TEST PLAN FOR ICTAK TRAINER MANAGEMENT SYSTEM

https://trainermanagement.herokuapp.com/

NOVEMBER 18, 2022

CREATED BY: - TESTER TEAM (VIDYA M MATHEW,
SASILEKHA SILOSHE T, SWATHY SAGAR, SNEHA N,
SRUTHI ASHISH)

Last Updated: - 18/11/2022

Table of Contents

1 OBJECTIVE

This test plan describes the testing approach and overall framework that will drive the testing of the ICTAK Trainer Management System (https://trainermanagement.herokuapp.com/). The objective of the test is to verify the functionality of the portal according to the SRS. The test will execute and verify the test scripts, identify, fix and retest all defects as per entry criteria. The application should enable the trainers to enrol in ICTAK Portal and thereby automatically generate the trainer profile. The Admin should be allowed to allocate the trainer to different courses and thereby block their calendars. The Test Plan document is created during the Planning Phase of testing the project. Its intended audience is the mentors of ICTAK. Some portions of this document may on occasion be shared with the other stakeholder whose input/approval into the testing process is needed.

1.1 QUALITY OBJECTIVE

A **primary objective** of testing "ICTAK Trainer Management System" is to: assure that the system meets the full requirements, including quality requirements (functional and non-functional requirements) and fit metrics for each quality requirement and satisfies the use case scenarios and maintain the quality of the product. At the end of the project development cycle, the user should find that the project has met or exceeded all of their expectations as detailed in the requirements. Any changes, additions, or deletions to the requirements document, Functional Specification, or Design Specification will be documented and tested at the highest level of quality allowed within the remaining time of the project and within the ability of the test team.

The **secondary objectives** of testing “ICTAK Trainer Management System” will be to: identify and expose all issues and associated risks, communicate all known issues to the project team, and ensure that all issues are addressed in an appropriate matter before release.

2 SCOPE

2.1 IN SCOPE

- Testing the overall functionality of ICTAK TRAINER MANAGEMENT SYSTEM website as per client specifications.
- Performance of the website needs to be tested.
- Both admin and trainer should be able to enrol to the system and needs to be tested.
- Sign up and login modules of admin and trainer needs to be verified.
- Trainer should be either approved or rejected by admin after enrolment and if it’s approved auto generated mail should be sent to registered Trainer’s Email ID and needs to be tested.
- Admin should be able to allocate trainer to course ID’s - DSA, FSD, or RPA as per client specification and mail should be sent to trainer’s email ID regarding the course details.
- Admin should be able to search trainer based on name, skill set, type of trainer, ICTAK courses.
- After trainer allocation their calendar needs to blocked and hence the feature needs to be verified.

2.2 OUT OF SCOPE

- Image testing
- The unexpected behaviour happened due to the website content is translated to another language using Google Translator.
- About us (As it is reviewed by Technical Writers)
- Problem related to browser extensions, ad-blockers or virus scanners (For Example, virus scanner blocking certain contents or even the execution of apps.)

3 TESTING METHODOLOGY

We follow Agile Methodology way to manage our project by breaking it into several phases. It's a process for managing a project that involves constant collaboration and working in iterations.

3.1 TYPES OF TESTING

- **GUI testing:** It's a testing that checks the Graphical User Interface of the application.
  Here we check whether the images that displayed is properly located is it override each other or not, whether the texts in the page is readable.
- **Black box testing**: It is some time called behavioural testing or Partitioning testing. This kind of testing focuses on the functional requirements of the application. Here we test whether the login, signup/enrol, acceptance and allocation to be performed by admin works properly. Also ensure the mail is correctly delivering.
- **Performance testing**: Checks the optimal time the page is loaded, Checks the operation under load. Here we test whether each pages take too much time to load and will test by giving more load simultaneously to the page.

