Mohamed Zainudeen

Software Engineer

Proactive software developer with 1.5 years of hands-on experience in Java. Quick to adapt to new technologies and eager to contribute to impactful and high-quality projects.

vazain786@gmail.com | 8056077976 | Chennai, India | leetcode.com/u/mohamed__zainudeen/

EDUCATION

Bachelor of Engineering
Velammal Institute Of Technology
07/2019 - 07/2023 *Chennai, TN*

Courses

- Coursework in Java ,C, C++, Console Based Applications, FrontEnd Applications

12th - State Board
Nazareth Matriculation Higher Secondary School
06/2018 - 06/2019 *Chennai, TN*

10th - State Board
Nazareth Matriculation Secondary School
06/2015 - 06/2017 *Chennai, TN*

Courses

- Coursework in Maths, Physics, Chemistry

Java Full Stack Development (CITIFY25)
SHOSHIN - LTIMindtree
10/2024 - 12/2024

Github (CITIFY25)
SHOSHIN - LTIMindtree

HOBBIES & INTERESTS

LEETCODE
Solved 200+ problems on LeetCode with strong grasp of DSA concepts including Graphs, Trees, Linked Lists, and Binary Search.

GITHUB - (DSA)
Published a comprehensive collection of solved LeetCode problems on GitHub, demonstrating proficiency in core DSA topics such as Sliding Window, Kadane's Algorithm, Permutations, Subsets, Merge Sort, and advanced String and Array manipulations.

Product-Based - (LLD)
Hands-on experience designing and implementing low-level systems like Library Management, Movie/Flight/Taxi Booking, Railway Reservation, Task Management, Sudoku, and Tic Tac Toe.

CERTIFICATION

Oracle Java SE 11 (10/2024 - Present)

SKILLS

Java | SpringBoot | SQL | MicroServices | 4G/5G Knowledge on packet flow | Maven | C# | Data Structures and Algorithms | Hibernate | HTML/BootStrap/CSS/CSS3

PROFESSIONAL PROJECTS (LTIMINDTREE)

Real-Time Margin Monitoring (RTMM), Wealth Management
JAVA SQL DSA (CITI)

- **Optimized Backend Solutions -** Designed and enhanced backend systems to support real-time margin monitoring in wealth management platforms. Enabled seamless and efficient tracking of diverse investment assets such as for eg: stocks, gold, and other financial instruments, ensuring high performance and reliability under dynamic market conditions.
- **Efficiency Improvements -** Implemented performance enhancements across backend processes to ensure faster data ingestion, reduced latency, and improved system responsiveness. Enabled accurate and timely profit/loss monitoring, supporting better decision-making in dynamic financial environments.
- **Code Optimization -** Refactored and streamlined backend code to improve execution efficiency, reduce processing delays, and enhance overall system performance. Contributed to smoother and faster financial transactions by eliminating bottlenecks and optimizing resource utilization.
- **Instant Front-End Updates** - Engineered backend functionalities to enable immediate synchronization with the front-end, providing real-time visibility into market movements. Ensured customers and bankers received up-to-date insights for timely and informed decision-making.

HEALTH CARE MANANGEMENT - Java SpringBoot Angular

- **Designed and developed a Healthcare Appointment Management System** using Java, Spring Boot, and Angular to streamline the management of patient appointments, doctor availability, and medical records.
- **RESTful APIs** for patient registration, doctor availability updates, appointment scheduling, and medical record retrieval, ensuring seamless communication between the frontend and backend.
- **Pagination and filtering** for fetching appointments and medical records to handle large datasets efficiently and improve performance.

BANK SAFE APPLICATION - Java SpringBoot Angular DSA

- **Designed and developed a Progressive Banking System** using Java, Spring Boot, and RESTful APIs to manage customers, accounts, transactions, and credit cards, ensuring modularity and scalability for financial operations.

@Qualifier("accountServiceImplJpa"))

- **Flexible Data Access** : Spring @Qualifier decoupled services, enabling JPA/JDBC/in-memory switching for environment adaptability and future ORM/database integration.

getAllCustomersSortedByNameFromArrayList()

- **Dynamic In-Memory Sorting/Storage** Built advanced in-memory operations for instant data manipulation, accelerating iterative testing and UI prototyping without database overhead.

Performance Focus:

- **In-memory sorting/storage** enables sub-millisecond operations for high-speed user testing, a rare optimization in typical CRUD systems.