

Fayaz Shaik









Developer (Java)

SUMMARY

Full Stack Developer with expertise in building scalable microservices architectures and optimizing performance through advanced data structures and algorithms. Skilled in both frontend and backend development, leveraging technologies such as React, Spring Boot, Redis, and Google Cloud Run to build high-performance applications. Experienced in designing modular microservices for seamless data handling and real-time user interaction.

SKILLS

- Programming Languages: Java
- Frontend Technologies: React, HTML5, CSS3, Redux, Material-UI
- Backend Technologies: Spring Boot, RESTful APIs, Maven
- Databases: MySQL, MongoDB
- Data Structures and Algorithms (DSA): Hash Maps, Arrays and Linked Lists, Graphs and Graph Traversal (BFS/DFS), Priority Queues and Heaps, Tries and Inverted Indexes, Caching Strategies (LRU Cache, Redis), Sorting and Pagination Algorithms, Load Balancing Algorithms
- Cloud Services: AWS (EC2, S3, RDS, SES), Google Authentication
- Libraries/Frameworks: Hibernate, JPA, Spring Data JPA/MongoDB, Spring Security, Lombok
- API Tools: Swagger, Axios, Fetch API
- Version Control: Git, GitHub
- Problem Solving: LeetCode, HackerRank

PROJECTS

MICROTWEET - SCALABLE SOCIAL PLATFORM WITH MICROSERVICES ARCHITECTURE

Technologies Used: React, Spring Boot, Redis, Google Cloud Run, Data Structures & Algorithms

Developed a microservices-based social media platform with high-performance functionality across user profiles, tweet handling, relationship management, and search capabilities. Each feature was implemented as a standalone microservice, deployed on Google Cloud Run for scalability and real-time data processing.

Project 1: User Profiles and Authentication Microservice

- · Frontend (React): Managed user authentication with Hash Maps for efficient form state handling; implemented Form Validation Algorithms (e.g., regex) for secure input validation.
- Backend (Spring Boot): Used BCrypt hashing for secure password encryption and Hash Maps to store user sessions and tokens, ensuring fast authentication and session management.

Project 2: Tweet Posting and Retrieval Microservice

- Frontend (React): Employed Arrays for dynamic tweet storage and applied Sorting Algorithms to display tweets by timestamp, prioritizing the latest tweets.
- · Backend (Spring Boot): Utilized B-Trees for efficient tweet indexing and storage, with Linked Lists maintaining tweet order in memory. Implemented Pagination Algorithms for smooth data retrieval in batches.

Project 3: Following System Microservice

- Frontend (React): Represented user relationships as Graph Data Structures (adjacency list) and optimized API calls with Debounce/Throttling Algorithms for follow/unfollow actions.
- · Backend (Spring Boot): Modeled relationships as a Graph (Adjacency List) and used Hash Maps for caching, enhancing follow suggestion efficiency. Employed Graph Traversal Algorithms (BFS/DFS) to suggest mutual followers.

Project 4: Timeline Feed Generation Microservice

- Frontend (React): Managed feed sorting using Priority Queues for timestamp prioritization and implemented Merge Sort to merge multiple tweet lists from followed users.
- Backend (Spring Boot): Utilized Priority Queues/Heaps for sorting and Merge Sort algorithms for feed unification. Applied Redis caching for optimized feed retrieval and response times.

Project 5: Hashtags and Search Functionality Microservice

- Frontend (React): Developed hashtag autocomplete with a Trie Data Structure for efficient prefix matching; utilized an Inverted Index for local search.
- · Backend (Spring Boot): Leveraged Tries for hashtag storage and Inverted Index for keyword-to-tweet mapping, optimizing backend search performance.

Project 6: Integration, Review, and Cloud Deployment

- Frontend (React): Integrated Sets for tracking active users and applied LRU Cache for frequently accessed UI components.
- Backend (Spring Boot): Used Sets for active session tracking, implemented LRU Cache with Redis for high-demand data, and configured Load Balancing Algorithms on Google Cloud Run for request distribution across services.

ADDITIONAL PROJECTS

OTT PLATFORM Project Link Github Repo Github Repo



Technologies Used

React, Redux, Material-UI, HTML, CSS, SpringBoot, Restful APIs, MySql, Axios, Swagger, Postman, Maven

Developed a dynamic OTT platform with TMDB API integration to provide up-to-date movie data. Created a responsive, userfriendly frontend using React, displaying movie information like titles, genres, release dates, and ratings. Implemented Redux for efficient state management, enhancing the app's scalability and performance.

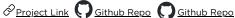
WEATHER APP



Technologies Used React, Redux, Material-UI, Weather API, HTML, CSS, SpringBoot, MongoDB, MySql, Axios.

Built a real-time weather application with a responsive frontend using React, integrating a weather API for live data (temperature, humidity, etc.) worldwide. Enabled users to search locations and instantly view current weather updates, with a focus on smooth user experience and efficient API handling.

MR ENGINEERING WORKS



Technologies Used

React, Redux, Material-UI, HTML, CSS, SpringBoot, Restful APIs, MongoDB, Axios, Postman, Maven

Developed a comprehensive welding services website with a responsive, user-friendly interface using React, designed to engage customers and showcase the team's expertise. Visitors can browse detailed service descriptions, view high-quality images of completed projects, and learn about the team's industry experience.

The contact form is integrated with a RESTful API, sending inquiries directly to a Spring Boot email service for prompt response. For added convenience, customers can quickly connect via clickable links to email, WhatsApp, or phone. This setup ensures seamless communication and a professional online presence, optimized for performance and scalability.

KEY CONTRIBUTIONS

- Optimized Performance: Enhanced application speed and efficiency by implementing data structures and algorithms, including Hash Maps, Arrays, Linked Lists, Graphs, Priority Queues, Tries, and various caching mechanisms.
- Scalable Architecture: Designed a modular microservices structure deployed on Google Cloud Run, enabling real-time data processing and efficient request management across distributed services.

CERTIFICATIONS

Program Completion Certification & Certificate XACADEMIX

Python fullStack Insternship at Kodnest 🔗 Certificate

Data Science And Machine Learning at Ybi 🔗 Certificate

web developement at internshala 🔗 Certificate

Java Certificate & Certificate HackerRank

EDUCATION

B-Tech - CSE (Computer Science and Engineering) - 2024 - 7.69

Jawaharlal Nehru Technological University Kakinada.