

**Telerik Test Studio**

**Course:** Software Testing

**Instructor:** Dr. Shaarbaf

**Student:** Alireza Dastmalchi Saei

**Student ID:** 993613026

Contents

[**Introduction** 3](#_Toc154261413)

[**Supported Testing Methodologies** 3](#_Toc154261414)

[**Feature Overview** 4](#_Toc154261415)

[**Testing with Sample Code** 5](#_Toc154261416)

[**1.** **Sample App for Test** 5](#_Toc154261417)

[**2.** **Writing Tests** 7](#_Toc154261418)

[**Load Test** 7](#_Toc154261419)

[**Web Test** 8](#_Toc154261420)

[**Test Failures** 8](#_Toc154261421)

[**Advantages and Disadvantages** 9](#_Toc154261422)

[**Explanation Video Link** 10](#_Toc154261423)

[**Resources** 10](#_Toc154261424)

# **Introduction**

Test Studio is an easy-to-use tool for functional UI, load/performance, and API testing for web and WPF desktop applications. Whether you go codeless or use Test Studio’s code-based automation capabilities, you’ll always ensure optimal app quality and deliver outstanding results. Test Studio provides test automation solutions for the entire team, empowering everyone—from junior testers to senior developers, product managers, and QA leads—to achieve maximum productivity in agile software delivery environments.

Telerik Test Studio Icon

# **Supported Testing Methodologies**

Ensuring your application is free from defects requires a testing strategy that integrates several testing methodologies and combines a variety of tests, each of which proves some aspect of your application is ready for deployment. Test Studio supports all kinds of functional UI testing, covering any web technology—the entire .NET stack for the web, including legacy and modern technologies like Blazor, WPF desktop, and JavaScript-based front-end technologies like Angular, React, and jQuery. Automating frequent and repetitive tests with Test Studio supports efficient and cost-effective regression testing and enables you to set up tests for all environments you need to test against.

Test Studio also supports load and performance testing by allowing you to convert functional into load tests or quickly build load tests from scratch, without code, based on Fiddler Everywhere logs. The API testing capabilities in Test Studio let you create verifications against all common RESTful API requests with the possibility of using API tests as steps in automated UI tests. Thanks to Test Studio’s integration capabilities, tests can be scheduled to run as part of any CI/CD setup, in Azure pipelines using Microsoft-hosted agents, or even in Docker containers.

# **Feature Overview**

Test Studio functional testing allows you to create, run, and automate functions with various productivity features that save you time. Some of our most prominent features are:

* **Intuitive recorder with cross-browser support** - with its intuitive click-and-record functionality, recording steps have never been more straightforward. Significant browsers, including Chrome, Edge, Firefox, and Headless Chrome for headless test execution, are proactively supported in Test Studio.
* **Intelligent hybrid element detection** - by creating automation elements that use both web locators and images, flaky and fragile tests are a thing of the past.
* **Element repository** – during test recording, elements are added to the centralized Element Repository, allowing you to manage these easily and reuse them across tests and projects, eliminating redundancies and making things easier.
* **Remote scheduling and concurrent Runs** - use out-of-the-box scheduling functionality to perform everyday tasks such as running simultaneous tests on multiple remote machines.
* **Executive dashboard** - monitor automation results and reports with an easy-to-use web feature. Everyone on your team can access it on the web without the need to have a dedicated Test Studio license
* **Data-driven testing** – bind the data source to test commands without writing and maintaining code.
* **Step failure details** – use failure reports and intelligent suggestions to identify and fix failing tests quickly.
* **Validating PDFs** – add PDF validation steps during test recording or whenever you need to during automation.

