NITTE EDUCATION TRUST
N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India
(ISO 9001:2015 Certified), Accredited with 'A' Grade by NAAC
08258 - 281039 – 281263, Fax: 08258 – 281265
Department of Computer Science and Engineering
B.E. CSE Program Accredited by NBA, New Delhi from 1-7-2018 to 30-6-2021

REPORT ON

LAMBDATEST

Course Code: 18CS701

Academic Year – 2021-2022

Course Name: **SOFTWARE TESTING**

Semester: 7 Section: C

Submitted To,
Course Instructor:

Mrs. Rajalaxmi S Hegde
Assistant Professor-II
Department of CSE,
NMAMIT, Nitte.

Submitted By:

1. **USN**: 4NM18CS140 **NAME**: S DHRUVA
2. USN: 4NM18CS152 **NAME**: SAMIT D MANVAR

Date of submission: 30-11-2021

TABLE OF CONTENTS

INTRODUCTION

What is Cross Browser Compatibility?

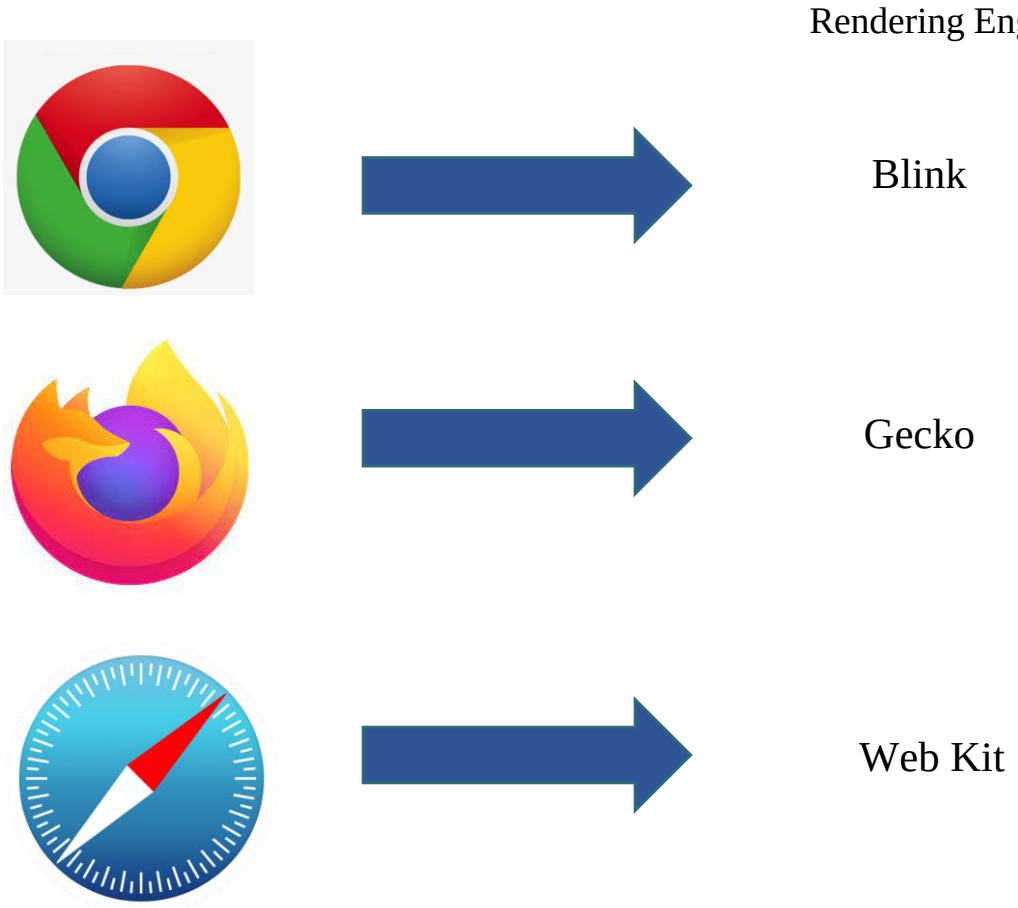
Cross Browser compatibility is the ability of a website or web application to function across different browsers and operating systems. Compatibility testing is a part of non-functional testing.

Why do we need Cross Browser Compatibility?

Once the application is stable, we moved it to the production, it may be used or accessed by multiple users on the different platforms, and they may face some compatibility issues, to avoid these issues, we do one round of compatibility testing. A simple website is comprised of 3 major technologies HTML, CSS and JAVASCRIPT. In the backend different technologies are used. But overall, in the front end in the rendering only these 3 main technologies are used. Each browser used different rendering engine. We need to understand that cross-browser compatibility is more related to site's functionality than its looks. This means the site should perform equally under various web environments. Every website owner should ensure cross-browser compatibility of their website or applications. Each individual has its own preferred web browser hence there are fewer chances that the website developer and site visitor will use the same browser.

INTRODUCTION

A rendering engine is software that draws text and images on the screen. The engine draws structured text from a document (often HTML), and formats it properly based on the given style declarations (often given in CSS). Examples of layout engines: Blink, Gecko, Edge HTML, Web Kit.



Cross Browser testing is a type of non-functional testing that lets you check whether your website works as intended when accessed through:

- **Different Browser-OS combinations** i.e., on popular browsers like Firefox, Chrome, Edge, Safari—on any of the popular operating systems like Windows, macOS, iOS and Android.
- **Different devices** i.e., users can view and interact with your website on popular devices—smartphones, tablets, desktops and laptops etc.
- **Assistive Tools** i.e., the website is compatible with assistive technologies like screen readers for individuals who are differently abled.

LAMBDA TEST

What is Lambda Test?

- Lambda Test is a scalable cloud-based cross browser testing platform.
- Designed to offer all website or web app testing need to cloud infrastructure.
- Lambda Test platform helps you to ensure your web app elements (such as JavaScript, CSS, HTLM5, Video etc.) render seamlessly across every desktop and mobile web browser
- Can test on 2000+ Real Browsers and operating systems.

Features

- It supports automated cross-browser testing across 2,000+ real browsers, platforms, and device emulators.
- Supports Parallel Execution
- It lets you check the responsiveness of the website or web application across different devices (iOS and Android) and viewport sizes.
- Locally hosted web testing.
- Supports automation testing with Selenium 4, the latest release of the Selenium framework.

Types of testing

- Real time Testing
- LT Browsing Testing
- Automation Testing
- Screenshot Testing
- Integration Testing

DEMONSTRATION

- Since it is a cloud based cross browser testing no additional installation is needed.
- In order to use Lambda Test, visit the website https://www.lambdatest.com/
- Lambda Test application provides free version to users. To avail free version, you need to create an account either by signing up with google account or by email.
- Once your account is successfully created you are ready to use the application to perform different tests.

