

Software Tester

Project 1: Automated Testing Framework for a Web Application

- **Objective:** Create an automated testing framework for end-to-end testing of a web application.
- **Description:** Develop an automated testing suite using Selenium or a similar framework to test the functionality of a web application. Implement test cases for UI, functionality, and performance testing. Generate reports and analyze test results.
- **Technologies to use:** ISTQB 4, Jira, Postman, Automated Testing, Selenium, Test Case Development, Performance Testing.

Week 1: Setup and Test Case Development

Tasks:

- **Set up testing environment:** Install and configure **Selenium WebDriver** or an equivalent framework for automated testing.
- **Project structure setup:** Organize the folder structure for test scripts and reports.
- **Create sample test cases:** Develop initial test cases for basic UI functionality (e.g., navigation, button clicks, form submissions).
- **Postman integration:** Set up **Postman** for API testing and write basic test cases to verify API responses.
- **Test management integration:** Integrate **Jira** for managing test cases and tracking issues found during testing.

Deliverables:

- Functional testing environment with Selenium or an equivalent.
- Basic folder structure for organizing test scripts.
- Initial test cases developed for UI and API functionality.
- Postman set up for API testing.
- Jira integration completed for tracking test cases and bugs.

Week 2: Functional Testing and Report Generation

Tasks:

- **Develop functional test cases:** Write detailed test cases to cover key functional workflows (e.g., user login, form submission, data processing).
- **Automated test scripts:** Automate the execution of functional tests using Selenium.
- **Implement report generation:** Integrate tools like **TestNG** or **JUnit** to generate detailed test reports with test results.
- **API test cases:** Expand API testing in Postman by adding more endpoints and verifying data.
- **Performance testing setup:** Install and configure tools for performance testing (e.g., **JMeter**, **Gatling**).

Deliverables:

- Comprehensive functional test cases written and automated.

- Test reports generated for UI and functional test cases.
- Expanded API testing in Postman.
- Performance testing setup ready for execution.

Week 3: Performance and UI Testing

Tasks:

- **Develop performance test cases:** Write test cases for load and stress testing using tools like **JMeter** or **Gatling**.
- **Execute performance tests:** Run performance tests on the web application to analyze response times, load handling, and scalability.
- **UI testing:** Enhance UI test cases to cover more complex scenarios (e.g., responsive design, cross-browser compatibility).
- **Bug tracking:** Log and track issues in **Jira** discovered during the execution of performance and UI tests.

Deliverables:

- Performance tests executed and results analyzed for scalability and load performance.
- Detailed UI test cases covering different browsers and device types.
- Bug tracking in Jira with proper issue management workflows.

Week 4: Final Testing, Optimization, and Deployment

Tasks:

- **Optimize test framework:** Refactor the test framework to optimize performance and reusability (e.g., using page object models).
- **Execute full testing suite:** Run the complete suite of tests (UI, functional, API, performance) on the web application.
- **Analyze test results:** Review and analyze the results, identifying any remaining issues.
- **Generate final reports:** Create detailed reports summarizing the results of all tests, including performance metrics.
- **Documentation:** Write detailed documentation on the automated testing framework, how to use it, and how to add new test cases.
- **Deploy the framework:** Make the testing framework ready for continuous integration and deployment (CI/CD) pipelines (e.g., integration with Jenkins).

Deliverables:

- Fully optimized and functional automated testing framework.
- Complete execution of UI, functional, API, and performance tests.
- Final test reports and bug analysis.
- Comprehensive documentation for using and extending the testing framework.
- Testing framework integrated into a CI/CD pipeline for continuous testing.