MICHIGAN STATE UNIVERSITY

Project Plan Presentation

Code Quality Intelligence System

The Capstone Experience

Team Auto-Owners

Austin Kim
Rohan Kochhar
Eva Shaska
Omer Sipahioglu
Amelia Smallfied
Sagar Ashwathanarayana Kikkeri

Department of Computer Science and Engineering
Michigan State University

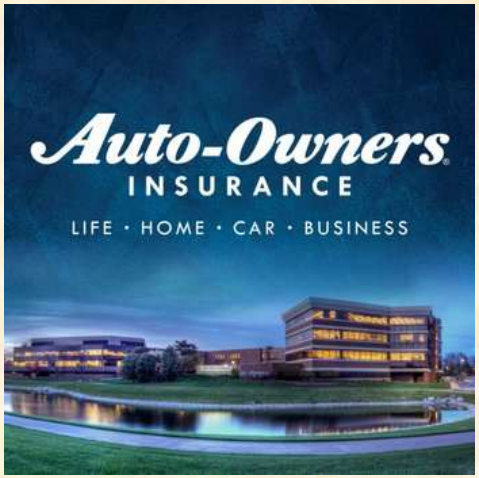
Spring 2026



From Students…
…to Professionals

Project Sponsor Overview

- Fortune 500 Company
- Headquarters in Lansing, MI
- Life, Auto, Home, and Business