Dashboard

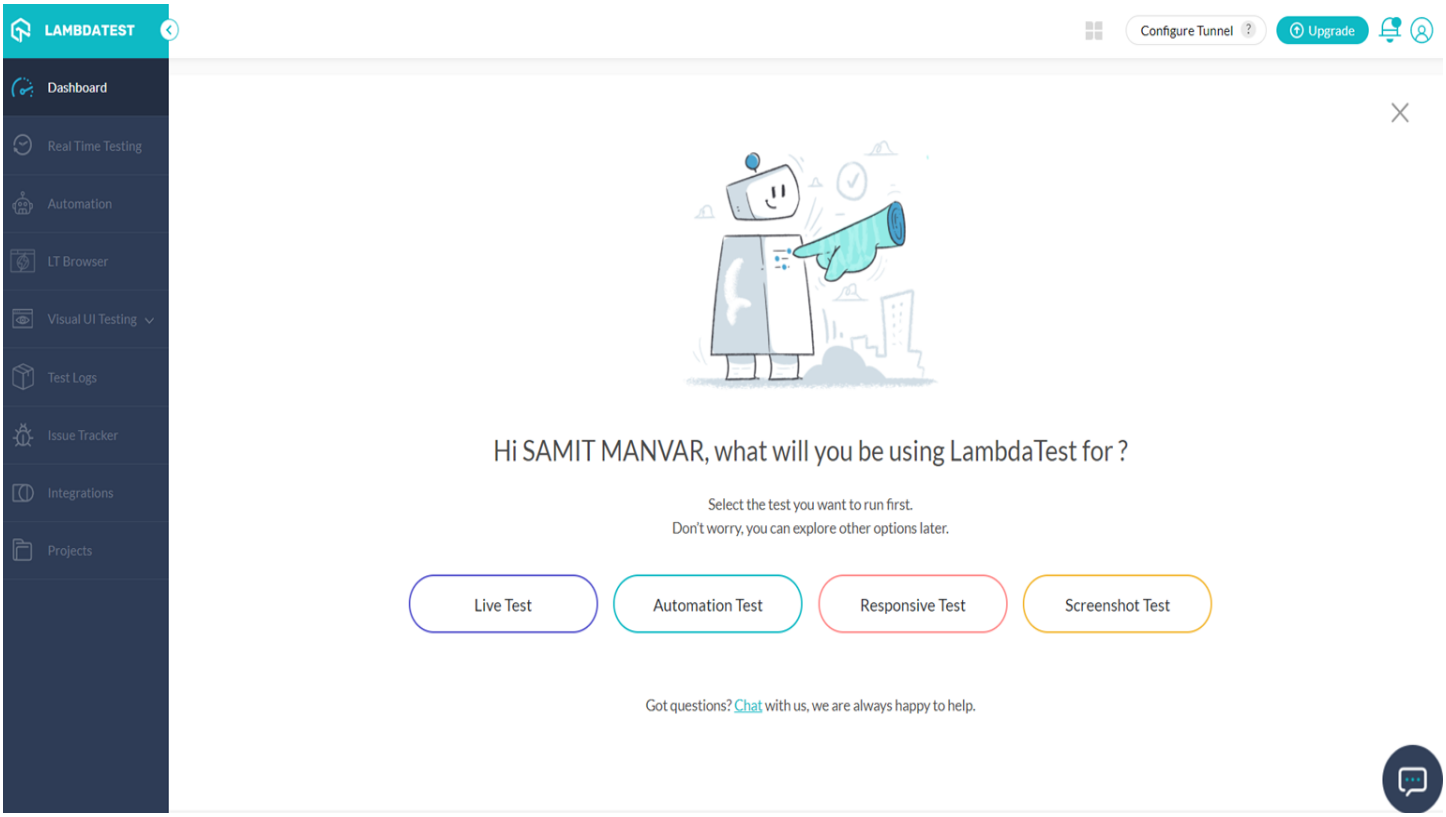


Fig 1: Dashboard of Lambda Test

REAL TIME TESTING (LIVE TESTING)

Real-Time Testing allows you to perform live-interactive cross browser testing of your website on more than 2000+ real browsers & browser versions running on various operating systems in the cloud. Lambda Test provides you real-time testing experience by virtual machines hosted on our cloud servers.

In order to perform real time testing place the URL of the web application you want to test , select the browser , browser version , operating system and resolution and then click on start.

