REVIEW -3

WECARE

Software Testing

Tanishq, Dhruv Sood
19BIT0275, 19BIT0234

CONTENTS

PROBLEM STATEMENT

There are many NGOs about which people are unaware, due to which they don't receive donations. The people who wish to donate, are unable to find a proper NGO. They are unable to read proper articles, blogs, and information regarding social services. If people wish to donate daily required items such as hand sanitizers, masks, etc. then there's no option available for that too.

In the present system people need to search for various NGOs and then go through their separate information in the about us section, and then check if they are genuine or not. Then they decide to volunteer or donate.

If a person wishes to read some blogs or write some blogs, then he needs to search individually for them, and for writing, he would have to create his blogpage.

In the present system, if a person wishes to buy necessary items for the needy people of the NGO, then they would have to physically purchase all the items and then go to the NGO of their choice for giveaway.

When purchasing all of the items proper credentials are to be taken and no misuse of given information should be done.

In present system, if a person wishes to check weather of a certain part of the world, the he has to search that on google.

The donations which are made physically have very low reliability and no transparency, which could be used for corruption or unnecessary means.

There should be a website which can be accessed by everyone from anywhere, having number of NGOs together. So when a user visits he/she could check all the details regarding that NGO, and decide accordingly if he/she wants to donate or volunteer in that NGO.

In the same system, there should be blogs available for awareness,

which people could read, and develop a sense of social service from them. And if a person wishes to create a blog of his her choice, then they could easily create.

In case a person doesn't want to donate money, instead he/she wishes to donate utilities then there should be option for that too, where a person ca choose and add to cart certain number of items that he wishes to donate in the NGO, then he could easily checkout and donate.

A weather API attached with the website would help users to identify whats te current weather of the city in which the NGOs are operating. So that user can get idea what kind of stuff he/she can donate.

All these features should be accessible from the homepage, where all the navigating links are present. This will enable user to easily look through whatever he wishes to do and do the good deed.

SOFTWARE REQUIREMENT SPECIFICATION

1. INTRODUCTION

1.1. PURPOSE

The purpose of this document is to present a detailed description of the NGO. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

1.2. SCOPE

The software system will provide its users who wish to help the cause. This system is aimed for efficient interactivity of the users and various NGOs.

Any user can use this website to donate for an NGO, view the work they do, volunteer, shop for utilities, view weather. They can login and signup too.

The users can view different products for display and purchase them, there is a checkout page for taking all the credentials required for purchasing the product.

1.3. DEFINITIONS, ACRONYMS AND ABBREVITIONS

CFD: – Context Flow Diagram

DFD: – Data Flow Diagram

IDE: – Integrated Development Environment

SQL: – Structured Query Language

SRS: – Software Requirement Specification.

GUI: - Graphical User Interface

1.4. REFERENCES

i. Software Engineering by K.K.Aggrawal, Singh, Yogesh.

ii. Ian Somerville, Software Engineering, Third Edition

1.5. OVERVIEW

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document iswritten primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

2. OVERALL DESCRIPTION

2.1. PROJECT PROSPECTIVE

The system will consist of two parts: one client side and one server side.

This application is for users who wish to do social service. And this website is available to everyone via internet.

The application will be a platform independent developed in HTML, CSS, JavaScript, NodeJS, AngularJS and MongoDB. And can be operated using any browser.

2.1.1. SOFTWARE INTERFACES

Operating System: Windows XP or higher

Front End: HTML, CSS, JavaScript, AngularJS, Bootstrap

Back End: MongoDB

2.1.2. HARDWARE

INTERFACES **Processor:**

Pentium IV and above**RAM:**

512 MB or above

Hard Disk: 40 GB or above

Input Devices: Keyboard and Mouse

Output Devices: Monitor

2.1.3. MEMORY CONSTRAINTS

The software is expected to use not more than 150 MB of RAM and 0.75 GB of internal storage for the server client application. However, a database server to store all the patient detail is required as per the number of patients registered in the hospital.

2.1.4. OPERATIONS

This product will not cover any automated housekeeping aspects of database. The DBA at client site will be manually deleting old/ non required data. Database backup and recovery will also have to be handled by DBA.

2.2. PRODUCT FUNCTIONS

The system will allow access only to any user with specific roles who wish to do social service. Depending upon the user's intent, he/she will be able to access only specific modules of the system.

A summary of the major functions that the software will perform:

- A login facility for enabling users to authorize themselves. It is necessary to authorize themselves to add a new blog.
- After user sign ups he can add a blog and can donate money.
- In the blog section, there are three sections
- It is necessary for a user to login before uploading a new blog and donating money.
- Only blogs that have been uploaded in past can be seen by user without logging in.
- There is a products page from which users can buy products which are in high demand in this covid-19 situation in very cheap price made by these NGOs .
- There is a weather API in the website through which a user can check the real time weather of any city.
- When a user enters the website, he/she can view different NGOs and read about them, what work they do and volunteer or donate accordingly.

2.3. USER CHARACTERISTICS

Web Portal User:

Education Level: Basic understanding of English

Technical Expertise: Should be a middle level individual with basic knowledgeon using web portal.

2.4. DESIGN & IMPLEMENTATION CONSTRAINTS

- **Operating System:** The Operating System shall be Windows.
- **Web Based:** The system shall be a web based application.
- **Maintenance:** The system shall be networked and a network admin hasto take care of the maintenance issues.

2.5. ASSUMPTIONS & DEPENDENCIES

- It is assumed that the NGOs are currently in working conditions and no money is being misused.
- An updated version of the system with additional functionalities shall bereleased within 6 months after the release of the first version.

3. SPECIFIC REQUIREMENTS

3.1. EXTERNAL INTERFACES

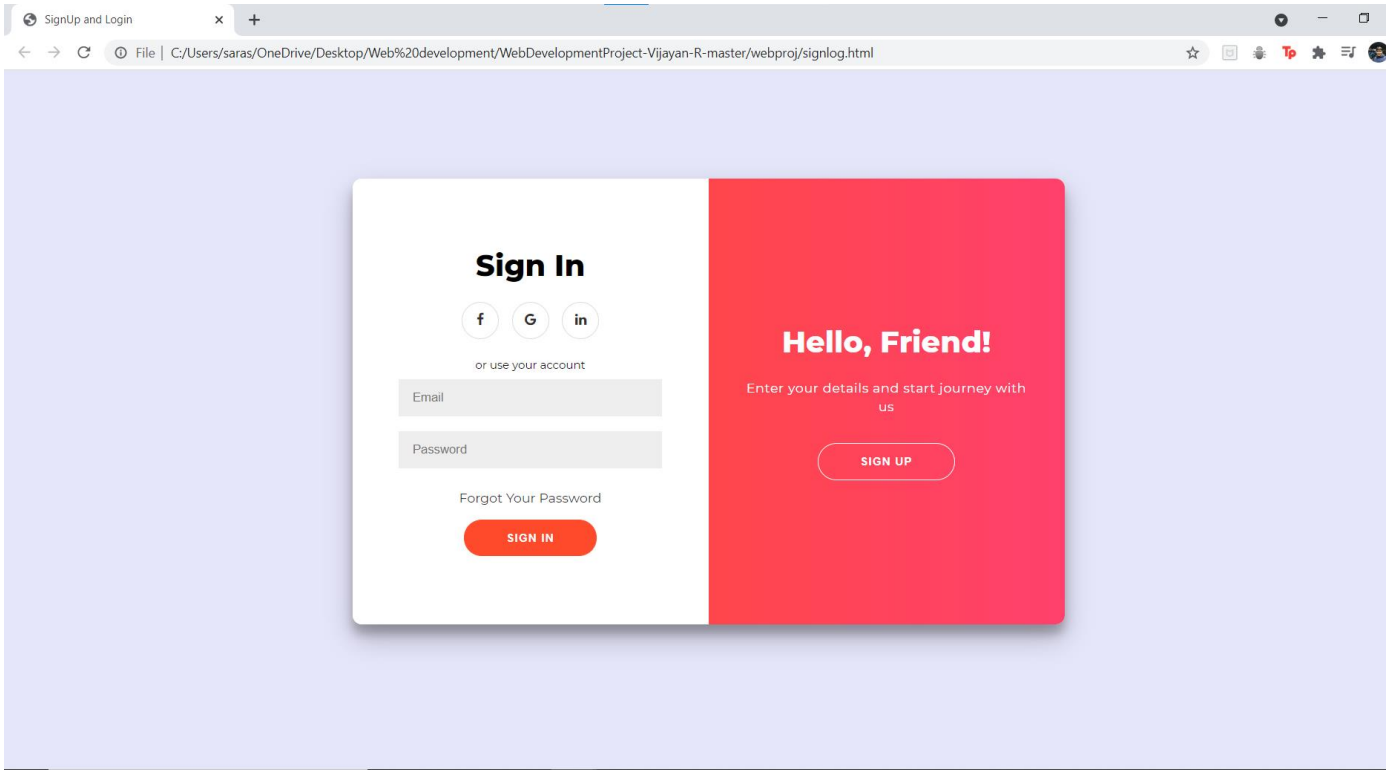
3.1.1. COMMUNICATION INTERFACES

The website is connected to a database which stores all the information, regarding login signup details. When people are making donations their information shall be recorded.

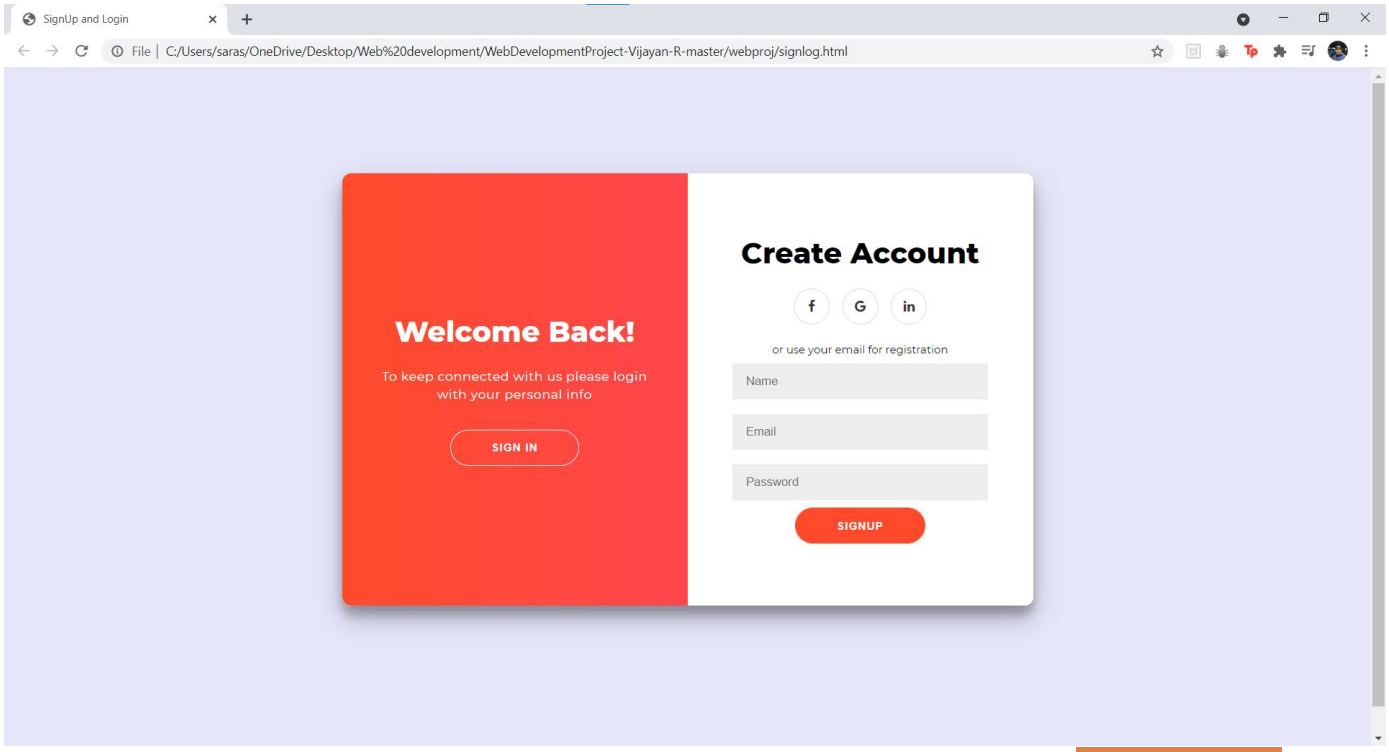
Blogs can be written with title, snippet, body and option to delete. All this shall be saved in database. And user would be able to read other blogs too.

3.1.2. USER INTERFACE

Users can login/signup to write blogs. For donations various inputs shall be taken.



The user can access different sections using the buttons on top of the application. Each button takes them to different functionalities of the system.



3.2. PERFOMANCE REQUIREMENTS

The system shall use minimal amount of internal memory and most of the data shall be stored in the server so that it can be retrieved by multiplesystems.

The web portal for the patients should be responsive and shall run on mobile platform too.

3.3. LOGICAL DATABASE REQUIREMENTS

The proposed information system contains the following data tables in its database collection.

i. Blogs details Table

ii. Donation Details Table

iii. Signup / Login Details Table

3.4. FUNCTIONAL REQUIREMENTS

- **Login Module**

  Anyone who uses the system has to login to the system and this module records only user ID and password.

- **Home Page Module**

  This page contains various navigating buttons to move to different sections of the website. Where about us, aim, and there reach has been displayed. In the footer section there is a contact us form.

- **About other NGOs Module**

  Here people can read about various NGOs and watch their videos. It contains two buttons, Volunteer and donate.

- **BLOG Module**

  The blog module contains various blogs written by users and there is option to write their own blogs.

- **Product Module**

  User can purchase different types of products and donate it to the NGOs. There is option to add the product to cart.

- **Billing Module**

  Here user enters his/her credentials and checkout purchasing all the products added to the cart.

- **Weather checker Module**

  Here user can check weather of different parts of the world.

- **Donation Module**

  Here user enter all his details required to make donation.

3.5. SOFTWARE SYSTEM ATTRIBUTES

3.5.1. PERFOMANCE

- **Response Time:**
  The system shall give responses in less than 1 second after checking thepatients information.
- **Capacity:**

  The system must support 1000 people at a time.

- **User-Interface:**

  The user-interface screen shall respond within 5 seconds.

- **Conformity:**

  The system must conform to Microsoft Accessibility

3.5.2. SECURITY

- **Logon ID:**
  Any user who uses the system shall have a Unique patient ID andPassword.
- **Modification:**
  Any modification (insert, delete or update) for the database shall be synchronized and only by the administrator in the ward.
- **Administrators' Rights:**

Administrators shall be able to view and modify all information in it.

3.5.3. RELIABILITY

How general the form generation language is simplified vs. functionality ofthe form language

3.5.4. AVAILABILITY

The system shall be available all the time.

3.5.5. SAFETY

Humans are error-prone, but the negative effect of common errors shouldbe limited.

e.g., users should realize that a given command will delete data, and be askedto confirm their intent or have the option to undo.

3.5.6. SOFTWARE QUALITY

Good Quality of the framework shall produce robust, bug free softwarewhich contains all necessary requirements Customer satisfaction.

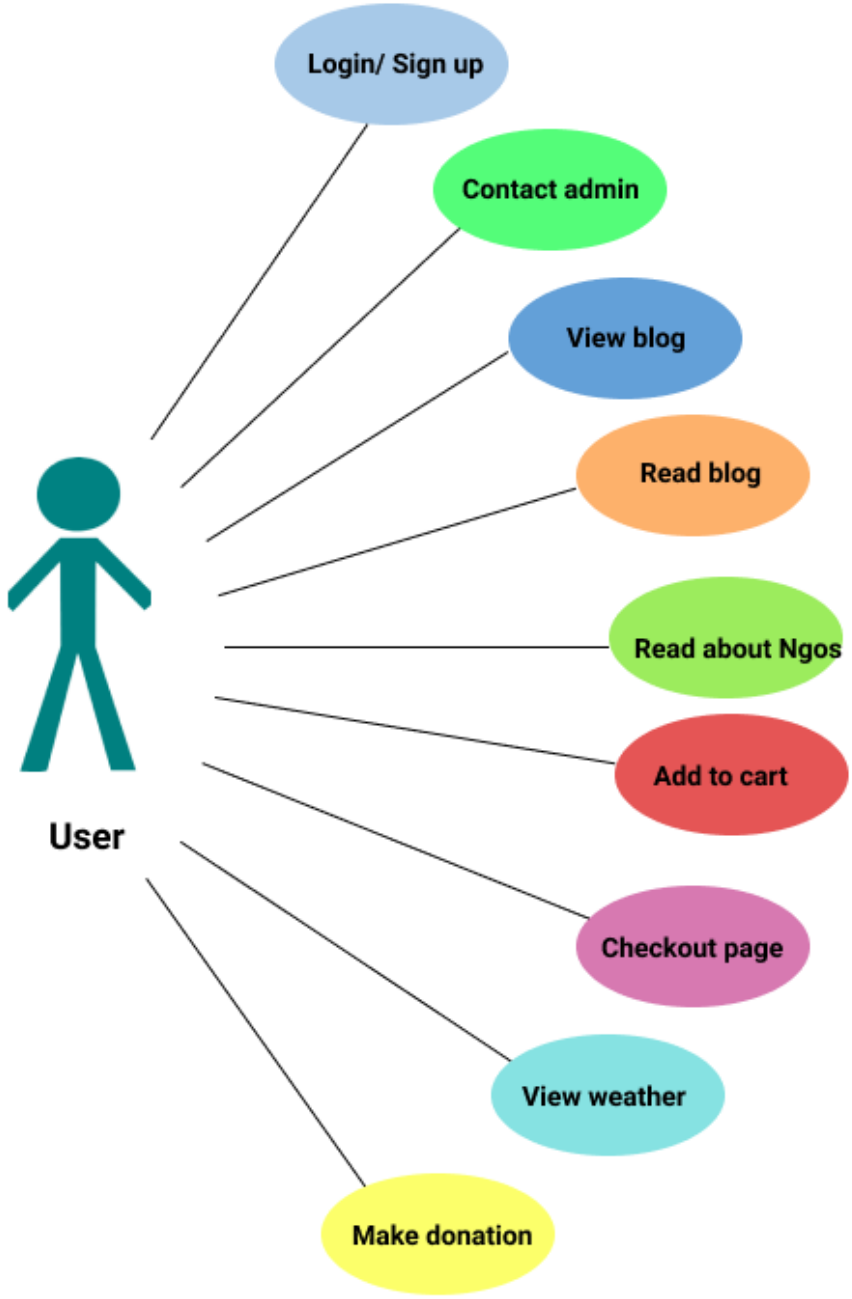
3.5.7. REUSABILITY

The software shall have simple and independent code modules that can bereused.

