

Alex Parker

Principal Full Stack .NET Developer & Problem Fixer

Turns production fires into boring, fast, maintainable systems.

Seattle, WA alex.parker.dev@exampleemail.com (555) 213-9874 linkedin.com/in/alex-parker-dev
github.com/alexparker-dotnet

PROFILE

Principal-level full stack engineer with 14 years of experience building and rescuing .NET systems end to end. Deep focus on performance, refactoring, and production firefighting, from SQL Server query plans to vanilla JavaScript on the DOM. Equally comfortable writing C# APIs, optimizing SSIS packages, and jumping into tangled legacy code to make it fast, testable, and boring again. Known for speaking both “business” and “tech” and for shipping changes that move actual numbers, not just ticket counts.

CORE STRENGTHS

.NET Core / .NET Framework

C# API Design

Vanilla JavaScript, HTML5, CSS3

Performance Profiling & Tuning

Legacy Code Refactoring

SQL Server Query Optimization

SSIS / SSRS

Distributed Systems

Technical Leadership

Architecture & Design

Stakeholder Collaboration

Incident Response / Root Cause

TECHNICAL SKILLS

Backend: C#, .NET 6/7, .NET Framework, ASP.NET MVC/Web API, REST, LINQ, Entity Framework, Dapper

Frontend: Vanilla JS (ES6+), HTML5, CSS3/Flexbox/Grid, minimal frameworks, Webpack, TypeScript (working knowledge)

Data: SQL Server, T-SQL, indexing strategies, execution plans, table partitioning, SSIS, SSRS

Cloud & Infra: Azure (App Service, Functions, SQL DB), IIS, Windows Server, CI/CD (Azure DevOps, GitHub Actions)

Practices: TDD when it matters, clean architecture, domain-driven thinking, code review, mentoring, pair programming

Tooling: Visual Studio, VS Code, Git, Postman, SQL Profiler, Redgate tools, Fiddler, Chrome DevTools

SELECTED IMPACT HIGHLIGHTS

- Refactored a monolithic .NET order processing engine, cutting average order time from 14 minutes to under 90 seconds and reducing nightly failures by 93%.
- Rewrote a set of core T-SQL reports and SSRS dashboards for finance, reducing run times from 40+ minutes to under 3 minutes and eliminating timeouts during quarter close.
- Led modernization of a legacy WebForms app to ASP.NET Core + vanilla JS, maintaining feature parity while reducing page size by 60% and support tickets by 30% within six months.

- Introduced lightweight performance budgets, profiling, and review practices that surfaced and eliminated several six-figure-per-year infrastructure costs.

PROFESSIONAL EXPERIENCE

Principal Full Stack .NET Developer / Team Lead

Northwind Logistics Solutions, Seattle, WA

2020 – Present

- Lead a team of 6 developers owning a mission-critical shipment tracking platform built on ASP.NET Core, SQL Server, and vanilla JS/CSS.
- Redesigned core tracking API and database schema, improving p95 API latency by 72% and reducing deadlocks to near zero using better indexing and isolation levels.
- Rewrote heavy client-side frameworks with targeted vanilla JS, cutting initial load from 4.2s to 1.6s on average and improving Lighthouse performance scores from 58 to 93.
- Created a “performance clinic” process: running SQL Profiler traces, .NET profilers, and browser audits to attack hotspots; drove down production incident volume by 40% year over year.
- Partnered with operations and finance leads to prioritize technical work that directly supported SLAs and margin targets instead of just “cool refactors.”

Senior Full Stack Engineer / Solutions Architect

BrightWave Health Systems, Bellevue, WA

2016 – 2020

- Owned architectural direction for a suite of clinical workflow apps built on ASP.NET MVC, Web API, and SQL Server with SSIS-driven data ingestion.
- Refactored a brittle SSIS-based ETL pipeline processing 10M+ rows nightly, introducing partitioned loads and incremental updates; reduced run time from 6 hours to under 90 minutes.
- Redesigned patient-facing UI from server-heavy Razor views to slim HTML + JS with REST APIs, reducing page reloads by 50% and increasing task completion rates in usability tests.
- Served as escalation point for production issues; implemented runbooks, better logging, and root-cause reviews that cut “mystery outages” by more than half.
- Mentored 4 mid-level developers into senior roles, emphasizing reading execution plans, understanding browser behavior, and tying work to clinical and business outcomes.

Senior .NET Developer / Lead

CityMarket Retail Technologies, Portland, OR

2013 – 2016

- Led development for a high-traffic promotions and pricing engine used by multiple retail chains.
- Identified and fixed a set of poorly performing T-SQL queries (cart pricing and inventory lookups) by adding proper indexes, rewriting joins, and splitting long-running reports; improved throughput by 3x.
- Optimized legacy ASP.NET MVC controllers and view logic, caching expensive operations and cutting homepage load time by over 50% without adding heavy JS frameworks.
- Introduced automated load tests that caught regressions early and justified infrastructure scaling decisions with data instead of hunches.

Full Stack .NET Developer

Harborline Insurance Group, Seattle, WA

2010 – 2013

- Developed internal underwriting and claims tools using ASP.NET WebForms/MVC, C#, and SQL Server, gradually refactoring toward cleaner layers and test coverage.

- Replaced Excel-based reporting with SSRS dashboards tied to optimized stored procedures, improving data freshness from weekly to hourly and increasing report adoption.
- Implemented a shared JavaScript utilities library (plain JS) to standardize common UI behaviors and reduce copy-paste code across legacy pages.

Software Developer (.NET)

Cascade Business Systems, Tacoma, WA

2009 – 2010

- Built and maintained line-of-business web apps in ASP.NET and SQL Server, working across the full stack from database to CSS.
- Got hooked on performance while tracking down a “random” timeout that turned out to be a full table scan—started a habit of reading query plans that never really stopped.

SAMPLE PROJECTS

- **Order Processing Engine Rewrite** — Extracted a tangled set of procedural C# services into a layered architecture with clear boundaries, async I/O, and tuned SQL access. Result: 10x faster throughput, simplified onboarding for new devs, and a noticeable drop in 2 a.m. calls.
- **Legacy UI Slim-Down** — Took a 10-year-old WebForms-based admin portal and replaced key flows with lean ASP.NET Core endpoints and vanilla JS frontends. Result: fewer reloads, happy power users, and a codebase small enough for humans to understand.
- **Reporting & Analytics Cleanup** — Consolidated dozens of ad-hoc T-SQL scripts into parameterized stored procedures powering SSRS reports, with standardized indexing and monthly performance reviews.

EDUCATION & CERTIFICATIONS

B.S. in Computer Science, University of Washington, Seattle, WA

Microsoft Certified: Azure Developer Associate (AZ-204)

Additional Training: Advanced T-SQL Query Tuning, Effective Legacy Code Refactoring

LEADERSHIP & COLLABORATION

- Regularly act as bridge between product owners, ops, and engineering to slice big technical initiatives into business-visible wins.
 - Run brown-bag sessions on “reading query plans without crying,” browser performance basics, and pragmatic refactoring.
 - Enjoy pairing with devs on “impossible” bugs and turning them into repeatable checklists and patterns for the team.
-