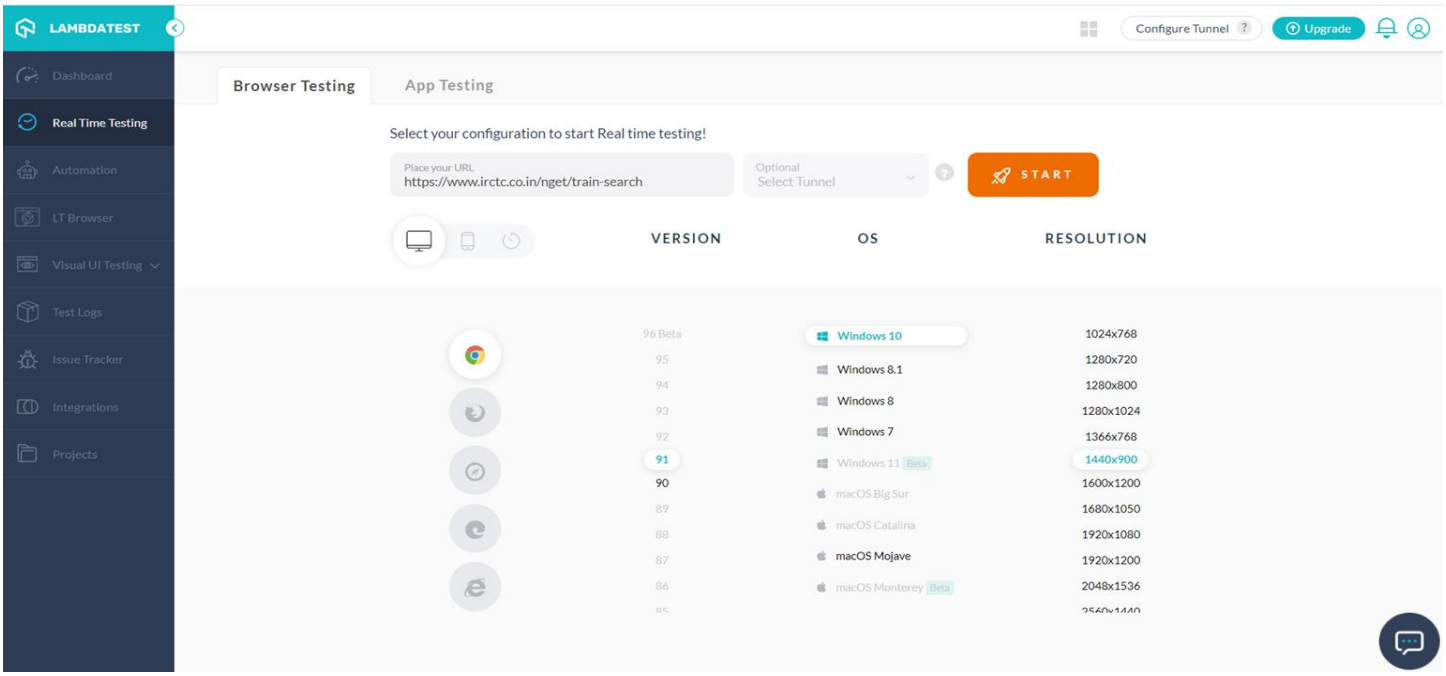


Fig 2: Real Time Testing

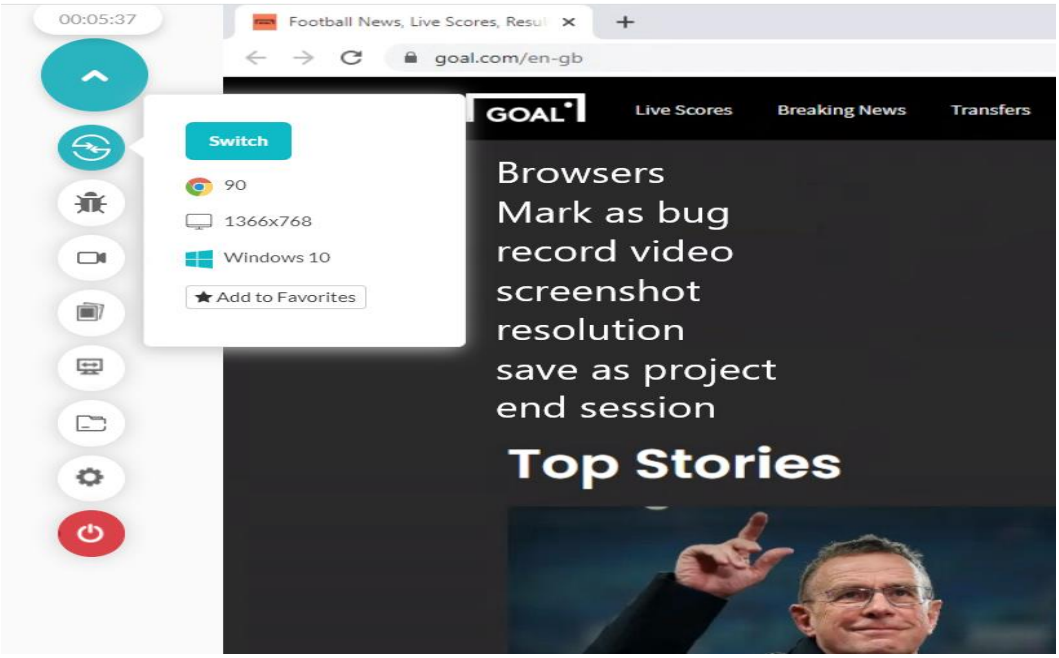


Fig 3: Different options available under Real Time Testing

We can also mark a bug by selecting mark as bug option available.

