### TEST SUMMARY REPORT

**-Learner Tracker**

**SUBMITTED BY:**

#### *SREEVIDHYA A G*

**CONTENTS**

1. **PURPOSE………………………………………………………. 3**
2. **APPLICATION OVERVIEW………………………………. 3**
3. **TESTING SCOPE……………………………………………… 3**
4. **METRICS…………………………………………………........ 4**
5. **TYPE OF TESTING PERFORMED………………………. 5**
6. **TEST ENVIRONMENT AND TOOLS…………………… 6**
7. **LESSONS LEARNED………………………………….……… 6**
8. **RECOMMENDATIONS…………………………………….. 7**
9. **BEST PRACTICES……………………………………………… 8**
10. **EXIT CRITERIA…………………………………………………. 8**
11. **CONCLUSION………………………………………………….. 8**

**1. PURPOSE**

This document explains various activities performed as part of testing of ‘**Learner Tracker’** application.

**2. APPLICATION OVERVIEW**

Learner Tracker application is a web based application to track the status of the learners who have enrolled to the ICTAK . The Status of the learners can be created,updated,viewed and deleted by Admin and Trainer Head. Admin have the overall control over the website with CRUD operations on Training Head & Placement Officer.Trainer Heard can Add new learner details using learner form and add bulk details of the learners by .csv file, also update and delete the details of the learners.Placement officer can only update the placement status of the learner, placement officer have no control over the CRUD operation of the application

**3. TESTING SCOPE**

###### IN SCOPE

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.

##### OUT OF SCOPE

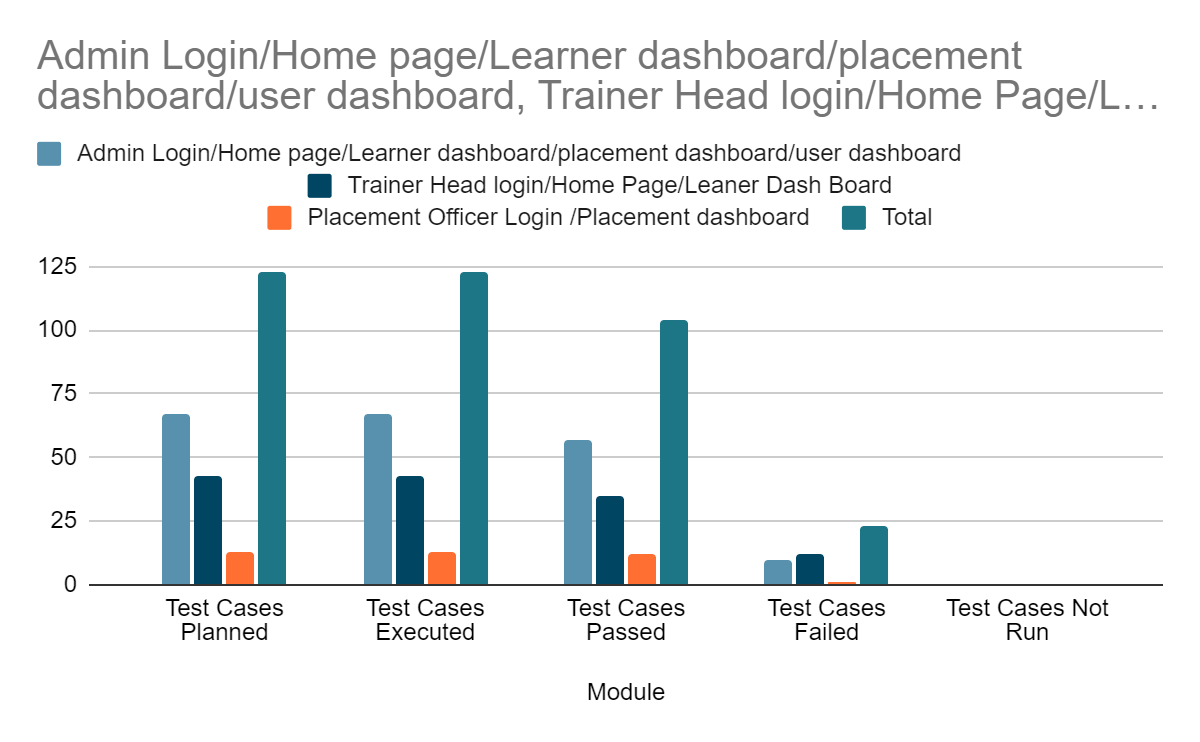
* Terms of service
* Privacy
* Security Testing
* Database Testing
* Page Number selection

