowerApps platform Suite to build out Kafka pipelines to stream real-time data to clusters

Projects

- **Creddit Hours**: Built an app that scrapes summaries of any subreddit within customizable date ranges giving moderators the current state of any subreddit and trends. Using TensorFlow Transformers, the app runs sentiment analysis to provide a 1-10 score for any specifiifed subreddit. Additionally, provides an analytical interface for moderators to track sentiment change and trends over time with graphs, helping them find specific posts, and track the most active users with the most sentiment changing posts. Built using TensorFlow, PyTorch, the Reddit API, FastAPI, React, and Firebase. Used by hundreds of reddit modertators to support Reddit Community.
- **EssayPilot.io**: Developed a full-stack essay analysis application that provides automated feedback by mapping AI-generated comments to specific essay sections. Integrated with 5+ LLM providers including OpenAI, Anthropic, and Google AI for diverse feedback perspectives. Built a responsive React frontend with real-time comment visualization and FastAPI backend with async processing, S3 for file storage, and a PostgreSQL database to handle user records. Scaled to serve 3,000+ daily active users processing 15,000+ essays monthly with sub-second response times.