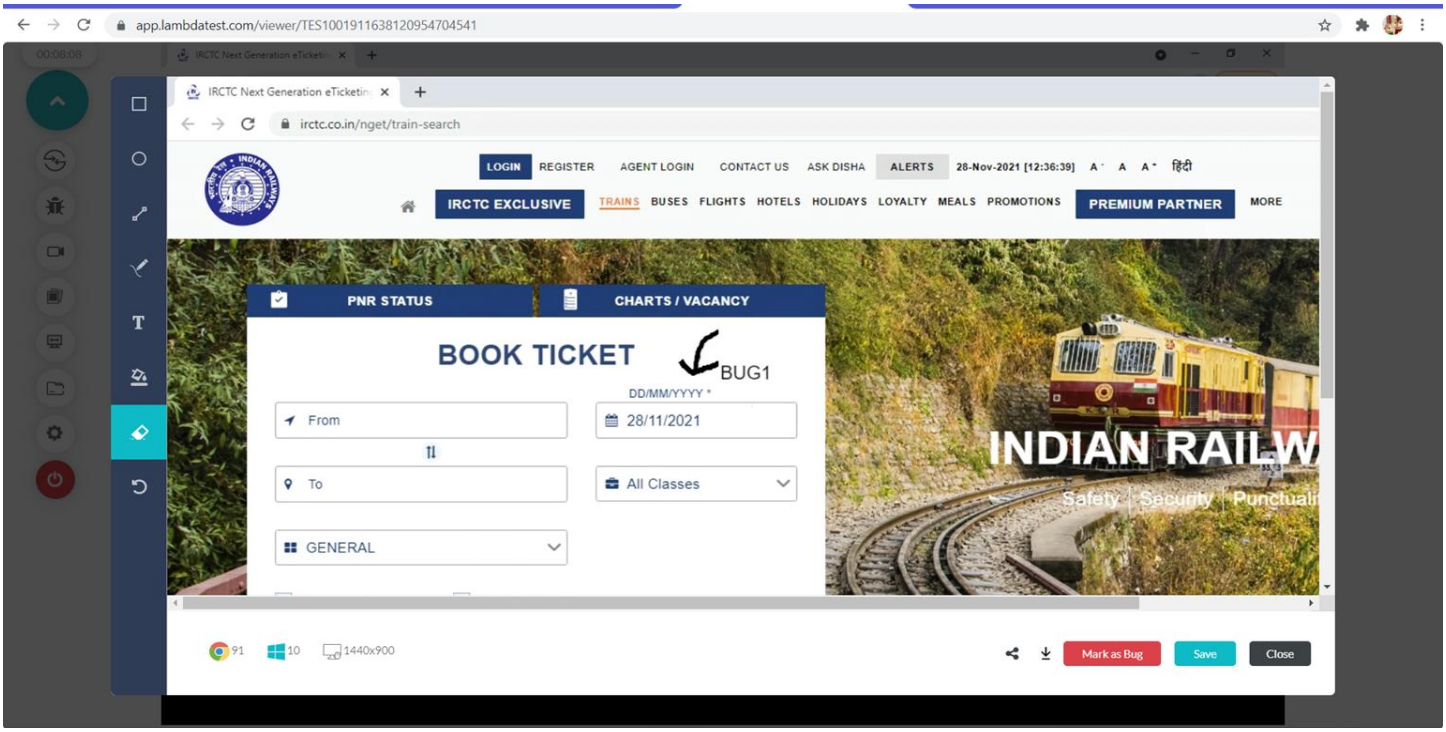


Fig 4: Mark a bug

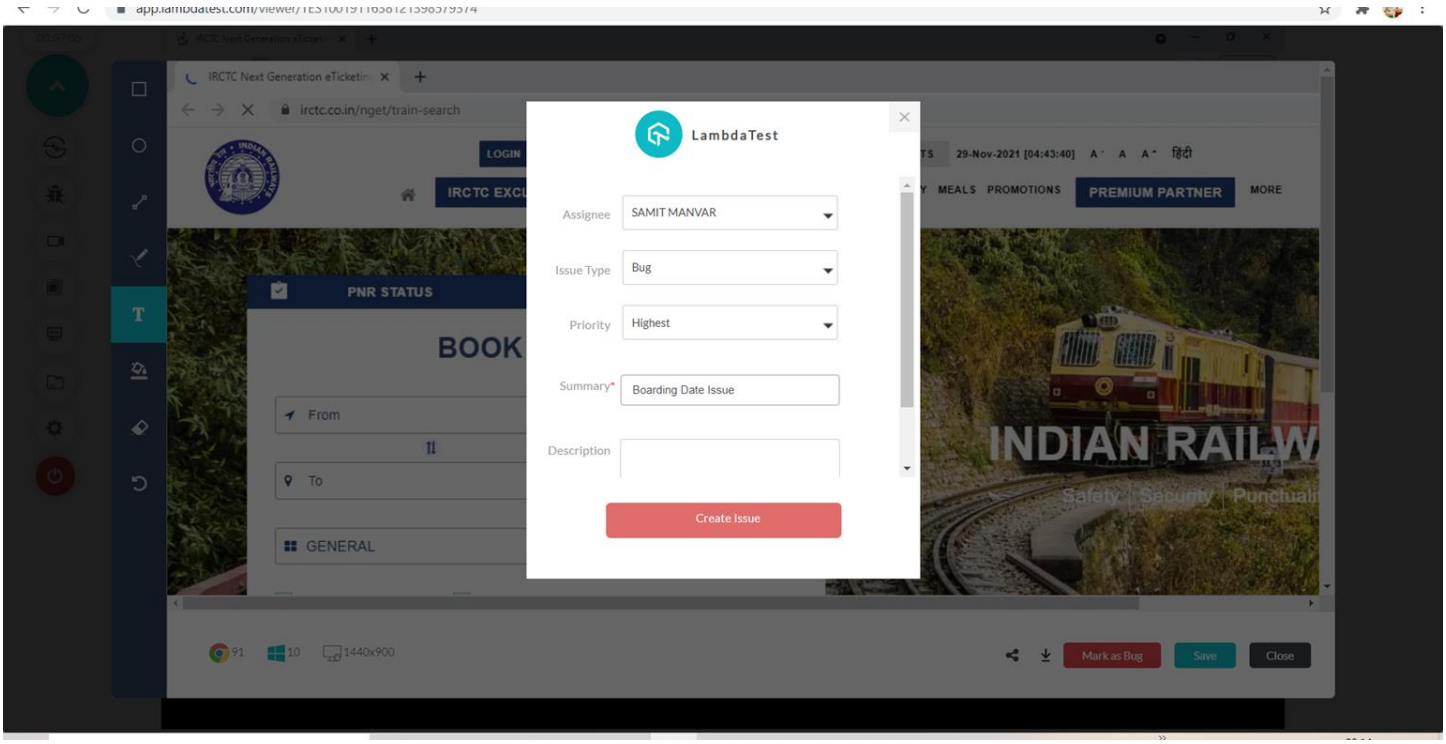


Fig 5: Write issues after finding a bug

LT BROWSER TESTING

LT Browser allows you to ensure your website's responsiveness over a variety of major devices and view ports. You can open a website in the LT Browser and perform live testing across 50+ pre-installed device viewports such as mobile devices, laptops, tablets etc.