- **Exploration testing**: It’s a type of testing where test cases are not created in advance but testers check system on the fly. Here we checks all the basic features of application is properly working or not.
- **Positive testing**: It’s a type of testing which is performed by providing the valid data sets as an input. As an example here when we provide the Admin ID and Password given in the SRS it works.
- **Negative testing**: It ensures that our application can gracefully handle invalid input or unexpected user behaviour. As an example here when we provide invalid Admin ID and Password it’s not working and warning message of invalid ID and password is showing.
- **User-Friendly Usability Testing**: This testing is conducted to ensure the application created enable users to do the things they need to do effectively and efficiently. Here manually by checking each of the features we testers approve that.
- **Compatibility testing**: It checks whether our application is capable of running on different hardware, operating systems, network environments or mobile devices. Here the application runs both in computers as well as mobile devices, different Operating Systems.
- **Cross platform testing**: It helps in identifying issues that may vary with platforms or configurations such as consistency, user interface, usability and performance issues. Here we are going to test in automation testing whether it supports in multiple platforms like chrome, Firefox etc.
- **Smoke testing**: Here we check whether the application working is stable or not. As a part we check the critical functionalities like url, signup, login works properly or not.
- **Component testing:** Also known as program or module testing which is done after unit testing. We check each modules individually, here home, login, sign-up all works properly or not.

- **Integration testing:** It checks the correctness of communication among all the modules. Here we check the data flow between the dependent modules in our application, like when the login details are entered whether it switch to next page or not.

4 APPROACH

The Test cases will be created during exploratory testing. This will cover all scenarios for requirements. The flow graph of the 'ICTAK Trainer Management System' is given below,