**4. METRICS**

Testing matrices are calculated to understand the efficiency, effectiveness, quality and performance of the software testing performed

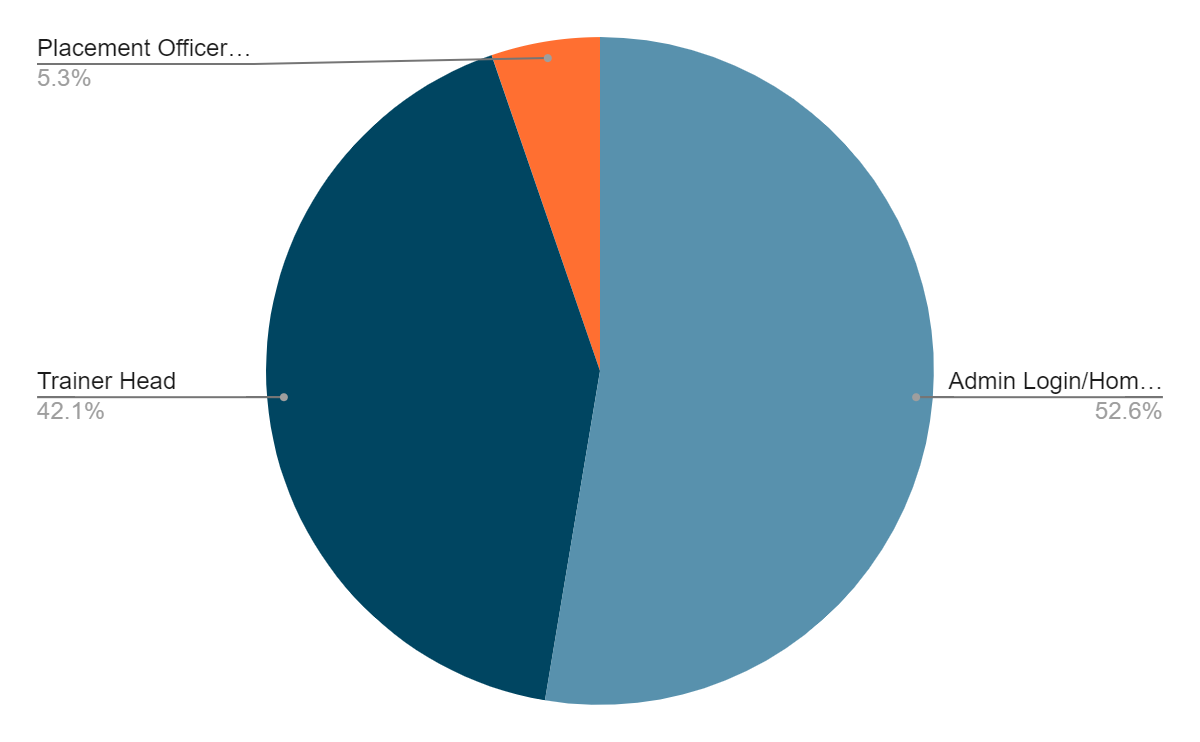
**NO. OF TEST CASES PLANED/EXECUTED/PASSED/FAILED/NOT RUN**

| **Module** | **Test Cases Planned** | **Test Cases Executed** | **Test Cases Passed** | **Test Cases Failed** | **Test Cases Not Run** |
| --- | --- | --- | --- | --- | --- |
| **Admin Login/Home page/Learner dashboard/placement dashboard/user dashboard** | **67** | **67** | **57** | **10** | **0** |
| **Trainer Head login/Home Page/Leaner Dash Board** | **43** | **43** | **35** | **12** | **0** |
| **Placement Officer Login /Placement dashboard** | **13** | **13** | **12** | **1** | **0** |
| **Total** | **123** | **123** | **104** | **23** | **0** |



**MODULE WISE BUG COUNT**

| **Admin Login/Home page/Learner dashboard/placement dashboard/user dashboard** | 10 |
| --- | --- |
| **Trainer Head login/Home Page/Learner**  **DashBoard** | 8 |
| **Placement Officer Login /Placement dashboard** | 1 |
| **Grand Total** | 19 |