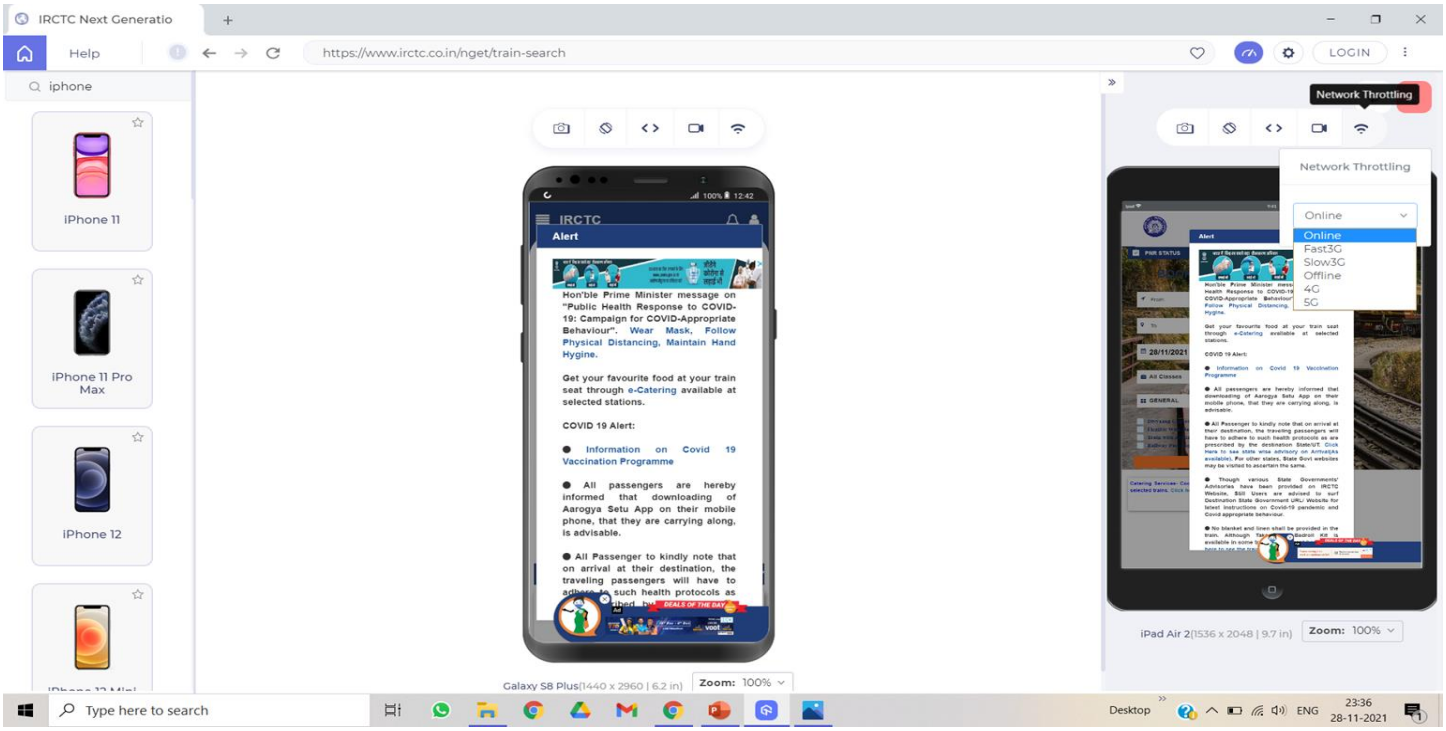


Fig 6: Responsive Test for mobile devices

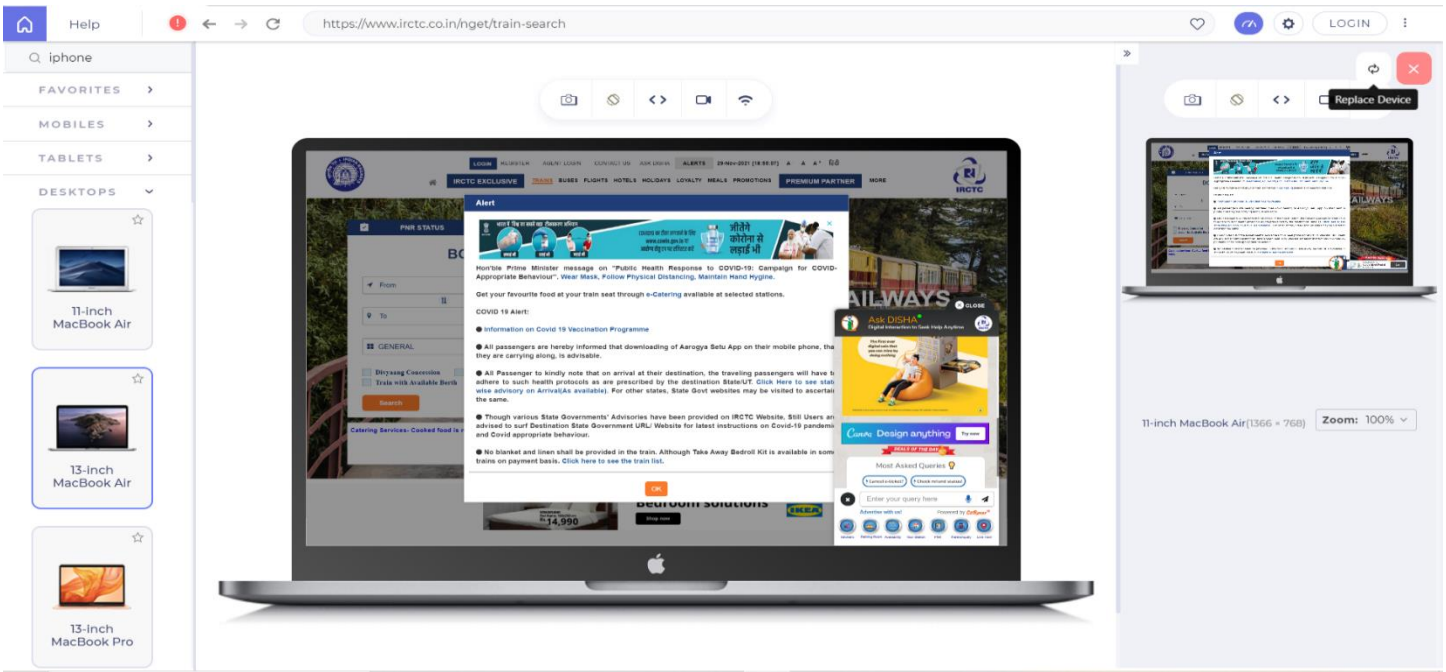


Fig 7: Responsive test for Laptop Devices

TEST LOGS

Test Logs is a library where you can find logs of all cross-browser testing performed by you on Lambda Test.

Accessing Test Logs

Step 1: Log in to your account.

Step 2: Select '**Test Logs**' from the menu bar on the left-hand side.

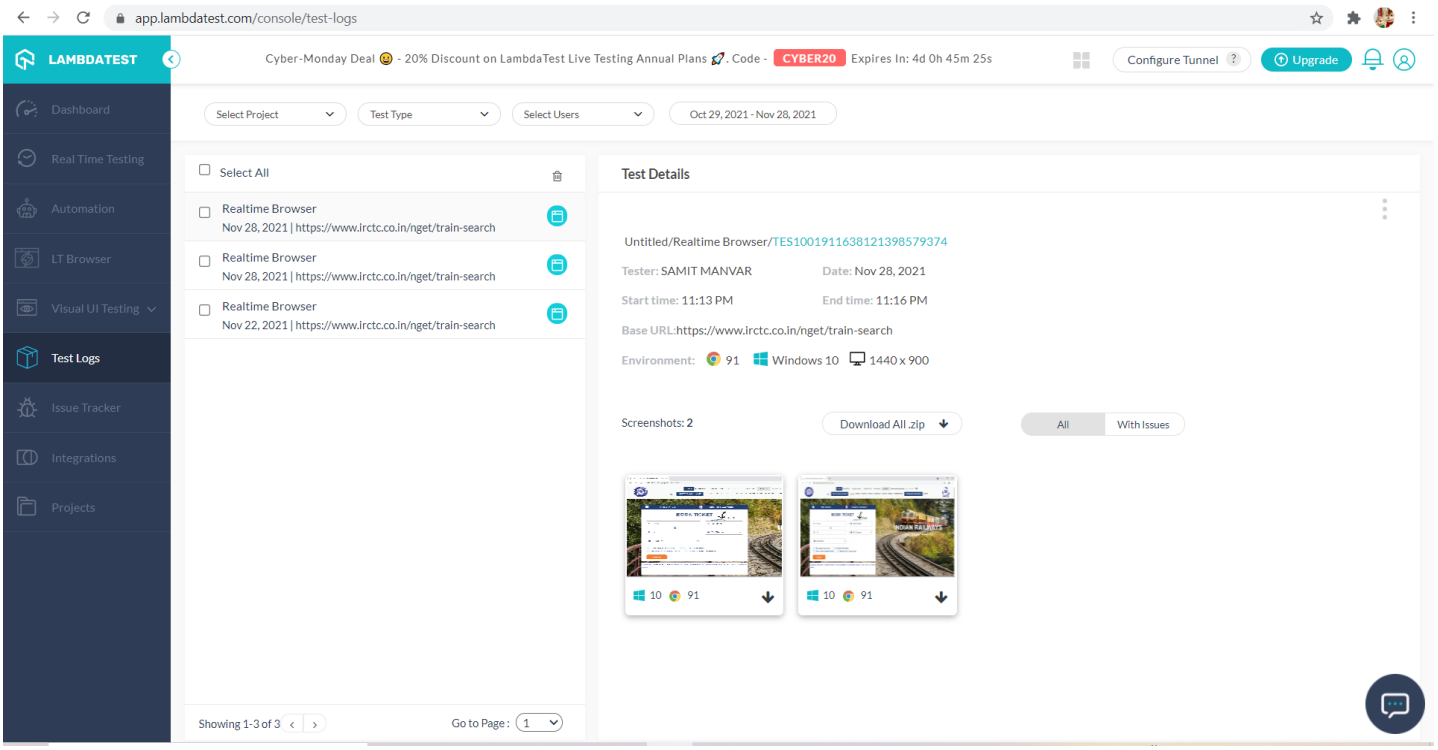


Fig 8: Test Logs with Test Details

AUTOMATION TESTING

Automation testing is the process of testing software and other tech products to ensure it meets strict requirements. Essentially, it's a test to double-check that the equipment or software does exactly what it was designed to do. It tests for bugs, defects, and any other issues that can arise with product development.

Automation testing with an example: -

Task is to mark two items as done in a to-do list website and also add a new item to the list.