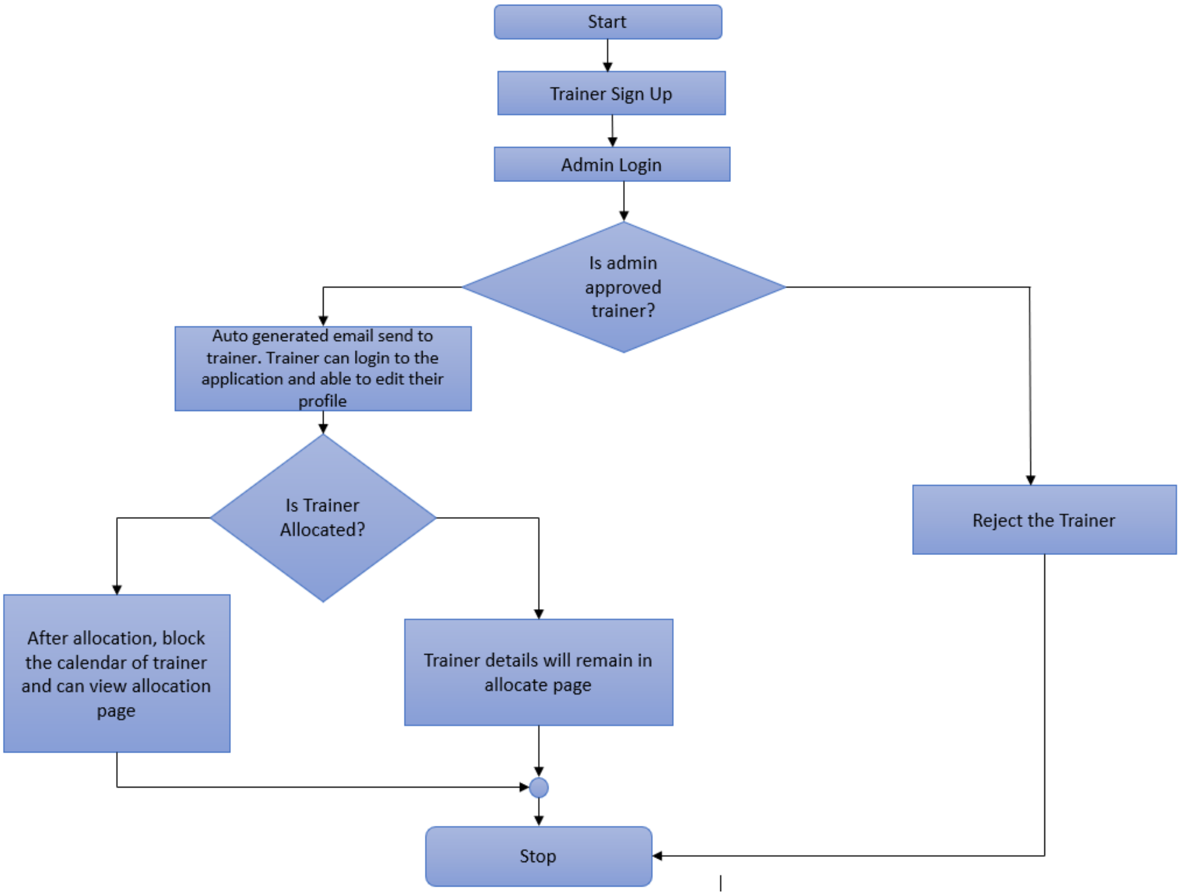

Start
Trainer Sign Up
Admin Login
Is admin approved trainer?
Auto generated email send to trainer. Trainer can login to the application and able to edit their profile
Reject the Trainer
Is Trainer Allocated?
After allocation, block the calendar of trainer and can view allocation page
Trainer details will remain in allocate page
Stop

Figure:- Work Flow Diagram

The project is using an agile approach, with weekly iterations. At the end of each week the requirements identified for that iteration will be delivered to the team and will be tested.

5 ASSUMPTIONS

- Testers will get proper documents like SRS, FRS etc.
- Testers will get proper support from development team.
- A proper communication will be there between testers and developers.
- All testing activities should sync with the test plan.
- Should identify the technical areas where assistance or training is needed.
- The necessary equipment should be there to carry out testing.
- Proper resource team should be available for test activities.

6 RISKS

Risk is the possibility of an event in the future which has negative consequences. Proper planning of risk elimination should be adopted.

- Requirement ambiguities
- Poor quality product
- Resource shortage
- Delay in the availability of project related documents
- Team member lacks the proper knowledge
- Project schedule is hectic
- Lack of proper communication among team members

7 MITIGATION PLAN

Here we take preventive measures to reduce the likelihood of the risk or reduce the impact of the risk in case it occurs.

Risk	Mitigation Plan
Requirement ambiguities	Arrange a review meeting for SRS before testing process
Poor quality product	Arrange a review meeting for SRS before testing process
Resource shortage	Request for more resources from management
Project schedule is hectic	Set Test priority to everyone in the team
Lack of proper communication among team members	Encourage and motivate each team member and inspire them for great efforts
Delay in the availability of project related documents	Arrange a review meeting among all stakeholders
Team member lacks the proper knowledge	Arrange a knowledge transfer session

8 ROLES & RESPONSIBILITIES

Role	Responsibilities
Swathy Sagar	• Prepared requirement document, RTM, Test Plan (Approach, Roles & responsibilities, Schedule, General learning), Test scenario, Bug Report and Test cases for Home Page, login page, Sign up and Admin Approval page. • Automated Sign Up and Login Page(Both admin and trainer). • Created Test Automation Execution Report for Sign up and Login page(Both admin and trainer). • Created Test Summary Report for Home Page, login page, Sign up and Admin Approval page. • Created Test Execution Report for Home Page, login page, Sign up and Admin Approval page. • Performance Testing of ICTAK website. • Created APDEX Report for ICTAK website. • Created Performance Analysis Report for ICTAK website. • Created Individual Summary Report.
Sneha N	• Prepared requirement document, RTM, Test plan (Objective, Scope, Testing methodology), Test scenario, Bug Report and Test cases for Allocate and Allocate Trainer Page • Automated View allocation Page and Log off. • Created Test Automation Execution Report for View allocation Page and Log off. • Created Test Summary Report for Allocate and Allocate Trainer Page. • Created Test Execution Report for Allocate and Allocate Trainer Page. • Performance Testing of ICTAK website. • Created APDEX Report for ICTAK website. • Created Performance Analysis Report for ICTAK website. • Created Individual Summary Report.

Sruthi Ashish	• Prepared requirement document, RTM, Test plan (Test automation, Entry/Exit Criteria and Test Environment) Test scenario, Bug Report and Test cases for View Allocation Page and Logout Page • Automated Login, Admin approval page and admin allocation Page • Created Test Automation Execution Report for Login, Admin approval page and admin allocation Page. • Created Test Summary Report for View Allocation Page and Logout Page. • Created Test Execution Report for View Allocation Page and Logout Page. • Performance Testing of ICTAK website. • Created APDEX Report for ICTAK website. • Created Performance Analysis Report for ICTAK website. • Created Individual Summary Report.
Sasilekha Siloshe T	• Prepared requirement document, RTM, Test plan (Defect tracking, deliverables, template), Test scenario, Bug Report and Test cases for Trainer-edit profile Page • Automated Trainer-edit profile Page • Created Test Automation Execution Report for Trainer-edit profile Page. • Created Test Summary Report for Trainer-edit profile Page. • Created Test Execution Report for Trainer-edit profile Page. • Performance Testing of ICTAK website. • Created APDEX Report for ICTAK website. • Created Performance Analysis Report for ICTAK website. • Created Individual Summary Report.

