Dustin Daniel Lee

Software Engineer

Senior Software Engineer with **8+ years** of proven expertise building enterprise-scale applications using **.NET Core, ASP.NET, and Azure/AWS** cloud platforms. Specialized in architecting microservices, designing Clean Architecture patterns, and delivering high-performance, multi-tenant systems. Demonstrated success modernizing legacy systems, implementing **CI/CD pipelines**, and leading technical initiatives that drive business outcomes. Deep proficiency across the entire stack: **C# backend systems, REST/CQRS APIs, React/Angular frontends, and cloud-native** infrastructure with **ECS, Lambda, and Kubernetes.**

Location	Email	Phone	Website	LinkedIn
Asutin, TX, 78735	dustlee.dev@gmail.com	+1 (430) 901-5970	https://dustinlee.online/	https://www.linkedin.com/in/dustin-daniel-lee-1b20803b0

Work experience

Booktix Jun 2022 - Present (3 years 9 months)

Senior Software Engineer Full-time - East Brunswick, New Jersey

C# · .NET 6/Core · ASP.NET MVC · Razor Pages · React · Angular 11 · SignalR · SQL Server · Entity Framework Core · AWS ECS Fargate · Docker · Kubernetes · GitLab CI/CD · OpenTelemetry · CloudWatch · Elasticsearch · CQRS

- Modernize a legacy Angular 1.x monolith serving high-concurrency ticket transactions while maintaining zero downtime and seamless user experience across 200+ partner organizations.
- Led front-end modernization of legacy Angular 1.x app, architecting hybrid Angular 11 + ASP.NET MVC interface. Implemented incremental module-by-module migration strategy to eliminate risk and enable rollback, maintaining uptime across 50M+ annual transactions.
- Designed and built real-time interactive seat selection interface for ticket buyers (UX modeled after industry-standard pretix/seats.io patterns). Implemented with React frontend, .NET Core backend, and SignalR for bi-directional real-time synchronization. System handles 10K+ concurrent users with sub-100ms latency
- Architected and maintained global multi-tenant point-of-sale platform serving Fortune 500 companies. Backend built on .NET Core using Clean Architecture with CQRS pattern for separation of concerns. Implemented tenant isolation, custom workflows, and audit logging. Handles 2M+ daily transactions across 50+ regions.
- Developed Blazor and React-style component library enabling mobile-first UIs with 95%+ code reuse. Implemented real-time sync patterns via SignalR and ASP.NET Core APIs, reducing mobile development time by 40%.
- Built React dashboards for internal admin and B2B clients featuring real-time reporting, WebSocket-driven notifications, and secure component loading integrated with .NET APIs. Implemented real-time data refresh reducing reporting latency from 5 minutes to under 500ms.
- Managed frontend and backend CI/CD pipelines with GitLab, reducing deployment time from 30 minutes to 8 minutes. Integrated OpenTelemetry with CloudWatch for distributed tracing, enabling rapid incident response and performance diagnostics.

Maverick Capital Mar 2020 - May 2022 (2 years 2 months)

Software Engineer Contract - Dallas, Texas

C# · ASP.NET Web API · .NET 2.0/3.0/3.5 · React · SQL Server · Entity Framework · Bloomberg API · Rhino Mocks · xUnit · TFS · LINQ

- Build high-performance financial analytics tools for quantitative traders requiring sub-second data refreshes, complex risk calculations, and real-time reconciliation across Bloomberg API feeds.
- Designed and implemented RESTful ASP.NET Web API backend serving internal React dashboards for trade reconciliation and risk analytics. Optimized API response times by 60% through caching strategies, LINQ query optimization, and connection pooling for 100+ concurrent analyst connections.
- Built interactive React dashboards for risk analytics and trade reconciliation with real-time Bloomberg API integration. Implemented client-side state management and responsive design patterns enabling analysts to monitor 10K+ positions simultaneously with under 1 second refresh rates.
- Introduced comprehensive frontend and backend test coverage (unit, integration) using Rhino Mocks and xUnit, increasing code coverage from 12% to 78%. Established responsive design patterns and accessibility standards for cross-platform financial tools.
- Tasked with defining shared component strategies and architecture decisions across all C# software projects (.NET 2.0, 3.0, 3.5 legacy systems). Documented best practices, code review standards, and migration pathways for modernization initiatives.
- Conducted 30+ technical candidate interviews, evaluating full-stack capabilities and mentoring junior developers on .NET fundamentals and API design patterns.

Accenture Jul 2017 - Feb 2020 (2 years 8 months)

Full-stack engineer Full-time - Redmond, WA

C# · ASP.NET · React · .NET Framework · SQL Server · Entity Framework · Razor · jQuery (legacy) · TFS · Bloomberg API · Rhino Mocks

- Migrate nationwide real estate servicing platform from legacy jQuery/ASP monolith to modern SPA architecture serving 500+ concurrent users processing thousands of transactions daily.
- Led front-end logic transformation from legacy jQuery/ASP.NET into dynamic Single Page Applications using React and Razor integration. Implemented component-based architecture reducing code duplication by 45% and improving developer velocity by 50%.
- Created modular React components for complex routing algorithms and scheduling apps in nationwide real estate servicing platform. Implemented form state management, data validation, and error handling supporting multi-step workflows with 99.5% uptime.
- Coordinated with backend developers to shift computationally expensive client-side logic to optimized .NET APIs, reducing frontend JavaScript payload by 35% and improving time-to-interactive by 40%. Implemented server-side caching and query optimization reducing database load.
- Designed RESTful .NET APIs for seamless React integration, establishing clear contracts and versioning strategy enabling independent frontend/backend deployments. Mentored junior developers on API design patterns and React best practices.

Education

The University of Texas at Austin June 2015 - June 2017

Master's degree in computer science - Computer Science

The University of Texas at Austin Aug 2011 - May 2015

Bachelor's degree in computer science - Computer Science and Applied Mathematics