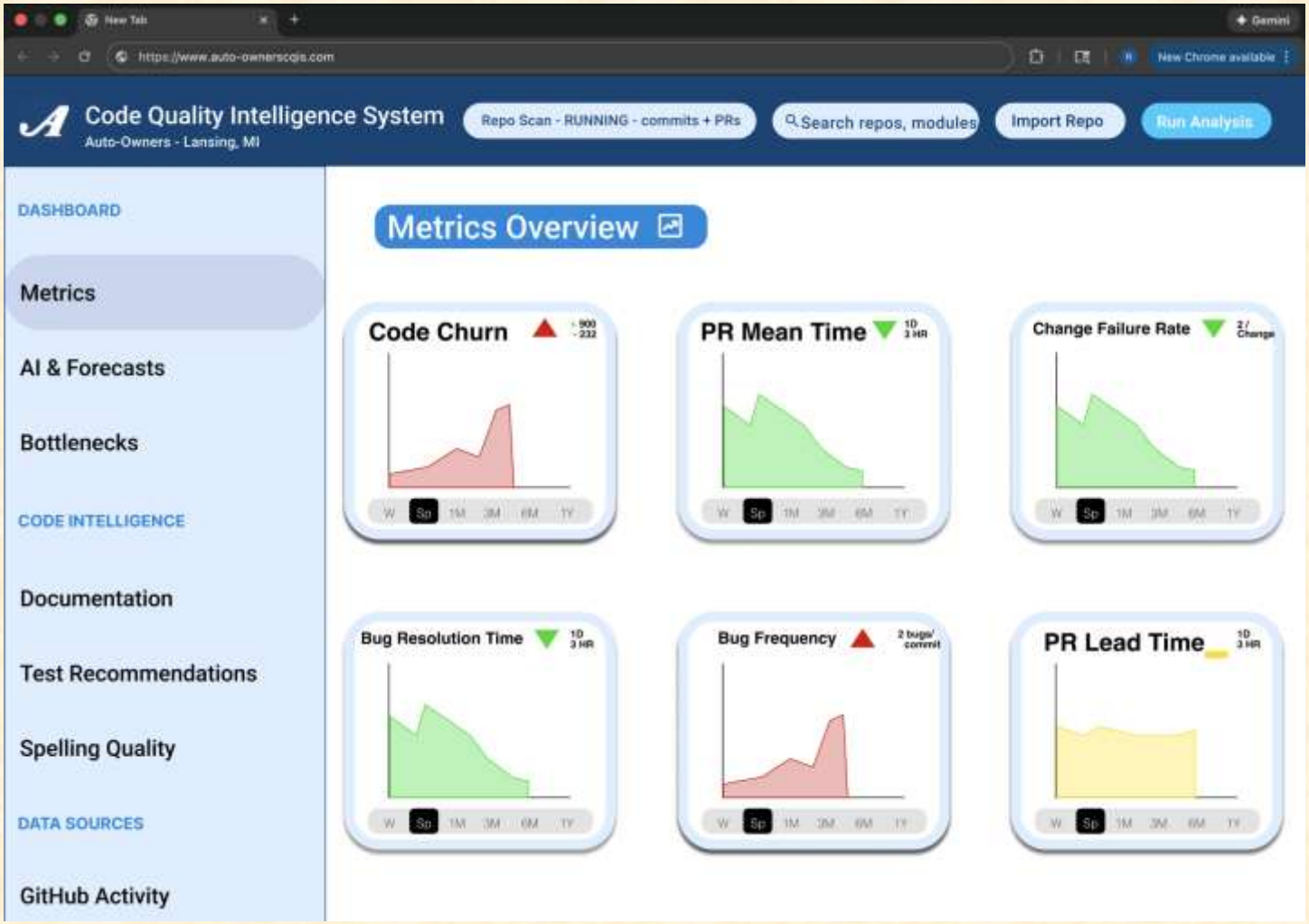
Project Functional Specifications

- Analyzing code quality through interactive dashboard
- Find inefficiencies and bugs
- Improve quality of code and efficiency of development