Vidya M Mathew	• Prepared requirement document, RTM, Test plan (Risk, Assumption, Mitigation plan), Test scenario, Bug Report and Test cases for My allocation details Page and Logout Page • Automated My allocation details Page and Logout Page • Created Test Automation Execution Report for My allocation page and Logout page. • Created Test Summary Report for My allocation page and Logout page. • Created Test Execution Report for My allocation page and Logout page. • Done performance Testing of ICTAK website. • Created APDEX Report for ICTAK website. • Created Performance Analysis Report for ICTAK website. • Created Individual Summary Report.

9 SCHEDULE

Task Name	Start	Finish	Effort	Comments
Understanding and Analysing Requirements	14/11/2022	15/11/2022	2 Days	
Test Planning	16/11/2022	18/11/2022	3 Days	
Review Meeting	19/11/2022		1 Day	
Understanding Test Scenarios	20/11/2022	21/11/2022	2 Days	
Test Case Documentation, Test Data Collection, Test Environment Setup	22/11/2022	23/11/2022	2 Days	
Review Meeting	24/11/2022		1 Day	
Manual Test Execution, Defect reporting	25/11/2022	26/11/2022	2 Days	
Review Meeting	27/11/2022		1 Day	
Automating the Test Scenarios	28/11/2022	4/12/2022	7 Days	
Review Meeting	5/12/2022		1 Days	
Test Execution, Performance Testing, Generating Reports	6/12/2022	8/12/2022	3 Days	
Review Meeting	9/12/2022		1 Day	
Final Deployment	10/12/2022	11/12/2022	2 Days	

10 DEFECT TRACKING

A **Defect in Software Testing** is a variation or deviation of the software application from end user's requirements or original business requirements. A software defect is an error in coding which causes incorrect or unexpected results from a software program which does not meet actual requirements. Testers might come across such defects while executing the test cases. Defect tracking tool used in Trainer management system is Jira. Defect Management is a systematic process to identify and fix bugs. A defect management cycle contains

- Discovery of Defect
- Defect Categorization
- Fixing of Defect by developers
- Verification by Testers
- Defect Closure
- Defect Reports at the end of project

10.1 BUG SEVERITY AND PRIORITY

Bug Severity and Priority fields are both very important for categorizing bugs and prioritizing if and when the bugs will be fixed. The bug Severity and Priority levels will be defined as outlined in the following tables below. Testing will assign a severity level to all bugs. The Test Lead will be responsible to see that a correct severity level is assigned to each bug.

The QA Lead, Development Lead and Project Manager will participate in bug review meetings to assign the priority of all currently active bugs. This meeting will be known as "Bug Triage Meetings". The QA Lead is responsible for setting up these meetings on a routine basis to address the current set of new and existing but unresolved bugs.

The type of severity that we follow are listed below,

Types of Severity	
Critical	This defect indicates complete shut-down of the process, nothing can proceed further
Major	It is a highly severe defect and collapses the system. However, certain parts of the system remain functional
Medium	It causes some undesirable behaviour, but the system is still functional
Low	It won’t cause any major break-down of the system

The type of priority that we follow are listed below,

Types of Priority	
Low	The Defect is an irritant but repair can be done once the more serious Defect has been fixed
Medium	During the normal course of the development activities defect should be resolved. It can wait until a new version is created
High	The defect must be resolved as soon as possible as it affects the system severely and cannot be used until it is fixed

11 TEST ENVIRONMENT

Test Environment Setup	
Operating System	Windows 7 & above
Browsers	Mozilla Firefox Google Chrome Internet explorer
Tools	Jira (Project management tool) Github(Collaboration Tool) Selenium Web driver(Automation Tool) Jmeter(Performance Tool) Microsoft Excel(Test Case Creation) Microsoft Excel (Defect Report) PDF (Test Reporting)

12 ENTRY/EXIT CRITERIA

12.1 ENTRY CRITERIA

- All the related documents (design and requirement) should be available which should allows the tester to analyse the system behaviour thoroughly
- Standard Software Tools must have installed successfully and working as expected.
- White box testing should be finished
- Build must be ready.
- Necessary resources must be ready
- Test data should be available
- QA resources have no requirement ambiguity

- Team clearly understood overall system functionality.
- Reviewed test scenarios, test cases and RTM.