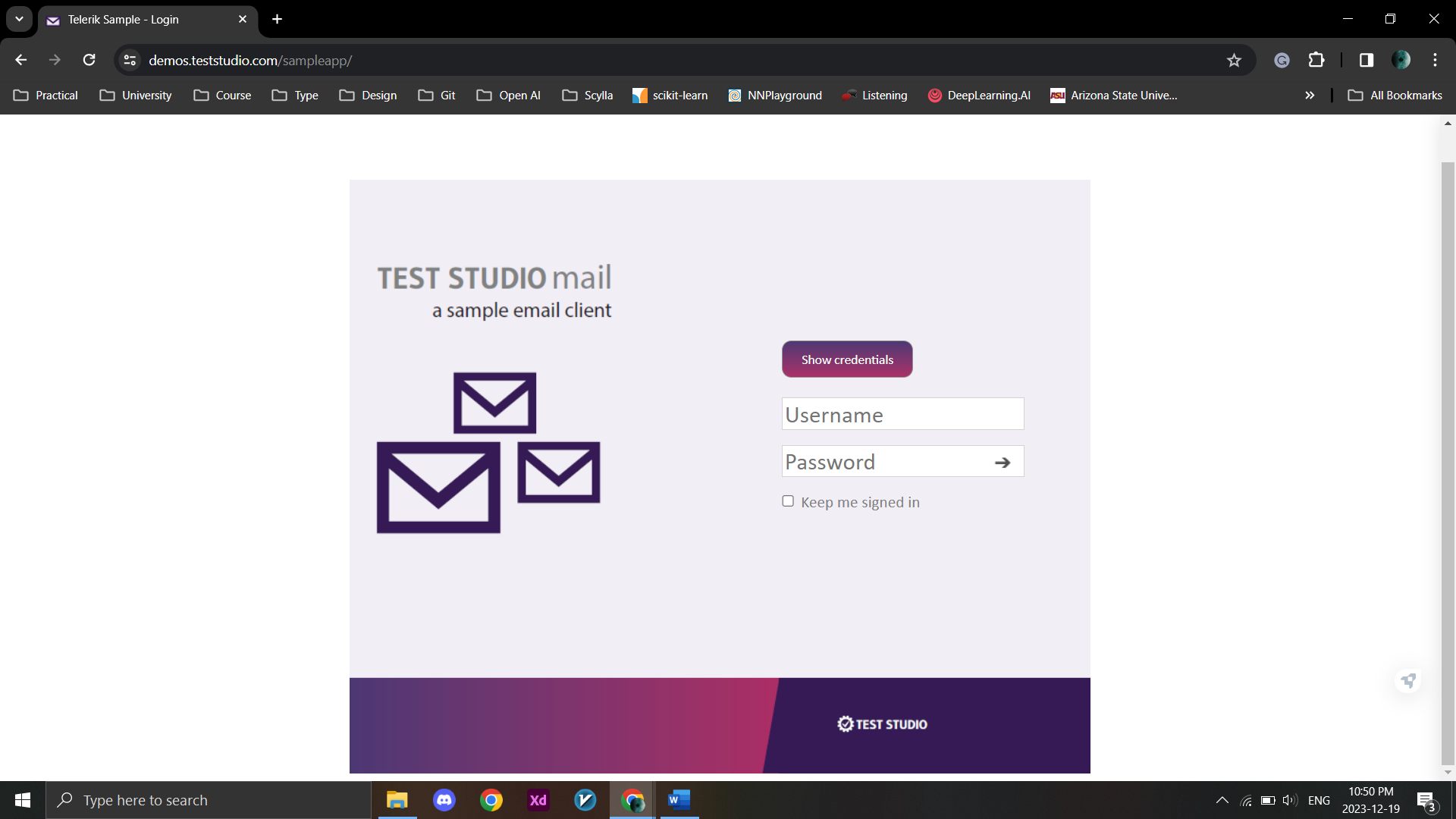
# **Testing with Sample Code**

## **Sample App for Test**

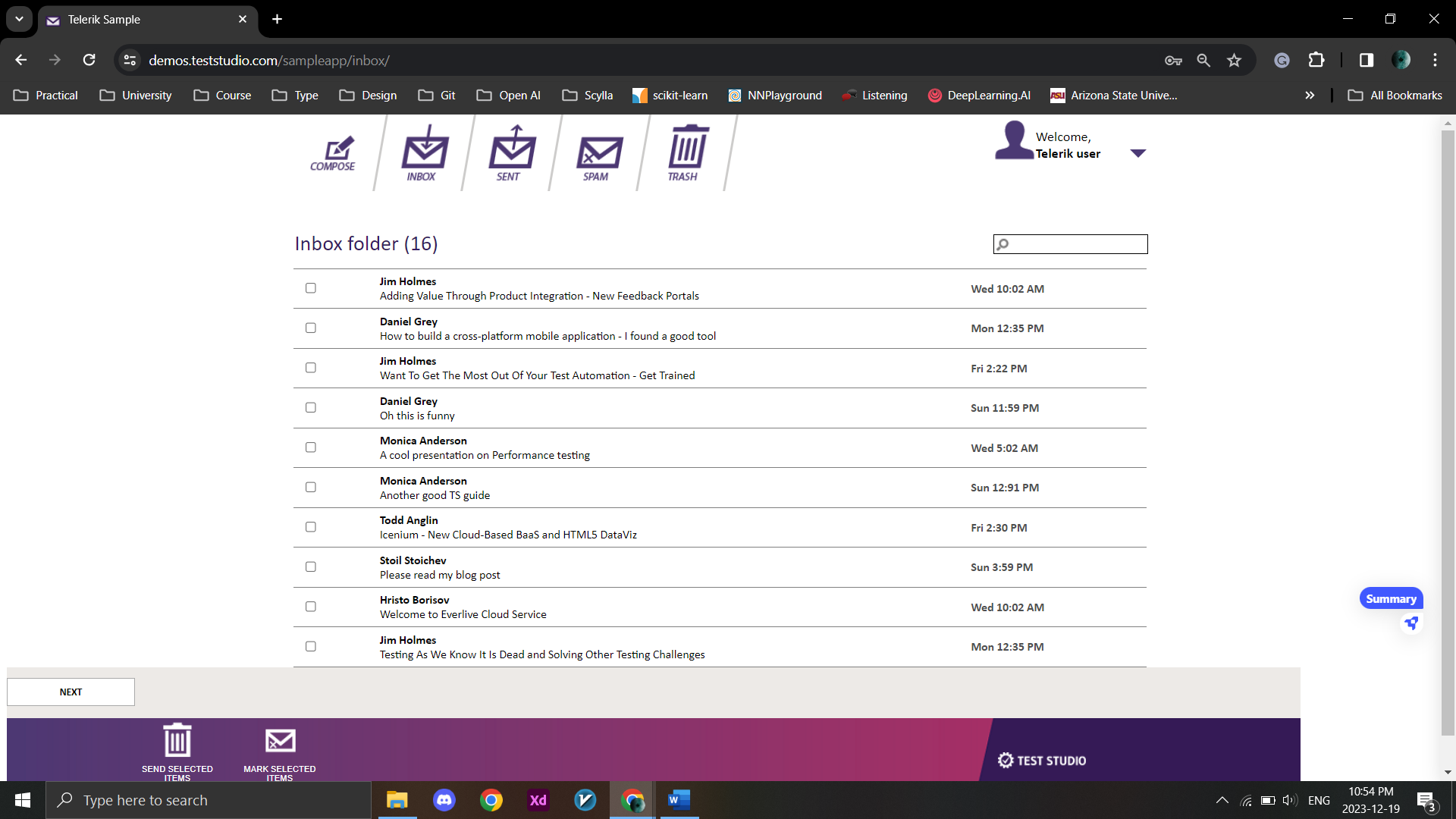
Telerik Test Studio gives the advantage to delve in testing apps quickly without having a code to test. By choosing, template option, you will get default tests for a project from the following URL: [**https://demos.teststudio.com/sampleapp/**](https://demos.teststudio.com/sampleapp/)

