

Automated Test Suite

Client

Brian Flannery, Data Verity, Inc. bflannery@dataverity.net

Background

DataVerity develops enterprise level customer relationship management and business intelligence tools. We have a current need to be able to automatically test an existing web database application. We need it to be easy for a developer to validate their own changes before suggesting merges. If we can come up with a suite of quick-running quality checks, we can make it easy for developers - even new ones - to start making changes without fear of inadvertently causing bugs.

Project Goals and Requirements:

To populate two suites with tests and quality checks, where one suite is integration testing (testing across modules to confirm pieces fit well together) and the other user interface testing. The goal is not to obtain 100% code coverage but rather to have a viable and extensible test suite that adds value to the development process.

Students can determine which test tool works best for their target suite. Because the application is a web app, tests of JavaScript or the interactive frontend overall may become another suite. Making it easy to run tests and receive meaningful feedback from them will be important. Speed will also be valuable: parallelizing tests for speed and optimizing a suite of fast-running tests (separate from slower, longer, integration/frontend acceptance tests).

Students will learn about testing and automation. They will also encounter language nuances and their impact upon testing options. They will come away able to distinguish between different categories of tests. More generally, they may learn the value of feedback cycles in software development.

Suggested Team Size:

3-4 students

Paid internships are available after Field Session for students that perform well.

Work can be done from CSM campus (connecting to our remote dev environment) or at our office in Westminster, CO, between ping pong games.

Skills/Experience for CSM Students:

- Web application testing
- JavaScript-based browser testing
- Exposure to:
 - Front-end testing, including tools like Cypress.io
 - Behavior-driven design (BDD) test dialects ("describe-it syntax")