12.2 EXIT CRITERIA

- All the test cases must be executed.
- Most of the test cases must be passed.
- Assure that there is no blocker, critical, major and minor bugs. Schedule has been achieved.
- All remaining defects are either cancelled or documented as change requests for a future release.

13 TEST AUTOMATION

- Selenium web driver is used for test automation.
- TestNG framework is used as part of test automation.
- Maven is used as a project management tool.
- Jmeter is used for performance testing.

13.1 FEATURES TO BE TESTED

- Url testing should be done.
- Login page should be tested for admin and trainer with given test data.
- Sign up feature must be tested and assure that sign up successfully message is displaying and auto generated email should be sent.
- After login trainer management system page should be tested.
- In the home page, test whether admin will be able to approve or reject the trainer.
- Verify whether admin will be able to allocate trainer to a course.
- Test email notification to trainer feature is working perfectly.

- Test the view allocation link.
- Logout page should be tested.
- Verify after successful trainer login he/she can view trainer profile and course information.

13.2 FEATURES NOT TO BE TESTED

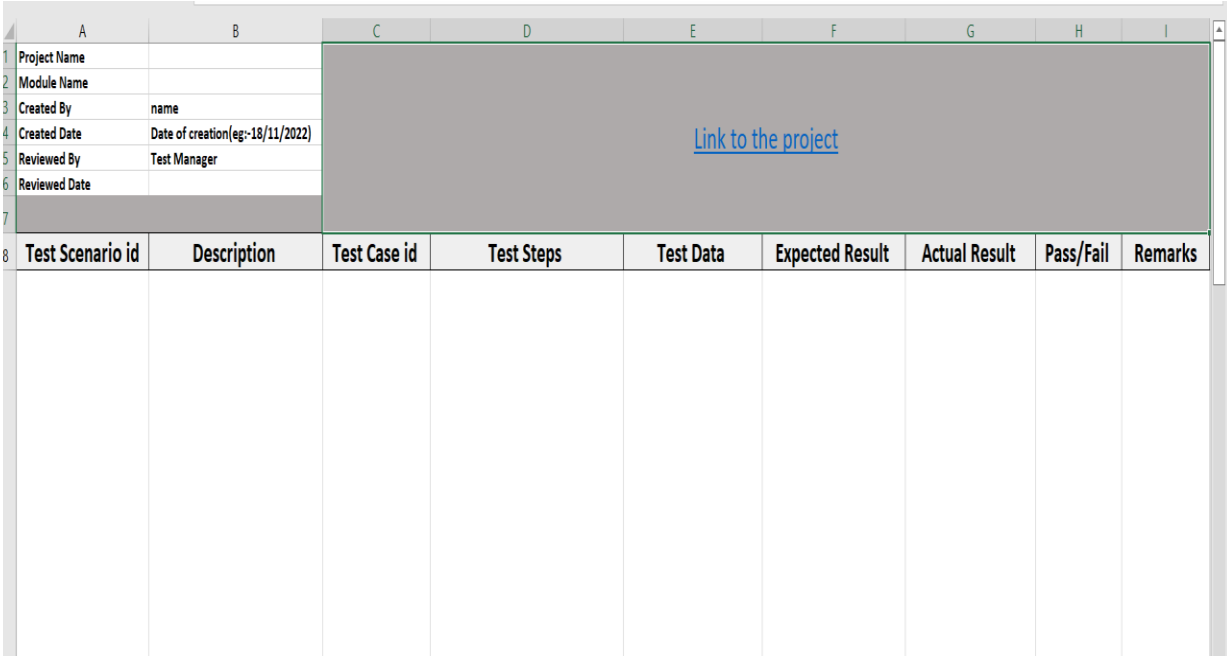
- Content testing or about us is out of scope as it’s been reviewed by technical writers.
- Problems related to browser extensions, ad-blockers or virus scanners are out of scope. For example, blocking certain contents or even the execution of application.
- Website content should not be changed. For example, if someone uses Google translator service for translation it may lead to unexpected behaviour.