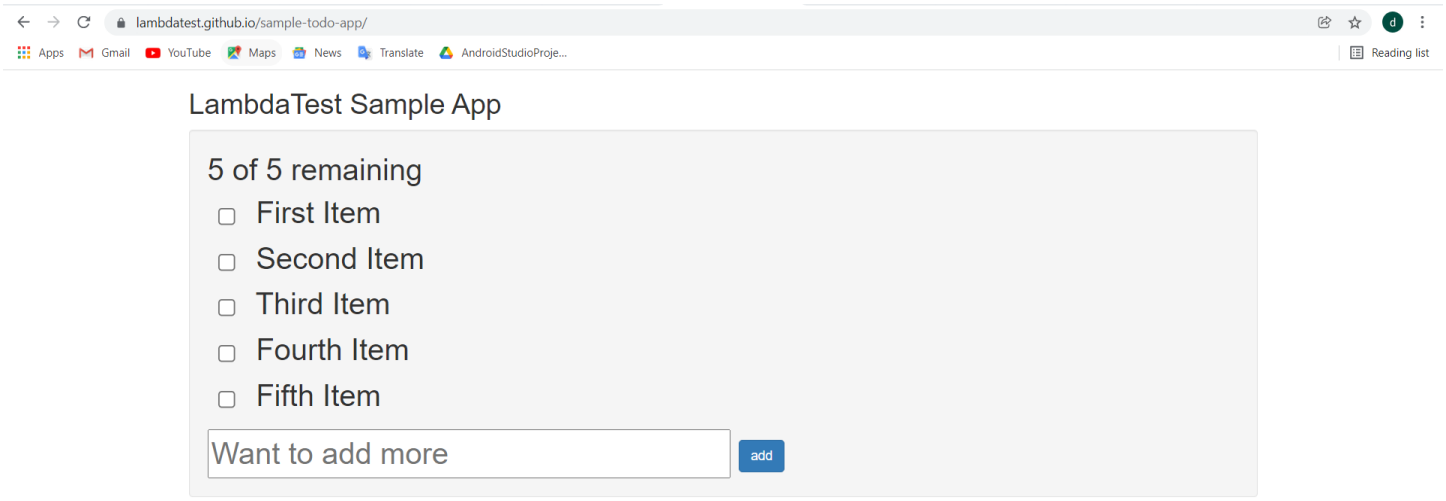


Fig 9: Task Performed by Automation Testing

STEPS INVOLVED

- 1)Driver code
  The driver code with lambda test credentials.

- 2)setting up
  Setup to select the browser, operating system.

- 3)Testing
  Actual testing.
  Open the website and mark two items as done.
  Then add a new item to the list.
  Now check if there is a new item added to the list.

- 4)finishing
  If new item is added to the list the test is completed successfully and the result is logged.
  Or else the test is marked as failed in the logs.

CODE

The java code written with the help of selenium ide and lambda test is as follows.

```
import java.net.MalformedURLException;
import java.net.URL;
import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.remote.DesiredCapabilities;
import org.openqa.selenium.remote.RemoteWebDriver;
public class JavaTodo {
    String username = "dhruwaah";
    String accesskey = "eRKxKjJ7gjD7wn1StYm3f2g0Hj0r7r28bDQhKoH8wOZNtnkrZk";
    static RemoteWebDriver driver = null;
    String gridURL = "@hub.lambdatest.com/wd/hub";
    boolean status = false;
    public static void main(String[] args) {
        new JavaTodo().test();
    }
```

```
    public void test() {
        // To Setup driver
        setUp();
        try {
              //Change it to production page
            driver.get("https://lambdatest.github.io/sample-todo-app/");

              //Let's mark done first two items in the list.
              driver.findElement(By.name("li1")).click();
            driver.findElement(By.name("li2")).click();

             // Let's add an item in the list.
              driver.findElement(By.id("sampletodotext")).sendKeys("Yey, Let's add it to list");
            driver.findElement(By.id("addbutton")).click();

              // Let's check that the item we added is added in the list.
            String enteredText = driver.findElementByXPath("/html/body/div/div/div/ul/li[6]/span").getText();
            if (enteredText.equals("Yey, Let's add it to list")) {
                status = true;
            }
        } catch (Exception e) {
            System.out.println(e.getMessage());
        } finally {
            tearDown();
        }
    }
```

```
private void setUp() {
    DesiredCapabilities capabilities = new DesiredCapabilities();
    capabilities.setCapability("browserName", "chrome");
    capabilities.setCapability("version", "70.0");
    capabilities.setCapability("platform", "win10"); // If this cap isn't specified, it will just get any availa
    capabilities.setCapability("build", "LambdaTestSampleApp");
    capabilities.setCapability("name", "LambdaTestJavaSample");
    try {
        driver = new RemoteWebDriver(new URL("https://" + username + ":" + accesskey + gridURL), capabilities);
    } catch (MalformedURLException e) {
        System.out.println("Invalid grid URL");
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
}
private void tearDown() {
    if (driver != null) {
        ((JavascriptExecutor) driver).executeScript("lambda-status=" + status);
        driver.quit(); //really important statement for preventing your test execution from a timeout.
    }
}
```

The record of the test execution is saved in test logs as follows: -

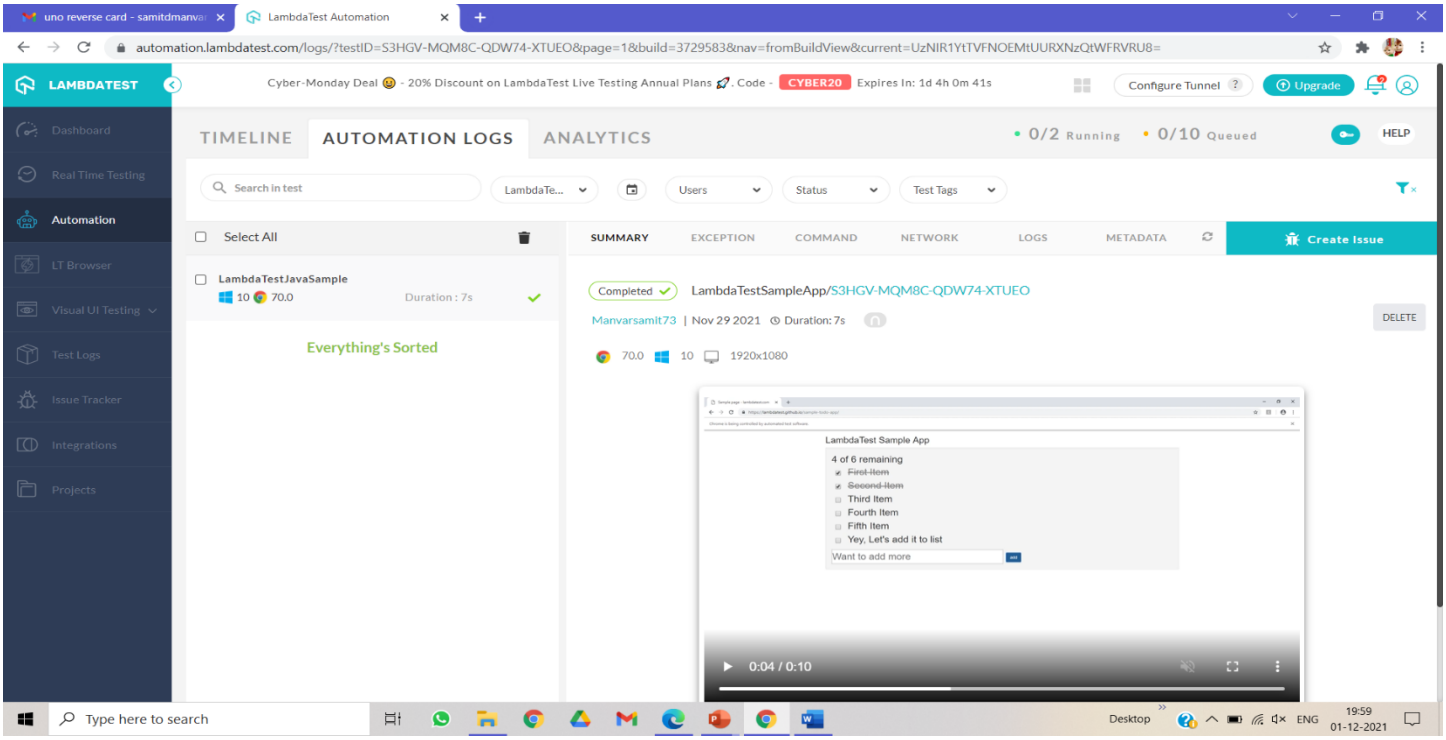


Fig 10: Test Logs of Automation Testing

SCREENSHOT TESTING

Screenshot Testing is a technique used by end-to-end testing which captures a screenshot from an URL and then compares the result with an expected image.