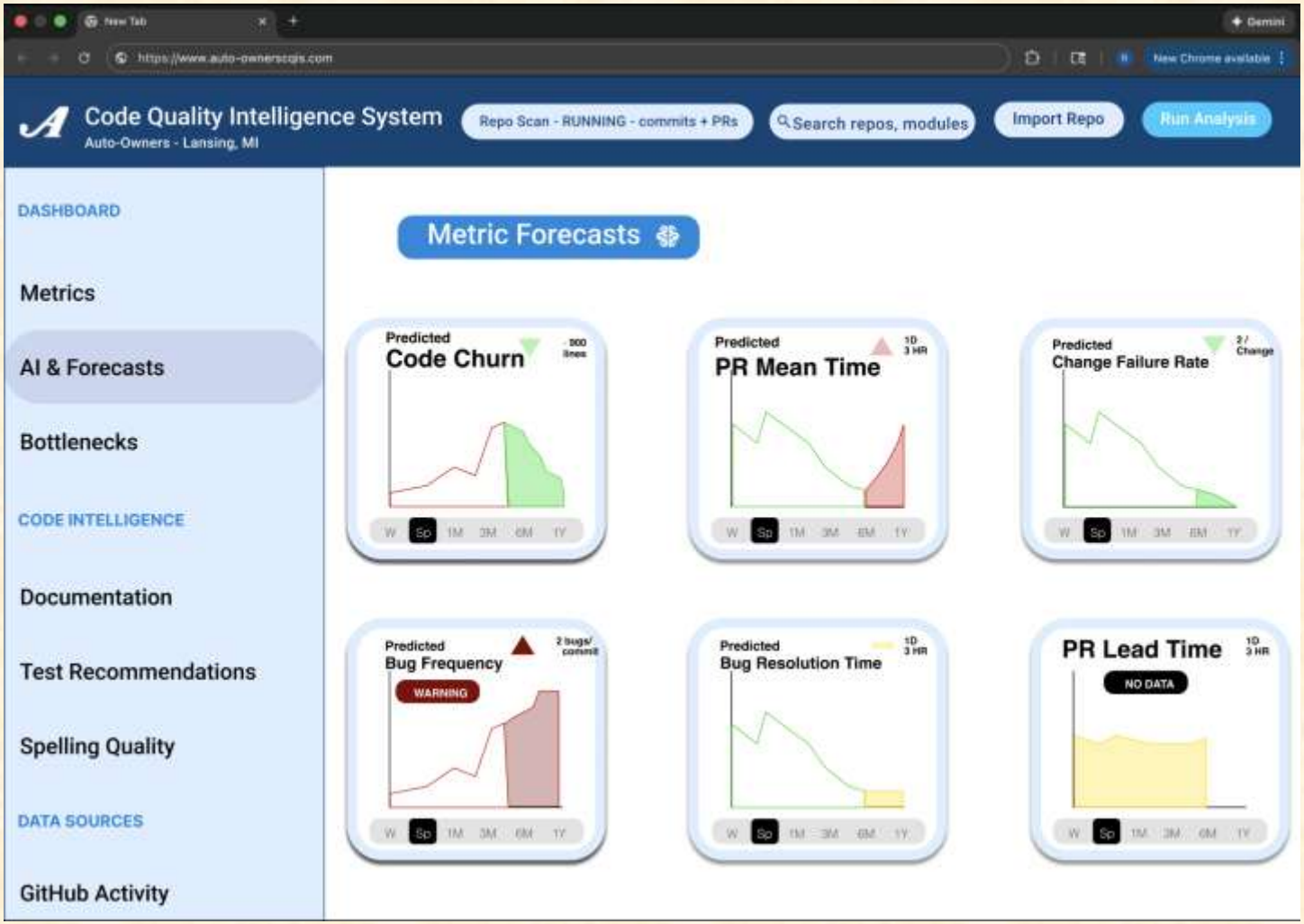
Project Design Specifications

- Code Quality Metrics
- Unit Test Recommendations
- Documentation
- Spelling Check
- GitHub Activity
- Bottlenecks