**5. TYPES OF TESTING PERFORMED**

1. **FUNCTIONAL TESTING**

* **Integration Testing**: Integration testing was performed to check if the individual software components work together as expected.
* Tested the integration of different modules and components.
* Validated the flow of data and the interaction between various parts of the application.
* **System Testing**: System testing was performed to verify whether the complete software product to be delivered meets the specifications described in the requirement document.
* Verified overall functionality of the Digital Resume builder.
* Tested all features including creating, updating, downloading and sharing the link options.
* **Automation Testing**: It is a method where software tools execute pre-scripted tests on a software application, rather than manual execution. It's efficient, accurate, and reusable, significantly reducing testing time and effort. Automation tests ensure consistent coverage, accuracy, and integration with development pipelines, making them essential for modern software development practices.
* **User acceptance Testing**: User acceptance testing is a type of testing in the software development cycle that runs after every change to ensure that the change introduces no unintended breaks.
* Evaluate the user interface and overall user experience.

1. **NON-FUNCTIONAL TESTING**

In non-functional testing the web pages non-functional aspects such as performance such as speed, scalability under load, response time etc were tested using the tool JMeter.

* **Performance Testing:**Measured the Digital Resume Builder application’s performance under different load conditions using JMeter.
* Tested the response time, throughput, and resources utilization of the application.
* Identified and resolved   performance bottlenecks and generate reports.

**6. TEST ENVIRONMENT & TOOLS**

**Test Environment**

* **Operating System**

Microsoft Windows 11

* **Browser**

Microsoft Edge

Google Chrome

Mozilla Firefox

* **Hardware**

Laptop

* **Other**

High speed internet connection

**7. LESSONS LEARNED**

 The difficulties we encountered during the projects, the solutions we

implemented, and the lessons learned are outlined below.

| **S. No** | **Challenge** | **Solutions** | **Lessons Learned** |
| --- | --- | --- | --- |
| 1 | How to map each requirement to a testcase to guarantee complete test coverage | Requirement Traceability Matrix | A requirement Traceability Matrix helps to ensure that all the possible test cases corresponding to every requirement in the SRS have been planned, executed and defects if any captured. |
| 2 | Bug tracking, test case management and collaboration with team members | JIRA | JIRA is a bug tracking and project management tool which also makes collaborative learning very easy. |
| 3 | Selenium exceptions, new web elements and methods. | Explicit wait command,  JavaScript Executor interface and scrollIntoView() method | Explicit wait and JavaScript executor were used to overcome the element not clickable exception and the method scrollIntoView() to identify the object below our current location on the page. |
| 4 | Sharing automation project code and also the project deliverables with team members | GitHub | GitHub is an online software development platform used for storing, tracking, and collaborating on software projects. |

**8. RECOMMENDATIONS**

The Below conditions or activities should be completed for the test activities.

* Forget password option is not implemented, this is an important feature which should be implemented.
* Search a perticular user or learner by his/her name is not implemented
* Learner or User with same details can be done’ n ‘number of times, no alert message is displayed ,like “ **already existing user or learner**”.
* To select the page number we have to scroll horizontally to the end there is no previous or next button for the page number selection.
* If we want to update or delete particular user or learner details we have no idea on which page that particular learners or user details are shown ,and also there is no search option to find the learner with his/her learner id or name .
* Selection of items from the drop-down list should also be done using Tab key

**9. BEST PRACTICES**

The following best practices were followed during the testing phase in order to save time and ensure efficiency.

* Test plan and test cases were well documented so that the testing phases would proceed smoothly.
* Jira tool was used to track each phase of testing.
* Automation scripts were used

**10. EXIT CRITERIA**

The below conditions or activities should be completed for the test activities :

1. All test cases should be executed - **Yes**
2. Sufficient test coverage of requirements/functionalities are achieved - **Yes**
3. All the defects identified are fixed retested and closed–**Defects are corrected severity/priority and are tracked in Jira**
4. No high priority/severity defects have been left out–**Yes**
5. Test deliverables within the scope of this test plan are ready–**Yes**

**11. CONCLUSION**

As the exit criteria have been satisfied and test deliverables are ready this phase can be concluded.