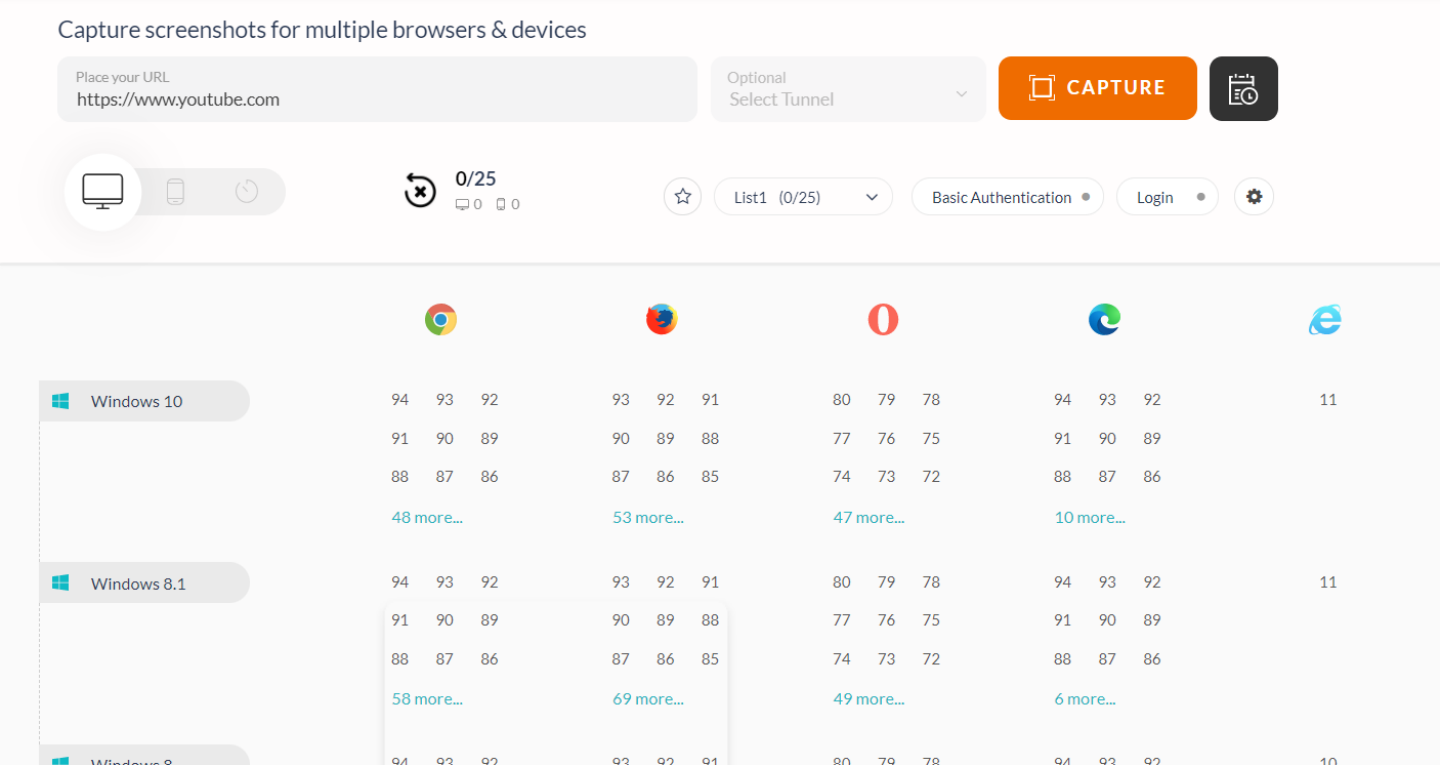


Fig 11: Screenshot Testing

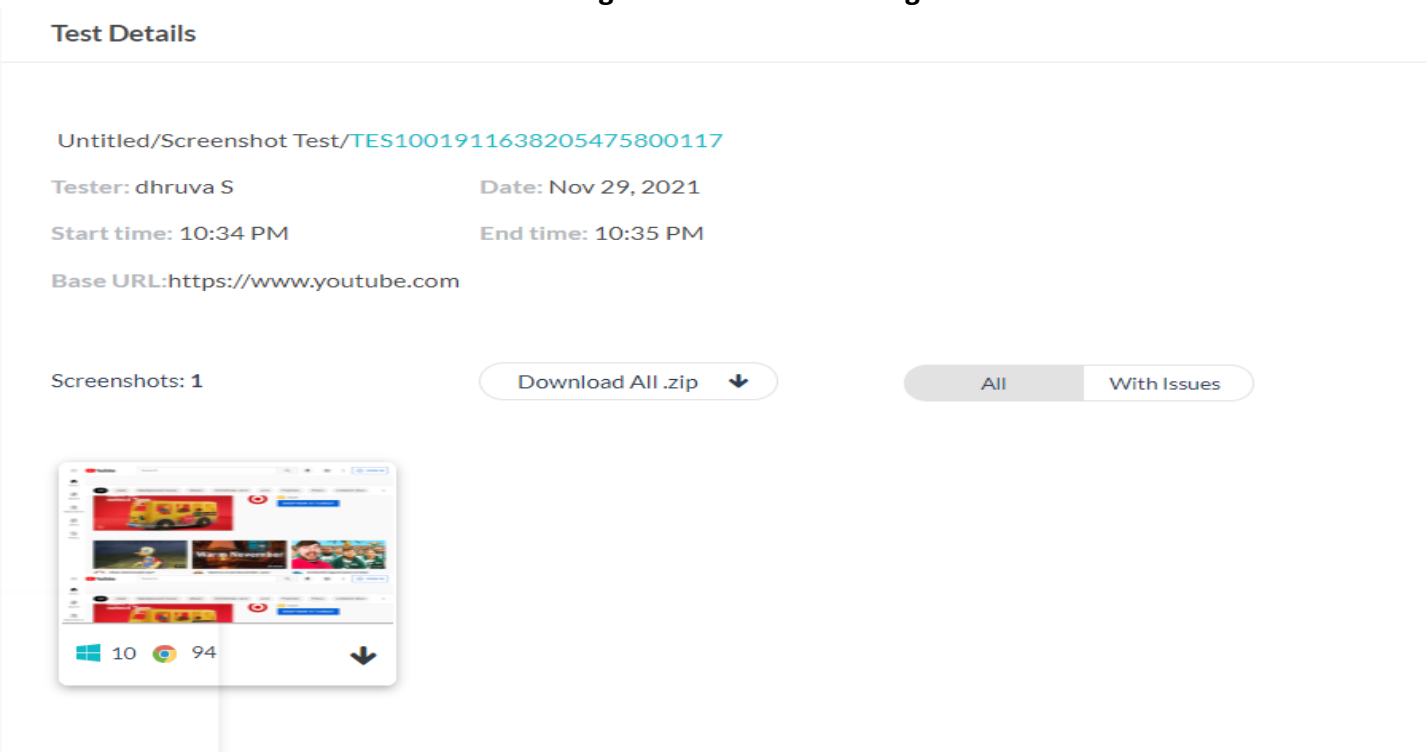


Fig 12: Test Details of screenshot testing

INTEGRATION TESTING

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group.