3.5.8. MAINTAINABILITY

- **Backup:** The system shall provide the capability to back-up the Data.
- **Errors:** The system shall keep a log of all the errors.

3.6. USE CASES


Login/ Sign up
Contact admin
View blog
Read blog
Read about Ngos
Add to cart
Checkout page
View weather
Make donation
User

USECASE DESCRIPTION:

1. LOGIN

1.1. BRIEF DESCRIPTION:

This use case describes how the user logs into the WECARE website

1.2. ACTORS:

The following actors are involved in the use case:

- User

1.3. FLOW OF EVENTS

1.3.1. BASIC FLOW:

This use case starts when the actor research to login to the website

- The system asks to actor to enter the name, password and login if new user then system asks to sign up.
- Then the system validates and logs the actor into the system.

1.3.2. ALTERNATIVE FLOW:

If in the login page the user enters invalid username and password, thenthe system should show the error message.

1.4. SPECIAL REQUIREMENT:

(No special requirement)

1.5. PRE CONDITION:

All users must have a user account i.e. user id, password the role createdfor then in the system through administrator, prior to executing this use case.

1.6. POST CONDITION:

If the use case was successful, the actor is logged info system; if not

the system, state is unchanged.

2. Contact Admin

2.1. BRIEF DESCRIPTION:

This use case describes how the user contacts the admin for any type of help.

2.2. ACTORS:

The following actors are involved in the use case:

- User

2.3. FLOW OF EVENTS

2.3.1. BASIC FLOW:

This use case starts when the user wants to contact the admin of the website

2.3.2. ALTERNATIVE FLOW:

If in the page-required fields like mail id, message are not entered then the system shows an error page.

2.4. SPECIAL REQUIREMENT:

(No special requirement)

2.5. PRE CONDITION:

All users must have a user account i.e. user id, password.

2.6. POST CONDITION:

If the use case was successful, a message is displayed saying you will be contacted soon.

3. View Blog:

3.1. BRIEF DESCRIPTION:

This use case describes how the user can view the blogs.

3.2. ACTORS:

The following actors are involved in the use case:

- User

3.3. FLOW OF EVENT

3.3.1. BASIC FLOW:

This use case starts when the actor wants to read the pre written blogs on the website.

3.3.2. ALTERNTIVE FLOW:

(No alternative flow)

3.4. SPECIAL REQUIREMENT

(No special requirement)

3.5. PRE CONDITION

All users must have a user account i.e. user id, password.

3.6. POST CONDITION

If the use case was successful, then the actor should able to see the blogs

4. Add blogs

4.1. BRIEF DESCRIPTION:

This use case describes how if the actor has to add more blogs he can add to the website.

4.2. ACTORS:

The following actors are involved in the use case:

- User

4.3. FLOW OF EVENT

4.3.1. BASIC FLOW:

This use case starts when the actor wants to add blogs to the website.
He should go to the blog section and click on add blog on top right corner in the website.

4.3.2. ALTERNTIVE FLOW:

(No alternative flow)

4.4. SPECIAL REQUIREMENT:

(No special requirement)

4.5. PRE CONDITION:

All users must have a user account i.e. user id, password.

4.6. POST CONDITION:

If the use case was successful, the blog should be added to the list.

5. Read about Ngos

5.1. BRIEF DESCRIPTION:

This use case describes how actor can read more about the NGOs.

5.2. ACTORS:

The following actors are involved in the use case:

- User

5.3. FLOW OF EVENT

5.3.1. BASIC FLOW:

This use case starts when the actor wants to view about the NGOs. He should navigate to the NGO page for more information.

5.3.2. ALTERNTIVE FLOW:

(No alternative flow)

5.4. SPECIAL REQUIREMENT:

(No special requirement)

5.5. PRE CONDITION:

All the users must have a user account i.e. user id, password.

5.6. POST CONDITION:

If the use case was successful, the actor can learn more about the NGOs.

6. Add to cart

6.1. BRIEF DESCRIPTION:

This use case describes how the actor can add the products to the cart.

6.2. ACTORS:

The following actors are involved in the use case:

- User

6.3. FLOW OF EVENT

6.3.1. BASIC FLOW:

This use case starts when the actor wants to buy stuff from the website. He/ she should go to products page.

6.3.2. ALTERNTIVE FLOW:

(No alternative flow)

6.4. SPECIAL REQUIREMENT:

(No special requirement)

6.5. PRE CONDITION:

All users must have a user account i.e. user id, password.

6.6. POST CONDITION:

If the use case was successful, the actor can easily go to the checkout page for the payment.

7. Checkout Page:

7.1. BRIEF DESCRIPTION:

This use case works for payment of the products.

7.2. ACTORS:

The following actors are involved in the use case:

- User

7.3. FLOW OF EVENT

7.3.1. BASIC FLOW:

This use case starts when the actor add some products in the cart and now want to generate bill for it.

The use case will ask the user to enter all details of credit/debit card for the payment information.

7.3.2. ALTERNTIVE FLOW:

(No special requirement)

7.4. SPECIAL REQUIREMENT:

(No special requirement)

7.5. PRE CONDITION:

All the users must have a user account i.e. user id, password.

7.6. POST CONDITION:

If the use case was successful, payment successful, you will receive message saying the product will be delivered within 2-3 working days.

8. View weather

8.1. BRIEF DESCRIPTION:

This use case works for displaying the weather of the country/city.

8.2. ACTORS:

The following actors are involved in the use case:

- User

8.3. FLOW OF EVENT

8.3.1. BASIC FLOW:

This use case starts when the actor wants to view the weather of different country or city.

The use case will ask the city name from the user.

8.3.2. ALTERNTIVE FLOW:

(No alternative flow)

8.4. SPECIAL REQUIREMENT:

(No special requirement)

8.5. PRE CONDITION:

All the city names must be stored in DB.

8.6. POST CONDITION:

If the use case was successful, the actor can easily view the weather.

9. Make Donation

9.1. BRIEF DESCRIPTION:

This use case works for donating money to NGOs

9.2. ACTORS:

The following actors are involved in the use case:

- User

9.3. FLOW OF EVENT

9.3.1. BASIC FLOW:

This use case starts when the User wants to donate money to NGOs

The use case will ask necessary information for the donation.

9.3.2. ALTERNTIVE FLOW:

(No alternative flow)

9.4. SPECIAL REQUIREMENT:

(No special requirement)

9.5. PRE CONDITION:

All the users must have a user account i.e. user id, password.

9.6. POST CONDITION:

If the use case was successful, a message should be received saying we received your donation.

1. Introduction

1.1. Purpose

This test plan describes the testing approach and overall framework that will drive the testing of WECARE site. The document introduces:

- Test Strategy: rules the test will be based on, including the givens of the project (e.g.: start / end dates, objectives, assumptions); description of the process to set up a valid test (e.g.: entry / exit criteria, creation of test cases, specific tasks to perform, scheduling, data strategy).
- Execution Strategy: describes how the test will be performed and process to identify and report defects, and to fix and implement fixes.
- Test Management: process to handle the logistics of the test and all the events that come up during execution (e.g.: communications, escalation procedures, risk and mitigation, team roster)

1.2. Project Overview

My Info Module is a powerful tool providing employees of the company with the ability to view relevant information such as personal information and updating personal information with an internet enabled PC without having to involve the HR department .

The functionality of this module spans through the entire system, making information available anywhere, anytime. All information is subject to company's defined security policy, where he/she can only view the information he/she is authorized to. An ESS-User can only edit certain fields in the ESS Module, maintaining the security and confidentiality of employee information.

1.3. Audience

- Project team members perform tasks specified in this document, and provide input and recommendations on this document.
- Project Manager Plans for the testing activities in the overall project schedule, reviews the document, tracks the performance of the test according to the task here in specified, approves the document and is accountable for the results.
- The stakeholders' representatives and participants (individuals as identified by the PMO Leads) may take part in the UAT test to ensure the business is aligned with the results of the test.
- Technical Team ensures that the test plan and deliverables are in line with the design, provides the environment for testing and follows the procedures related to the fixes of defects.
- Business analysts will provide their inputs on functional changes.

2. Test Strategy

2.1. Test objectives

The objective of the test is to verify that the functionality of WECARE works according to the specification.

The test will execute and verify the test scripts, identify, fix and retest all

high and medium severity defects per the entrance criteria, prioritize lower severity defects for future fixing via CR.
The final product of the test is two fold:
•A production-ready software;
•A set of stable test scripts that can be reused for Functional and UAT test execution.

2.2. Test assumptions

Key Assumptions

- Production like data required and be available in the system prior to start of Functional Testing
- In each testing phase, Cycle 3 will be initiated if the defect rate is high in Cycle 2.

General

- Exploratory Testing would be carried out once the build is ready for testing
- Performance testing is not considered for this estimation.
- All the defects would come along with a snapshot JPEG format
- The Test Team will be provided with access to Test environment via VPN connectivity
- The Test Team assumes all necessary inputs required during Test design and Development/BUSINESS ANALYSTs will support execution appropriately.
- Test case design activities will be performed by QA Group
- Test environment and preparation activities will be owned by Dev Team
- Dev team will provide Defect fix plans based on the Defect meetings during each cycle to plan. The same will be informed to Test team prior to start of Defect fix cycles
- BUSINESS ANALYST will review and sign-off all Test cases prepared by Test Team prior to start of Test execution
- The defects will be tracked through HP ALM only. Any defect fixes planned will be shared with Test Team prior to applying the fixes on the Test environment
- Project Manager/BUSINESS ANALYST will review and sign-off all test deliverables
- The project will provide test planning, test design and test execution support
- Test team will manage the testing effort with close coordination with Project PM/BUSINESS ANALYST
- Project team has the knowledge and experience necessary, or has received adequate training in the system, the project and the testing processes.
- There is no environment downtime during test due to outages or defect fixes.
- The system will be treated as a black box; if the information shows correctly online and in the reports, it will be assumed that the database is working properly.
- Cycle 3 will be initiated if there are more defects in Cycle 2.

Functional Testing

- During Functional testing, testing team will use preloaded data which is available on the system at the time of execution
- The Test Team will perform Functional testing only on WECARE

UAT

- End users (L1, L2and L3) will perform UAT test execution and QA Group will provide their support on creating UAT script.

2.3. Test Principles

- Testing will be focused on meeting the business objectives, cost efficiency, and quality.
- There will be common, consistent procedures for all teams supporting testing activities.
- Testing processes will be well defined, yet flexible, with the ability to change as needed.
- Testing activities will build upon previous stages to avoid redundancy or duplication of effort.
- Testing environment and data will emulate a production environment as much as possible.
- Testing will be a repeatable, quantifiable, and measurable activity.
- Testing will be divided into distinct phases, each with clearly defined objectives and goals.
- There will be entrance and exit criteria.

2.4. Data Approach

- In functional testing, WECARE will contain pre-loaded test data and which is used for testing activities.

2.5. Scope and Levels of Testing

2.5.1. Exploratory

- **PURPOSE**: the purpose of this test is to make sure critical defects are removed before the next levels of testing can start.
- **SCOPE**: First level navigation, dealer and admin modules
- **TESTERS**: Testing team.
- **METHOD**: this exploratory testing is carried out in the application without any test scripts and documentation
- **TIMING**: at the beginning of each cycle.

2.5.2. Functional Test

- **PURPOSE:** Functional testing will be performed to check the functions of application. The functional testing is carried out by feeding the input and validates the output from the application.

- **Scope:** The below excel sheet details about the scope of Functional test. Note: The scope is high level due to changes in the requirement.
- **TESTERS**: Testing Team.
- **METHOD**: The test will be performed according to Functional scripts, which are stored in HP ALM.
- **TIMING**: After Exploratory, test is completed.

TEST ACCEPTANCE CRITERIA

- Approved Functional Specification document, Use case documents must be available prior to start of Test design phase.
- Test cases approved and signed-off prior to start of Test execution
- Development completed, unit tested with pass status and results shared to Testing team to avoid duplicate defects
- Test environment with application installed, configured and ready to use state

Sign off
• Approved Functional Specification Document • Approved Use cases • Approved Test cases

Readiness
• Development completed & unit tested • Application deployed and system ready for testing on Test environment • Production like data is available to test all functionalities. • Defect fixes planned based on Defect triage (Unit Testing) and evaluation criteria

TEST DELIVERABLES

S.no	Deliverable name	Author	Reviewer
1	Test Plan	Test lead	Project Manager/Business Analyst
2	Functional test cases	Test team	Business Analyst's Sign off
3	Logging Defects in HL ALM	Test team	Test Lead/ Programming Lead
4	Daily/weekly status report	Test Team/ Test Lead	Test Lead/ Project Manager
5	Test Closure report	Test Lead	Project Manager

MILESTONE LIST
The milestone list is tentative and may change due to below reasons
a) Any issues in the System environment readiness
b) Any change in scope/addition in scope
c) Any other dependency that affects efforts and timelines

2.5.3. User Acceptance Test (UAT)

- **PURPOSE**: this test focuses on validating the business logic. It allows the end users to complete one final review of the system prior to deployment.
- **TESTERS**: the end users (L1, L2 and L3) perform the UAT.
- **METHOD**: Since the business users are the most indicated to provide input around business needs and how the system adapts to them, it may happen that the users do some validation not contained in the scripts. Test team write the UAT test cases based on the inputs from End user (L1, L2 and L3 users) and Business Analyst's.
- **TIMING**: After all other levels of testing (Exploratory and Functional) are done. Only after this test is completed the product can be released to production.

TEST DELIVERABLES

S.no	Deliverable Name	Author	Reviewer
1	UAT Test case	Test team	Business Analyst's Sign off

3. EXECUTION STRATEGY

3.1. Entry and Exit Criteria

- The entry criteria refer to the desirable conditions in order to start test execution; only the migration of the code and fixes need to be assessed at the end of each cycle.
- The exit criteria are the desirable conditions that need to be met in order proceed with the implementation.
- Entry and exit criteria are flexible benchmarks. If they are not met, the test team will assess the risk, identify mitigation actions and provide a recommendation. All this is input to the project manager for a final "go-no go" decision.
- Entry criteria to start the execution phase of the test: the activities listed in the Test Planning section of the schedule are 100% completed.
- Entry criteria to start each cycle: the activities listed in the Test Execution section of the schedule are 100% completed at each cycle.

Exit criteria	Test team	Technical team	Notes

100% Test Scripts executed			
95% pass rate of Test Scripts			
No open Critical and High severity defects			
95% of Medium severity defects have been closed All remaining defects are either cancelled or documented as Change Requests for a future release			
All expected and actual results are captured and documented with the test script			
All test metrics collected based on reports from HP ALM			
All defects logged in HP ALM			
Test environment clean up completed and a new back up of the environment			

3.2. Test Cycles

- There will be two cycles for functional testing. Each cycle will execute all the scripts .
- The objective of the first cycle is to identify any blocking, critical defects, and most of the high defects. It is expected to use some work-around in order to get to all the scripts.
- The objective of the second cycle is to identify remaining high and medium defects, remove the work-around from the first cycle, correct gaps in the scripts and obtain performance results.
- UAT test will consist of one cycle.

3.3. Validation and Defect Management

- It is expected that the testers execute all the scripts in each of the cycles described above. However, it is recognized that the testers could also do additional testing if they identify a possible gap in the scripts. This is especially relevant in the second cycle, when the Business analyst's join the TCOE in the execution of the test, since the BUSINESS ANALYSTs have a deeper knowledge of the business processes. If a gap is identified, the scripts and traceability matrix will be updated and then a defect logged against the scripts.
- The defects will be tracked through HP ALM only. The technical team will gather information on a daily basis from HP ALM, and request additional details from the Defect Coordinator. The technical team will work on fixes.
- It is the responsibility of the tester to open the defects, link them to the corresponding script, assign an initial severity and status, retest and close the defect; it is the responsibility of the Defect Manager to review the severity of the defects and facilitate with the technical team the fix and its implementation, communicate with testers when the test can continue or should be halt, request the tester to retest, and modify status as the defect progresses through the cycle; it is the responsibility of the technical team to review HP ALM on a daily basis, ask for details if necessary, fix the defect, communicate to the Defect Manager the fix is done, implement the solution per the Defect Manager request.