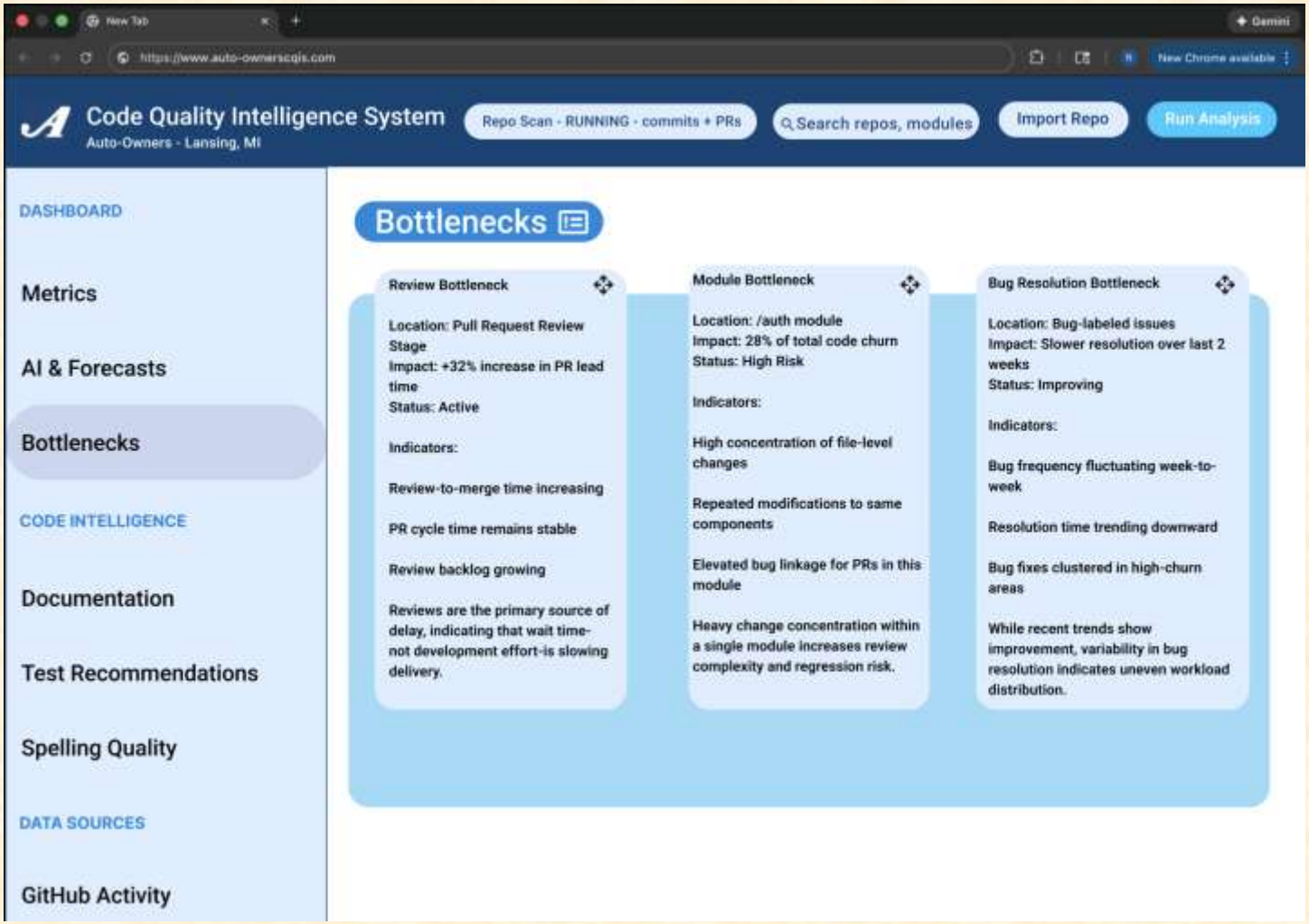
Screen Mockup: Metrics page of the CQIS



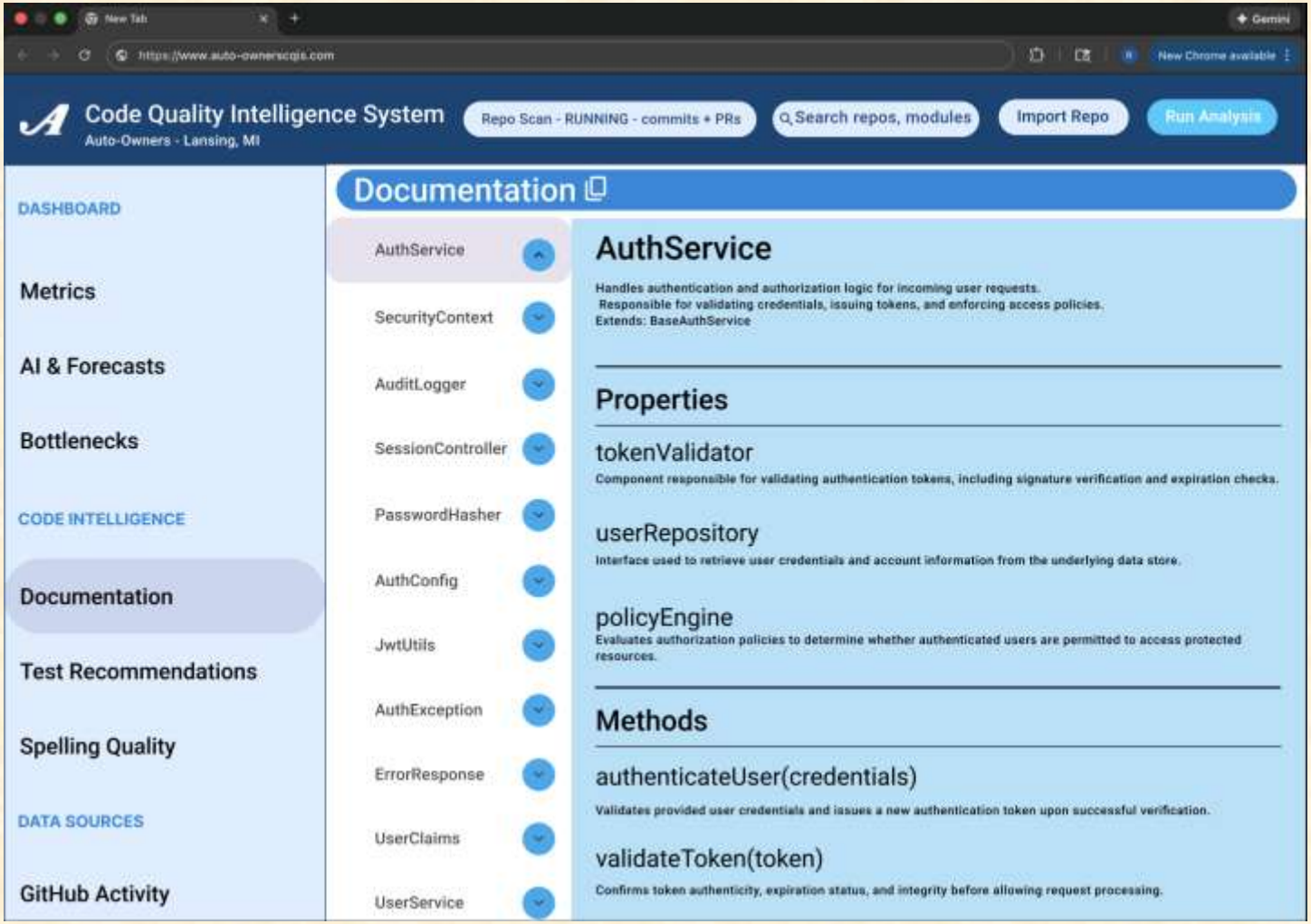
Screen Mockup: AI Forecast page of the CQIS