You can inspect the project, explore different parts of it, and write your custom tests for it using Test Studio. In this report, we will test this application from a demo app that is a primary email client. Some screenshots from crucial parts of the project have been given below:

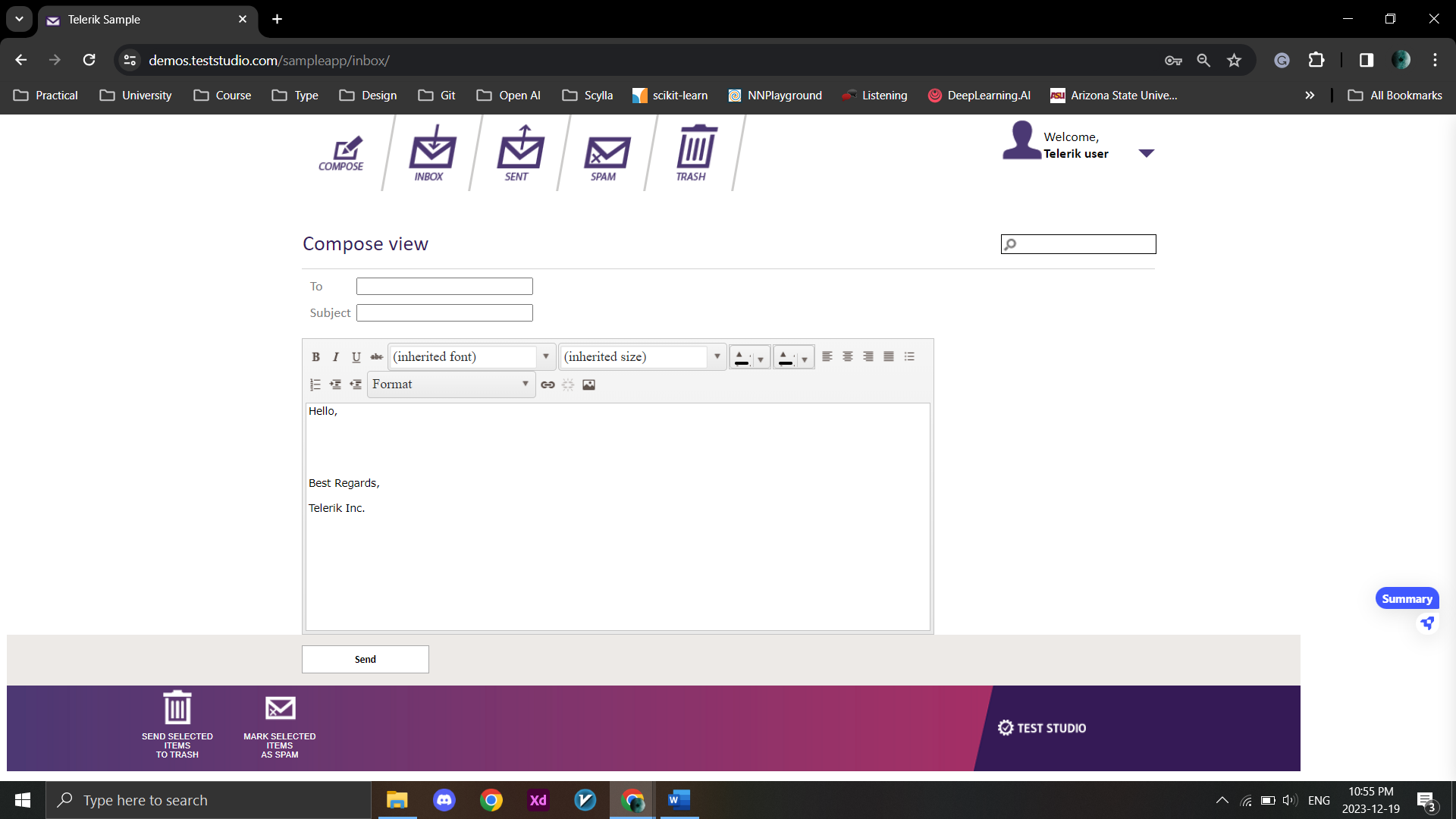
1. Landing Page:



1. After login:



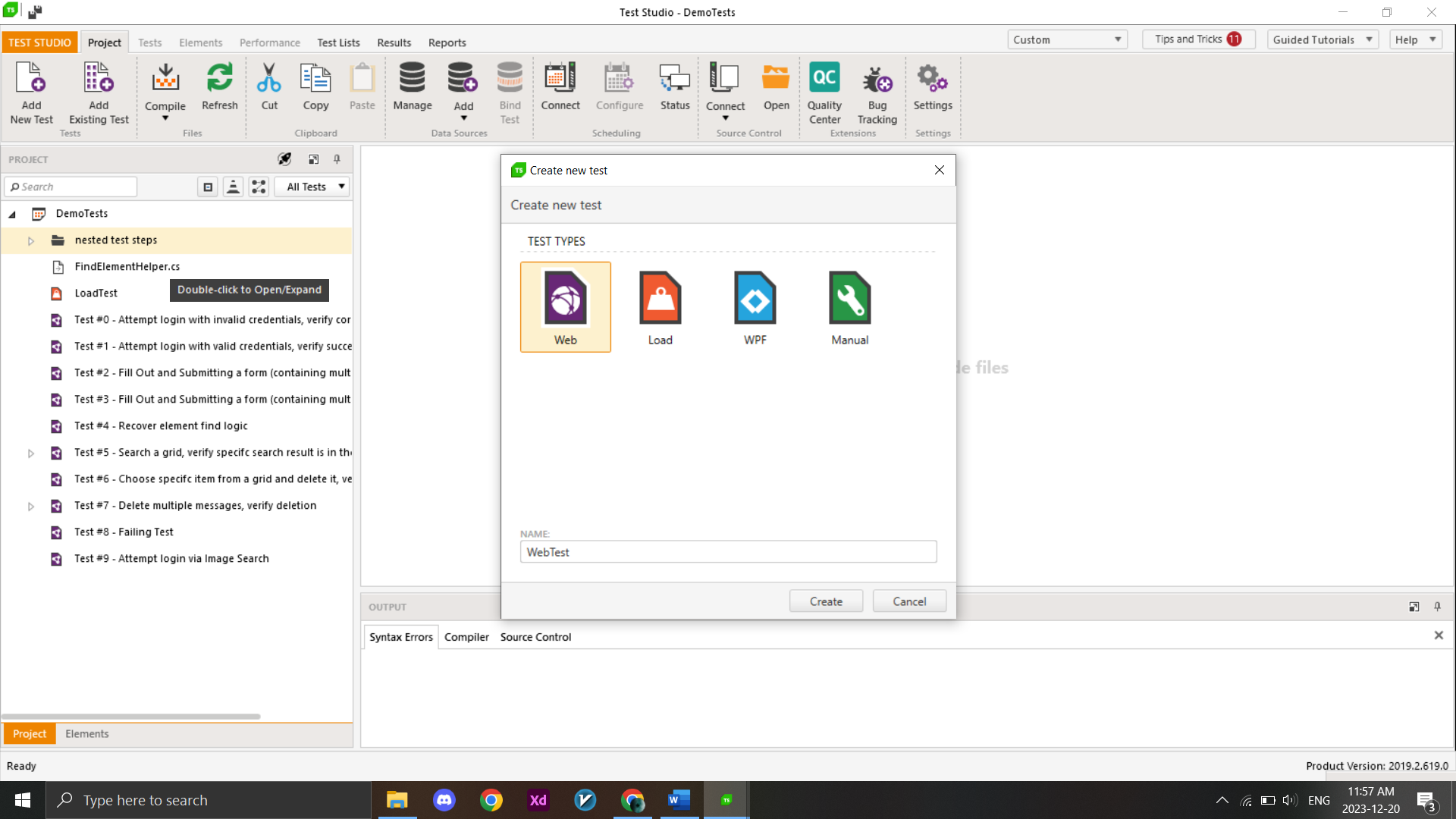
1. Composing an email:



1. Other parts of the project will be further explained and discussed in explanation video.

## **Writing Tests**

After getting familiar with the sample application, we must test it to ensure it is working properly. In Telerik Test Studio, tests can be created in two ways: **Manual** and **Automatic (Web, Load, WPF)**.



