**Test Plan for HCL CODING LABS**

**PROJECT NAME: HCL Coding Labs**

**PREPARED BY:** Simran Kumari (TEST ENGINEER)

**DATE:**

**REVIEWED BY:** Nageswarrao G, Srirangam Bhavani

1. **AIM/OBJECTIVE:**

- To provide a platform for individuals to enhance their coding and development skills.

- It’s helping individuals gain practical experience that can be valuable for their careers in technology and software development.

- In addition, encouraging a culture of coding and problem solving among scholars, professionals and enthusiasts.

1. **SCOPE OF TESTING:**

**a) Within The Scope:**  
- Scholar login and authentication  
- Navigation through topics, sections, and assessments  
- Progress tracking and percentage completion  
- Rank display within batches  
- Quizzes and coding assessments  
- Online proctoring for assessments  
- Time tracking feature

**b) Out of the Scope:**

- End-to-end system testing outside the coding scope.

- Hardware testing or infrastructure –related testing.

- Compatibility testing

- Backend server and database testing

- Security penetration testing

1. **TEST STRATEGY:**

**a) Levels of Testing:  
  
Unit Testing**: Individual components such as login, navigation buttons, progress tracking, and rank display will undergo unit testing to ensure their correctness and functionality.  
  
**Integration Testing**: The interaction between components, such as the connection between sections and assessments, will be tested to ensure smooth data flow and interaction.  
  
**System Testing**: End-to-end testing will be performed to validate the complete user journey, from login to assessment completion, while considering various scenarios.  
  
**b) Types of Testing**:  
  
**Functional Testing**: Each feature, such as scholar login, navigation through sections, assessment taking, and rank display, will be thoroughly tested to ensure they work as intended.  
  
**Usability Testing**: The user interface will be assessed for its user-friendliness and ease of navigation.  
  
**Performance Testing**: The responsiveness of the application, especially during peak loads, will be tested to ensure smooth user experience.  
  
**Regression Testing**: After any changes or updates, previously working functionalities will be retested to ensure new changes do not introduce regressions.  
  
**c) Test Design Techniques**:  
  
**Equivalence Partitioning**: Input fields such as login credentials and assessment answers will be tested with valid and invalid inputs.  
  
**Boundary Value Analysis**: Numeric inputs like percentage completion and time tracking will be tested with boundary values to ensure accuracy.  
  
**Decision Table Testing**: Different scenarios, such as different assessment types, will be documented and tested to validate various combinations of inputs and outputs.  
  
**d) Terminology Used**:  
  
**Topics:** Distinct areas of study available for practice.  
**Sections (Explore, Mandatory, Bridges, HOTs):** Segments within topics offering various types of content.  
**Assessments (Quizzes, Coding Assessments):** Different types of exercises available for practice.  
**Online Proctoring**: Remote monitoring during assessments.  
**Percentage Completion**: Progress tracking for each topic and section.  
**Rank**: Scholar's standing within their batch.   
**Time Tracking**: Recording the time spent using the application.  
  
**e) Configuration Management Tool**: GIT as the configuration management tool for storing project-related documents and providing access to stakeholders.  
  
**f) Area Planned for Automation**:  
  
**Regression Testing**: Automate test cases to ensure quick validation of previously tested functionalities.  
  
**Performance Testing**: Automate load testing to simulate multiple users and assess application responsiveness.  
  
**g) List of Automation Tools:** Selenium, Junit, Test-NG

**4) ENTRY AND EXIT CRITERIA**:  
  
**a) Entry Criteria**:  
  
**1. Functional Requirements**: All functional requirements for the "coding-labs" web application should be documented and approved by the stakeholders.  
  
**2. Design and UI**: The user interface design and layout should be finalized and approved.  
  
**3. Test Environment**: The testing environment should be set up and configured with the necessary browsers and devices.  
  
**4. Test Data**: Mock data for assessments, quizzes and login credentials should be available for testing purposes.  
  
**5. Test Plan**: The comprehensive test plan for the application should be reviewed and approved by the QA Lead.  
  
**b) Exit Criteria**:  
**1. Test Case Execution**: All identified test cases must be executed.  
  
**2. Defect Resolution**: Any critical defects identified during testing must be resolved and retested.  
  
**3. Functionality**: All features, sections, assessments, and tracking mechanisms should function as intended.  
  
**4. Usability**: The user interface should be intuitive and user-friendly.  
  
**5. Performance**: The application should respond quickly and efficiently during various user interactions.  
  
**6. Regression Testing**: After defect resolution, regression testing should be conducted to ensure new changes do not cause any regressions.  
  
