**TEST PLAN DOCUMENT**

**Learner Tracker App**

SUBMITTED BY:

SREEVIDHYA A G

**TABLE OF CONTENTS**

**1.**  **INTRODUCTION................................................................................3**

**2.** **TEST OBJECTIVES............................................................................3**

**3.** **SCOPE OF TESTING..........................................................................5**

**4.** **APPROACH .......................................................................................6**

**5.** **PASS/FAIL CRITERIA........................................................................8**

**6.** **EXIT/ENTRY CRITERIA.....................................................................9**

**7.** **TEST DELIVERABLES .....................................................................9**

**8.** **RESPONSIBILITIES..........................................................................12**

**9.** **RESOURCE PLANNING ..................................................................14**

**10.** **SCHEDULES....................................................................................15**

**11.** **RISKS...............................................................................................15**

**12.** **MITIGATION PLAN .........................................................................16**

**13.** **TEST CLOSURE..............................................................................17**

**1. INTRODUCTION**

This test plan document describes the objectives, scope, approach and focus of the testing of the **Learner Tracker App** ( <https://learnertracker.netlify.app/> ) web application. It lists the high-level features to be tested as part of the testing cycle. The objective of the test plan is to define the activities to perform the testing, define the test deliverables documents involved in testing. The document also lists the features that will not be tested. It then discusses the high-level testing schedule and the resource responsible for testing each of the features. It then briefly describes the pass/fail criteria and risks associated with the testing.

**2. TEST OBJECTIVES**

The Learner Tracker App is an online tool or application enable the Training Head to view the status of each student enrolled with ICTAK.

**1. Login Functionality**:

* Verify that the login functionality allows authorized users (Training Head and Placement Officer) to access the system securely.
* Ensure that only valid login credentials are accepted, and incorrect credentials result in appropriate error messages.
* Confirm that users are redirected to the Learner Dashboard upon successful login.

**2 Learner Dashboard**:

* Validate the functionality to add individual learners through the provided form.
* Verify the bulk addition of learners by uploading a CSV file.
* Confirm that learner details are accurately displayed on the Learner Dashboard.

1. **Placement Status Update**:

* Ensure Placement Officers can update the placement status of each student.
* Validate the dropdown menu for updating placement status, including options (Placed, Job Seeking, NOT Interested)

**4 Learner Form Details:**

* Confirm that learner details, including ID, Name, Course Name, Project, Batch, and Course Status, are accurately captured.
* Validate the dropdown functionality for Course Name, Project, Batch, and Course Status fields.

**5 Admin Controls:**

* Verify that the admin has control over Training Head and Placement Officer accounts, including the ability to add, edit, and delete user details.
* Ensure CRUD operations are functioning correctly.

**6 System Integration:**

* Validate the integration of the Learner Tracker with the database.
* Confirm accurate storage and retrieval of learner details from the database.

**7** **Security**:

* Ensure that only authenticated users have access to the system.
* Verify that sensitive information is securely stored and transmitted.

**8 Usability:**

* Confirm that the user interface is intuitive and user-friendly.
* Validate the functionality of form fields and buttons.

**9 Error Handling**:

* Test error messages for accuracy and clarity.
* Confirm that appropriate error messages are displayed for incorrect login attempts and other potential issues.

**10 Performance**:

* Test the system's performance under various loads, especially during bulk data uploads.
* Validate that the system responds within acceptable time limits.

**11 Compatibility**:

* Confirm that the application works seamlessly across different browsers and devices.
* Validate compatibility with various operating systems.

**12** **Documentation**:

* Verify the accuracy and completeness of user manuals and documentation.

**13** **Accessibility**:

* Ensure that the application is accessible to users with disabilities.

**3. SCOPE OF TESTING**

The scope of a test defines what areas of a customer's product are supposed to get tested, what functionalities to focus on, what bug types the customer is interested in, and what areas or features should not be tested by any means. The scope of this document is to test Learner Tracker App, to view and update the status of each student enrolled with ICTAK.

**Functions to be tested**

1. Verify the login functionality for Admin, Training Head and Placement Officer.
2. Test the redirection to the Learner Dashboard upon successful login.
3. Test the addition of individual learners through the provided form.
4. Verify the bulk addition of learners by uploading a CSV file.
5. Confirm that learner details are accurately displayed on the Learner Dashboard.
6. Test Placement Officer's ability to update the placement status of each student.
7. Test the capture and display of learner details, including ID, Name, Course Name, Project, Batch, and Course Status.
8. Verify that the admin has control over Training Head and Placement Officer accounts
9. Test CRUD operations for adding, editing, and deleting user details.