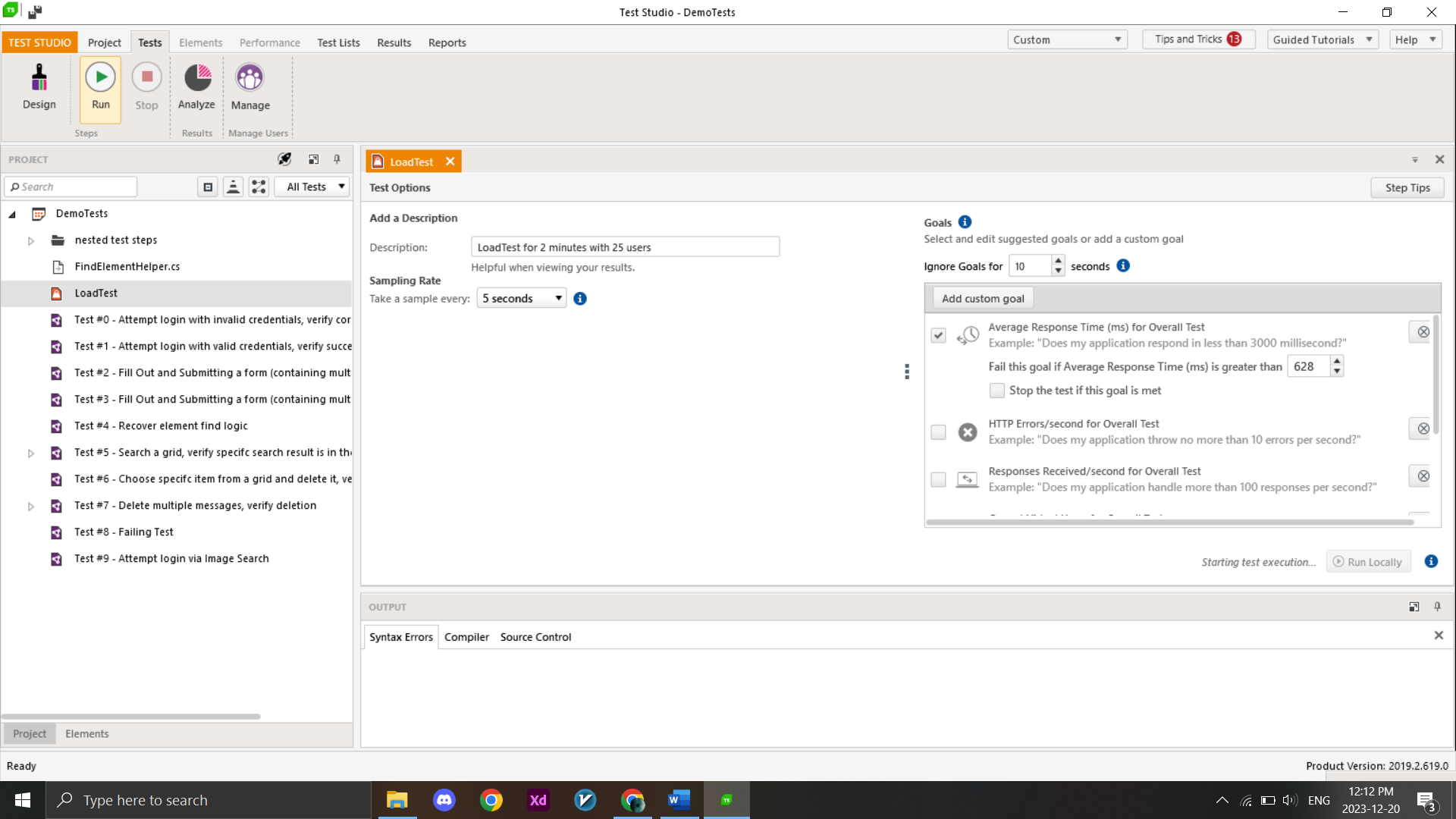
Types of Tests (We will be working with Web Tests)

First, we need to create a test and specify the type of test we want to create. Telerik Studio offers various test types, such as Unit Tests, Functional Tests, Performance Tests, etc. We can select the appropriate test type based on our requirements.

