

Unwan Khan

New Delhi, India | unwanazmi@gmail.com | +916387711532 | [Leetcode](#) | [LinkedIn](#) |

[Naukri](#) | [GitHub](#)

Professional Summary

Entry-level Software Developer skilled in Java, Spring Boot, and JavaScript (ES6+), with hands-on experience in building responsive web apps, RESTful APIs, and scalable backend systems. Strong in OOP, data structures & algorithms, and concurrency control, with practical exposure to JSON persistence (Jackson), Gradle, JUnit, and Git.

Education

University of Lucknow, Master Of Computer Application

- GPA: 8.16/10.0 [View Academic Transcript](#)

Certifications & Achievements

- Coding & Programming**, Samsung Innovation Campus — Certificate of Completion, 2022
- Introduction to Java**, Coding Ninjas — Certificate of Excellence & Certificate of Completion, 2023 [View](#)
- Data Structures in Java**, Coding Ninjas — Certificate of Excellence & Certificate of Completion, 2023 [View](#)
- Consistently solved problems on **LeetCode** for 3+ years; solved **1000** problems with a **global rank in top 0.2%** (24K / 12.2M users) and maintained an **870+ day ongoing streak (POTD)**

Projects

IRCTC Ticket Booking — Java Ticketing Simulator

[View on GitHub](#)

- Built a Java-based IRCTC ticket booking simulator implementing signup/login, train search, PNR generation/validation, end-to-end booking and cancellation flows.
- Designed OOP domain model and modular services (TrainService, UserBookingService); implemented secure authentication with BCrypt password hashing, and JSON persistence with Jackson + JavaTimeModule using atomic NIO file I/O to prevent data corruption.
- Developed pluggable seat-allocation strategy (first-fit; extensible to group-best-fit & waitlist), added unit-test scaffolding and automated build with Gradle.

Tech: Java · Jackson · Java Time API · BCrypt · NIO File I/O · Gradle · JUnit · Git

GroupChat — Multi-threaded Java TCP Chat Application

[View on GitHub](#)

- Built a multi-threaded Java TCP chat application (ChatServer + console ChatClient) supporting broadcast and private messaging with a thread-per-connection model.
- Implemented core service modules — MessageRouter, ClientHandler and UserRegistry — plus a command processor (user list, private message commands) to enable robust message routing and user management.
- Used Java I/O and socket programming with careful concurrency handling and clear modular structure to simplify maintenance and testing.

Tech: Java · TCP sockets · Multithreading · Java I/O · Collections · Git

Technologies

Programming Languages: Java · JavaScript (ES6+) · C · SQL

Web Technologies: Spring Boot · HTML5 · CSS3 (Flexbox, Grid) · JSON · Axios

Databases & Connectivity: MySQL · JDBC · MongoDB

Tools: Git/GitHub · Postman · Chrome DevTools · JMeter · Jira

Architectural Skills: System Design (Scalability, High-level UML, Caching, Load Balancing)