**7. Documentation**: Test cases, test execution reports, and defect reports should be updated and finalized.  
  
**8. Approval**: The Test Lead should review the test execution reports and approve the readiness for deployment.  
  
**c) Suspension and Resumption Criteria**:  
**- Suspension**: Testing may be suspended if critical defects affect the core functionalities, preventing further testing until the issues are resolved.  
  
**- Resumption**: Testing can be resumed once the critical defects are resolved, and the application is stable for testing again.  
  
**d) Pass/Fail Criteria:**   
**- Pass Criteria**: The application must pass all functional, usability, and performance tests without any critical defects.  
**- Fail Criteria**: The application will be considered a fail if critical defects prevent core functionalities from working as intended.

**5) TEST DELIVERABLES**:   
  
**Test Case and Test Script Documents:**   
**- Purpose**: Provide a detailed outline of the test scenarios, steps, and expected results for each feature and functionality of the "coding-labs" web application.  
**- Content**: Clearly defined test cases and scripts for scholar login, navigation, progress tracking, assessments, and other features.

**Test Execution Reports:**   
**- Purpose**: Document the results of the test cases executed during various testing phases.  
**- Content**: A comprehensive report showcasing the test case ID, description, status (pass/fail), any defects found, and remarks.  
  
**Defect Reports**:   
**- Purpose**: Document and track defects found during testing, aiding developers in addressing and resolving issues.  
**- Content**: Detailed description of each defect, including steps to reproduce, severity, priority, and screenshots.   
  
**Traceability Matrix:**   
**- Purpose**: Establish a clear link between requirements and the corresponding test cases, ensuring comprehensive coverage.  
**- Content**: A matrix that maps each requirement to its associated test case(s).  
  
**Regression Test Suite:**   
**- Purpose**: Compile a set of test cases to be executed during regression testing to ensure new changes do not cause regressions.  
**- Content:** List of test cases specifically focused on retesting modified or impacted functionalities.  
  
**Performance Test Reports:**   
**- Purpose:** Document the results of performance testing, including response times, resource utilization, and scalability analysis.  
**- Content:** Performance metrics, graphs, and analysis showcasing application responsiveness and stability under various loads.

**Usability Test Reports:**  
**- Purpose:** Evaluate user-friendliness and user satisfaction with the application's interface.  
**- Content**: User feedback, observations, and recommendations to enhance the user experience.  
  
**Test Summary Report:**  
**- Purpose:** Provide an overall summary of the testing efforts, outcomes, and key findings.  
**- Content:** Overview of testing phases, pass/fail status, defect trends, and any outstanding issues.  
  
**Exit Criteria Checklist:**  
**- Purpose:** Verify that all exit criteria have been met before the application is considered ready for deployment.  
**- Content**: A checklist indicating whether each exit criterion has been fulfilled.  
**Documentation Updates:**  
**- Purpose**: Ensure that all test-related documentation is updated with the latest information and results.  
**- Content**: Revised test cases, scripts, and documents to reflect any changes during the testing process.

6**. ROLES AND RESPONSIBILITIES**:

|  |  |
| --- | --- |
| **ROLEES** | **RESPONSIBILITIES** |
| Test Manager | * Develop the overall test strategy, including test levels, types, and techniques. * Coordinate with stakeholders to understand project goals and priorities. * Allocate resources, assign tasks, and monitor progress throughout testing phases. * Review and approve the test plan and test deliverables. * Oversee risk assessment and mitigation strategies. * Ensure testing aligns with project schedules and goals. |
| Domain Expert | * Provide insights into the educational context and coding concepts for effective test design. * Collaborate with the QA team to ensure assessments align with real-world student needs. * Verify the accuracy of assessment content in line with coding curriculum. |
| Test Lead | * Develop detailed test plans, specifying testing scope, objectives, and methodologies. * Coordinate with the Test Manager to align testing efforts with project goals. * Assign test cases to team members and oversee test execution. * Monitor progress and ensure testing milestones are met. * Review and approve test case documentation and test scripts. * Report testing progress, defects, and issues to stakeholders. |
| Senior QA Tester | * Design and document detailed test cases based on requirements and specifications. * Execute test cases, record results, and report defects. * Mentor junior testers, guiding them in test case design and execution. * Participate in defect triage meetings and contribute to defect resolution. * Conduct regression testing and verify defect fixes. |
| Test Engineer | * Execute test cases as per the test plan and report defects with accurate details. * Collaborate with Senior Test Engineer to enhance testing skills. * Participate in test case reviews and provide feedback for improvement. * Contribute to test documentation, including test execution reports. |
| Automation Tester | * Design, develop, and maintain automated test scripts using selected tools. * Execute automated test scripts and analyze results. * Identify areas suitable for automation and implement automation frameworks. * Perform regular regression testing using automated scripts. * Collaborate with the QA team to integrate automation into the testing process. |
| Performance Tester | * Develop performance test plans, scenarios, and scripts. * Execute load, stress, and performance tests to identify bottlenecks and issues. * Analyze performance test results and recommend improvements. * Collaborate with development and infrastructure teams to optimize application performance. * Ensure the application's responsiveness under varying loads. |

**7) RISK AND MITIGATION**:   
  
**a) Risks and Contingency**:   
  
**1. Lack of Online Proctoring Integrity:**   
**- Risk**: The online proctoring system might face vulnerabilities leading to integrity issues during assessments.  
**- Contingency**: Implement secondary identity verification steps such as facial recognition and identity document submission.  
  
**2. Performance Issues with Multiple Users:**   
**- Risk:** Application might slow down or crash during peak usage times.  
**- Contingency:** Conduct thorough load testing to identify performance bottlenecks and optimize server resources accordingly.  
  
**3. Data Privacy Concerns:**  
**- Risk:** User data and assessment results might be vulnerable to breaches.  
**- Contingency:** Implement robust encryption methods, regular security audits, and comply with data protection regulations.  
  
**b) Risks and Mitigation:**   
  
**1. Data Privacy Concerns**:   
**- Risk**: User data, especially assessment results, may be accessed by unauthorized parties.  
**- Mitigation**: Implement strict role-based access controls and encryption to ensure data confidentiality. Regularly audit access logs for anomalies.  
  
**2. Platform Compatibility Issues**:   
**- Risk**: The application might not function as expected on various browsers and devices.  
**- Mitigation**: Conduct extensive cross-browser and cross-device testing to identify and fix compatibility issues.  
  
**3. User Experience Hurdles:**   
**- Risk:** Users might find the interface confusing or cumbersome, affecting their engagement.  
**- Mitigation**: Perform usability testing to gather user feedback and make necessary interface adjustments for a smoother experience.  
  
**4. Inaccurate Assessment Results**:   
**- Risk**: Technical glitches might lead to incorrect assessment grading.  
**- Mitigation**: Implement a comprehensive validation process to ensure accurate assessment results. Conduct thorough testing of assessment grading algorithms.  
  
**5. Loss of Time Tracking Data:**   
**- Risk**: The time tracking feature might fail, leading to inaccurate time data.  
**- Mitigation**: Regularly back up time tracking data and implement fail-safes to prevent data loss.

**6. Incomplete Progress Tracking:**   
**- Risk**: Users might not receive accurate percentage completion due to glitches.  
**- Mitigation**: Conduct thorough testing of the progress tracking mechanism to ensure it accurately updates based on user actions.

8**) SCHEDULE**:

|  |  |
| --- | --- |
| **TASK** | **DURATION** |
| Requirement Analysis | 10 Working Days  - Start Date: August 14, 2023 (Monday) - End Date: August 25, 2023 (Friday) |
| Test Planning Phase | 5 Working Days  - Start Date: August 28, 2023 (Monday) - End Date: September 1, 2023 (Friday) |
| Test Case Preparation | 10 Working Days  - Start Date: September 4, 2023 (Monday) - End Date: September 15, 2023 (Friday) |
| Test Execution | 15 Working Days  - Start Date: September 18, 2023 (Monday) - End Date: October 6, 2023 (Friday) |
| Defect Resolution and Retesting | - Ongoing during Test Execution Phase |
| Regression Testing | 7 Working Days  - Start Date: October 9, 2023 (Monday) - End Date: October 17, 2023 (Tuesday) |
| Performance Testing | 7 Working Days  - Start Date: October 18, 2023 (Wednesday) - End Date: October 26, 2023 (Thursday) |
| Usability Testing | 5 Working Days  - Start Date: October 27, 2023 (Friday) - End Date: November 2, 2023 (Thursday) |
| Documentation and Reporting | - Ongoing throughout Testing Phase |
| Test Summary and Review | 3 Working Days  - Start Date: November 3, 2023 (Friday) - End Date: November 7, 2023 (Tuesday) |

**9) HIRING AND TRAINING:**

**Testing Team Composition**:   
  
**1. Test Manager**:   
**- Experience**: 7+ years of QA experience, including leadership and managerial roles.  
**- Skills**: Test planning, strategy development, team management, stakeholder coordination.  
  
**2. Domain Expert**:   
**- Experience**: 5+ years of experience in education or coding instruction.  
**- Skills**: In-depth knowledge of coding concepts, curriculum, and assessment design.  
  
**3. Test Lead**:   
**- Experience**: 5+ years of QA experience, including leadership roles.  
**- Skills**: Test case design, execution, team coordination, reporting.  
  
**4. Senior QA Testers (2):**  
**- Experience:** 3+ years of QA experience.  
**- Skills**: Test case design, execution, defect reporting, mentorship.  
  
**5. Test Engineers (4):**  
**- Experience**: 1+ years of QA experience.  
**- Skills**: Test case execution, defect reporting, test documentation.  
  
**6. Automation Tester:**   
**- Experience**: 2+ years of experience in test automation.  
**- Skills**: Automation tools (e.g., Selenium), scripting languages, framework development.  
  
**7. Performance Tester:**   
**- Experience**: 2+ years of experience in performance testing.  
**- Skills**: Performance testing tools (e.g., JMeter), load testing, performance analysis.  
  
**Hiring Process:**  
**1. Test Manager**:   
- Recruit candidates with extensive QA experience, proven leadership in managing testing projects, and effective stakeholder management skills.  
  
**2. Domain Expert:**  
- Seek individuals with a background in coding education, curriculum design, and understanding of student needs.  
  
**3. Test Lead:**   
- Identify candidates with leadership experience, strong test planning skills, and the ability to coordinate testing efforts.  
  
**4. Senior QA Testers:**   
- Look for candidates experienced in manual testing, test case design, and mentoring junior testers.  
  
**5. Test Engineers:**   
- Consider candidates with foundational testing knowledge and a willingness to learn and grow.  
  
**6. Automation Tester:**   
- Recruit individuals with expertise in test automation, scripting languages, and framework development.  
  
**7. Performance Tester:**   
- Source candidates with a background in performance testing, load testing, and performance analysis.  
  
**Training Plan:**   
  
**1. Test Manager:**   
- Offer leadership training, focusing on project management, effective communication, and strategic planning.  
  
**2. Domain Expert:**   
- Provide an orientation to the application's educational context and align the testing efforts with real-world student needs.  
  
**3. Test Lead:**   
- Provide training on test planning, coordination, and reporting, along with mentorship skills to guide the team effectively.  
  
**4. Senior QA Testers:**   
- Provide advanced training on test case design, execution, and defect reporting, emphasizing mentorship and leadership skills.  
  
**5. Test Engineers:**   
- Conduct training on testing fundamentals, including test case execution, defect reporting, and test documentation.  
  
**6. Automation Tester:**   
- Offer hands-on training on automation tools (e.g., Selenium), scripting languages, and framework development.  
  
**7. Performance Tester:**   
- Provide training on performance testing methodologies, load testing tools (e.g., JMeter), and performance analysis techniques.