### 

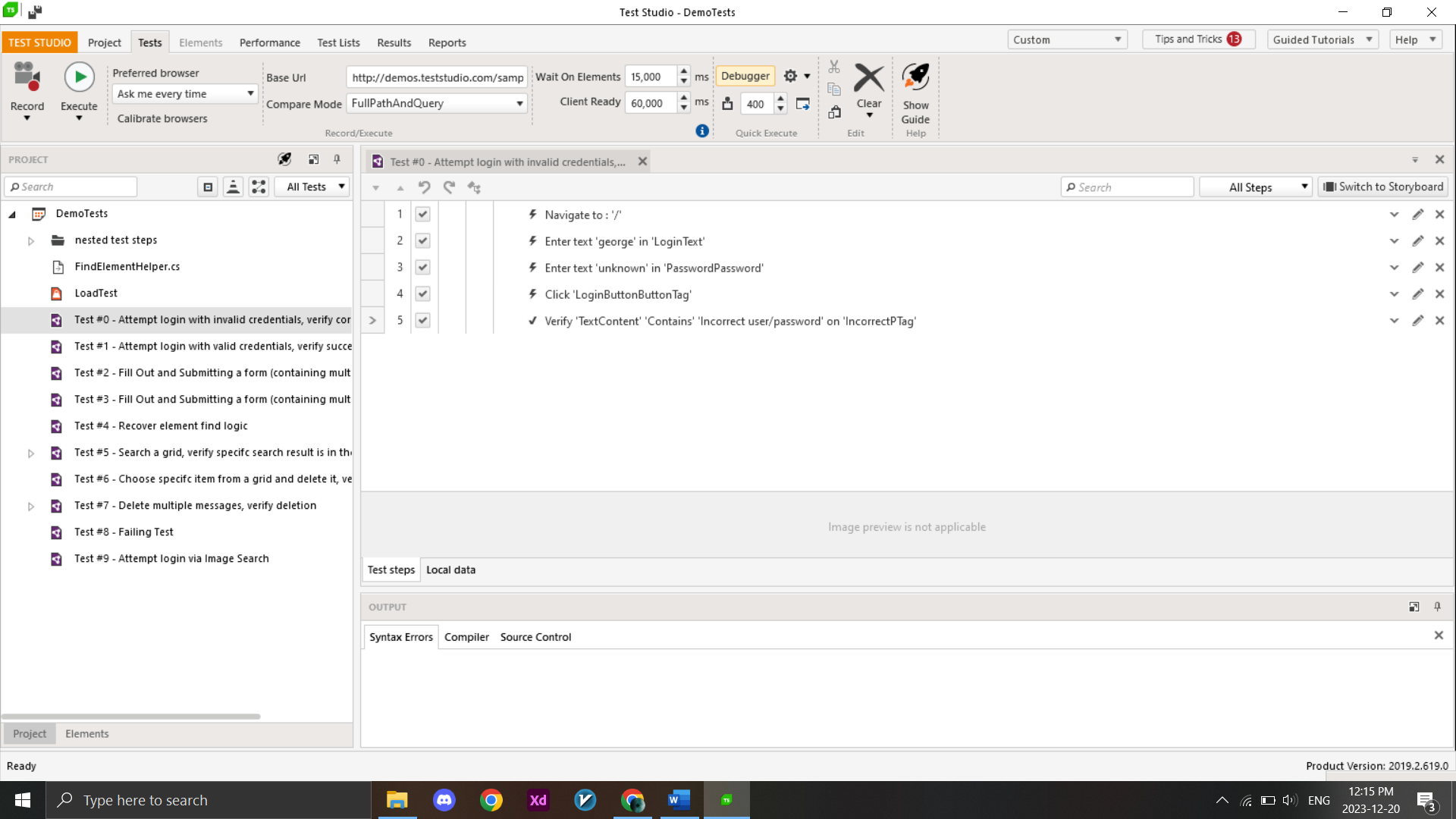
### **Load Test**

We can write load tests for our project to check the load that the application can handle. In load tests, a description, sampling rate and various goals must be defined. Also, some predefined goals can be used, or new goals can be defined in the test environment.