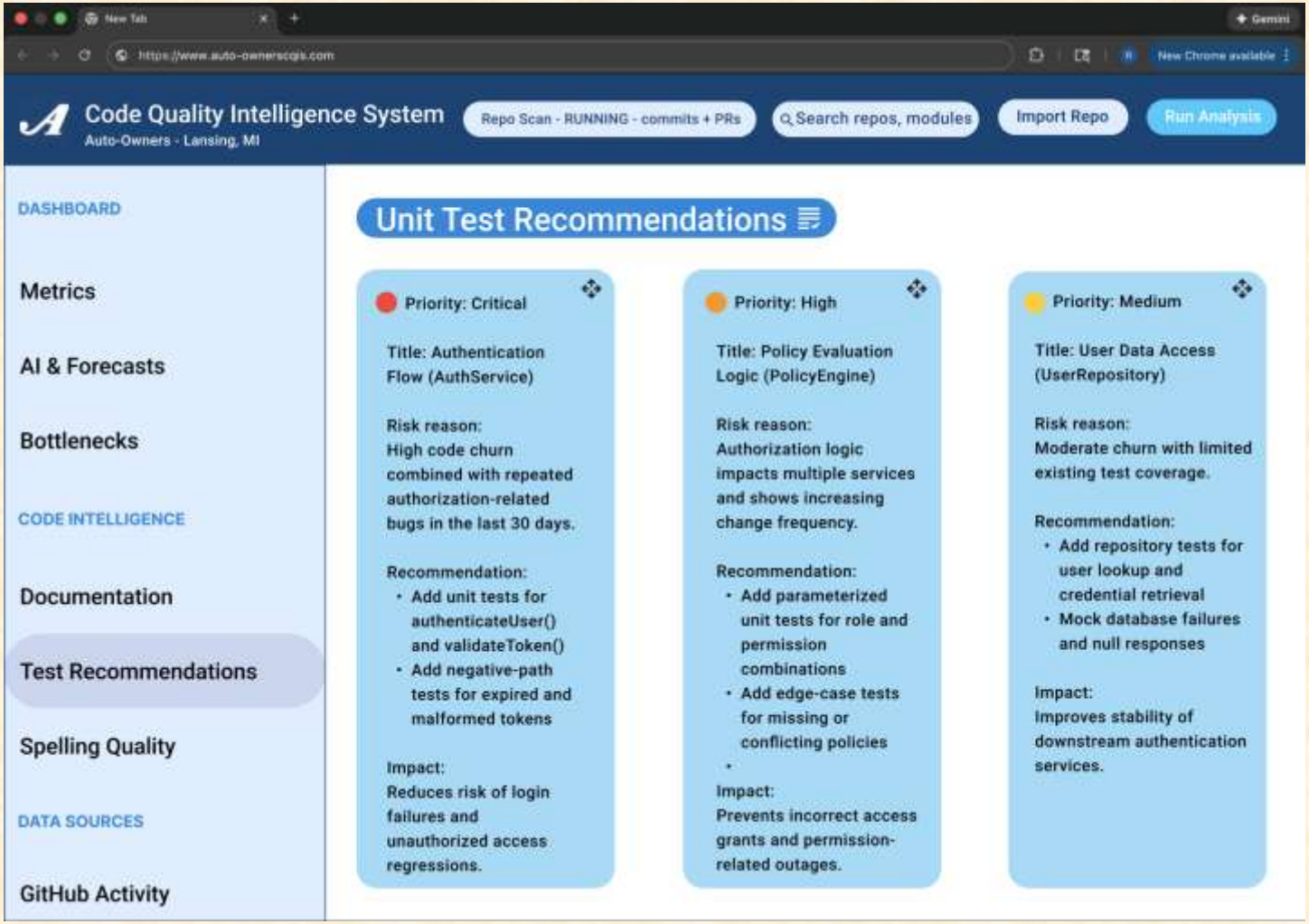
Screen Mockup: Bottlenecks page of the CQIS



