

BHUVANESWAR REDDY MARREDDY

AI Solutions Engineer | Full Stack .NET & Agentic AI Development

bhuvaneshwar.marreddy@gmail.com | 704-318-6272 | Houston, TX | [LinkedIn](#) | [bhuvaneshwar.me](#)

PROFESSIONAL SUMMARY

Full Stack .NET Engineer with 4+ years building enterprise web applications and APIs, now deep in applied AI for software development. Build agentic development workflows and internal tooling with Claude Code and the Claude API — custom agents and subagents, MCP servers with guardrailed data access, and indexed flow/sprint context that speeds up delivery. Also experienced building LLM-powered applications — RAG, tool/function calling, and conversational UIs — through a generative-AI prototype and personal mobile projects. Strong foundation in C#, ASP.NET Core (.NET 6–9), REST, and SQL/NoSQL data stores, with hands-on Azure across Container Apps/App Service, Azure DevOps (Git + YAML pipelines), Application Insights, Azure AI Search (vector/semantic), Azure Data Factory, and Service Bus worker services. Solid UI skills across React, Angular, Blazor, and mobile (Flutter/React Native), plus secure authentication/authorization using Azure AD (Entra ID), OAuth2/OIDC, and JWT.

CORE SKILLS

AI & LLM Engineering: Claude API, Claude Code, MCP (Model Context Protocol), OpenAI (ChatGPT) API, ElevenLabs (text-to-speech), AI agents & subagents, tool/function calling, agentic workflows, RAG, prompt engineering, guardrailed LLM tooling, AI-assisted development

Backend: C#, .NET Core / ASP.NET Core (.NET 6–9), Web API, REST, EF Core / Dapper, ADO.NET, LINQ

Frontend: Blazor (Server/WASM), Razor Pages, React.js, Angular (9/10/12), TypeScript, JavaScript, HTML5, CSS/SCSS, RxJS

Mobile: Flutter, Dart, React Native, Expo, cross-platform app development

Data: SQL Server, Cosmos DB, MongoDB, MySQL, Azure AI Search (vector/semantic retrieval), Firebase/Firestore, Supabase

Cloud & DevOps: Azure (App Service, Container Apps, Application Insights, Azure AI Search, Azure Data Factory, Service Bus), Azure DevOps (Git, YAML CI/CD, release pipelines), Docker, GCP, AWS (EC2/S3 – exposure), Vercel, Git/GitLab

Security: Azure AD (Entra ID) SSO, OAuth 2.0 / OIDC, JWT, role-based access, secure credential handling

Messaging & Jobs: Azure Service Bus, .NET Worker Services, background/scheduled jobs

Tools & Practices: SSMS, Swagger/OpenAPI, Postman, Jira, Playwright (E2E testing), SonarQube, Agile/Scrum, code reviews, production support

PROFESSIONAL EXPERIENCE

The Canton Group — .NET Developer

Feb 2024 – Present

Project: Election Management Platform

AI Engineering & Agentic Tooling

- Build internal AI-assisted development workflows using **Claude Code** with custom agents and subagents, orchestrating Git worktrees for parallel, multi-task development across the team.
- Design and develop **MCP (Model Context Protocol) servers** exposing guardrailed, least-privilege access to SQL Server and Cosmos DB, enabling safe, scoped LLM-driven data reads without granting broad database access.
- Implement MCP-based **Application Insights** integrations for automated log retrieval and analysis, accelerating production incident investigation and root-cause tracing.
- Integrate Claude agents with **Azure DevOps** to assist pull-request monitoring and review, PBI/backlog planning, and sprint workflows, improving review consistency and reducing manual triage.
- Author reusable agent **skills and tool-calling patterns on the Claude API**, establishing internal standards for grounded, guardrailed, and repeatable AI-assisted engineering.
- Continuously refine agent definitions, guardrails, and prompt patterns, applying lessons from prototype builds to harden day-to-day team development workflows.

- Built an indexed memory/context layer capturing application flows and sprint-wise development history, enabling agents in later sprints to retrieve prior context, understand existing flows faster, and reduce rework and onboarding time.
- Integrated automated Playwright end-to-end tests as a final local validation gate — run after SonarQube static analysis and before raising a PR — to catch UI/flow regressions early in the cycle.

Platform Engineering (.NET / Azure)

- Develop and enhance a large-scale election management platform using C#, ASP.NET Core (.NET 6–9), Web APIs, and SQL Server in an Agile/Scrum environment.
- Design and maintain REST API endpoints and integrations with consistent request/response contracts, validation, and versioning across services.
- Build reusable, component-based UI with Blazor/Razor Pages backed by shared .NET class libraries (DTOs/models, repository layer) consumed across UI, API, and worker services.
- Deploy and operate services with Docker on Azure App Service and Container Apps, managing environment configuration and rollouts through Azure DevOps Git + YAML CI/CD pipelines.
- Implement observability and reliability with Application Insights — structured logging, dependency monitoring, performance tracing, health checks, and centralized exception middleware.
- Deliver sub-second search across ~13M records by implementing and tuning Azure AI Search indexing and query patterns, including filtering and wildcard scenarios.
- Implement asynchronous and scheduled workflows using Azure Service Bus with .NET Worker Services, and support data movement/migration via Azure Data Factory pipelines.
- Design data models and query patterns for NoSQL stores (Cosmos DB, MongoDB) to support performance and scale where applicable.
- Secure APIs with Azure AD (Entra ID) SSO, OAuth2/OIDC, and JWT role-based access; validate contracts using Swagger/OpenAPI and Postman.
- Act as go-to engineer for critical bug investigations and production escalations, and perform peer PR reviews enforcing coding standards, architecture consistency, and release quality.