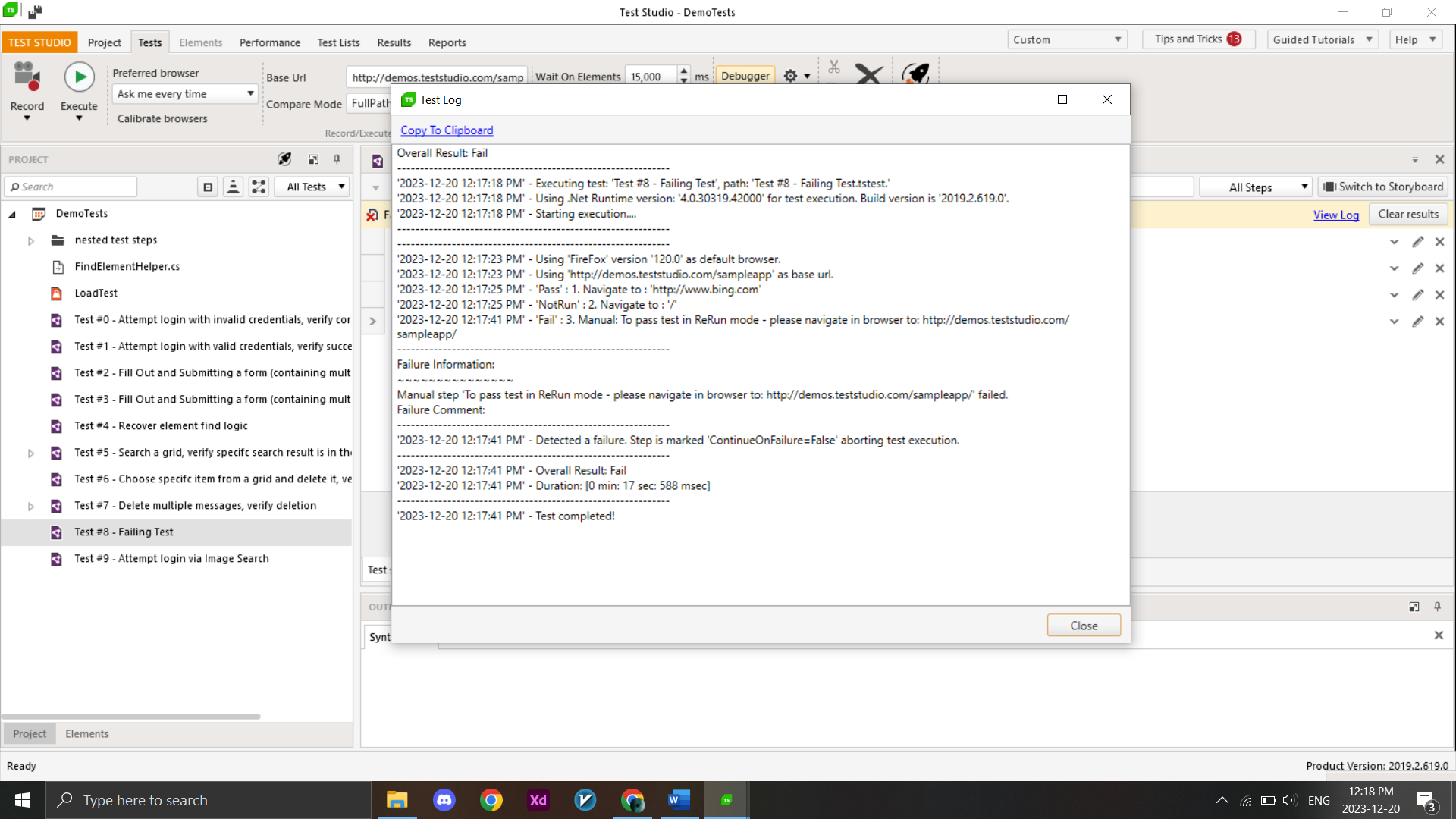
### **Web Test**

For web applications web tests can be defined to check the functionality, UI, or logic of the app. Nested tests can also be written to be used inside other tests for preventing test repetition.