Defects found during the Testing will be categorized according to the bug-reporting tool “Mercury HP ALM” and the categories are:

3.4. Test Metrics

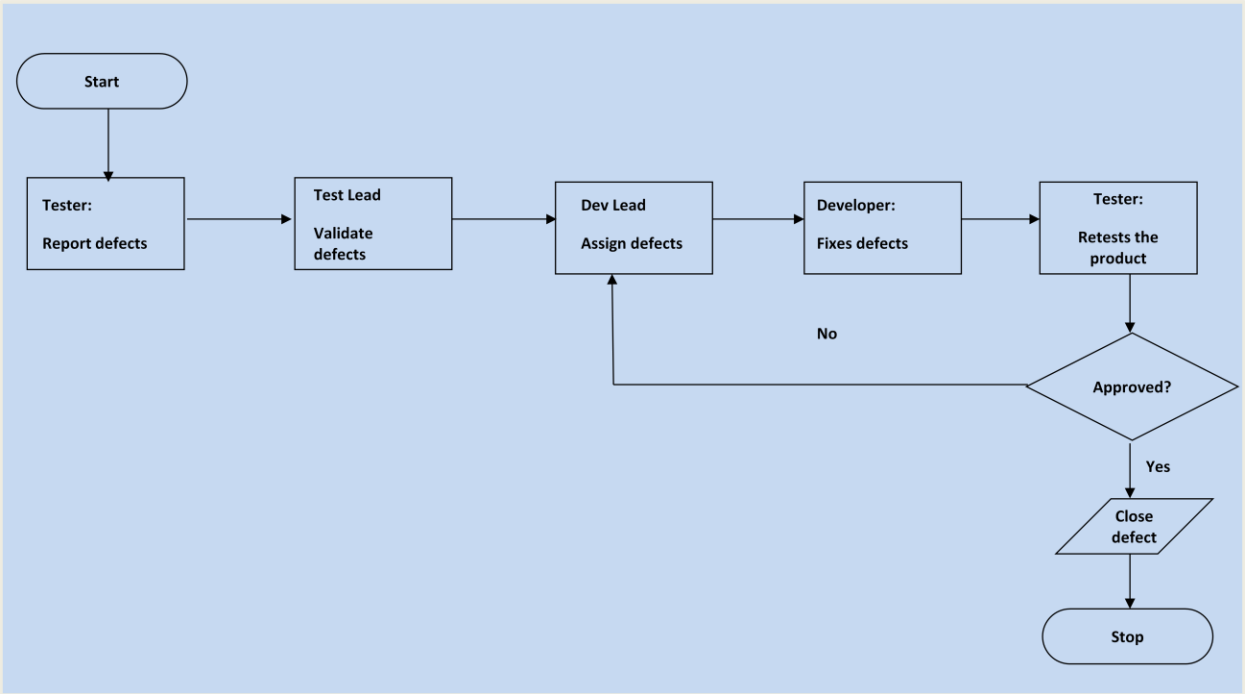
Severity	Impact
1 (Critical)	▪ This bug is critical enough to crash the system, cause file corruption, or cause potential data loss ▪ It causes an abnormal return to the operating system (crash or a system failure message appears). ▪ It causes the application to hang and requires re-booting the system.
2 (High)	▪ It causes a lack of vital program functionality with workaround.
3 (Medium)	▪ This Bug will degrade the quality of the System. However there is an intelligent workaround for achieving the desired functionality - for example through another screen. ▪ This bug prevents other areas of the product from being tested. However other areas can be independently tested.
4 (Low)	▪ There is an insufficient or unclear error message, which has minimum impact on product use.
5(Cosmetic)	▪ There is an insufficient or unclear error message that has no impact on product use.

Test metrics to measure the progress and level of success of the test will be developed and shared with the project manager for approval. The below are some of the metrics

Report	Description	Frequency
Test preparation & Execution Status	To report on % complete, %WIP, % Pass, % Fail Defects severity wise Status – Open, closed, any other Status	Weekly / Daily (optional)
Daily execution status	To report on Pass, Fail, Total defects, highlight Showstopper/ Critical defects	Daily
Project weekly status report	Project driven reporting (As requested by PM)	Weekly – If project team needs weekly update apart from daily and there is template available with project team to use.

3.5. Defect tracking & Reporting

Following flowchart depicts Defect Tracking Process:



4. TEST MANAGEMENT PROCESS

4.1. Test Management Tool

HP Application Lifecycle Management is the tool used for Test Management. All testing artifacts such as Test cases, test results are updated in the HP Application Lifecycle Management (ALM) tool.

4.1.1. Project specific folder structure will be created in HP ALM to manage the status of this DFRT project.

4.1.2. Each resource in the Testing team will be provided with Read/Write access to add/modify Test cases in HP ALM.

4.1.3. During the Test Design phase, all test cases are written directly into HP ALM. Any change to the test case will be directly updated in the

HP ALM.

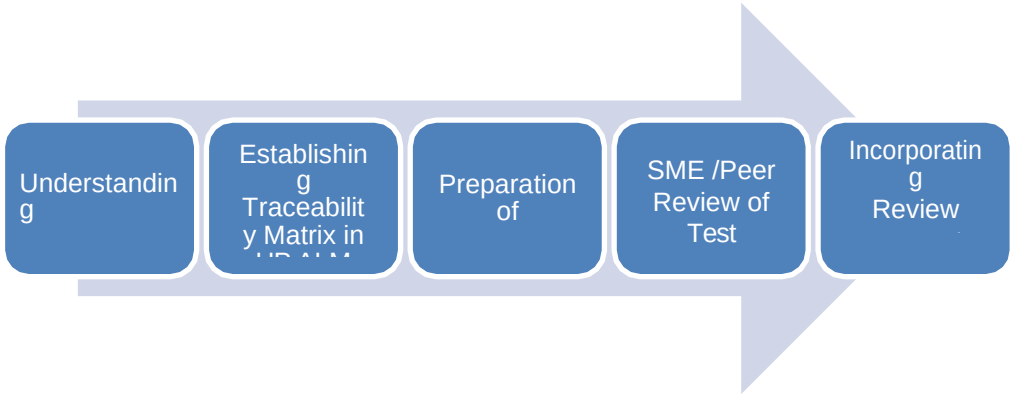
4.1.4. Each Tester will directly access their respective assigned test cases and update the status of each executed step in HP ALM directly.

4.1.5. Any defect encountered will be raised in HP ALM linking to the particular Test case/test step.

4.1.6. During Defect fix testing, defects are re-assigned back to the tester to verify the defect fix. The tester verifies the defect fix and updates the status directly in HP ALM.

4.1.7. Various reports can be generated from HP ALM to provide status of Test execution. For example, Status report of Test cases executed, Passed, Failed, No. of open defects, Severity wise defects etc.

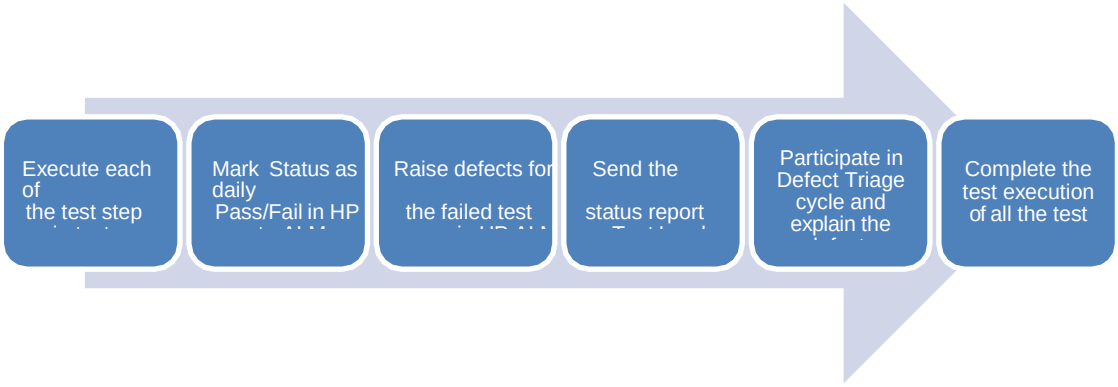
4.2. Test Design Process



- The tester will understand each requirement and prepare corresponding test case to ensure all requirements are covered.
- Each Test case will be mapped to Use cases to Requirements as part of Traceability matrix.
- Each of the Test cases will undergo review by the BUSINESS ANALYST and the review defects are captured and shared to the Test team. The testers will rework on the review defects and finally obtain approval and sign-off.
- During the preparation phase, tester will use the prototype, use case and functional specification to write step by step test cases.
- Testers will maintain a clarification Tracker sheet and same will be shared periodically with the Requirements team and accordingly the test case will be updated. The clarifications may sometimes lead to Change Requests or not in scope or detailing implicit requirements.
- Sign-off for the test cases would be communicates through mail by Business Analyst's.
- Any subsequent changes to the test case if any will be directly

updated in HP ALM.

4.3. Test Execution Process



4.3.1. Once all Test cases are approved and the test environment is ready for testing, tester will start a exploratory test of the application to ensure the application is stable for testing.

4.3.2. Each Tester is assigned Test cases directly in HP ALM.

4.3.3. Testers to ensure necessary access to the testing environment, HP ALM for updating test status and raise defects. If any issues, will be escalated to the Test Lead and in turn to the Project Manager as escalation.

4.3.4. If any showstopper during exploratory testing will be escalated to the respective development SPOCs for fixes.

4.3.5. Each tester performs step by step execution and updates the executions status. The tester enters Pass or Fail Status for each of the step directly in HP ALM.

4.3.6. Tester will prepare a Run chart with day-wise execution details

4.3.7. If any failures, defect will be raised as per severity guidelines in HP ALM tool detailing steps to simulate along with screenshots if appropriate.

4.3.8. Daily Test execution status as well as Defect status will be reported to all stakeholders.

4.3.9. Testing team will participate in defect triage meetings in order to ensure all test cases are executed with either pass/fail category.

4.3.10. If there are any defects that are not part of steps but could be outside the test steps, such defects need to be captured in HP ALM and map it against the test case level or at the specific step that issue was encountered after confirming with Test Lead.

4.3.11. This process is repeated until all test cases are executed fully with Pass/Fail status.

4.3.12. During the subsequent cycle, any defects fixed applied will be tested and results will be updated in HP ALM during the cycle.

As per Process, final sign-off or project completion process will be followed

4.4. Test Risks and Mitigation Factors

Risk	Prob.	Impact	Mitigation Plan
SCHEDULE Testing schedule is tight. If the start	High	High	• The testing team can control the preparation tasks (in advance)
of the testing is delayed due to design tasks, the test cannot be extended beyond the UAT scheduled start date.			and the early communication with involved parties. • Some buffer has been added to the schedule for contingencies, although not as much as best practices advise.
RESOURCES Not enough resources, resources on boarding too late (process takes around 15 days.	Medium	High	Holidays and vacation have been estimated and built into the schedule; deviations from the estimation could derive in delays in the testing.
DEFECTS Defects are found at a late stage of the cycle or at a late cycle; defects discovered late are most likely be due to unclear specifications and are time consuming to resolve.	Medium	High	Defect management plan is in place to ensure prompt communication and fixing of issues.
SCOPE Scope completely defined	Medium	Medium	Scope is well defined but the changes are in the functionality are not yet finalized or keep on changing.
Natural disasters	Low	Medium	Teams and responsibilities have been spread to two different geographic areas. In a catastrophic event in one of the areas, there will resources in the other areas needed to continue (although at a slower pace) the testing activities.
Non-availability of Independent Test environment and accessibility	Medium	High	Due to non availability of the environment, the schedule gets impacted and will lead to

			delayed start of Test execution.
Delayed Testing Due To new Issues	Medium	High	During testing, there is a good chance that some “new” defects may be identified and may become an issue that will take time to resolve. There are defects that can be raised during testing because of unclear document specification. These defects can yield to an issue that will need time to be resolved. If these issues become showstoppers, it will greatly impact
			On the overall project schedule. If new defects are discovered, the defect management and issue management procedures are in place to immediately provide a resolution.

5.1. Communications Plan and Team Roster

5.2. Role Expectations

The following list defines in general terms the expectations related to the roles directly involved in the management, planning or execution of the test for the project.

	Roles	Name	Contact Info
1.	Project Manager		
2.	Test Lead		
3.	Business Analyst		
4.	Development Lead		
5.	Testing Team		
6.	Development Team		
7.	Technical Lead		

5.2.1. Project Management

- Project Manager: reviews the content of the Test Plan, Test Strategy and Test Estimates signs off on it.

5.2.2. Test Planning (Test Lead)

- Ensure entrance criteria are used as input before start the execution.
- Develop test plan and the guidelines to create test conditions, test cases, expected results and execution scripts.
- Provide guidelines on how to manage defects.
- Attend status meetings in person or via the conference call line.
- Communicate to the test team any changes that need to be made to the test deliverables or application and when they will be completed.
- Provide on premise or telecommute support.
- Provide functional (Business Analysts) and technical team to test team personnel (if needed).

5.2.3. Test Lead

- Develop test conditions, test cases, expected results, and execution scripts.
- Perform execution and validation.
- Identify, document and prioritize defects according to the guidance provided by the Test lead.
- Re-test after software modifications have been made according to the schedule.
- Prepare testing metrics and provide regular status.

5.2.4. Development Team

- Review testing deliverables (test plan, cases, scripts, expected results, etc.) and provide timely feedback.
- Assist in the validation of results (if requested).
- Support the development and testing processes being used to support the project.
- Certify correct components have been delivered to the test environment at the points specified in the testing schedule.
- Keep project team and leadership informed of potential software delivery date slips based on the current schedule.
- Define processes/tools to facilitate the initial and ongoing migration of components.
- Conduct first line investigation into execution discrepancies and assist test executors in creation of accurate defects.

- Implement fixes to defects according to schedule.

6. TEST ENVIRONMENT

WECARE will be hosted on localhost

A windows environment with Internet Explorer 8, 9 and 10, and with Firefox 27.0, as well as Google Chrome32.0 and later should be available to each tester.

7. APPROVALS

The Names and Titles of all persons who must approve this plan.

Signature:	
Name:	
Role:	
Date:	

Signature:	
Name:	
Role:	
Date:	

4) Design Positive Test scenarios and Negative Test Scenarios to test the functionality of the system.

Login Page:

Test Case	Description	Status	Message
1.	User enters correct username and password and is able to login	Successful	Successful login
2.	User enters username correct but password wrong	Successful	Error message displayed
3.	User enters username wrong but password correct	Successful	Error message displayed
4.	User enters both credentials wrong	Successful	Error message displayed
5.	User did not enter credentials	Successful	Error message displayed

Sign Up Page:

Test Case	Description	Status	Message
1.	User enters valid details	Successful	Successful sign up
2.	User enters invalid name but correct email	Successful	Error message displayed
3.	User enters email wrong but name correct	Successful	Error message displayed
4.	User enters both credentials wrong	Successful	Error message displayed
5.	User did not enter credentials	Successful	Error message displayed

Contact us module:

Test Case	Description	Status	Message
1.	User enters correct email	Successful	Successful

	id		
2.	User enters wrong email id	Successful	Error message displayed
3.	User doesn't enters email id	Successful	Error message displayed

Weather Page:

Test Case	Description	Status	Message
1.	User enters correct city name	Successful	Successfully displays weather
2.	User enters wrong (invalid city name)	Successful	Error message displayed
3.	User did not enter name	Successful	Error message displayed

Donate:

Test Case	Description	Status	Message
1.	User fills all the details correct	Successful	Successful
2.	User fills wrong details	Successful	Error message displayed
3.	User did not fill the details	Successful	Error message displayed

5. Black box testing

Boundary value checking of the payment page amount text field

Input	Expected outcome
0	Invalid
1	Valid
10000000	Valid
9999999	Valid
5000000	Valid

Random testing Weather API

Input	Expected Outcome
New Delhi	22
123	Invalid
@@	Invalid
Vellore	30

Vllore	Invalid
N E W D E L H I	Invalid
NeW DeLhI	22
new delhi	22
NEW DELHI	22
New1 delhi	Invalid
New@ delhi	Invalid
New Delhi Vellore	Invalid
New New Delhi	Invalid

Cause and effect graph technique

Login feature

Situation:

The user must enter correct and valid Email Id as well as password to be able to login to their respective accounts.

Causes:

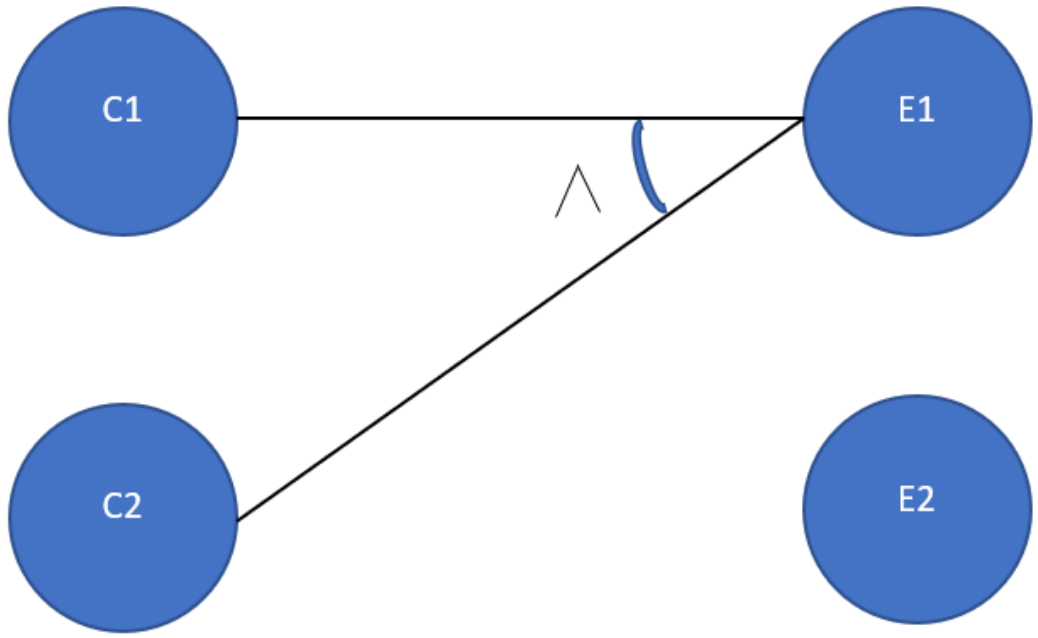
C1 – Correct Email ID

C2 – Correct password

Effect:

E1 – user logged in successfully

E2 – Not allowed to login due to incorrect email or password and display error message.