Tech Stack: C#, ASP.NET Core (.NET 6–9), Web API, SQL Server, Cosmos DB, MongoDB, Azure AI Search, Azure Data Factory, Azure Service Bus, Application Insights, Docker, Azure App Service/Container Apps, Azure DevOps (YAML pipelines), Claude API, Claude Code, MCP, Playwright, SonarQube, Git, Jira, Postman

Dsilo — Full Stack Developer (Contract / Project-Based)

Jan 2023 – Feb 2024

Project: ProcureGPT (AI Chatbot / Procurement Assistant Prototype)

- Built an AI chatbot prototype integrating a React.js conversational UI with ASP.NET Core Web APIs, supporting chat sessions, multi-turn context handling, streaming-style responses, and workflow-based interactions.
- Implemented backend endpoints for prompt handling, tool/function routing, and LLM service integration; shaped response contracts to keep the UI fast and predictable.
- Applied LangChain-style patterns (prompt templates, chains/agents, retrieval-augmented generation) to ground responses in knowledge sources and reduce hallucinations.
- Iterated on prompt and retrieval behavior through systematic evaluation and tuning (without model training) to improve response grounding and reliability for procurement workflows.
- Built a multi-tenant experience with secure session handling, route guards, and role-based access using OAuth2/OIDC and JWT token patterns.
- Implemented complex React state management and reusable UI components (hooks/context patterns), focusing on performance, maintainability, and consistent CSS/SCSS styling.
- Containerized services and supported environment-specific deployments on GCP, including configuration management and integration testing for stable demos and handoff.
- Delivered end-to-end execution on GCP — deployments, configuration management, integration testing, and release readiness for stakeholder demos.
- Accelerated onboarding and team velocity by simplifying complex flows, improving code structure, and documenting key integration points and runbooks.

Tech Stack: C#, ASP.NET Core Web API, React.js, TypeScript, REST, OAuth2/OIDC, JWT, LLM integrations, RAG, LangChain-style concepts, Docker, GCP, Git

Infosys Private Limited — .NET Developer

Oct 2021 – Dec 2022

Project: InfyMe (Internal Employee Services Portal)

- Developed and maintained internal enterprise modules using C#, ASP.NET Core, and REST APIs to support employee self-service workflows in a centralized portal.
- Built UI features using Angular and TypeScript and integrated API data to improve usability across screens.
- Implemented and consumed RESTful services with structured JSON contracts; handled validation, error handling, and UI-state management consistently across layers.
- Worked with SQL Server for queries and data fixes; ensured backward-compatible schema changes and stable releases.
- Participated in Agile ceremonies, sprint planning, and code reviews; collaborated with QA and product teams to deliver production-ready features and support releases.
- Supported defect triage and production fixes by analyzing logs, reproducing issues, and delivering clean patches with regression awareness.
- Performed peer code reviews and collaborated with cross-functional teams to improve code quality, API consistency, and sprint delivery.
- Validated APIs using Swagger/OpenAPI and Postman, and contributed to secure access patterns (JWT-based authorization and basic encryption practices where required).
- Assisted with CI/CD and release activities through Azure DevOps, ensuring stable deployments and smooth handoffs to QA and support teams.

Tech Stack: C#, ASP.NET Core, Web API, Angular, TypeScript, SQL Server, Azure DevOps/Git, Postman, Jira

PERSONAL PROJECTS — MOBILE APP DEVELOPMENT

Cross-Platform Mobile Apps (Flutter / React Native) — *Self-initiated, built with a co-developer*

- Designed, built, and shipped two cross-platform mobile apps from concept to beta release on the App Store, owning UI design, implementation, and deployment alongside a co-developer.
- Built backends and data layers with Firebase/Firestore and Supabase, using serverless deployment on Vercel for fast, zero-cost iteration.
- Integrated vector databases for semantic search and retrieval-augmented (RAG) AI features — the same retrieval patterns applied professionally with Azure AI Search.
- Integrated OpenAI (ChatGPT) to power conversational and generative in-app features, shaping prompts and response handling for a responsive mobile experience.
- Added text-to-speech with the ElevenLabs API to convert generated content into natural-sounding audio for users.
- Built monitoring-style background functions for automated, user-customizable notifications — triggering alerts based on configurable conditions and events.
- Used Expo for React Native development, on-device testing, and preview/distribution, streamlining the build-and-feedback cycle.
- Implemented user authentication and real-time data sync using Firebase and Supabase, with secure access rules and cloud-hosted data stores.
- Collaborated via Git workflows (branching, PR review, shared task planning) and applied AI-assisted development (Claude Code, agentic patterns) to prototype and iterate faster.

Tech Stack: Flutter, Dart, React Native, Expo, Firebase/Firestore, Supabase, vector databases (RAG), OpenAI (ChatGPT) API, ElevenLabs, Vercel, REST APIs, Git

EDUCATION

Master of Science (M.S.), Computer Science — University of Central Missouri, USA — 2024