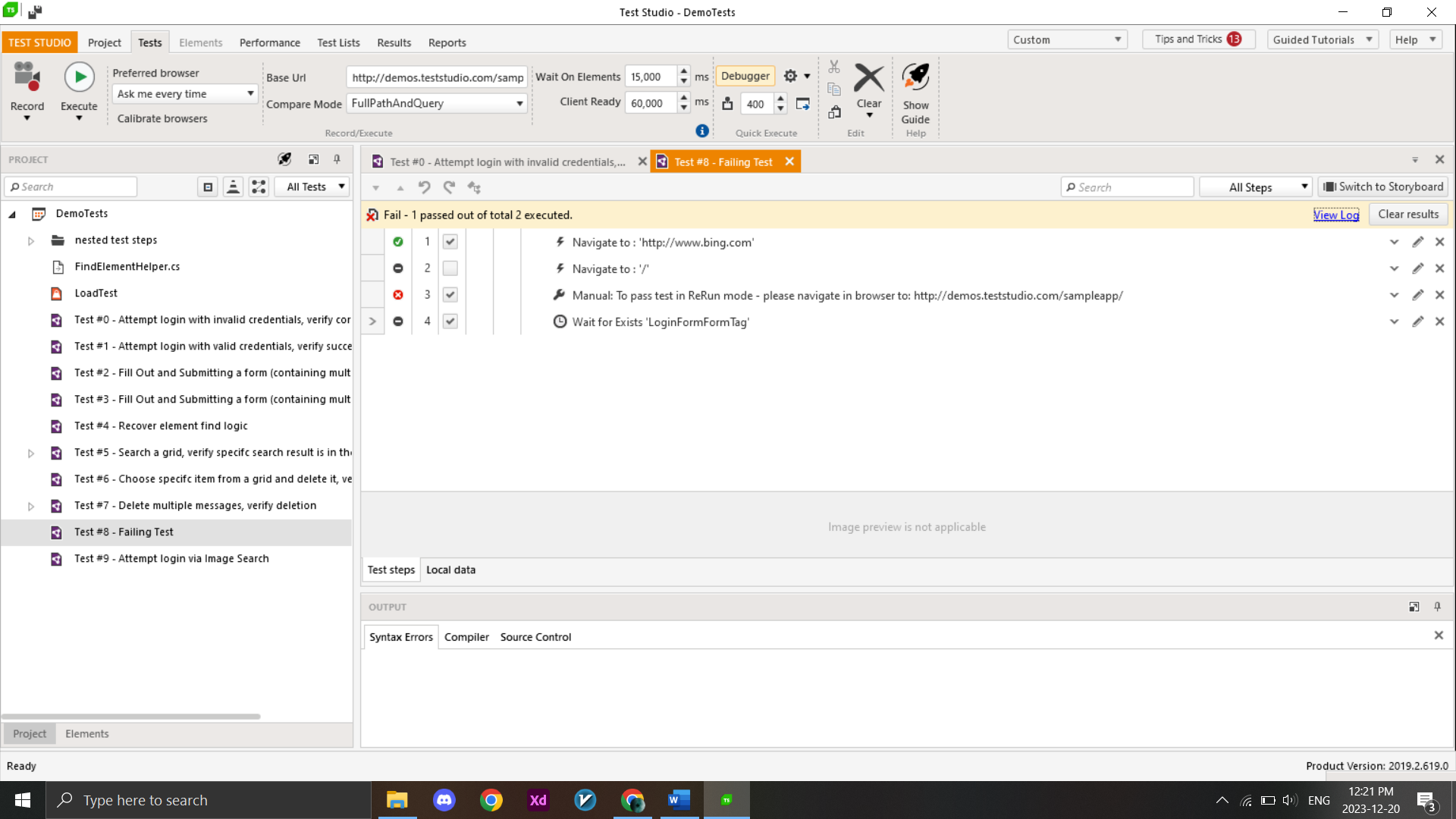


### **Test Failures**

During the test phase, some tests may fail. Telerik Test Studio offers a detailed log for checking the reasons that led the test to failure. An example test failure log can be seen below:



Test Failure Log



Sub Tests

# **Advantages and Disadvantages**

Advantages of Telerik Test Studio are:

* Telerik Test Studio is very user friendly and easy to learn.
* Good language support, Test Studio doesn’t require you to write code in a lot of scenarios. However, if you do need to it supports C# and VB.NET.
* Team Collaboration, Testers can design and maintain tests and pass them to developers through source control to assist with more complex, edge-case scenarios.
* Test Studio comes with rich support for data-driven testing. All recorded test steps have data-related properties that allow you to bind them to a data source. Test Studio supports various data sources: Excel, CSV, XML, and Database. In addition, it has a built-in data grid that allows you to quickly create your data source right inside your test without having to revert to external sources.
* Extensive HTML and Silverlight Control Suite: besides native support for Telerik controls, the Test Studio software testing solution also includes an extensive suite of HTML and Silverlight control translators, which abstracts the control specifics. These translators allow testers to build automated tests for complex control-based applications quickly and efficiently.
* JavaScript and JSON support: Test Studio supports JavaScript function invocation and validation directly from your code. The testing tool also understands JSON objects, can handle strongly typed objects returned from JavaScript, and has access to JQUERY API.
* You can run automated tests on real devices and emulators without writing a single line of code.

Now let's list some of its disadvantages:

* Test Studio is standalone, and if you need to use the VS plugin, you need an extra VS professional or higher license.
* You can't use elements of one project to another, so you have to create only 1 project, and it becomes heavy in due course of time. But this depends upon your application size. You can copy and paste the content from one task to another as a workaround.
* You can convert all your steps to code but can't revert them.
* Issue with the usability of the "If-else" statement: as for using the If-else condition, your element in the "If" situation must be present. If not, the whole test case fails.
* It doesn't support Android and Desktop application testing (on desktop only, WPF is supported).
* For customized reports, if required, you need to write code.
* Need a powerful computer to run all capabilities
* Quite a lot of customization options are available, but they are time-consuming to set up.

# **Explanation Video Link**

<https://drive.google.com/drive/folders/1KLHFdvIY3ka2m1kVdXFB1HiRp3izZeG4?usp=drive_link>

# **Resources**

* **Sample App Used for Test:** <https://demos.teststudio.com/sampleapp/>
* **Telerik Documentations:** <https://docs.telerik.com/teststudio/getting-started/first-project>
* **YouTube Tutorial:** <https://youtu.be/Lk5ZYUjeNMM?si=VG2oa4HWFvUSgnEn>