Decision table-based testing

Creating decision table from cause effect graph

Actions	TC1	TC2	TC3	TC4

C1	1	0	1	0
C2	1	1	0	0
E1	1	0	0	0
E2	0	1	1	1

Writing Test cases from the decision table

TC ID	TC Name	Description	Steps	Expected Results
TC1	Login test case 1	Allow the user to login if email and password are correct	1. Open website 2. Go to sign up/sign in page 3. Enter the correct email and password	Successfully logged in
TC2	Login test case 2	Do not allow the user to login if Email is wrong and password maybe correct	1. Open website 2. Go to sign up/sign in page 3. Enter the correct email and incorrect password	Error while logging in
TC3	Login test case 3	Do not allow the user to login if Email is correct and password is incorrect	1. Open website 2. Go to sign up/sign in page 3. Enter the incorrect email and correct password	Error while logging in
TC4	Login test case 4	Do not allow the user to login if Email is incorrect and password is incorrect	1. Open website 2. Go to sign up/sign in page 3. Enter the incorrect email and incorrect password	Error while logging in

Requirements based testing
Sample requirements Traceability Matrix

S No.	**Requirements Identifier**	**Description**	**Priority (high, med, low)**
1.	BR-01	Correct Email and password should facilitate successful login of the user	H
2.	BR-02	Incorrect Email or Password should not allow to login and display error	H
3.	BR-03	User can register	H

		themselves using sign up	
4.	BR-04	when user decides to Donate for a certain NGO, the they should be redirected towards the pay now page	H
5.	BR-05	When user adds item purchased to the cart, the amount should be updated dynamically	M
6.	BR-06	When user enters details in the Pay Now page should be validated before taking payment from the user	H
7.	BR-07	The website should support regional languages too.	L

Req. ID	Description	Priority (H, M, L)	Test Conditions	Test Case IDs	Phase of testing
BR-01	Correct Email and password should facilitate successful login of the user	H	Email: sample@gmail.com Password: sample	T-001	Unit Component
BR-02	Incorrect Email or password should not allow to login and display error	H	Email: sample@gmail.com Password: 1234	T-002	Unit Component
BR-03	For signing up for the website, user should enter their name, email, and password	H	Name: Tanishq Email: tanishq@gmail.com Password: 1111	T-003	Unit Component
BR-04	When user decided to donate for a certain NGO, By clicking	H	Clicking on Donate now button Redirection towards Pay now page	T-004	Unit Component

	on Donate now button, he should be taken to respective payment page.				
BR-05	User adding items to cart, the cart price displayed should be updated dynamically	M	Selecting item and number of items Displaying cart price	T-005	Unit component
BR-06	User adding details to the pay now page, all the fields should be validated, before proceeding to payment	H	Amount: 1000 DOB: 25-11-200 Name: Tanishq Email: tanishq@gmail.com Mode of payment: credit card Phone no: 8181818181 Address: B-121, Janak puri City: New Delhi State: New Delhi	T-006	Unit Component
BR-07	The website should support other regional languages	L			System

Sample Test Execution Data

S No.	Req. ID	Priority	Test Cases	Total Test Cases	Test Cases Passed	Test Cases failed	% Pass	No. Of Defects
1.	BR-01	H	T-001	1	1	0	100	0
2.	BR-02	H	T-002	1	1	0	100	0
3.	BR-03	H	T-003	1	1	0	100	0
4.	BR-04	H	T-004	1	1	0	100	0
5.	BR-05	M	T-005	1	1	0	100	0
6.	BR-06	H	T-006	1	1	0	100	0
7.	BR-07	L	T-007					0

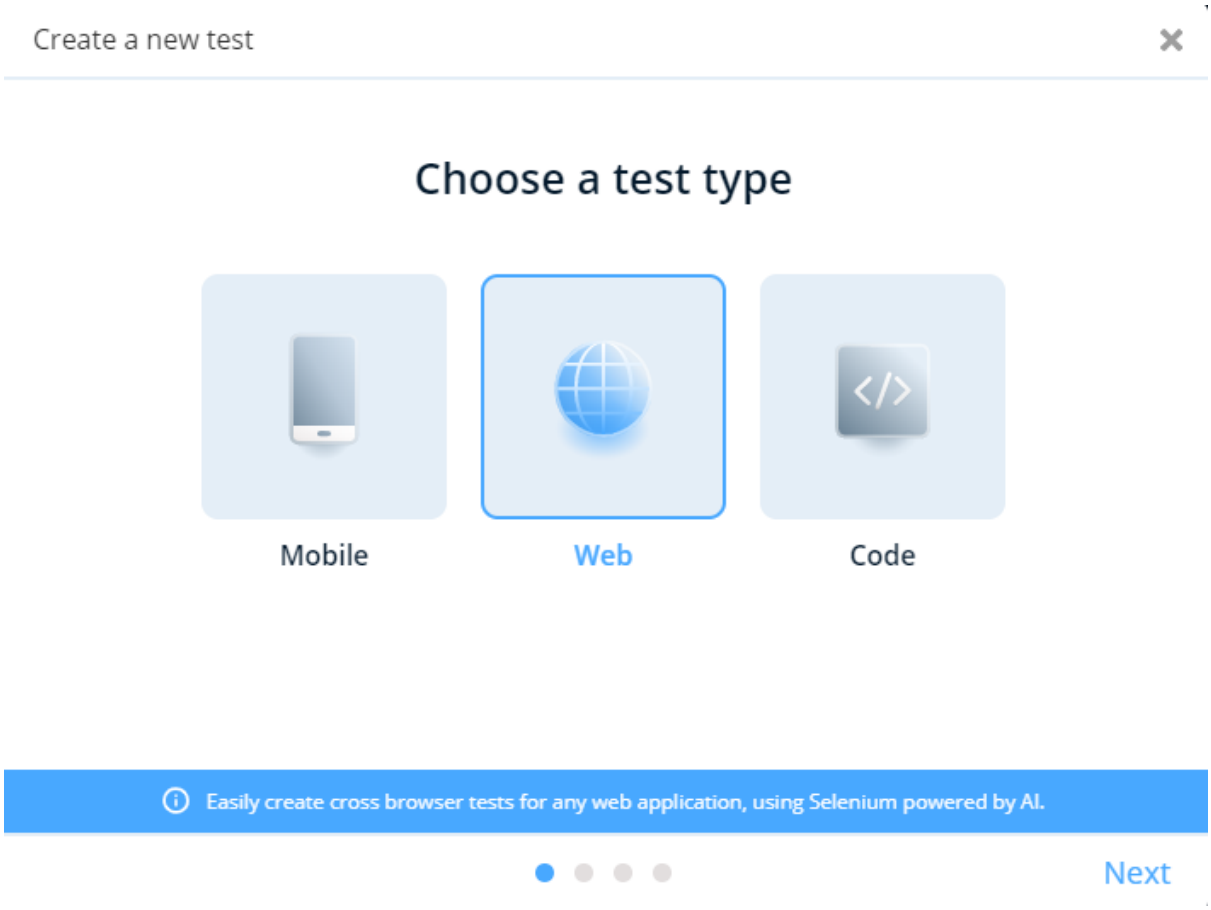
6. Use any open source-testing tool and execute the test cases meant for automation (Unit Testing) and take the snapshots for recording the test results.

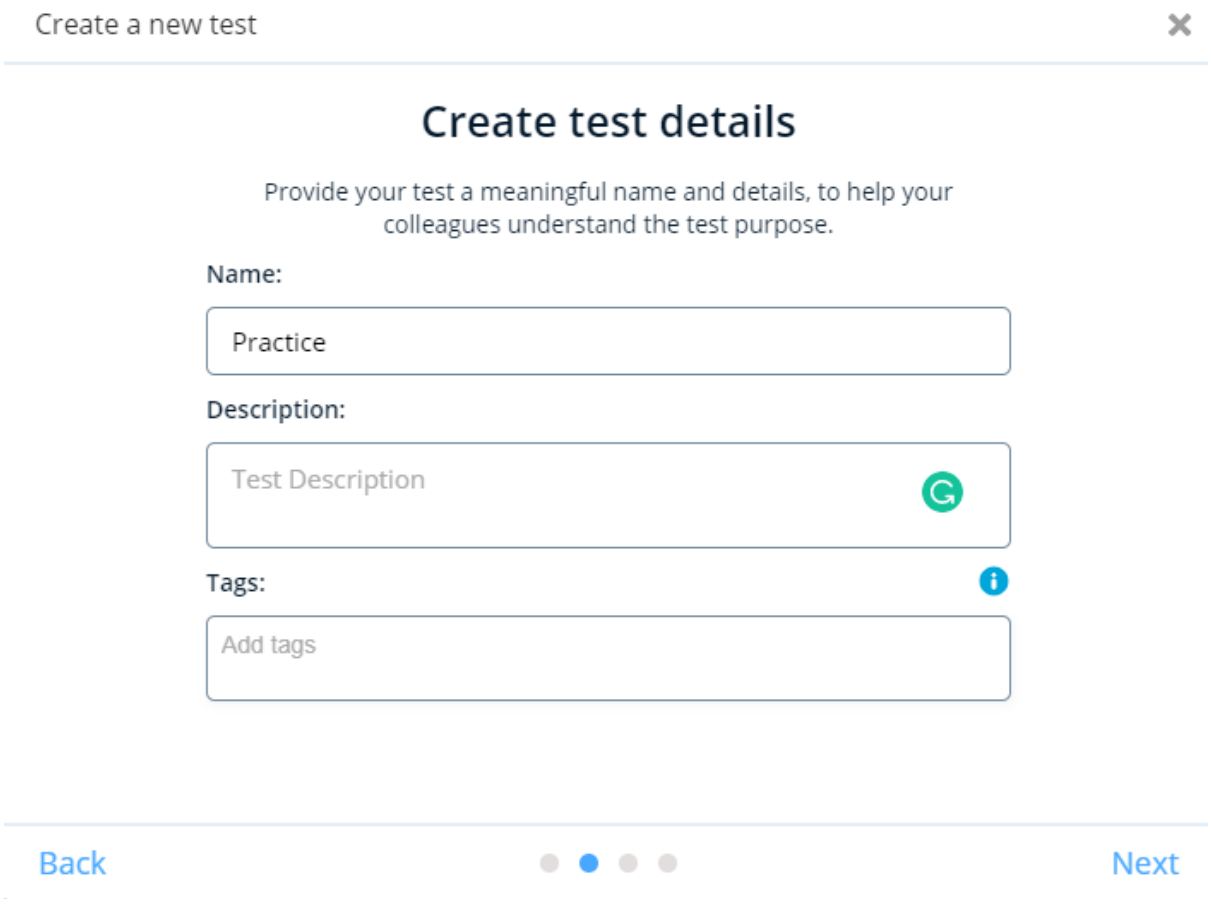
For unit testing we used 2 software to verify the results

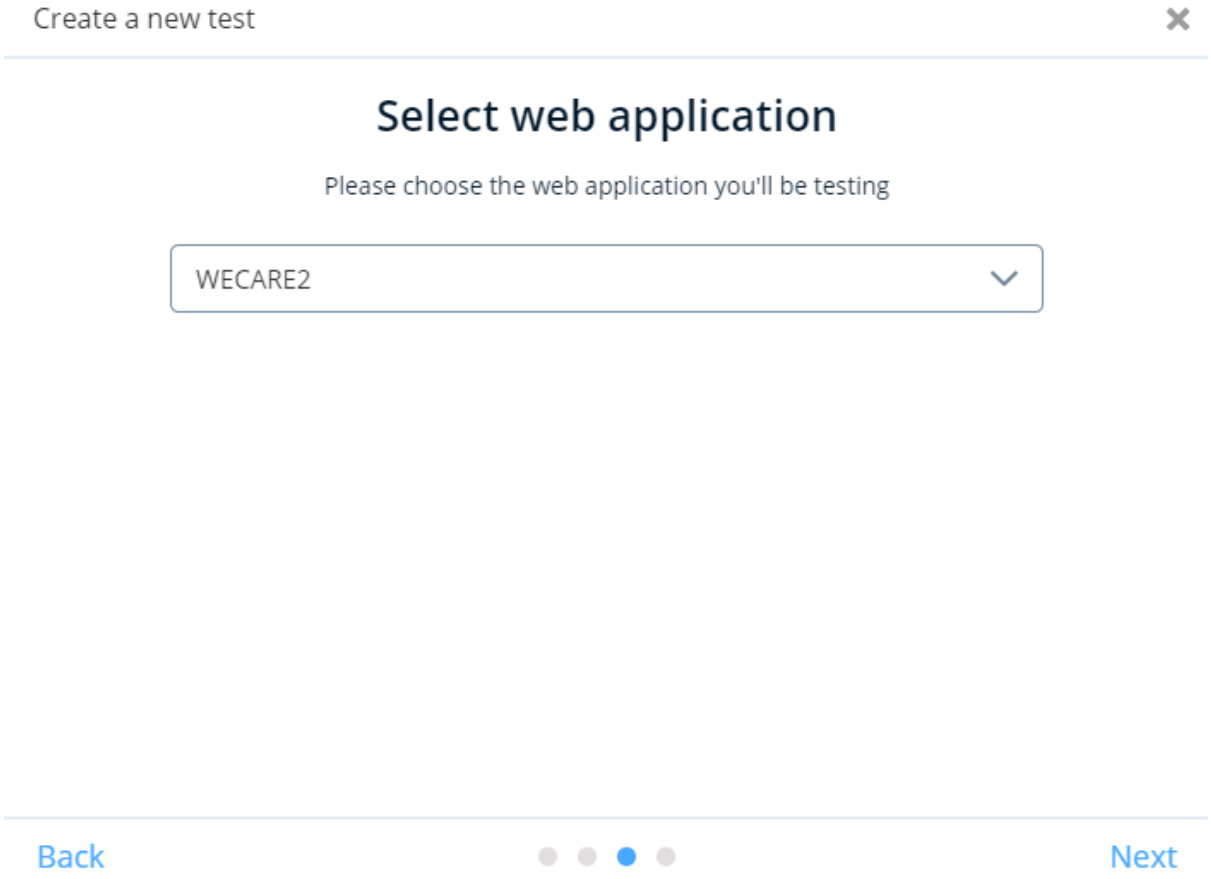
1. Test project
2. Selenium IDE

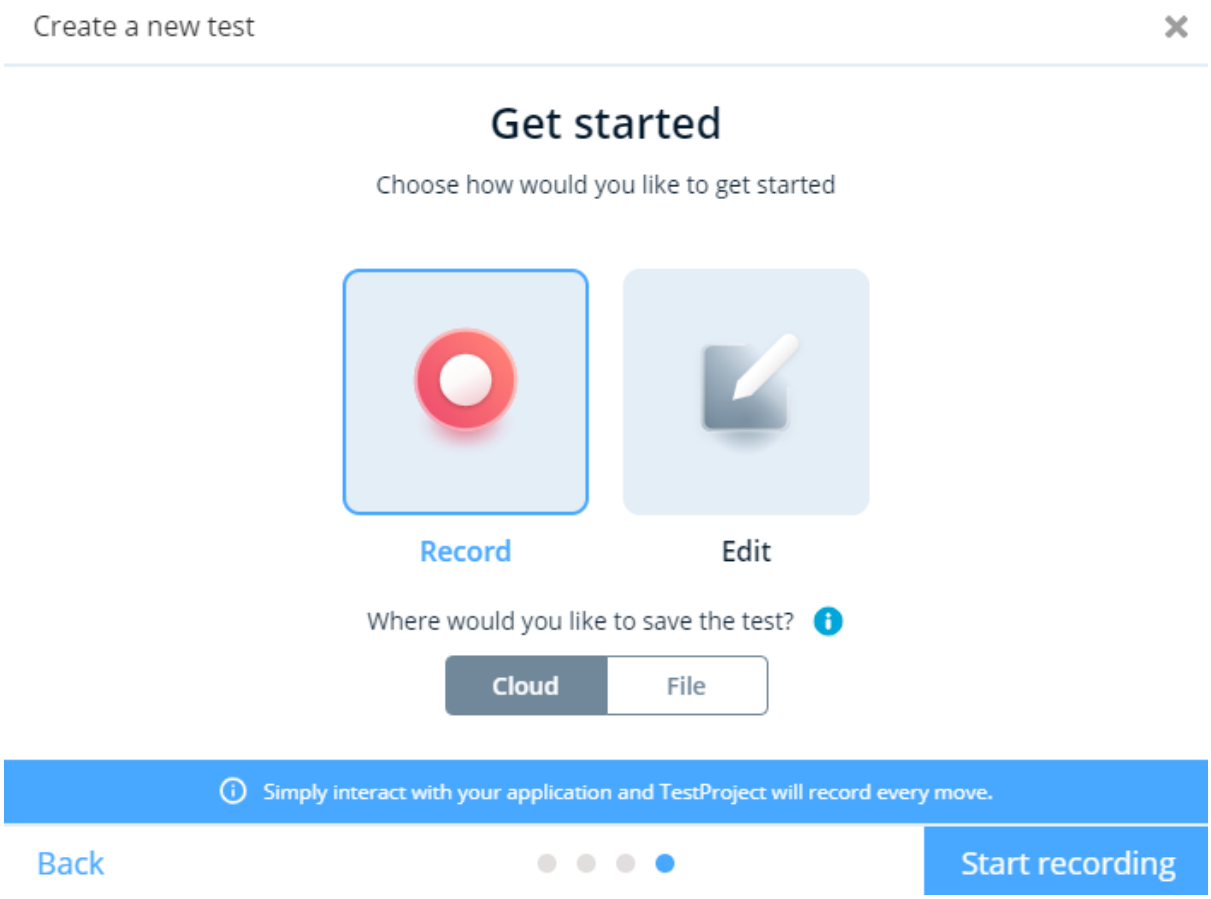
Creating test case to sign up:

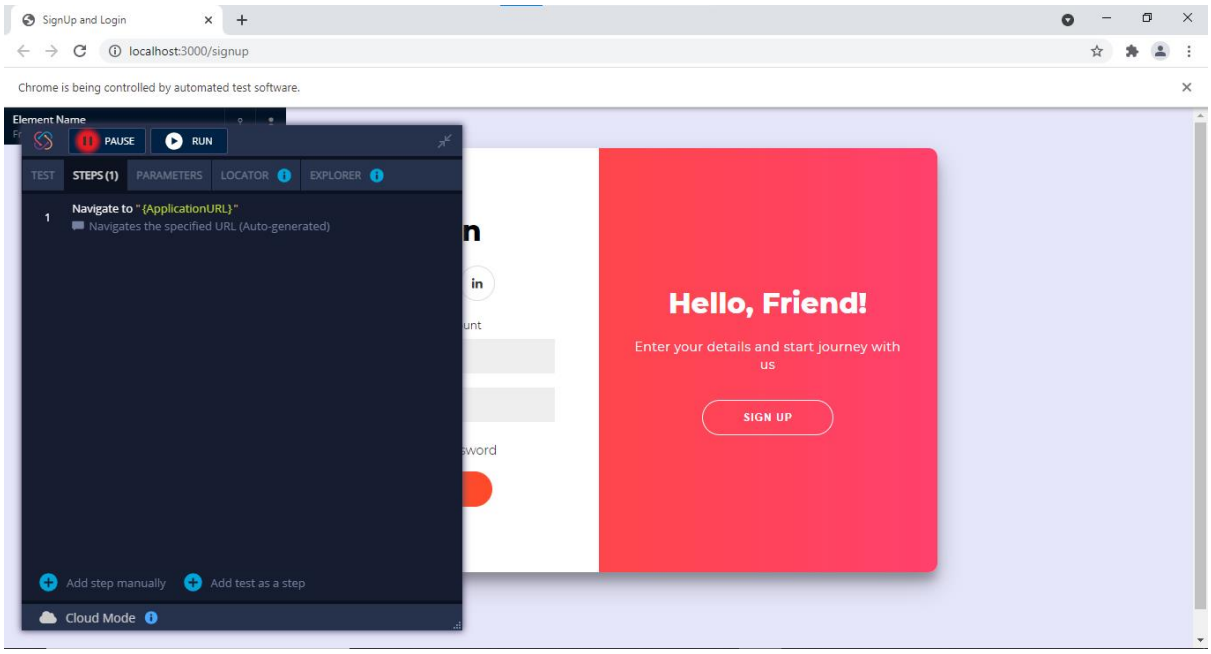
Creating using Test project

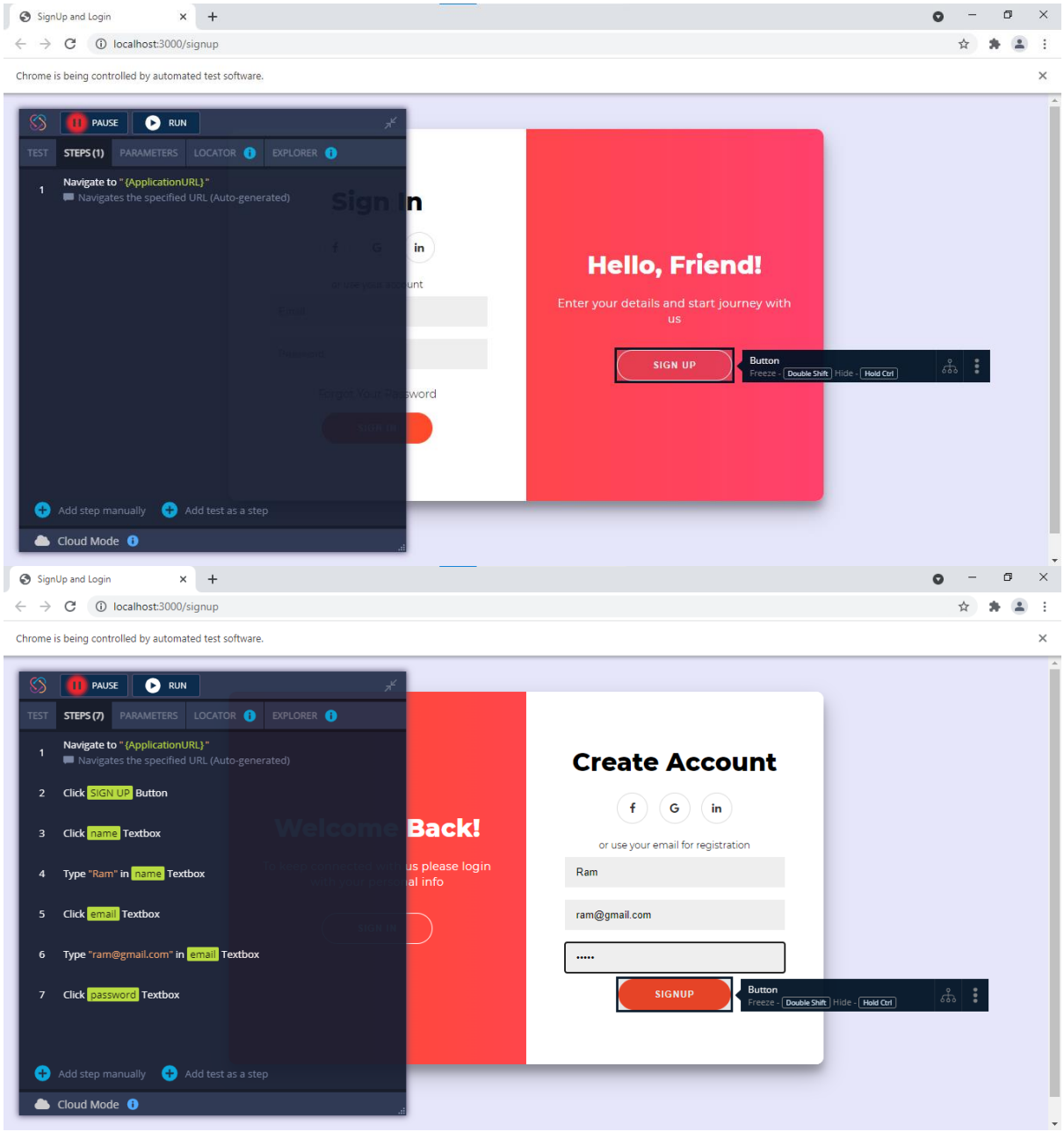


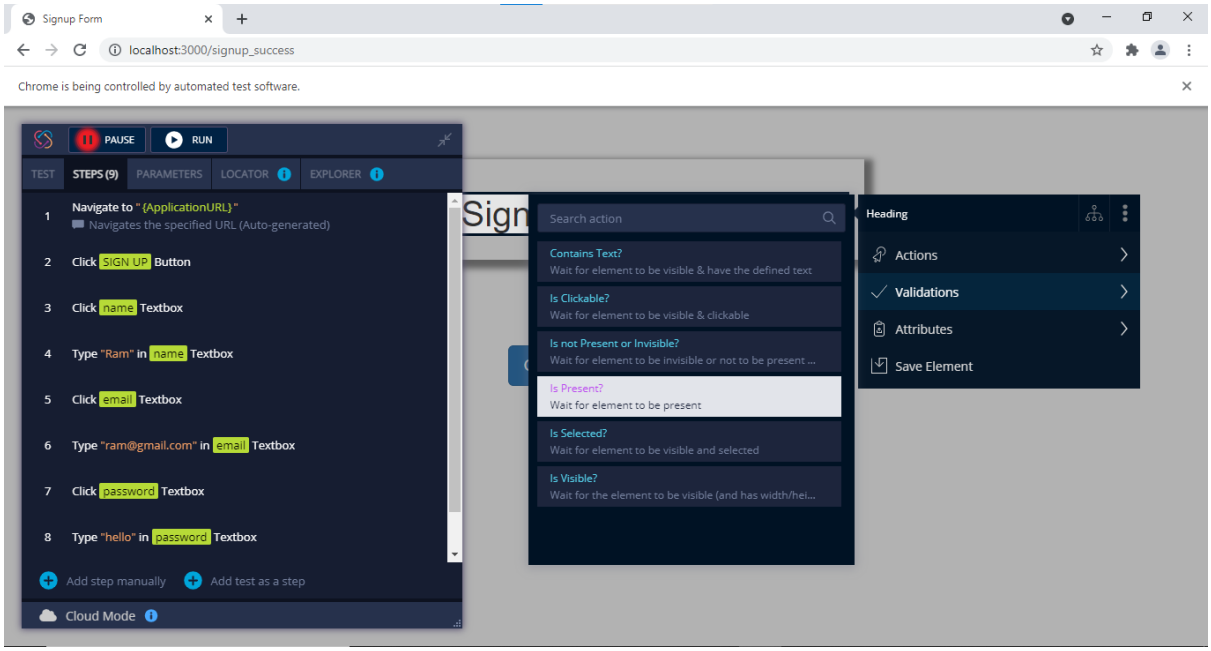

Create a new test
Create test details
Provide your test a meaningful name and details, to help your colleagues understand the test purpose.
Name:
Practice
Description:
Test Description
Tags:
Add tags
Back
Next


Create a new test
Select web application
Please choose the web application you'll be testing
WECARE2
Back
Next

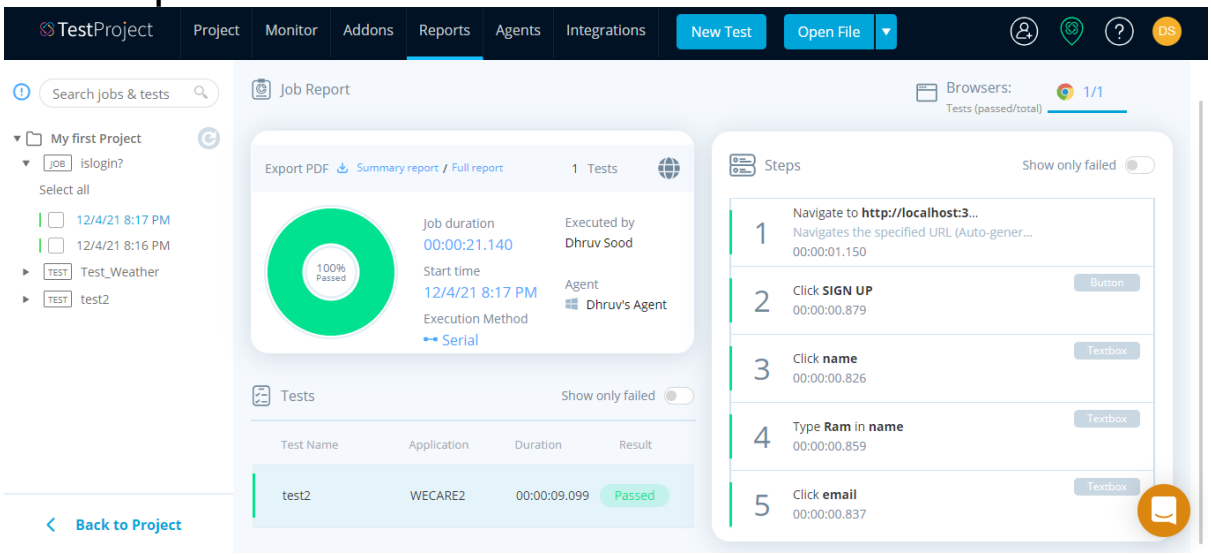

Create a new test
Get started
Choose how would you like to get started
Record
Edit
Where would you like to save the test?
Cloud
File
Simply interact with your application and TestProject will record every move.
Back
Start recording


SignUp and Login
localhost:3000/signup
Chrome is being controlled by automated test software.
PAUSE
RUN
TEST
STEPS (1)
PARAMETERS
LOCATOR
EXPLORER
Navigate to "{ApplicationURL}"
Navigates the specified URL (Auto-generated)
Add step manually
Add test as a step
Cloud Mode
Hello, Friend!
Enter your details and start journey with us
SIGN UP

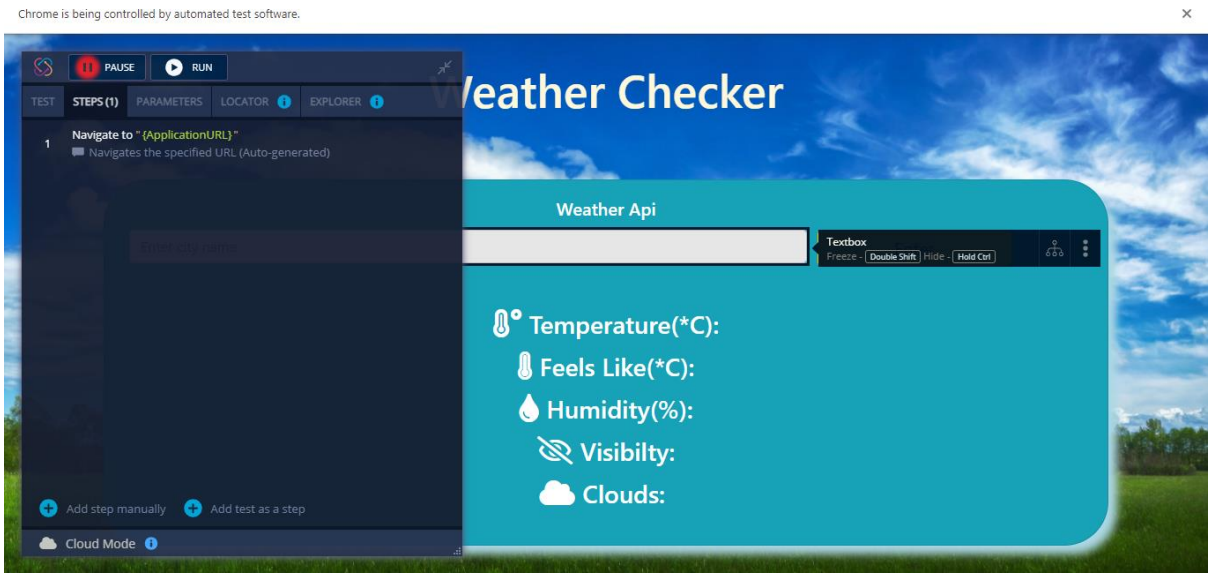

SignUp and Login
localhost:3000/signup
Chrome is being controlled by automated test software.
PAUSE
RUN
TEST
STEPS (1)
PARAMETERS
LOCATOR
EXPLORER
1
Navigate to "{ApplicationURL}"
Navigates the specified URL (Auto-generated)
Add step manually
Add test as a step
Cloud Mode
Hello, Friend!
Enter your details and start journey with us
SIGN UP
Button
Freeze - Double Shift Hide - Hold Ctrl
SignUp and Login
localhost:3000/signup
Chrome is being controlled by automated test software.
PAUSE
RUN
TEST
STEPS (7)
PARAMETERS
LOCATOR
EXPLORER
1
Navigate to "{ApplicationURL}"
Navigates the specified URL (Auto-generated)
2
Click SIGN UP Button
3
Click name Textbox
4
Type "Ram" in name Textbox
5
Click email Textbox
6
Type "ram@gmail.com" in email Textbox
7
Click password Textbox
Add step manually
Add test as a step
Cloud Mode
Create Account
or use your email for registration
Ram
ram@gmail.com
SIGNUP
Button
Freeze - Double Shift Hide - Hold Ctrl

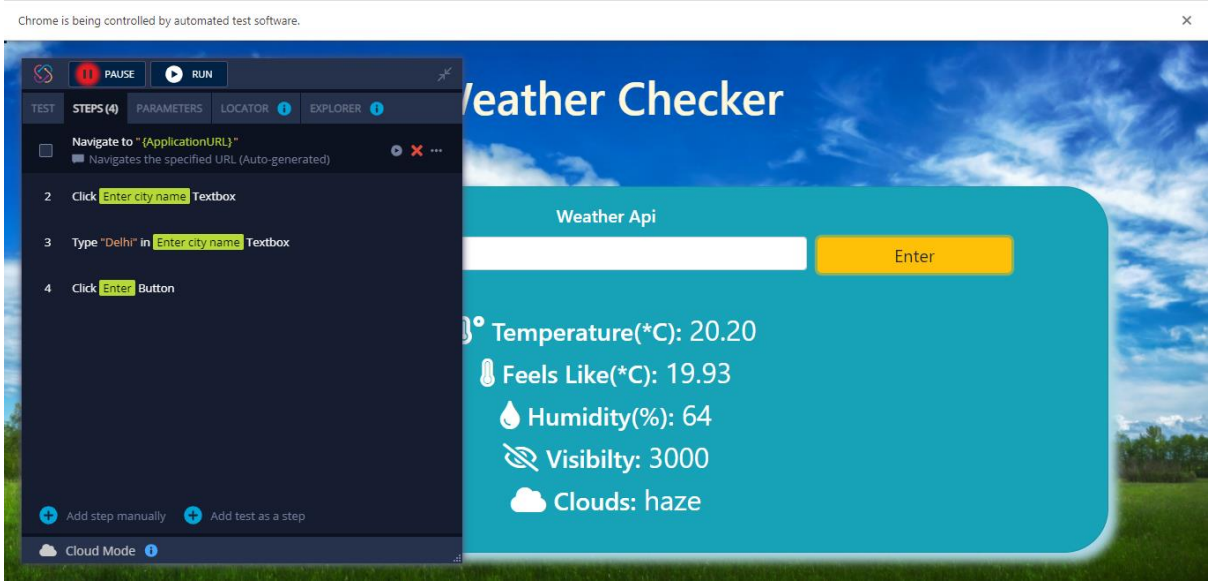


Test report:

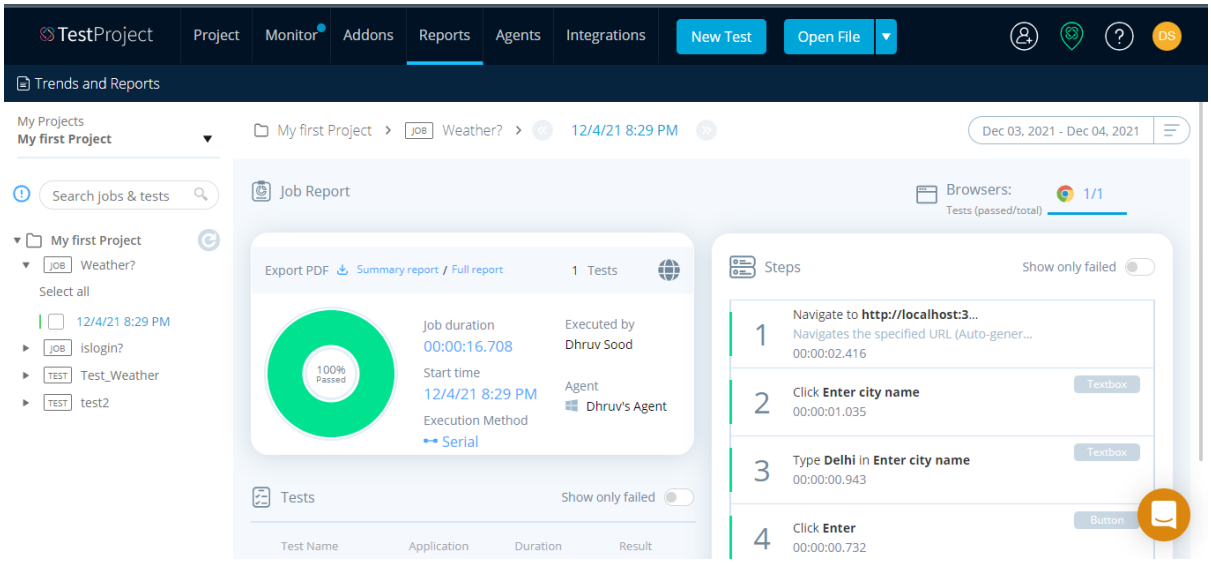


Creating test case to check weather:

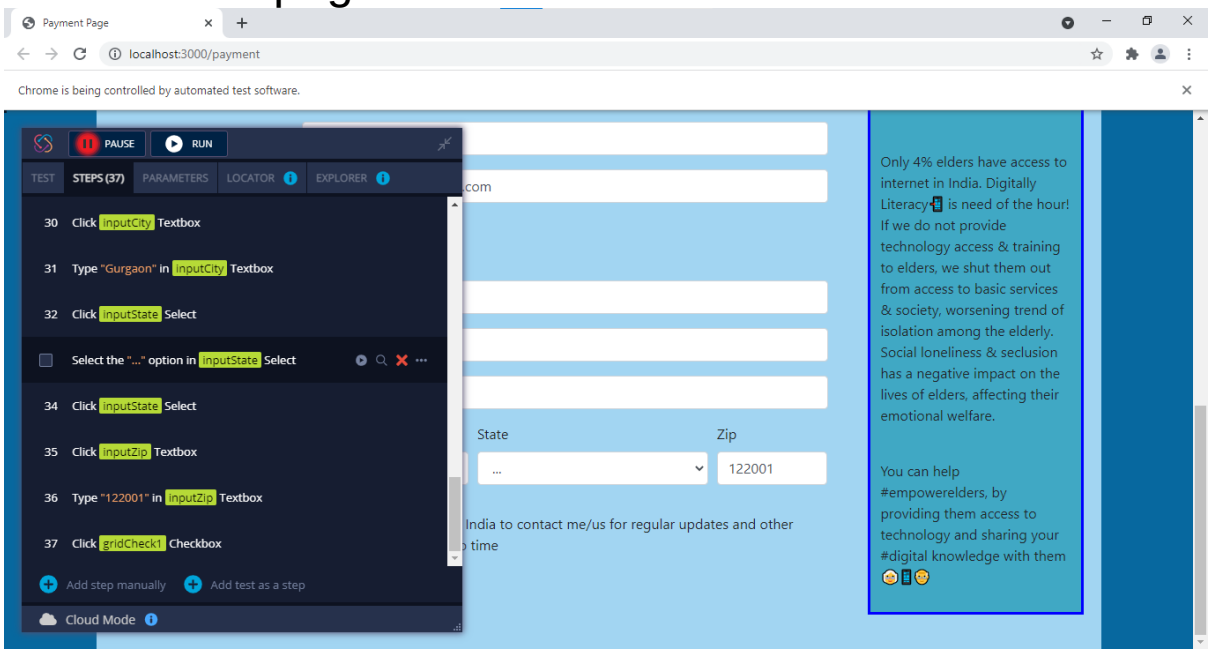




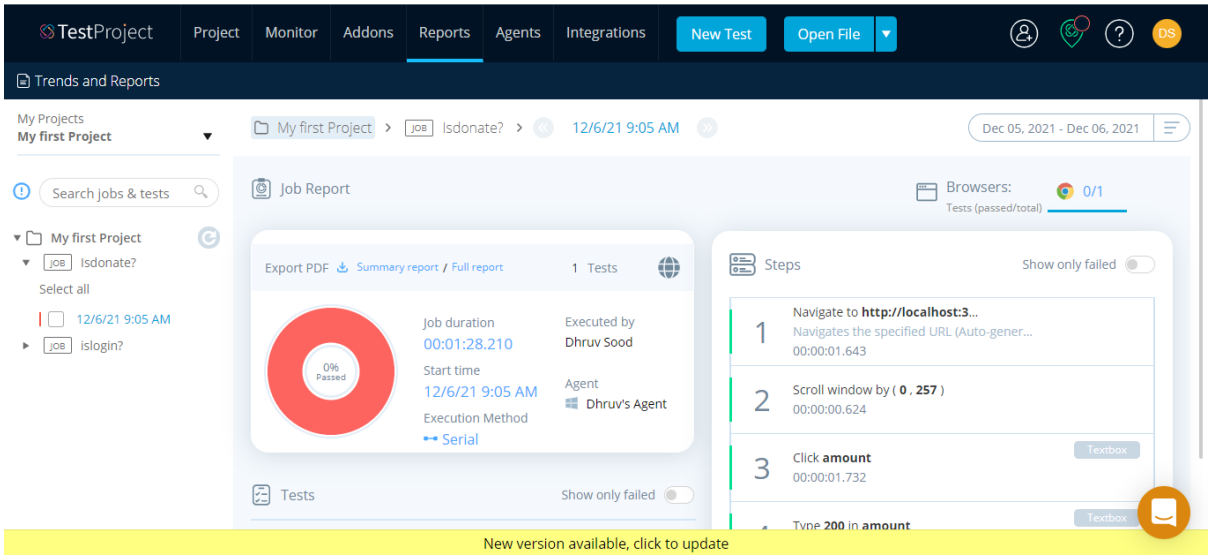
Test results:



Donate now page:

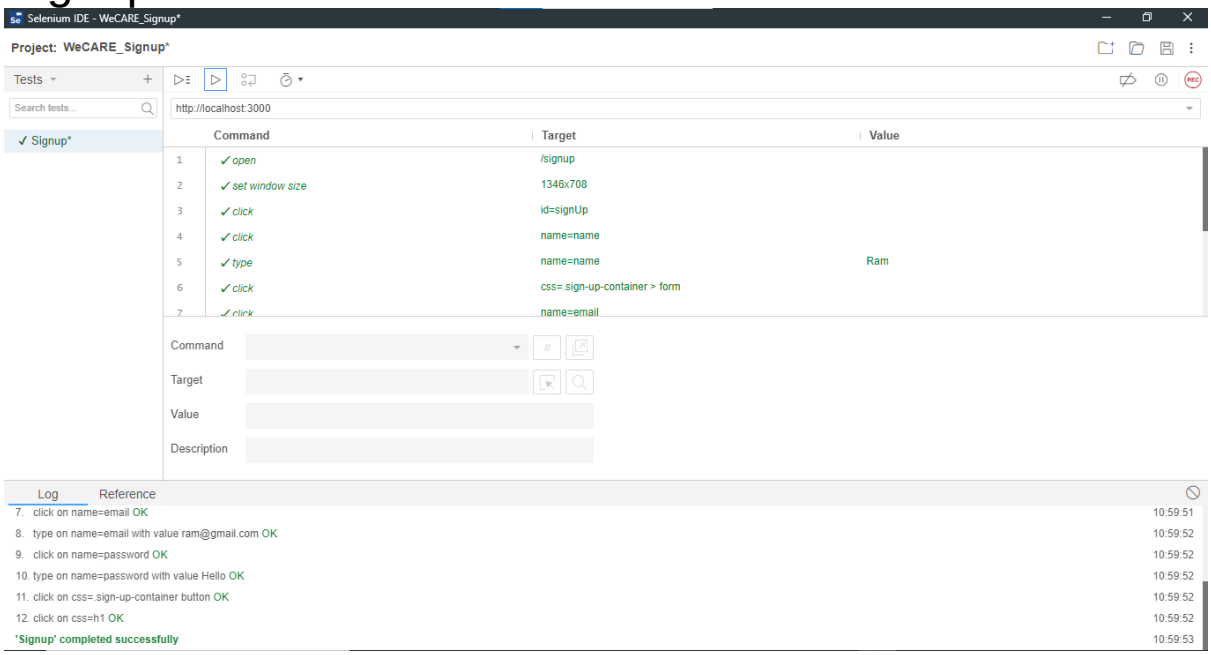


Test results:

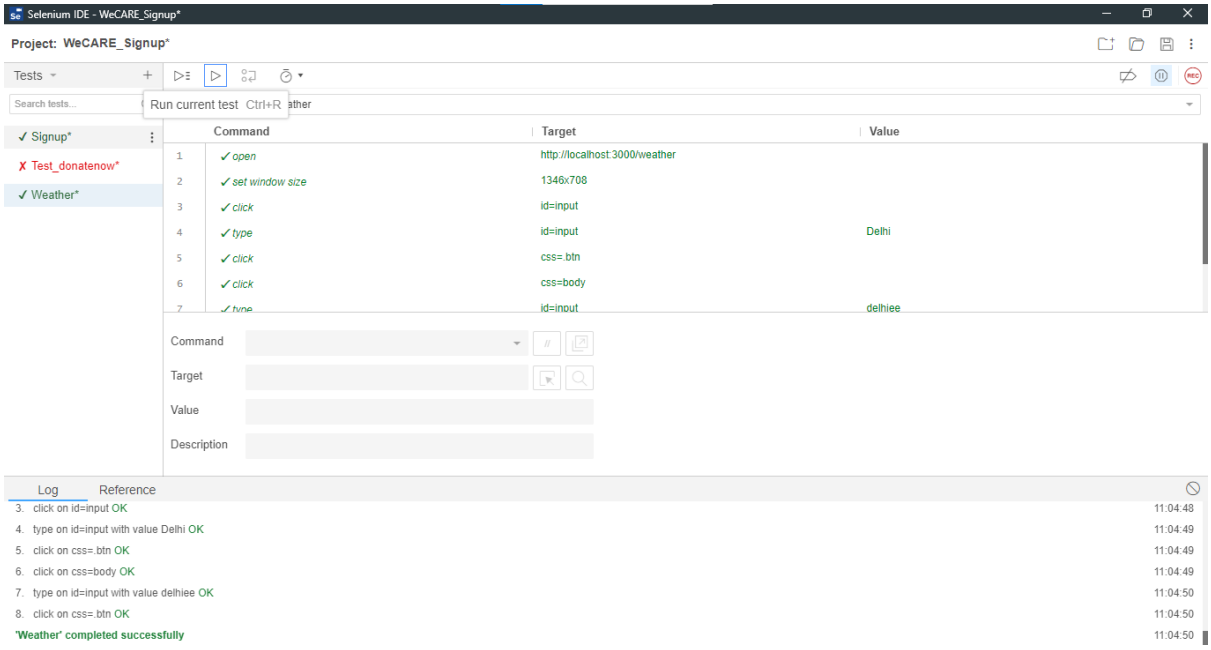


Testing using Selenium IDE:

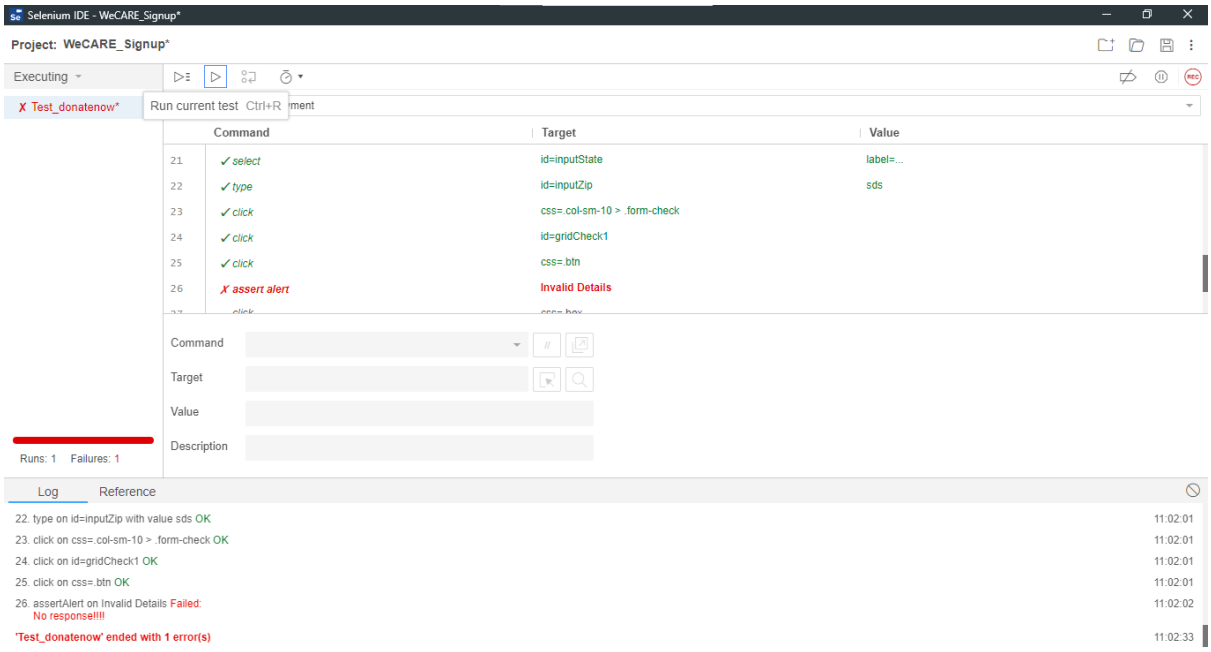
Signup Module:



Weather:



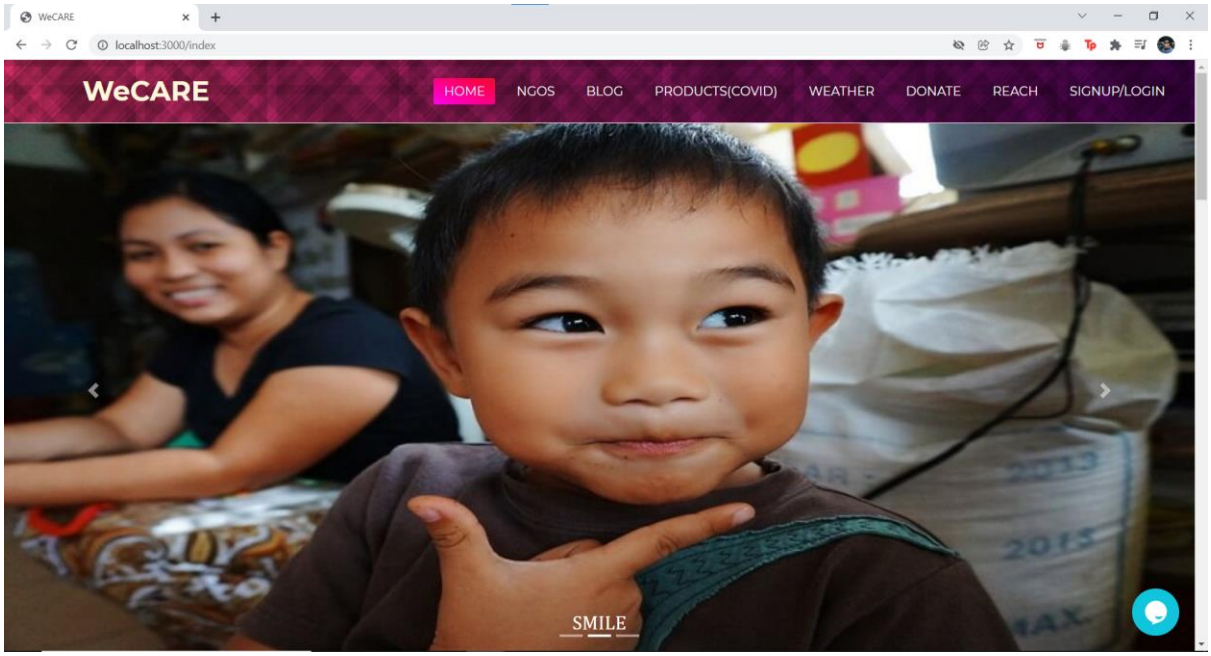
Donate Now:

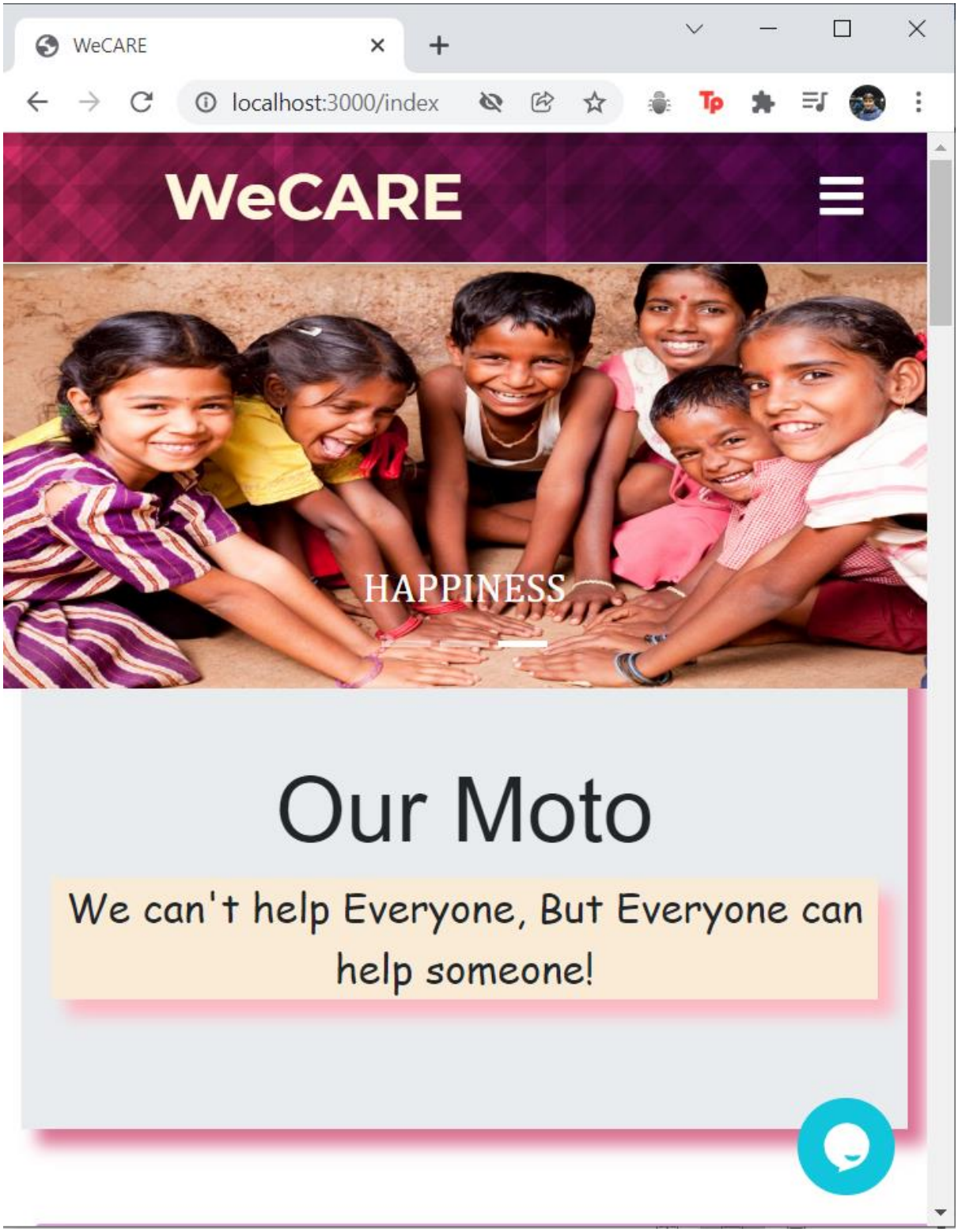


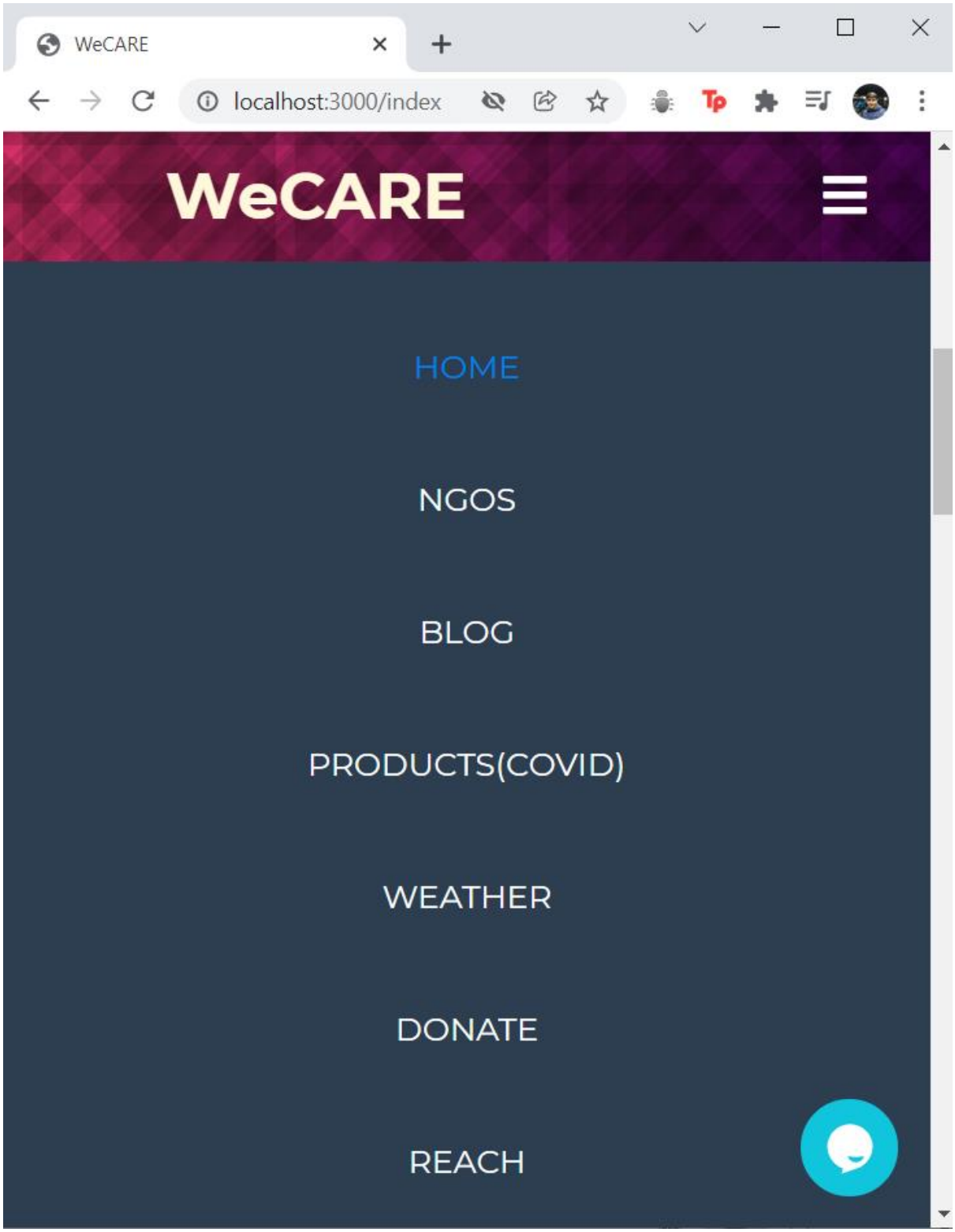
7. Performance Testing

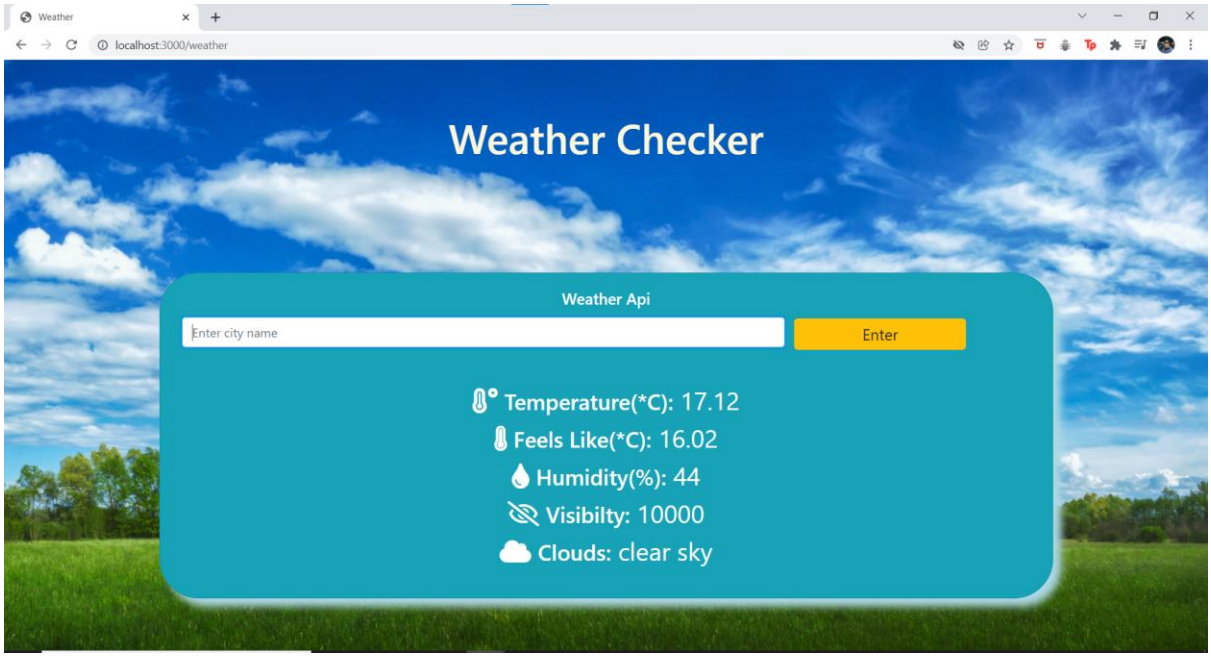
UI testing manually:

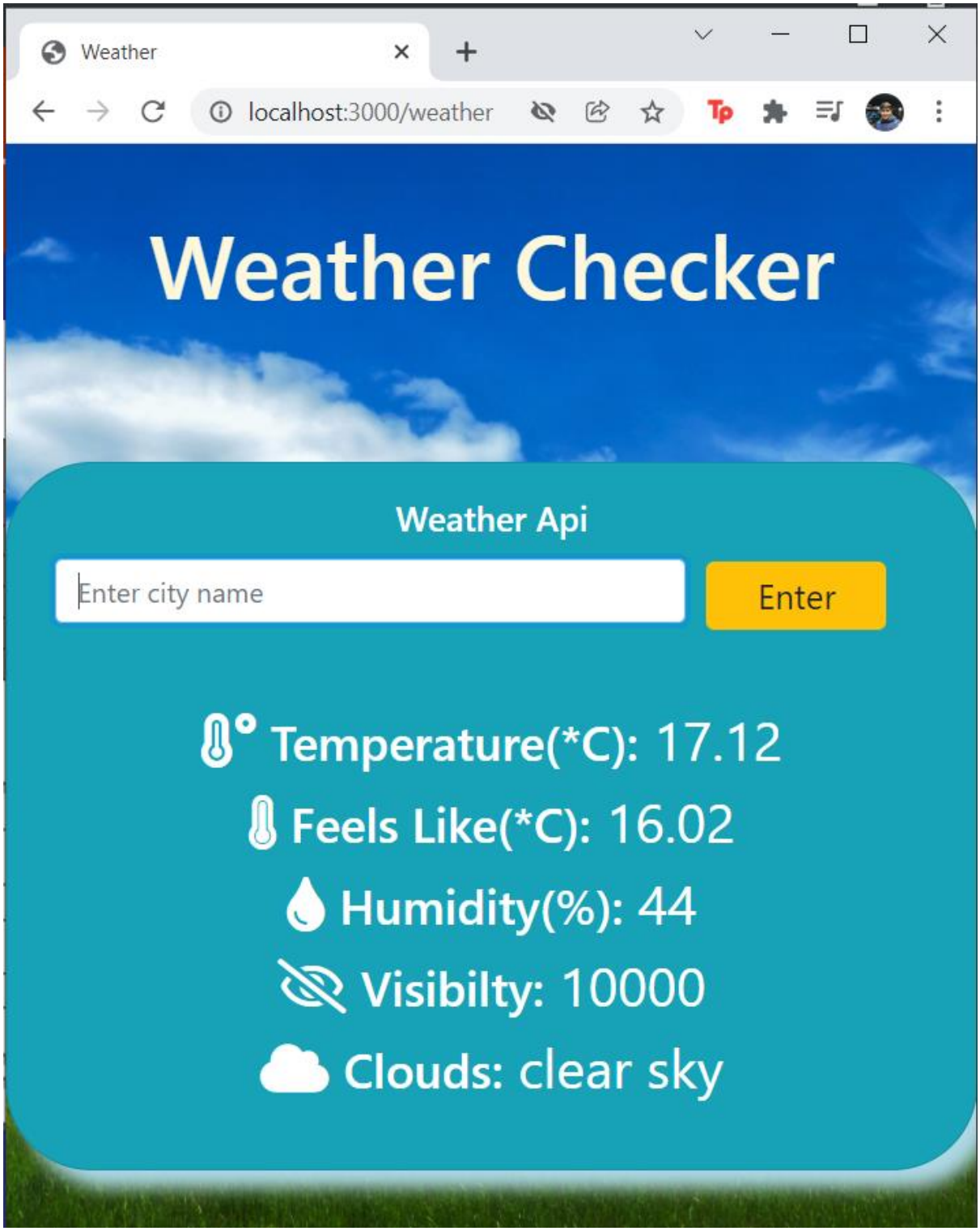
Check if the website is smart phone compatible.


WeCARE
localhost:3000/index
WeCARE
HOME
NGOS
BLOG
PRODUCTS(COVID)
WEATHER
DONATE
REACH
SIGNUP/LOGIN
SMILE

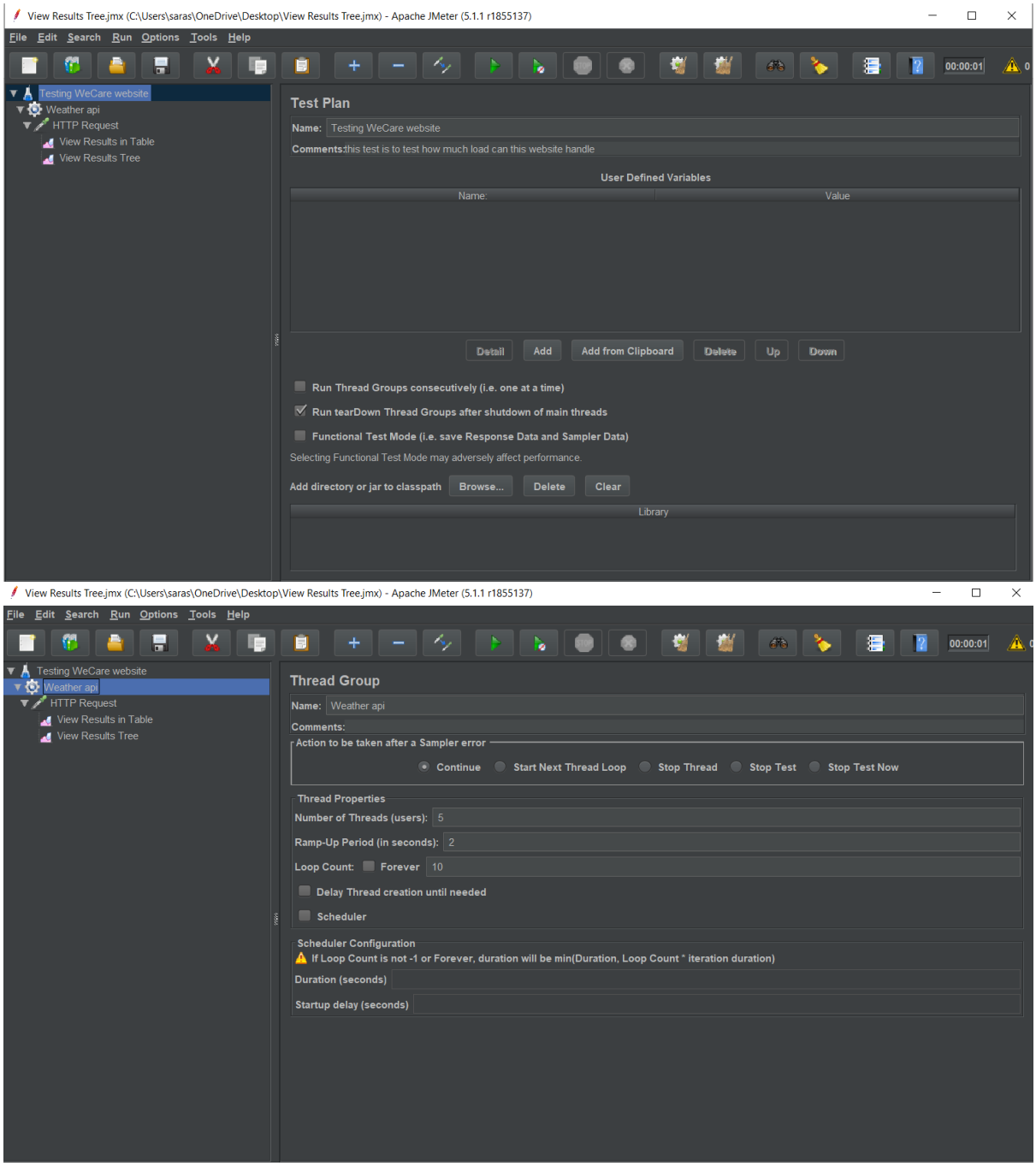

WeCARE
localhost:3000/index
WeCARE
HAPPINESS
Our Moto
We can't help Everyone, But Everyone can help someone!