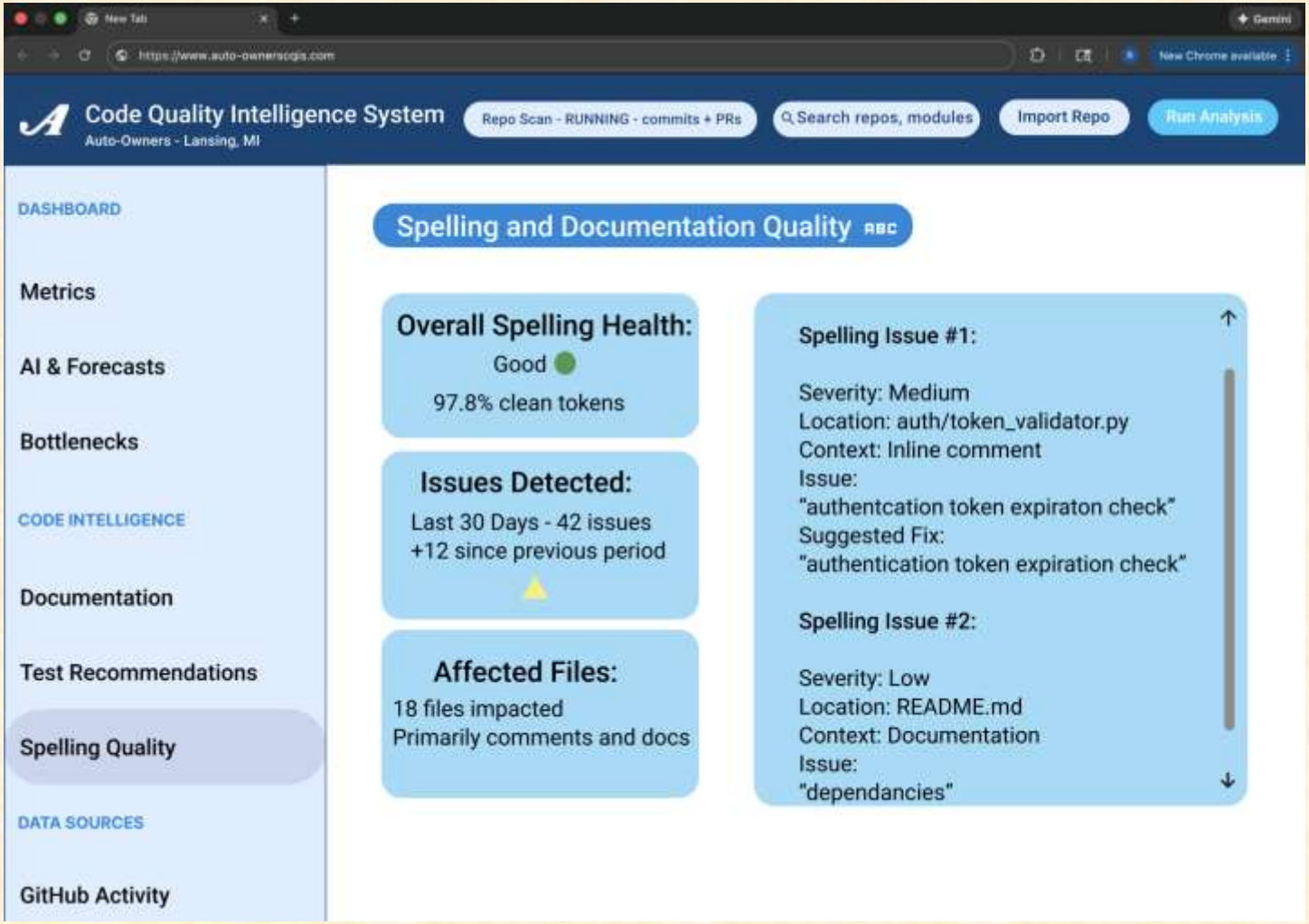
Screen Mockup: Documentation page of the CQIS



