**Google Map Automation Framework**

**Documentation**

# Introduction

This document describes the approach used in the automation framework to design, organize and perform the automation testing of Google Map Seach Address.

# Test Browsers

Although this automation framework is compatible to run automated tests in all the below mentioned browsers but still it is highly **recommended** to run tests in **Chrome browser**

|  |  |
| --- | --- |
| Browser | Version |
| Chrome | Latest |
| Firefox | Latest |
| Microsoft Edge | Latest |

# Automation Tools

|  |  |
| --- | --- |
| Tool | Comments |
| Visual Studio | IDE for automated case development |
| Specflow | Tool for BDD test case creation |
| NUnit | Unit test framework |
| Selenium | End-to-End Web UI tests |
| Log4Net | To log the output statement |
| Git | Source control |
| Extent Report | Detailed UI report for tests |
| GitHub Actions | Continuous Integration for test automation |

# Automation Framework Architecture Details

GoogleMapAutomation framework has been developed in a **BDD** paradigm using SpecFlow.

Repository link: [GoogleMapAutomation](https://github.com/SanjeetRautela/GoogleMapAutomation)

.Net Core, NUnit, Selenium is the base of the framework.

Different folders are created to keep the common files together.

Page Object Model also known as **POM**, design pattern used to create the object repository for storing all the web element and methods.

Tests inside the framework are compatible to run in cross browser platform.

Configuration can be updated in the appsettings Json file to run the tests according to the end user need.

Framework also consists of logging the details in a file which will be generated after the test is run and the location will be the bin folder. **Log4Net** is used to achieve this.

**Extent report** has been integrated in the framework so that user can see the test run execution report at the end. A folder name “TestReport” will be generated inside bin directory, which will contain the execution report.

**Screenshot** implementation is also present in the framework for every test failure. You can see the failed test with screenshot in HTML report generated by framework.

# Automation Tests Details:

* As of now there are five tests are present in this framework which you can see here:
* Tests are only related to search addresses.
* Tests are mix of positive and negative scenarios.

# Continuous Integration:

GitHub Action is used for CI pipeline. CI pipeline will run for every push in master branch to make sure that your code works smooth and does not break any existing behavior.

CI Pipeline also runs all the tests in headless mode and generate a HTML report at the end so that you can see result of pipeline execution.

CI pipeline: [GoogleMapAutomation\_CI\_Pipeline](https://github.com/SanjeetRautela/GoogleMapAutomation/actions/workflows/dotnet.yml)

# Extent Report Screenshot:

