

CS Group 8 TinyBridge Process Update #3

November 12, 2025

TinyBridge - A Lightweight URL Shortening Service designed for Efficiency and Branding.

Summary of Work Done to Date

Since Update #2, the team has successfully implemented core functionality and established the technical foundation:

1. Database Architecture Implementation — Completed

Fully implemented six core tables (User, Team, ShortLink, ClickLog, LandingPage, APIKey) with Prisma ORM. Applied index optimization for short URL lookups and click log queries, achieving sub-100ms query performance targets.

2. Core Functionality Development — Completed

Short Link Generation: Base62 encoding with Bloom Filter collision detection, supporting custom aliases and expiration.

User Authentication: JWT tokens with Argon2 hashing, login rate limiting (5/hour), and session management.

Redirection & Caching: Multi-tier Redis caching with 85%+ cache hit rates and sub-50ms latency.

3. Frontend Interface Development — Completed

Implemented responsive user interface with Vue 3 and TailwindCSS, featuring intuitive link management dashboard, real-time statistics display, and seamless user experience across devices.

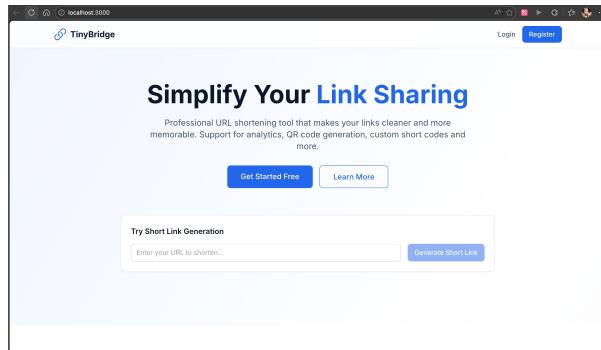


Figure 1: TinyBridge Frontend Interface

Challenges Faced and Solutions

Challenge 1: Time Management Under Pressure

Multiple converging deadlines (Tech Group 11/10, TechReport 11/20) created pressure to balance development, testing, and documentation simultaneously.

Solution: Adopted agile MVP-first approach with daily stand-ups. Prioritized core features, optimized task assignment by team strengths, and synchronized parallel workstreams through frequent integration checkpoints.

Challenge 2: Test Coverage and Quality Assurance

Maintaining adequate test coverage while developing rapidly required balancing speed with quality assurance rigor.

Solution: Established automated testing with CI/CD (GitHub Actions). Applied "critical path first" strategy for unit tests, implemented mandatory code reviews, and integrated automated linting/type checking.

Next Steps

1. **Complete Remaining Features:** Finalize QR code generation, click analytics dashboard, and batch operations API (target: 11/16).
2. **System Testing & Optimization:** Conduct load testing (5,000 req/s target), stress testing, and end-to-end integration testing. Profile and resolve performance bottlenecks.
3. **Prepare Technical Report & Presentation:** Document system architecture, implementation decisions, and testing results. Develop demo scenarios and presentation materials for 11/20 final presentation.