14 DELIVARABLES

1. Requirement Document
2. Test scenario
3. Test Cases
4. Test Execution Report
5. RTM
6. Defect report
7. Test Automation Execution Report
8. Performance Analysis Report
9. Test Summary Report

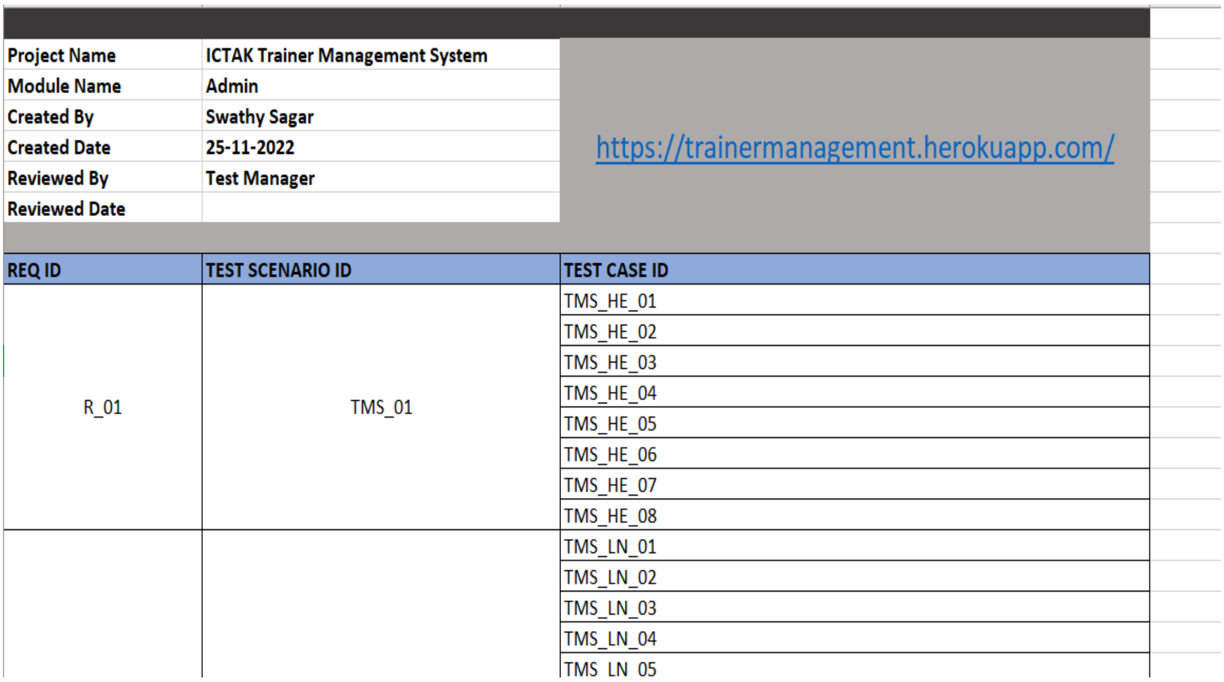
15 TEMPLATES

The templates used for creating test case document is given below,



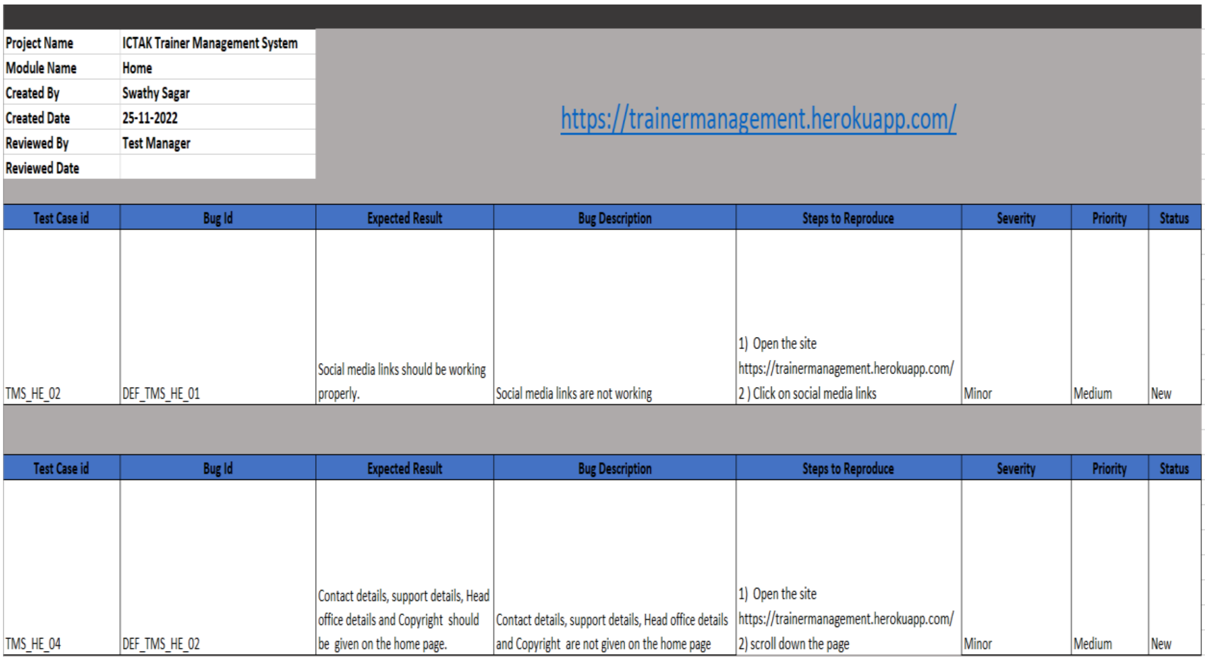
	A	B	C	D	E	F	G	H	I
1	Project Name		Link to the project						
2	Module Name								
3	Created By	name							
4	Created Date	Date of creation(eg:-18/11/2022)							
5	Reviewed By	Test Manager							
6	Reviewed Date								
7									
8	Test Scenario id	Description	Test Case id	Test Steps	Test Data	Expected Result	Actual Result	Pass/Fail	Remarks

