

AI-Driven Development Test: Production-Ready Banking REST Service

OVERVIEW:

Your task is to develop a REST service for a banking use case. To complete this efficiently, you are required to use AI-driven development practices. This assessment measures your ability to leverage AI tools to design and implement a banking service, as well as your capability to deliver production-grade features using AI-assisted workflows.

Project Requirements:

Core Components: You must develop the following components:

1. ****Service Interface****
 - Signing up
 - Authentication
 - Account Holders
 - Accounts
 - Transactions
 - Money Transfer
 - Cards
 - Statements
2. ****Database**** (using SQLite)
 - Database implementation
3. ****Test Suite****
 - Comprehensive test coverage for critical functionality
 - Unit tests for business logic
 - Integration tests for API endpoints
4. ****Containerization****
 - Dockerfile for the application
 - docker-compose.yml for local development
 - Environment variable configuration
 - Multi-stage builds (bonus)
5. ****Logging & Monitoring****
 - Structured logging implementation
 - Log levels and formatting
 - Error tracking and reporting
6. ****Health Checks****
 - Health check endpoint
 - Database connectivity checks
 - Service readiness probes
 - Graceful shutdown handling

Technology Stack:

- Your choices of technologies will be judged against the nature of the application and how effective those technologies are in modern software development ecosystems

Getting Started

1. Begin development using your AI tools of choice

Deliverables:

1. ****Source Code****
 - All components (service, database, tests, containers) in organized directories
 - Comprehensive test suite with good coverage
 - Docker configuration files
 - Configuration files for different environments
2. ****Documentation****
 - README.md with setup instructions, API documentation, and deployment guide
 - Security considerations document
 - Future considerations/roadmap document
 - Docker deployment instructions
3. ****AI Usage Report****
 - Tools used (Claude Code, ChatGPT, Cursor, Copilot, etc.)
 - Example prompts and iterations
 - Challenges faced and how AI helped solve them
 - Areas where manual intervention was necessary
4. ****Bonus Points****
 - Test client application demonstrating the flow
 - Simple frontend interface

Ways of Working

1. Commit regularly with meaningful messages
2. Include your AI usage log as a Markdown file

Additional Notes

- You MUST use any AI tools available to you
- External libraries and frameworks are allowed
- Focus on demonstrating effective AI-driven development
- Partial implementations are acceptable if well-documented
- Security is paramount - ensure no secrets are committed
- Test suite is a hard requirement - demonstrate testing best practices
- Production readiness is key - show understanding of operational concerns