Zaeem Rashid

Dallas, TX | US Citizen | (972) 213-6403 | careers@zaeemrashid.com
Linkedin | zaeemrashid.com | Github

EDUCATION

The University of Texas at Dallas ***Expected December 2027***
Masters of Computer Science

The University of Texas at Dallas ***December 2025***
Bachelor of Computer Science

SKILLS

- **Programming Languages:** Java, Python, SQL, C, C++, TypeScript
- **Frameworks/Libraries:** React, LangChain, FastAPI, JUnit, Spring Boot, Pytest, GraphQL, REST APIs
- **Application/Tools:** AWS, Azure, Git/GitHub, Linux/Unix, Docker, Kubernetes, Prometheus, Grafana
- **Databases:** MySQL, PostgreSQL, SQLite, Amazon RDS, MongoDB, Redis
- **AI/ML:** LLM orchestration, AI agents, tool-based AI workflows, API-driven model usage
- **Methodologies:** Agile, MVC Architecture, CI/CD, DevSecOps

EXPERIENCE

AI Software Engineering Intern ***June 2025 - December 2025***
Mercedes-Benz R&D North America *San Jose, CA*

- Improved service reliability and monitoring coverage across microservices, measured by end-to-end observability during live beta deployment, by integrating Prometheus and Grafana into FastAPI services within an Agile environment.
- Increased actionable in-car user feedback across 7,500+ users, measured by 1,800+ submissions per month, by building GenAI workflows that automatically detected user friction and generated classified feedback with real-time context for internal teams.
- Reduced debugging time during production incidents, measured by faster root-cause identification across distributed services, by implementing request tracing and service-to-service data flow propagation using parent task IDs.
- Improved log consistency and debuggability across FastAPI microservices by implementing a schema-validated logging framework using Pydantic models and context propagation to produce consistent JSON logs with trace IDs and request metadata.

Founding Software Engineer ***January 2024 - Present***
MyPlannr - EventFlow Platform *Dallas, TX*

- Owning and developing a multi-tenant SaaS event management platform for weddings, corporate events, and private gatherings, using a modular monolith architecture with tenant-based data isolation to support scalable event and guest management.
- Organized the FastAPI backend into clear domain modules to keep the codebase maintainable and make it easier to split functionality into separate services if needed as the system grows.
- Developed backend APIs using Python and FastAPI to support authentication, event creation, RSVP workflows, task management, and SMS-based bulk messaging via Twilio.
- Reduced authentication issues and abuse by implementing JWT-based organizer authentication, Twilio-powered OTP login for guests, and per-endpoint rate limiting.
- Designed PostgreSQL schemas for multi-tenant workloads by modeling organizers, events, invites, and guests while preventing cross-tenant data access.
- Improved system stability and visibility under load by adding request validation, structured logging, and monitoring with Prometheus and Grafana to track API latency, error rates, and traffic.
- Built a React + TypeScript frontend with strongly typed state management, dynamic routing, event dashboards, and real-time RSVP updates to improve usability for organizers and guests.

PROJECTS

JARVIS - *Individual Project* ***December 2025***
Python, LangChain, FastAPI, Whisper, ElevenLabs, Spotify, OpenWeatherMap, Home Assistant

- Built an AI assistant capable of executing multi-step workflows across external tools, as measured by successful end-to-end task completion, by implementing LLM-driven intent planning and dynamic tool selection.
- Improved conversational continuity and response quality by maintaining entity-aware context across interactions, using structured memory, summarization, and optional voice input/output support.

Autocomplete Sentence Builder - *Group Project* ***November 2025***
Java, JavaFX, MySQL

- Built a database-backed autocomplete system for text prediction by modeling token frequencies and word-pair relationships, enabling data-driven suggestion generation.
- Improved data import and query performance by ~30%, as measured by processing time, by applying multithreading to large text parsing and database inserts.