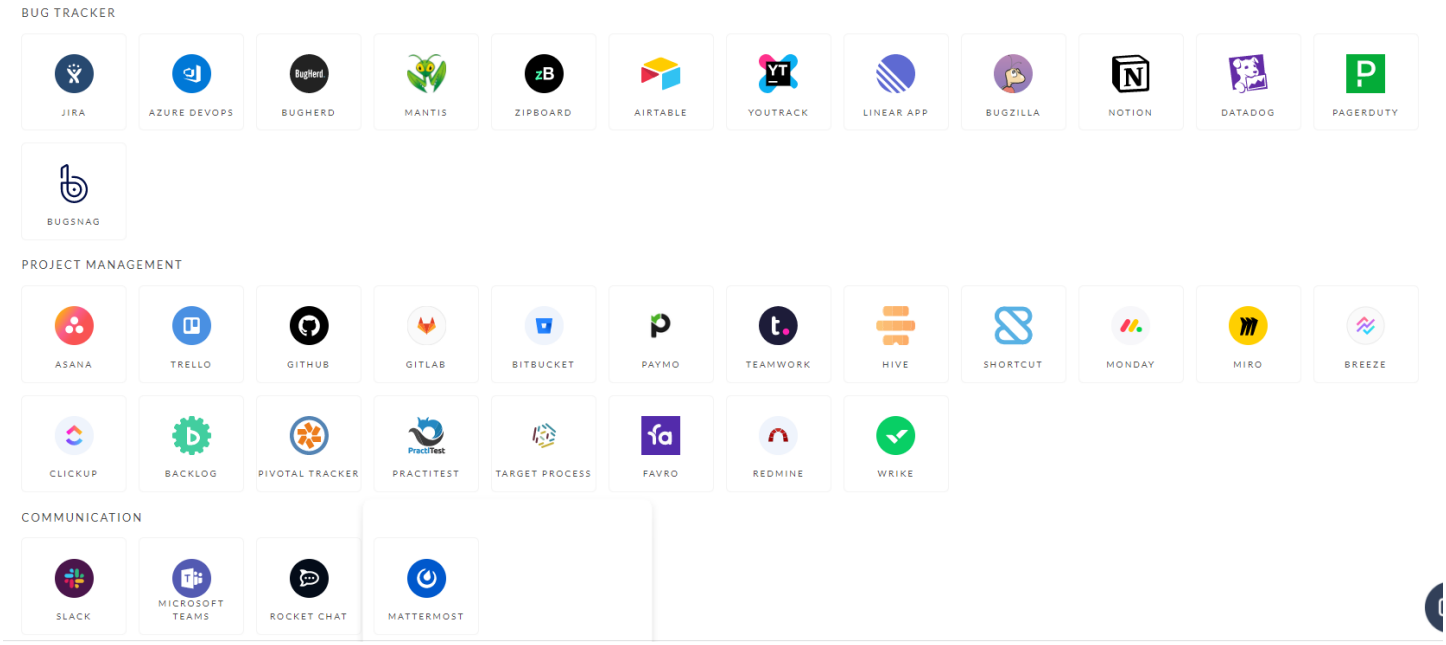


Fig 13

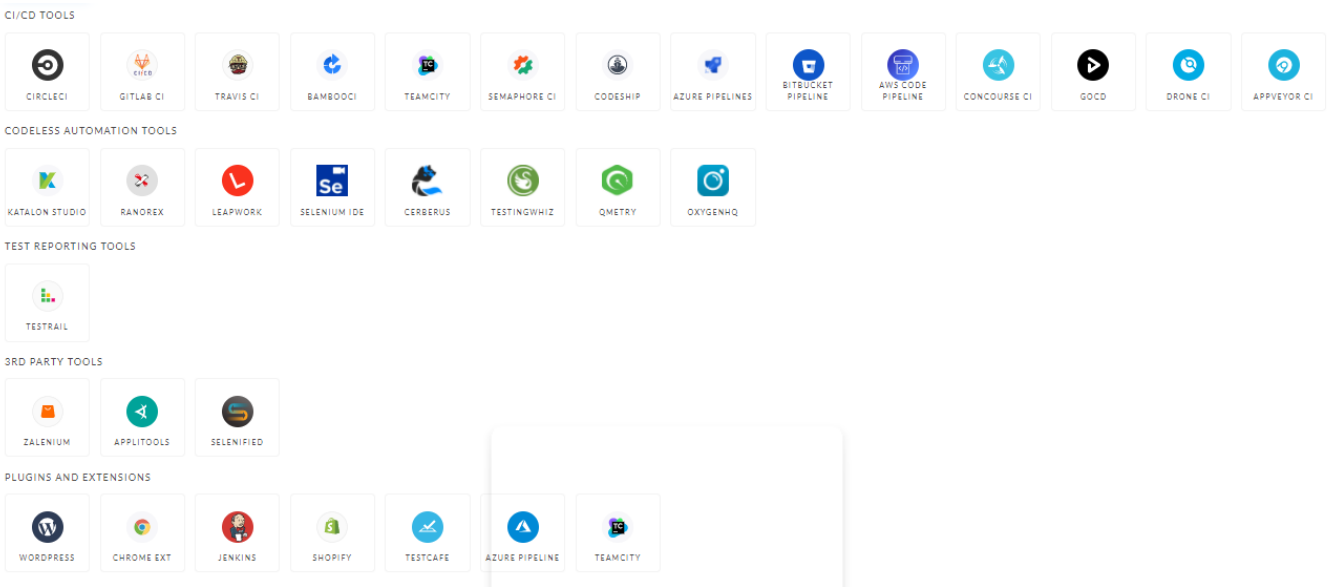


Fig 14

Fig 13 & Fig 14: Integration testing with different bug and project management tools

PROS AND CONS

Pros

- Easy to use
- No additional installation is needed since it is a cloud-based cross browser testing platform
- Secure
- Reliable

Cons

- Not available for free
- No native app testing
- No real device (only emulators)

CONCLUSION

- It's not just about infrastructure, you get a whole range of complimentary features that will make your cross-browser testing experience smoother and help you ship products faster.
- Also, it is trusted by 500,000+ Users.
- With Lambda Test we can choose from a wide range of Windows and Mac operating systems, along with all legacy and latest browsers.