

MUHAMMAD TAHIR

mtahirdars32@gmail.com · linkedin.com/in/muhammادتahir32 · github.com/Tahir-Dars

PROFESSIONAL SUMMARY

Backend-focused Software Engineering student at MUET with 4 production-grade Java projects demonstrating expertise in Spring Boot, REST API design, Spring Security, and Spring Data JPA. Proven ability to architect layered systems with strict separation of concerns and deliver full CRUD lifecycles with proper HTTP semantics. Actively seeking a Software Engineering role to contribute backend development skills to a product team.

TECHNICAL SKILLS

Languages: Java (Advanced) · C++ (Intermediate) · SQL (Advanced) · HTML5 · CSS3 · JavaScript
Frameworks: Spring Boot · Spring MVC · Spring Data JPA · Spring Security · Spring Actuator · Servlets · JSP
Databases: MySQL · PostgreSQL · JDBC (PreparedStatement, Transaction Mgmt) · H2 Embedded
Tools: Git / GitHub · Maven · Postman · Docker · Docker Compose · Swagger · Apache Tomcat · IntelliJ IDEA
Concepts: OOP · MVC Architecture · DAO / DTO Patterns · REST API Design · JUnit · Agile · API Testing

EDUCATION

B.E. in Software Engineering

Aug 2023 – Aug 2027 (Expected)

Mehran University of Engineering & Technology (MUET), Jamshoro, Sindh

Relevant Coursework: OOP · Data Structures · Database Systems · Software Engineering · Web Technologies · Discrete Mathematics

Intermediate in Pre-Engineering

2019 – 2021

AIMS School and College, Matli, Sindh

PROJECTS

LetsReadBlogs — Full-Stack Blog Platform

Feb 2026 – May 2026

github.com/Tahir-Dars/letsReadBlogs

Java · Spring Boot · Spring Security · JWT · MapStruct · PostgreSQL · React · TypeScript · Docker

- Designed a 7-package layered backend architecture — controller, service, repository, domain, mapper, security, config — enforcing strict separation of concerns across 148 commits.
- Secured all write operations with stateless JWT authentication; public read endpoints remain open, eliminating session management overhead entirely.
- Eliminated DTO↔Entity mapping boilerplate with compile-time MapStruct mappers, decoupling API contracts from internal domain models and improving build-time safety.
- Containerised the data layer with Docker Compose — PostgreSQL database paired with an Adminer admin panel — enabling zero-config local setup for any contributor.
- Integrated a React + TypeScript + Vite frontend (deployed on Vercel) with TipTap rich-text editor, Axios API client, and DOMPurify sanitisation — live at letsreadblogshere.vercel.app.

Banking Customer REST API

Jan 2026

github.com/Tahir-Dars/BankCustomerAPI

Java · Spring Boot · Spring Data JPA · Spring Security · H2 · Maven

- Built a 4-tier Spring Boot REST API — Controller → Service → Repository → Model — covering the full customer lifecycle (CRUD) with clean layer boundaries throughout.
- Enforced endpoint protection with Spring Security authentication and wired Spring Actuator health and metrics endpoints, establishing production observability from day one.
- Standardised all responses with correct HTTP semantics (200, 201, 400, 404, 500) and consistent JSON contracts, reducing integration friction for API consumers.

Full-Stack To-Do List Web Application

Nov 2025 – Dec 2025

github.com/Tahir-Dars/todolistUsingServlet

Java · Servlets · JSP · JDBC · MySQL · Maven · Apache Tomcat · Bootstrap 5

- Delivered an end-to-end task management application: designed the MySQL schema, implemented 3 Servlet endpoints for full CRUD, and deployed a WAR artifact to Apache Tomcat.
- Applied a 4-layer architecture — Controller → DAO → DB Connector → Entity — achieving zero coupling between HTTP handling and data access logic.

- Built a console banking application with deposit, withdrawal, and fund-transfer operations — each mutation wrapped in explicit JDBC transaction boundaries with manual commit and rollback.
- Validated zero data corruption across 50+ test scenarios, confirming that partial-transfer failures roll back atomically under all tested conditions.
- Eliminated SQL injection risk across all three transaction types by using PreparedStatement exclusively — no string-concatenated queries anywhere in the codebase.

KEY ACHIEVEMENTS

- **Full-Stack Delivery:** Extended from pure backend to full-stack product with React + TypeScript frontend, Dockerised infrastructure, and live Vercel deployment — all within a single project.
- **Rapid Progression:** Moved from raw JDBC console apps to JWT-secured Spring Boot REST APIs with containerised databases within 12 months of self-directed learning.
- **Data Integrity:** Achieved zero data corruption across 50+ banking system test scenarios through explicit JDBC transaction boundaries and atomic operation validation.
- **Security-First Practice:** Applied SQL injection prevention via PreparedStatement across 100% of database operations and stateless JWT authentication in all secured projects.

LANGUAGES

English — Professional Proficiency · Urdu — Native · Sindhi — Native