The template used for RTM document is given below,



Project Name	ICTAK Trainer Management System	https://trainermanagement.herokuapp.com/
Module Name	Admin	
Created By	Swathy Sagar	
Created Date	25-11-2022	
Reviewed By	Test Manager	
Reviewed Date		
REQ ID	TEST SCENARIO ID	TEST CASE ID
R_01	TMS_01	TMS_HE_01
		TMS_HE_02
		TMS_HE_03
		TMS_HE_04
		TMS_HE_05
		TMS_HE_06
		TMS_HE_07
		TMS_HE_08
		TMS_LN_01
		TMS_LN_02
		TMS_LN_03
		TMS_LN_04
		TMS_LN_05

The template used for Defect Report is given below,



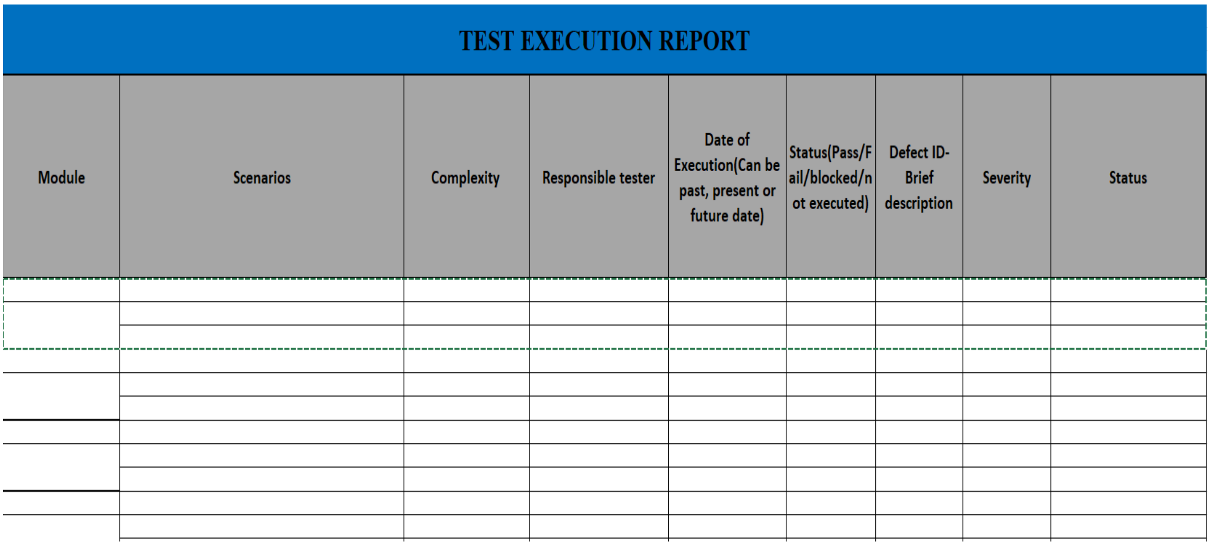
Project Name	ICTAK Trainer Management System
Module Name	Home
Created By	Swathy Sagar
Created Date	25-11-2022
Reviewed By	Test Manager
Reviewed Date	