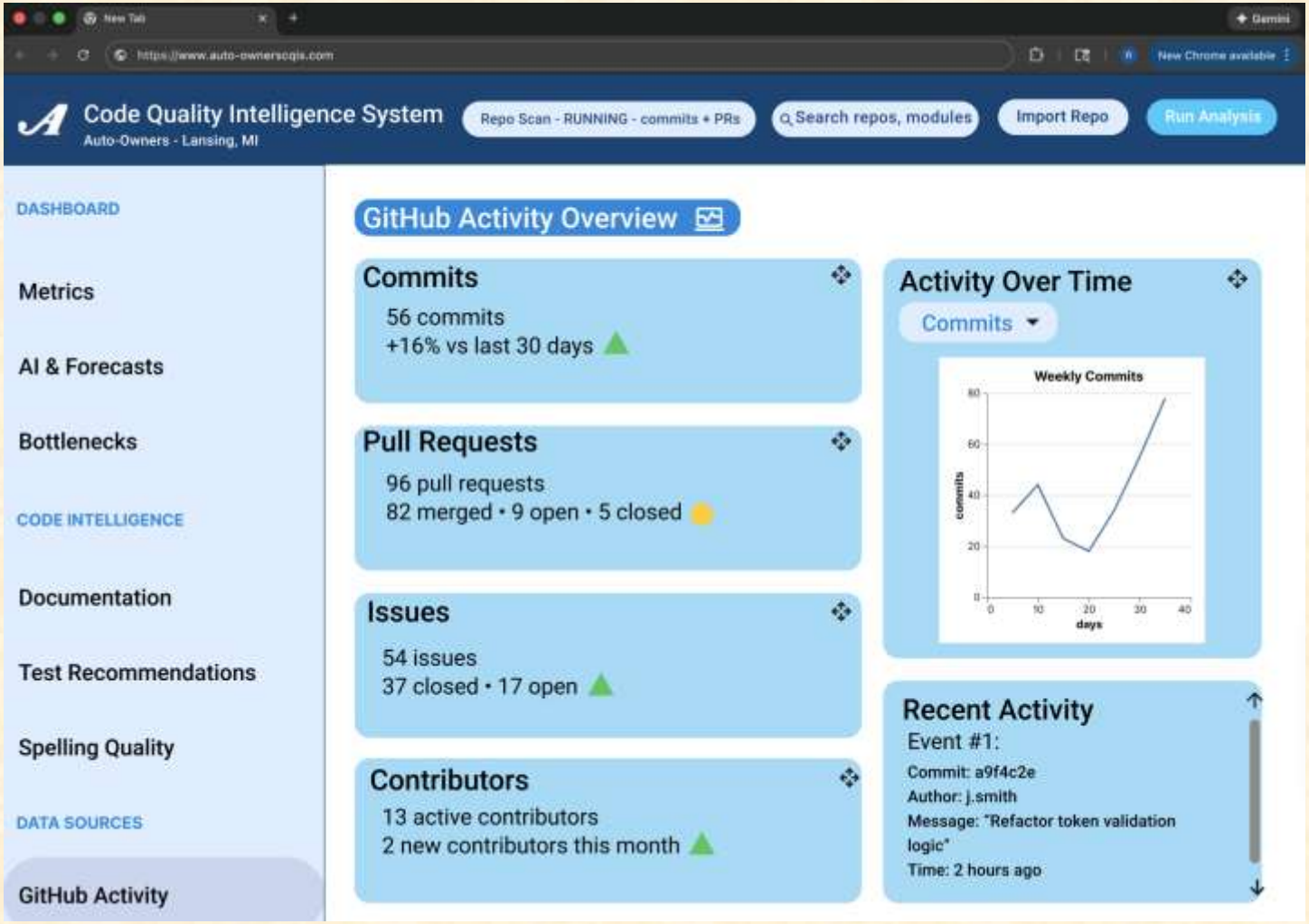
Screen Mockup: Test recommendations page of the CQIS