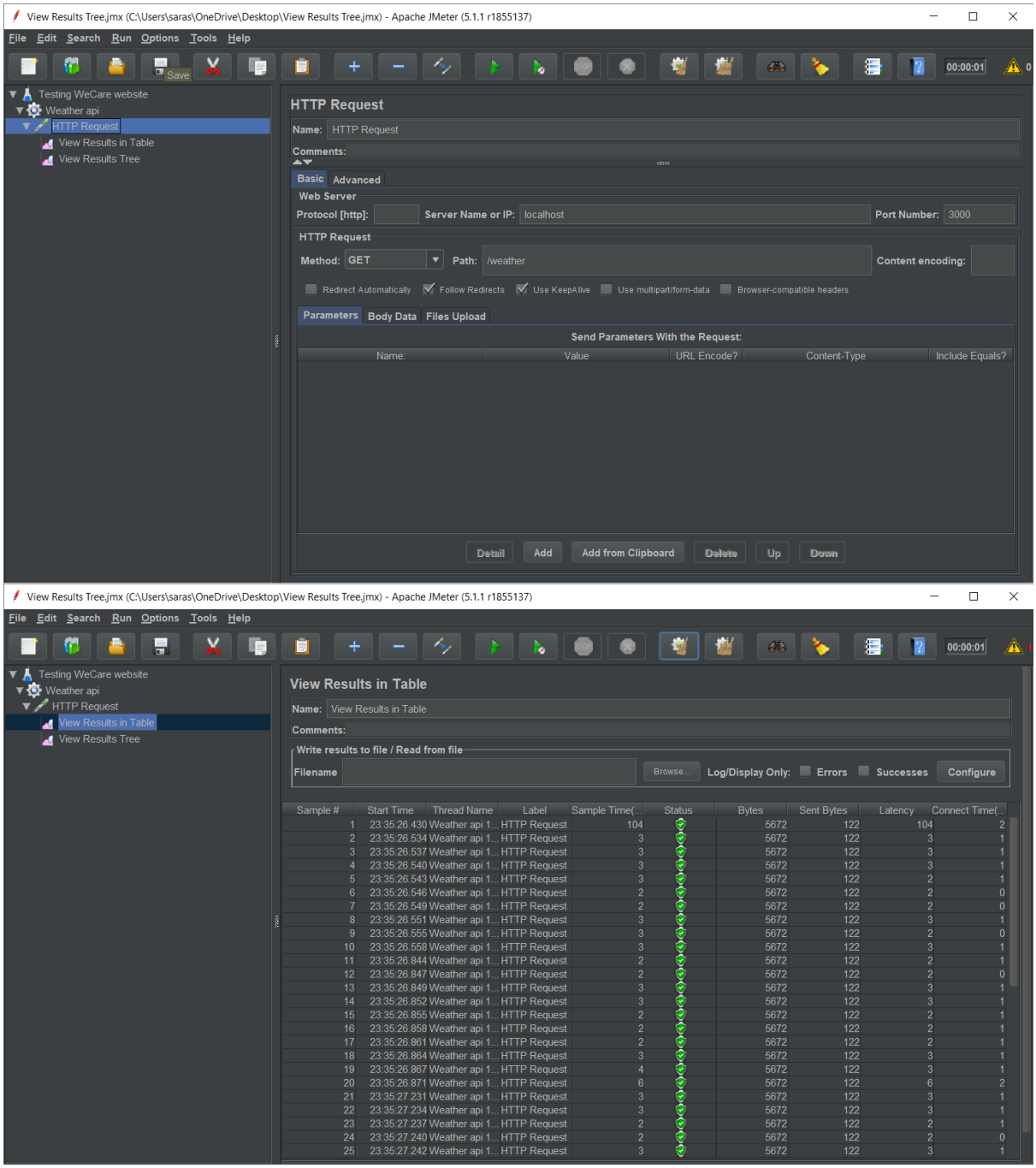

WeCARE
localhost:3000/index
WeCARE
HOME
NGOS
BLOG
PRODUCTS(COVID)
WEATHER
DONATE
REACH


Weather
localhost:3000/weather
Weather Checker
Weather Api
Enter city name
Enter
Temperature(*C): 17.12
Feels Like(*C): 16.02
Humidity(%): 44
Visibilty: 10000
Clouds: clear sky

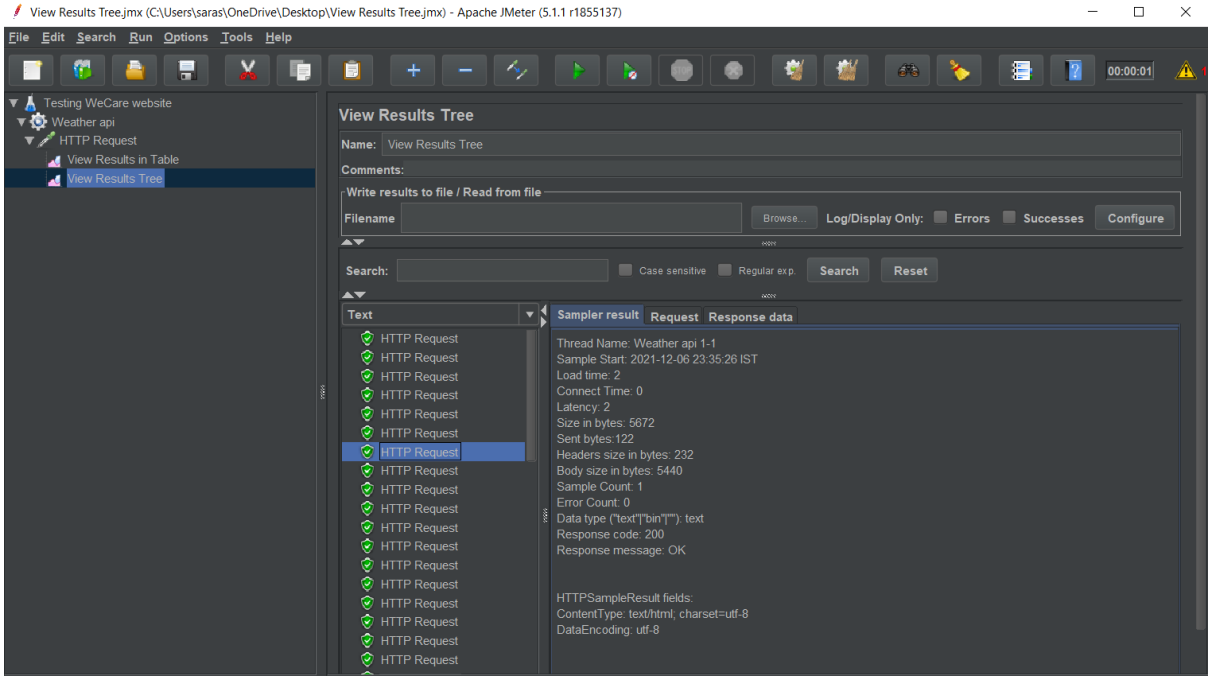


Load Testing WeCare website using jmeter

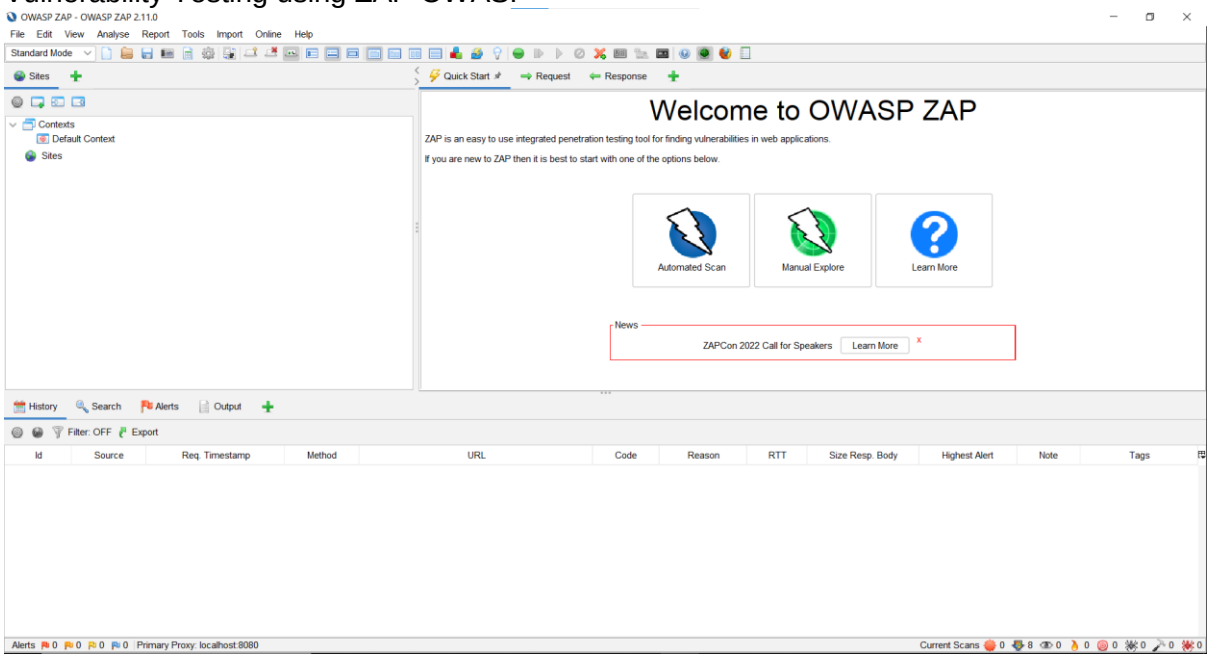

View Results Tree.jmx (C:\Users\saras\OneDrive\Desktop\View Results Tree.jmx) - Apache JMeter (5.1.1 r1855137)
File Edit Search Run Options Tools Help
00:00:01
Testing WeCare website
Weather api
HTTP Request
View Results in Table
View Results Tree
Test Plan
Name: Testing WeCare website
Comments: this test is to test how much load can this website handle
User Defined Variables
Name:
Value
Detail
Add
Add from Clipboard
Delete
Up
Down
Run Thread Groups consecutively (i.e. one at a time)
Run tearDown Thread Groups after shutdown of main threads
Functional Test Mode (i.e. save Response Data and Sampler Data)
Selecting Functional Test Mode may adversely affect performance.
Add directory or jar to classpath
Browse...
Delete
Clear
Library
View Results Tree.jmx (C:\Users\saras\OneDrive\Desktop\View Results Tree.jmx) - Apache JMeter (5.1.1 r1855137)
File Edit Search Run Options Tools Help
00:00:01
Testing WeCare website
Weather api
HTTP Request
View Results in Table
View Results Tree
Thread Group
Name: Weather api
Comments:
Action to be taken after a Sampler error
Continue
Start Next Thread Loop
Stop Thread
Stop Test
Stop Test Now
Thread Properties
Number of Threads (users): 5
Ramp-Up Period (in seconds): 2
Loop Count: Forever 10
Delay Thread creation until needed
Scheduler
Scheduler Configuration
If Loop Count is not -1 or Forever, duration will be min(Duration, Loop Count * iteration duration)
Duration (seconds)
Startup delay (seconds)

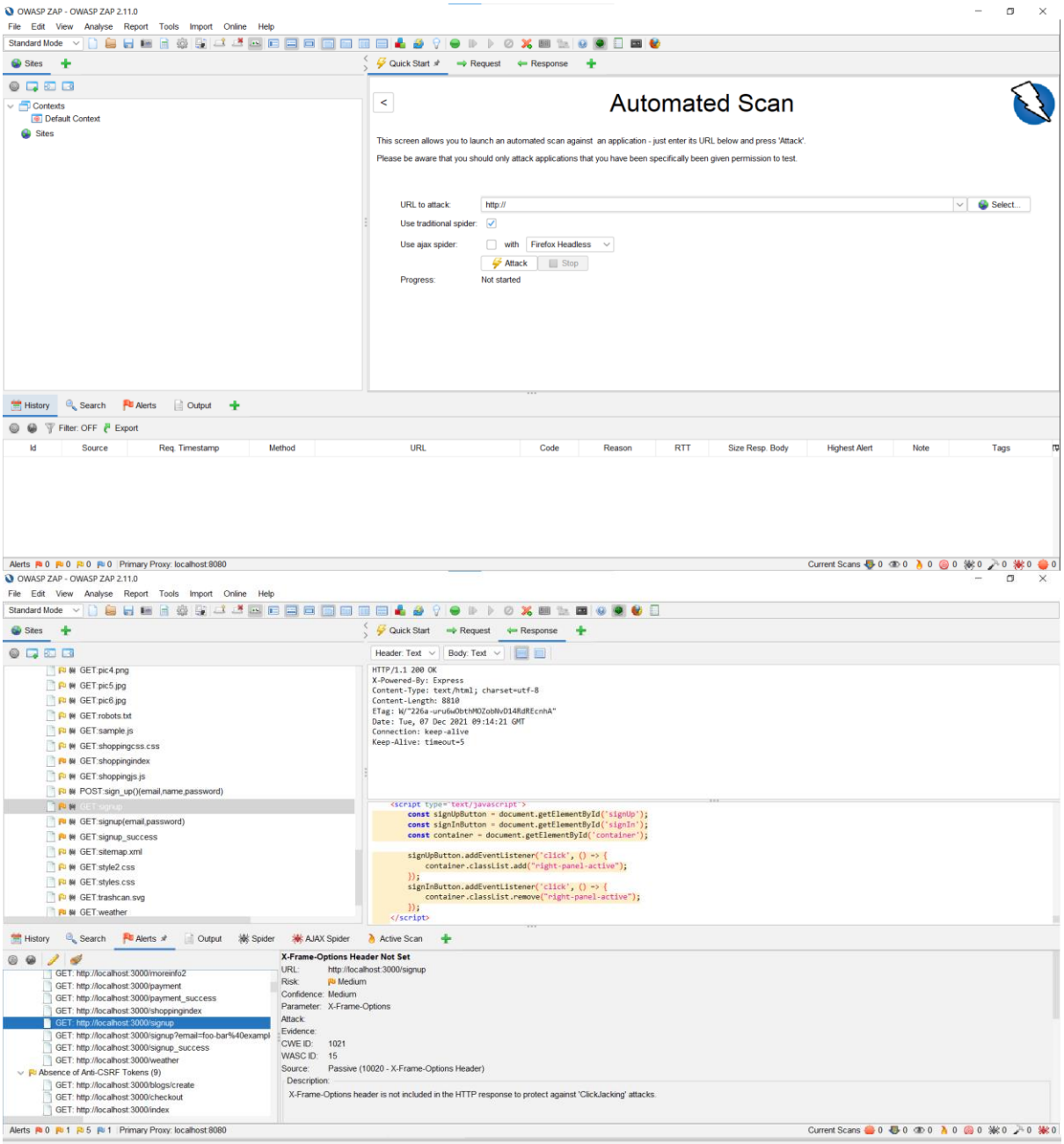

View Results Tree.jmx (C:\Users\saras\OneDrive\Desktop\View Results Tree.jmx) - Apache JMeter (5.1.1 r1855137)
File Edit Search Run Options Tools Help
Save
00:00:01
Testing WeCare website
Weather api
HTTP Request
View Results in Table
View Results Tree
HTTP Request
Name: HTTP Request
Comments:
Basic Advanced
Web Server
Protocol [http]:
Server Name or IP: localhost
Port Number: 3000
HTTP Request
Method: GET
Path: /weather
Content encoding:
Redirect Automatically
Follow Redirects
Use KeepAlive
Use multipart/form-data
Browser-compatible headers
Parameters Body Data Files Upload
Send Parameters With the Request:
Name:
Value
URL Encode?
Content-Type
Include Equals?
Detail
Add
Add from Clipboard
Delete
Up
Down

View Results Tree.jmx (C:\Users\saras\OneDrive\Desktop\View Results Tree.jmx) - Apache JMeter (5.1.1 r1855137)
File Edit Search Run Options Tools Help
00:00:01
Testing WeCare website
Weather api
HTTP Request
View Results in Table
View Results Tree
View Results in Table
Name: View Results in Table
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only:
Errors
Successes
Configure
Sample # Start Time Thread Name Label Sample Time(... Status Bytes Sent Bytes Latency Connect Time(...
1 23:35:26.430 Weather api 1... HTTP Request 104 5672 122 104 2
2 23:35:26.534 Weather api 1... HTTP Request 3 5672 122 3 1
3 23:35:26.537 Weather api 1... HTTP Request 3 5672 122 3 1
4 23:35:26.540 Weather api 1... HTTP Request 3 5672 122 3 1
5 23:35:26.543 Weather api 1... HTTP Request 3 5672 122 2 1
6 23:35:26.546 Weather api 1... HTTP Request 2 5672 122 2 0
7 23:35:26.549 Weather api 1... HTTP Request 2 5672 122 2 0
8 23:35:26.551 Weather api 1... HTTP Request 3 5672 122 3 1
9 23:35:26.555 Weather api 1... HTTP Request 3 5672 122 2 0
10 23:35:26.558 Weather api 1... HTTP Request 3 5672 122 3 1
11 23:35:26.844 Weather api 1... HTTP Request 2 5672 122 2 1
12 23:35:26.847 Weather api 1... HTTP Request 2 5672 122 2 0
13 23:35:26.849 Weather api 1... HTTP Request 3 5672 122 3 1
14 23:35:26.852 Weather api 1... HTTP Request 3 5672 122 3 1
15 23:35:26.855 Weather api 1... HTTP Request 2 5672 122 2 1
16 23:35:26.858 Weather api 1... HTTP Request 2 5672 122 2 1
17 23:35:26.861 Weather api 1... HTTP Request 2 5672 122 2 1
18 23:35:26.864 Weather api 1... HTTP Request 3 5672 122 3 1
19 23:35:26.867 Weather api 1... HTTP Request 4 5672 122 3 1
20 23:35:26.871 Weather api 1... HTTP Request 6 5672 122 6 2
21 23:35:27.231 Weather api 1... HTTP Request 3 5672 122 3 1
22 23:35:27.234 Weather api 1... HTTP Request 3 5672 122 3 1
23 23:35:27.237 Weather api 1... HTTP Request 2 5672 122 2 1
24 23:35:27.240 Weather api 1... HTTP Request 2 5672 122 2 0
25 23:35:27.242 Weather api 1... HTTP Request 3 5672 122 3 1



Vulnerability Testing using ZAP OWASP





X-Frame-Options Header Not Set

URL: http://localhost:3000/index

Risk: Medium

Confidence: Medium

Parameter: X-Frame-Options

Attack:

Evidence:

CWE ID: 1021

WASC ID: 15

Source: Passive (10020 - X-Frame-Options Header)

Description:

X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks.

- **Test log identifier:**

  TL1
- **Description**

  Login module in 3.4 Functional requirements in SRS document. The functional testing is done in online environment with internet.
- **Activity and Event Entries**

Date: 28.11.2021
Author: Dhruv Sood
Test case ID: T1
Name of personnel involved in testing: Dhruv Sood, Tanishq
The function was tested with the following inputs

Input		Results	Status
Email ID	Password		
ddhruv2@gmail.com	User1	Logged in	Passed
anishq@gmail.com	User2	Logged in	Passed

- **Test log identifier:**

  TL2
- **Description**

Home page contact us module in 3.4 Functional requirements in SRS document. The functional testing is done in online environment with internet.

- **Activity and Event Entries**

Date: 28.11.2021
Author: Dhruv Sood
Test case ID: T2
Name of personnel involved in testing: Dhruv Sood, Tanishq
The function was tested with the following inputs

Input		Results	Status
Email ID	Message		
ooddhruv2@gmail.com	Hi	Message sent	Passed
Tanishq@gmail.com	Hello please reply	Message sent	Passed

- **Test incident Identifier**

TI1

- **Description**

i. Date and time: 1-12-2021, 2.00pm
ii. Testing personnel names: Tanishq, Dhruv sood
iii. Environment: Online environment with X database
iv. Testing Inputs
v. Expected Outputs
vi. Actual Outputs
vii. Anomalies detected during the test
viii. Attempts to repeat the same test

Testing inputs	Expected outputs	Actual outputs	Anomalies detected	Attempts to

				repeat the same test
Clicking on volunteer now button	Redirection towards the Form page to collect all the details of the volunteer	No redirection.	Non - functioning button is placed	3
Adding blogs	Blog that are added should be proof checked and have some content	Blogs with no data or numeric data or gibberish can be added simply	No validations performed	4
Date of birth input in Payment page	Correct dates to be accepted	Dates like 31st November, which doesn't exist are accepted	Proper data time validation missing	3
Cities in payment page	Drop down menu should contain all the cities in the	Drop down menu doesn't have any cities listed	Because of this proper submission of form is difficult	5

	specific country		and it gives error every time while submitting the form	

- **Impact**

The severity value is medium 2

Thank you!