https://trainermanagement.herokuapp.com/

Test Case id	Bug Id	Expected Result	Bug Description	Steps to Reproduce	Severity	Priority	Status
TMS_HE_02	DEF_TMS_HE_01	Social media links should be working properly.	Social media links are not working	1) Open the site https://trainermanagement.herokuapp.com/ 2) Click on social media links	Minor	Medium	New

Test Case id	Bug Id	Expected Result	Bug Description	Steps to Reproduce	Severity	Priority	Status
TMS_HE_04	DEF_TMS_HE_02	Contact details, support details, Head office details and Copyright should be given on the home page.	Contact details, support details, Head office details and Copyright are not given on the home page	1) Open the site https://trainermanagement.herokuapp.com/ 2) scroll down the page	Minor	Medium	New

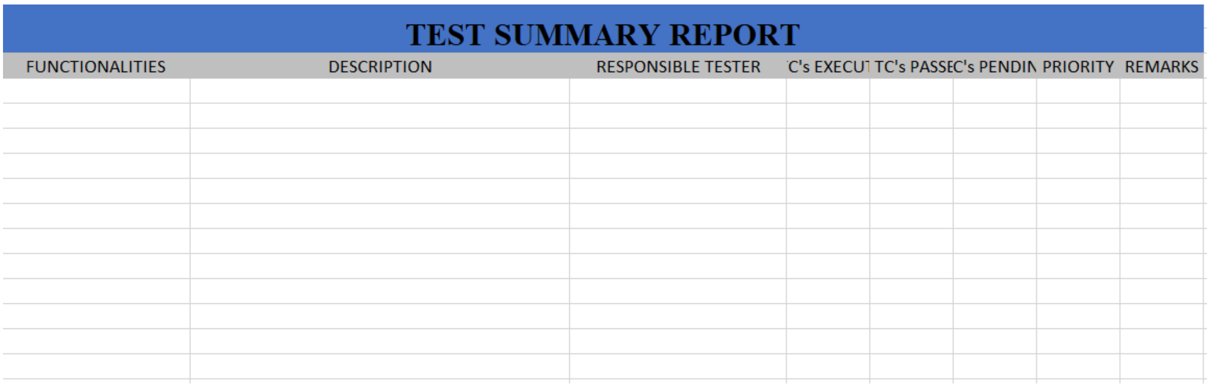
Template used for Test Execution Report is given below,



TEST EXECUTION REPORT

Module	Scenarios	Complexity	Responsible tester	Date of Execution(Can be past, present or future date)	Status(Pass/Fail/blocked/not executed)	Defect ID- Brief description	Severity	Status

Template used for Test Summary Report is given below,



TEST SUMMARY REPORT

FUNCTIONALITIES	DESCRIPTION	RESPONSIBLE TESTER	C's EXECUT	TC's PASSE	C's PENDIN	PRIORITY	REMARKS

16 GENERAL LEARNING

- Learned more about action class and waits in selenium.
- Requirement document structure.
- Requirement traceability matrix.
- Defect Report contents.
- Learned GitHub Integration.
- Learned Advanced Performance Analysis Report contents from Google.
- Learned test execution report contents from Google.
- Learned test summary report contents from Google.