Screen Mockup: Spelling quality page of the CQIS



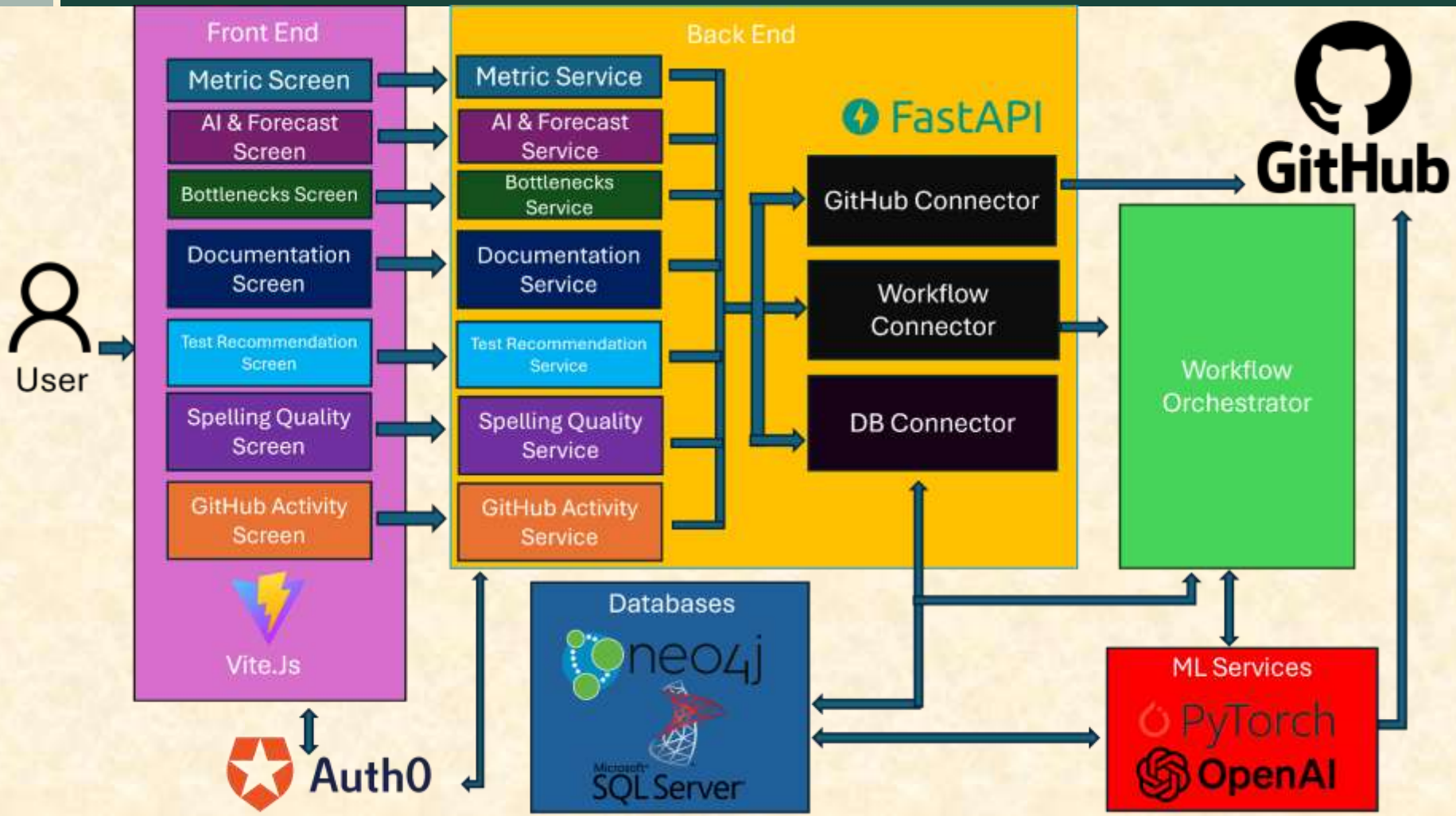
Screen Mockup: GitHub activity page of the CQIS



Project Technical Specifications

- Architectural Design
  - Vertically modular architecture
  - Asynchronous Separation Model
- Modules
  - Metric Modules - presents historical repository analytics through visual graphs
  - Insight Modules - Forecasting, bottlenecks
  - Documentation Modules – generates documents on codebase and spelling quality
  - Unit Testing Module
  - GitHub Activity Module
- Supporting Modules – Workflow Orchestrator, Authentication, ML Service, GitHub Connector, DB Connector, Workflow Connector

Project System Architecture



Project System Components

- Hardware Platforms
  - Microsoft SQL Server
  - Neo4j
  - Auth0
  - Docker
  - OpenAI API
- Software Platforms / Technologies
  - React(Vite.Js)
  - FastAPI
  - Python

Project Risks

- Risk 1
  - Description: LLMs may hallucinate when generating documentation and test cases.
  - Mitigation: Add guardrail request to the prompt (cite sources, change temperature parameter of LLM, give LLM access to previously generated test cases and documentation)
- Risk 2
  - Description: Not being able to tell if there is a bug from a PR with Git data alone
  - Mitigation: Only Flag failures when there is a clear link between a bug and a PR
- Risk 3
  - Description: Some Repos may not have enough history of data for a reliable Prediction
  - Mitigation: Show simple trends, use different models for different timeframes
- Risk 4
  - Description: Metrics Can be misrepresented and context dependent
  - Mitigation: Avoid labeling metrics as good or bad and adding clear explanations

Questions?