10. Verify that only authenticated users have access to the system.

1. .Test the user interface for intuitiveness and ease of navigation.

12. Verify the functionality of form fields and buttons.

13. Verify that appropriate error messages are displayed for incorrect login attempts and other potential issues.

14. Test the system's performance under various loads, especially during bulk data uploads.

15. Confirm that the system responds within acceptable time limits.

16. Test the application's compatibility with different browsers and devices.

17. Verify the accuracy and completeness of user manuals and documentation.

18. Test the application's accessibility for users with disabilities.

**Functions not to be tested**

No other than mentioned above

**4. APPROACH**

A test approach is the test strategy implementation of a project, defining how testing would be carried out. An approach in which the test design process is initiated as early as possible in order to find and fix the defects before the build is created. An approach in which the testing is not started until after design and coding are completed. Several different tests must be run to ensure that the **Learner Tracker App** website facilitates to view and update the status of each student enrolled with ICTAK.

**Documentation:**

* Maintain comprehensive documentation for test plans, test cases, and test results.
* Document lessons learned and any improvements for future testing efforts.

**Continuous Improvement:**

* Conduct a post-project review to analyze the testing process and identify areas for improvement.
* Use feedback to refine testing strategies and methodologies for future projects.

**Communication:**

* Maintain open communication with the development team, project managers, and stakeholders to ensure everyone is aligned on testing goals and progress.

**4.1 FUNCTIONAL TESTING**

Functional Testing checks if the website works in accordance with predetermined requirements. Functional tests comprise a variety of sub-categories:

**1. Unit Testing:** This process involves the testing of particular system components. These components are isolated from other portions and tested for their input, output, and module procedures.

**2. Integration Testing:** Integration testing procedures incorporate system components and how they perform together functionally between one another. System parts are built together forming new interfaces and these are tested to determine.

**3. System Testing:** System testing of software is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

**4. Regression testing:** Regression test is carried out to check modification in software has not caused unintended adverse side effects that affects the core functionality of the system.

**4.2 NON FUNCTIONAL TESTING**

**1. Performance Testing:** Performance testing evaluates how a certain software performs under different conditions. Performance testing is necessary to ensure that software operates at expected quality levels at all times. It checks parameters such as application output, data transfer speed, data processing speed, network bandwidth use, load-bearing capacity, memory consumption, command response times, etc.

**2. Security testing:** Security Testing is used to determine the data and resources are protected from the possible intruders. Security is the key to software testing.

**5. PASS/FAIL CRITERIA**

It defines the successful criteria of your tests in the test plan document. You can encounter three situations while executing the test plan- normal, suspension and resumption.

* 1. Suspension criteria: Any situation which impedes the ability to continue testing or value in performing testing leads to suspended testing activities.
  2. Resumption criteria: When the problem that caused the suspension had been resolved, testing activities can be resumed.
  3. Approval criteria: An item will be considered as ‘Pass’ if it meets the ‘Expected outcome’ defined in the corresponding test case.

Pass/fail criteria for this Learner Trackerapplication are:

1. 90% of test cases should pass.
2. No high priority or critical bug has been left out.

**6. EXIT/ENTRY CRITERIA**

**ENTRY CRITERIA**

* All development activities related to the **Learner Tracker App** are completed.
* Unit testing is conducted on individual components and modules of the application.
* The testing environment, including servers, databases, and necessary software, is set up and ready for testing.
* Valid and representative test data for various scenarios are prepared.
* The test plan has been reviewed and approved by relevant stakeholders.
* The testing team is trained on the features and functionalities of the resume builder application.

**EXIT CRITERIA**

* All test cases, including functional, usability, performance, and security, are executed and pass successfully.
* All identified defects are addressed, fixed, and retested. No critical or high-priority defects remain unresolved.
* User acceptance testing is conducted, and the application receives approval from end-users or stakeholders.
* Performance testing criteria, including acceptable response times and load handling, are met.
* Security testing is conducted, and the application complies with security standards and requirements.
* Usability testing is completed, and the application meets predefined usability criteria.
* All testing documentation, including test cases, test reports, and defect logs, is completed and reviewed.
* Relevant stakeholders, including product owners and project managers, provide formal approval for the completion of testing.
* Relevant stakeholders, including product owners and project managers, provide formal approval for the completion of testing.
* Test environments are cleaned up, and resources are released for other projects.
* Knowledge transfer sessions are conducted to share insights and learning from the testing process.
* A comprehensive final test report is generated, summarizing testing activities, outcomes, and recommendations

**7. TEST DELIVERABLES**

Test deliverables are provided as below:

**Before testing phase**

**1. Test Plan Document:**

An overall document outlining the testing approach, strategy, and scope.