**10.) TEST ENVIRONMENT:**  
The test environment for the "coding-labs" web application is a crucial component of the testing process. It ensures that the application is thoroughly tested across various platforms, browsers, and devices to deliver a seamless user experience. The following details outline the specifics of the test environment:   
  
**Supported Browsers:**   
- Google Chrome   
- Mozilla Firefox   
- Microsoft Edge  
- Apple Safari   
  
**Supported Devices:**- Desktop Computers  
- Laptops  
- Tablets (iOS and Android)  
- Smartphones (iOS and Android)  
  
**Operating Systems:**- Windows   
- macOS  
- iOS  
- Android  
  
**Browser Settings:**  
- Disable browser extensions or ad-blockers during testing.  
- Enable cookies and JavaScript for full application functionality.  
  
**Network Conditions:**   
- Test under various network conditions, including high-speed and low-speed connections.  
  
**Test Data:**   
- Utilize mock data for assessments, quizzes and login credentials during testing to avoid affecting real user data.  
  
**Screen Resolutions:**   
- Test across various screen resolutions to ensure responsiveness and layout consistency.

**11) ASSUMPTIONS**:

* The functional requirements provided are accurate and complete.
* The test environment will remain stable throughout the testing phases.
* The development team will promptly address and fix reported defects.
* Realistic and relevant test data will be available for testing.
* Stakeholders, including the domain expert and educational advisors, will be available for clarifications.
* Communication channels between the testing team, development team, and stakeholders will remain effective.
* The selected automation tools will be compatible with the application's technology stack.
* Target users will be available for usability testing during the specified timeframe.
* The required resources for conducting performance testing, including tools and test environment, will be available.
* The testing team members possess the necessary skills and are available for the specified testing phases.

**12) APPROVAL INFORMATION**:

* Test plan approval: The Test lead will approve the test plan before execution.
* Test case approval: Test cases will be reviewed and approved by the Test lead.
* Test execution approval: Test lead's approval will be sought after test cases have been executed and reported.
* Regression test approval: The Test lead will approve the results of regression testing.
* Performance test approval: Test lead and performance tester will jointly approve performance test results.
* Usability test approval: The Test lead will approve the usability test findings.
* Test summary and review approval: The Test lead will review and approve the comprehensive test summary report.
* Deployment approval: The Test Manager and Project Manager will jointly approve deployment readiness.

**13) TEST METRICS:**

**1. Test Case Coverage:**

- **Metric**: Percentage of test cases executed out of the total planned test cases.

- **Purpose**: Measures the extent to which different areas of the application have been tested.

**2. Defect Density:**

- **Metric**: Number of defects identified per unit of code or functionality.

- **Purpose**: Indicates the application's quality and helps identify problematic areas.

**3. Defect Rejection Rate:**

- **Metric**: Percentage of reported defects rejected after review.

- **Purpose**: Assesses the accuracy of defect reports and the effectiveness of defect tracking.

**4. Test Execution Progress:**

- **Metric**: Percentage of test cases executed out of the total planned test cases over time.

- **Purpose**: Tracks the progress of test execution and helps manage timelines.

**5. Defect Closure Rate:**

- **Metric**: Percentage of reported defects that have been resolved and closed.

- **Purpose**: Reflects the efficiency of the defect resolution process.

**6. Test Pass Rate:**

- **Metric**: Percentage of test cases passed out of the total executed test cases.

- **Purpose**: Measures the application's stability and adherence to requirements.

**7. Automation Test Coverage:**

- **Metric**: Percentage of test cases covered by automated tests.

- **Purpose**: Evaluates the extent of test automation and its impact on testing efficiency.

**8. Performance Test Metrics:**

- **Metrics**: Response times, throughput, error rates, and resource utilization during performance testing.

- **Purpose**: Assesses the application's performance under different loads and stress levels.

**9. Usability Test Metrics:**

- **Metrics**: User satisfaction scores, task completion rates, and user feedback.

- **Purpose**: Evaluates the user-friendliness and effectiveness of the application's user interface.

**10. Test Efficiency Metrics:**

- **Metrics**: Test case execution time, defects detected per hour of testing.

- **Purpose**: Measures the efficiency of the testing process and identifies bottlenecks.

**11. Time Spent on Test Preparation and Execution:**

- **Metric**: Time spent on creating test cases, executing tests, and defect management.

- **Purpose**: Helps manage resource allocation and identify areas for process optimization.

**12. Code Coverage:**

- **Metric**: Percentage of code covered by test cases during testing.

- **Purpose**: Measures the effectiveness of test cases in exercising application code.