Aleksei Beck

alekseibeck0711@gmail.com | linkedin.com/in/alekseibeck | github.com/AlekseiBeck

Education

Michigan State University **May 2025**
Bachelor of Science in Computer Science | GPA: 3.7 | Dean's List (6x) *East Lansing, MI*

Experience

Software Developer I **May 2025 – Present**
United Wholesale Mortgage (Intern → Associate Software Developer → Software Developer I) *Pontiac, MI*

- Earned Most Valuable Project Award (1 of 16 intern groups) for a BOLT code quality initiative that eliminated ∼1,900 code smells and raised test coverage from ∼70% to over 80% across 34+ microservices; authored README standards adopted as an official UWM engineering convention.
- Authored and maintained 15 C# NRules in the BOLT rule engine that evaluate loan objects (property, insurance, income, tradelines, liabilities) and fire underwriter-facing conditions, tracking Freddie Mac and investor guideline changes.
- Engineered C# matching algorithms with LLM-driven response parsing that link borrower liabilities to credit tradelines and loan documents to specific debt-related risks, backed by EF Core data access changes and SQL migration scripting.
- Drove CI/CD compliance across 4 BOLT repositories by authoring the unit, integration, contract, smoke, and end-to-end suites required to clear automated quality gates, advancing a company-wide initiative to certify every microservice for continuous deployment through PORT by the end of 2026.
- Delivered 70+ Jira tickets spanning high-priority production bugs and net-new feature work, using Dynatrace to monitor deployments and perform root-cause analysis across the BOLT microservice ecosystem.

Projects

Loan Info Tool | *C#, .NET, EF Core, TypeScript, SQL Server, OAuth, Dream UI* | *UWM Internal* **May 2026 – Present**

- Led the intern team that designed and built an internal loan diagnostics tool used across UWM's ∼1,500-person IT department, consolidating all related information for a single loan so developers, business analysts, and product owners can inspect test loans in integration and staging alongside production loans flagged by underwriters and operations specialists.
- Replaced a manual workflow that took hours to reconstruct a loan's state with a single lookup that traces the loan through BOLT's approval pipeline and pinpoints the exact service where a feature or production-breaking change fails.
- Built the C#/EF Core backend behind a consolidated GET endpoint secured by UWM's internal Global User Passport library and OAuth, and delivered the React/TypeScript frontend on the internal Dream UI component library; owned testing, drove UI iterations with UX designers, and shaped the feature roadmap.
- Shipped per-user search history, bookmarks, and custom loan aliases that map a loan to its Jira story, with granular and bulk history deletion, so engineers tracking multiple defects resume context instead of re-searching by loan number.

Stakdx | *TypeScript, React, Supabase, Finnhub, Claude API, Alpaca API* | stakdx.vercel.app **April 2026 – Present**

- Designed a budget-driven scan engine that returns AI-ranked longs, shorts, calls, and puts scored by confidence, combining real-time market data, global news sentiment, and a Claude-powered analysis layer that also handles conversational market questions and on-demand portfolio review.
- Validated AI trade recommendations against live market performance via the Alpaca paper trading API, and delivered opt-in push notifications for breaking news, price movement, and sell signals on tracked positions.
- Architected production cloud deployment across Vercel (frontend), Railway (backend), and Supabase (Postgres) with GitHub-based CI/CD, and built a responsive UI that adapts between desktop and mobile.

Technical Skills

.NET Stack: C#, .NET, EF Core, XUnit, SQL Server, SQL Server Management Studio
Web & Frontend: TypeScript, JavaScript, React, Next.js, Node.js, HTML/CSS, REST APIs, OAuth
Other Languages: Python, C++, SQL
Cloud & DevOps: Azure, Vercel, Railway, Supabase, Docker, Jenkins, CI/CD, Dynatrace, Git, GitHub, Bitbucket, Jira
AI / LLM: Custom Claude Code and GitHub Copilot agents, skills, and instructions, LLM integration, prompt engineering