

Muhammad Talha

(214) 909-3357 | talha.j771@gmail.com | linkedin.com/in/tjbravo | github.com/talha771

EDUCATION

M.S Computer Science (Incoming) <i>University of Texas at Dallas</i>	Jan 2026-Aug 2028
B.S Computer Science <i>Habib University</i>	Jun 2020-Jun 2024

TECHNICAL SKILLS

Languages: Python, JavaScript/TypeScript, SQL
Frameworks: React, Next.js, TailwindCSS, Redux, React Router, Node.js, Express, FastAPI, Redis
Databases: MySQL, PostgreSQL, NoSQL
Cloud/DevOps: AWS (EC2, S3, RDS), Docker, Git/GitHub, Unix/Linux

RELEVANT EXPERIENCE

Jacqueline Construction Wylie, TX <i>Full-Stack Developer</i>	Oct 2024-Present
- Led development of end-to-end job and time-tracking systems used by 50+ field and office staff, reducing manual reporting effort by 70%. - Built scalable APIs and frontend workflows using React, Redux, Node.js, and MySQL to enable real-time job status, labor cost tracking, and operational visibility, reducing daily reporting workload by 250+ hours annually. - Worked directly with operations and management to translate loosely defined business needs into shipped features on biweekly timelines.	
QLU.AI San Francisco, CA <i>Full-Stack Developer (Frontend-Focused)</i>	Jun 2024-Oct 2024
- Delivered multiple end-to-end features for a production SaaS platform using React, Next.js, TailwindCSS, and Node.js, supporting 300+ of daily active users. - Implemented structured frontend state management with Redux and React Router, reducing UI regressions by 30% - Built full-stack attachment functionality spanning React UI, backend services on EC2, and AWS S3, using pre-signed URLs to support secure, scalable file uploads and retrieval without routing files through the server. - Built a reverse-indexing and tokenization system on Apache Cassandra to enable search over deterministically encrypted messages, supporting fast lookup, automatic index updates on ingestion, and zero plaintext storage on the client or server.	

PROJECTS

To Vibe or Not to Vibe	May 2026 - Current
- Graduate security research with a UTD professor, targeting publication, evaluating AI-generated patch quality against human-written fixes using CodeQL static analysis across the SWE-bench Pro benchmark. - Built an end-to-end analysis pipeline that clones 11 open-source repos (JS, Python, Go), applies patch variants (control / golden / AI) via git worktrees, and runs CodeQL security-extended query suites automatically. - Engineered a SARIF diffing engine to compare findings before and after patching at the (ruleId, file, line) level, quantifying vulnerabilities fixed vs. regressions introduced per patch. - Leveraged high-performance computing (HPC) resources to scale CodeQL database creation and query execution across the full SWE-bench Pro test split, handling large multi-language codebases at volume.	