**2. Test Cases Document:**

Detailed test cases for each requirement, covering positive and negative scenarios.

* Verify the login functionality for Admin, Training Head and Placement Officer.
* Ensure that only authorized users can log in.
* Verify the redirection to the Learner Dashboard upon successful login.
* Verify that incorrect login attempts result in appropriate error messages.
* Verify the addition of individual learners through the provided form.
* Verify the bulk addition of learners by uploading a CSV file.
* Confirm that learner details are accurately displayed on the Learner Dashboard.
* Verify Placement Officer's ability to update the placement status of each student.
* Validate the dropdown menu for updating placement status.
* Test the capture and display of learner details, including ID, Name, Course Name, Project, Batch, and Course Status.
* Validate the dropdown functionality for Course Name, Project, Batch, and Course Status fields.
* Verify that the admin has control over Training Head and Placement Officer Accounts.
* Verify CRUD operations for adding, editing, and deleting user details.
* Validate the integration of the Learner Tracker with the database.
* Confirm the accuracy of data storage and retrieval.
* Verify the user interface for intuitiveness and ease of navigation.
* Verify the functionality of form fields and buttons.
* Verify error messages for accuracy and clarity.
* Verify that appropriate error messages are displayed for incorrect login attempts and other potential issues.

**3 Test Environment Setup Document**

* Detailed instructions for setting up the testing environment.
* Specify hardware, software, browsers, and configurations.

**During testing phase**

**1. Test Approach:**

* High-level testing strategy (Agile).
* Levels of testing (unit, integration, system, user acceptance).

**2. Test Data:**

* Test data for valid login credentials for Admin, Training Head and Placement Officer.
* Data to simulate the addition of individual learners through the provided form.
* Sample data for bulk addition of learners via CSV file upload.
* Data representing to test the dropdown functionality for updating
* Data for creating, updating, and deleting Training Head and Placement Officer Accounts.
* Data representing different scenarios for CRUD operations.
* Test data to validate the integration of the Learner Tracker with the database.
* Pre-existing data in the database to simulate real-world scenarios.

**3 Traceability Matrix:**

* A matrix mapping test cases to the corresponding requirements.
* Ensure that all requirements have associated test cases.

**4. Automation Scripts**

* Automated test scripts for repetitive or regression testing.

**After testing phase**

**1. Test Summary Reports:**

* Summarize overall test results, key findings, and metrics.

**2. Defect Reports:**

* Documented reports for any defects identified during testing.
* Include details such as steps to reproduce, severity, and priority.

**3. Test Logs:**

* Capture detailed activities during the testing phase.

**4. Performance Test Results**

* Summarize results from performance testing.

**5. Automation Test Scripts and Reports**

* Store automated test scripts and summarize results.

**6. Release Notes:**

* Provide information on the release and changes.

**7. Lessons Learned Document**:

* Purpose: Capture insights and improvements for future testing.

**8. Test Sign-off:**

* Formal approval indicating the completion of testing.

**8. RESPONSIBILITIES**

**Test Manager:**

* Defining the testing activities for subordinates – testers or test engineers.
* All responsibilities of test planning.
* To check if the team has all the necessary resources to execute the testing activities.
* To check if testing is going hand in hand with the software development in all phases.
* Prepare the status report of testing activities.
* Required Interactions with customers.
* Updating project managers regularly about the progress of testing activities.

**Testers:**

* To read all the documents and understand what needs to be tested.
* Based on the information procured in the above step, decide how it is to be tested.
* Inform the test lead about what all resources will be required for software testing.
* Develop test cases and prioritize testing activities.
* Execute all the test cases and report defects, define severity and priority for each defect.
* Carry out regression testing every time when changes are made to the code to fix defects.

**Developer in Test:**

* **Writing Testable Code**: Developers should write code that is easy to test. This involves breaking down the code into modular and well-defined units, making it easier to create and execute tests.
* **Participation in Test Planning:** Developers should actively participate in test planning sessions, providing insights into critical areas that need testing.
* **Test Data Preparation:** Assist in the creation and preparation of test data that is representative of real-world scenarios.
* **Security Best Practices:** Consider security aspects during development, and address any security-related findings identified during testing.
* **Collaboration with Testers:** Work closely with testing teams to understand test cases, provide insights into the expected behavior of the code, and address issues found during testing.

**Test Administrator:**

Establish, implement, and enforce test center policies and procedures, ensuring compliance with all applicable regulations and standards. They train and supervise staff, verifying they are well-informed of all mandatory protocols regarding ID verification, security, and academic integrity.

**SQA Member:**

Work with developers, testers, and other stakeholders to troubleshoot testing failures and bugs. Communicate with project managers and other stakeholders about QA systems, activities, and progress. Help monitor performance and uncover system vulnerabilities through tools and bug tracking systems.

**10. RESOURCE PLANNING**

**System Resources:**

|  |  |  |
| --- | --- | --- |
| **SI NO** | **Resource** | **Description** |
| 1 | Computer | Personal Computer which users often use to connect with the website.At least 4 computers of specification 4GB RAM, CPU 3.4 GHz, operating system Windows 7 or above. |
| 2 | Network | Internet to simulate the real business and user environment. Minimum speed 100 mbps |
| 3 | Test Tools | Testing tool to automate the testing, simulate the user operation and generate the test results Preferred tool: Selenium, Apache JMeter(performance testing) |
| 4 | Server | Install the web application under test This includes a separate database server and web server. Database server : My SQL  Web Server: Apache HTTP Server |

**10. SCHEDULES**

|  |  |  |
| --- | --- | --- |
| **TASK** | **MEMBER** | **ESTIMATE TIME** |
| Creating test plan | Sreevidhya | 10hrs |
| Developing test case scenarios | Sreevidhya | 35hrs |
| Manual testing | Sreevidhya | 10hrs |
| Automate the test scenarios using script | Sreevidhya | 35hrs |
| Functional test and validation | Sreevidhya | 10hrs |
| Performance Testing | Sreevidhya | 20hrs |
| Github deployment | Sreevidhya | 5hrs |

**11. RISKS.**

* Misunderstandings about the expected behavior may result in missing critical test cases.
* Lack of resources, hardware, software, or network configurations may lead to incomplete testing, and issues may not be discovered until later stages.
* Testing with inaccurate or incomplete data may result in overlooking potential defects, and the application's real-world performance might not be accurately assessed.
* Additional features or modifications may introduce new test scenarios, impacting the original testing plan and schedule.
* Limited availability of skilled testers, inadequate testing tools, or a shortage of testing environments can impact the thoroughness and efficiency of testing
* Keeping test cases updated, relevant, and efficient requires ongoing effort to ensure effective regression testing.
* Challenges in implementing and maintaining test automation, such as tool limitations, script maintenance issues, or a lack of skilled automation engineers, can hinder the efficiency and effectiveness of automated testing efforts.

**12. MITIGATION PLAN**

* Starting early with static testing and document verifications and other preparation tasks. Set test priority for each test activity.
* Some extra time added for contingencies while calculating time required to finish the testing tasks.
* Conduct thorough requirements analysis, seek clarification from stakeholders, and document clear, unambiguous requirements. Implement traceability matrices to link requirements to test cases.
* Reallocating to other staff who might be free.
* Plan a training course to skill up your members.
* Proper defect management plan in place.
* Using the right tools to raise defects and mark priority for developers to fix.
* Clearly define resource requirements, allocate resources based on project needs, and monitor resource availability throughout the testing process.
* Flexible test plan to incorporate new test cases and removal of redundant scenarios.
* Establish a well-defined test environment, closely resembling the production environment. Implement automated environment setup processes and conduct regular checks for stability.
* Proper communication between testing teams in different projects and planning in advance to make sure sufficient hours are made available and share resources.
* Develop a comprehensive regression testing strategy, automate repetitive regression tests, and prioritize test cases based on critical functionality and areas of change.
* Pay a lot of attention to project planning and constantly track and measure the progress.
* Regularly conduct retrospectives, analyze testing process metrics, and implement improvements based on lessons learned.

**13. TEST CLOSURE**

The test closure for testing the project **Learner Tracker App** will be finalized once the product quality is measured against the test completion criteria. The entry criteria is that execution of test cases is complete, test reports are available and the defects reports are ready.

The criteria for test completion includes the following:

* 90 % coverage has been achieved.
* No showstoppers or critical defects found.
* The medium and low priority defects are not severe.
* All Planned test cases executed during the test process.
* All defects found have been recorded.
* All high level risks have been resolved.
* Created Bug reports, Test Plan, Test Scenario, Test case Documents.
* Test Summary report issued to test holders.

The exit criteria for this phase is preparing test closure reports and matrices later signed off by the authorities and client. The test completion report is made ready in this stage which indicates quality, measures outstanding risks and identifies level of the tested software.