Software Unit Testing, part 4
Playwright

1 Introduction
In part 1 we discussed how Software Unit Testing is different from End-to-End testing. We briefly discussed Test Coverage and why you need it. There was a discussion of how you did automated software testing in the early days using a generic test framework, and an explanation of how automated software testing would be performed using JavaScript with the modern Web testing framework Cypress. In part 2 we discussed how to do Software Unit Testing in the C# language using .NET and the modern xUnit test framework. In part 3 we discussed how to do Software Unit Testing in the C++ language using a generic test framework as well as the modern GTEST framework. In this article we will discuss how to do Software Unit Testing using Playwright.

2 What is Playwright?
Playwright is a product of Microsoft and is actively maintained. [202] Playwright can test .NET, Java, JavaScript, Python and TypeScript. It can run on Linux, macOS and Windows. Playwright supports browsers based on Chromium, Firefox and WebKit. It also supports mobile devices (Chrome for Android and Mobile Safari). [196] One advantage of Playwright is that it supports TypeScript. [202]

Playwright runs in a real browser, not an emulated browser. Everything that runs in your browser will work. It was created for end-to-end testing, but also works well for unit testing. [202] [203] [206] Unit testing has been supported since 2022. [206] Code coverage can be provided using ***nyc***, which is part of ***Istanbul*** (a code coverage tool for JavaScript). [202]

3 Setting up Playwright
You need to install Node.js® on your local machine before you install Playwright. Node.js is a free, open-source, cross-platform JavaScript runtime environment that lets you run JavaScript on the server. [96] [97] It is used by millions of developers. [98] **Make sure that you have successfully installed Node.js before you begin.**

Create a directory for the project.

mkdir playwrightTest1

Change directory to the project directory.

cd playwrightTest1

Install Playwright :

npm install -D @playwright/test@latest

Install nyc in order to obtain line coverage:

npm install nyc --save-dev

Install monocart reporter :

npm i -D monocart-reporter

Create a directory called **src** for the code we wish to test.

mkdir src

If there is no directory called tests, create a directory called **tests** where we will create our tests.

mkdir tests

Change to the tests directory.

cd tests

Create a directory called **fizzbuzz** .

mkdir fizzbuzz

Change directory back to the project directory.

cd ..

4 Test Scenario

For our testing example let's use the children's game "FizzBuzz", which is sometimes used as a coding test. FizzBuzz works like this:

Output the numbers from 1 to 100.

1. If the number is not divisible by either 3 or 5, just display the number.
2. If the number is divisible by 3, display the word "FIZZ" instead of the number.
3. If the number is divisible by 5, display the word "BUZZ" instead of the number.
4. If the number is divisible by both 3 and 5, display the word "FIZZBUZZ" instead of the number.

So, our code must meet the four requirements listed above. In order to do so, the code must make several decisions based on the input value. We would therefore need a minimum of 4 test cases to do testing based on the requirements. This function has the following code paths:

path 1 - the input IS NOT divisible by EITHER 3 or 5, return the input value
path 2 - the input IS divisible by BOTH 3 and 5, return "FIZZBUZZ"
path 3 - the input IS divisible by 3 but IS NOT divisible by 5, return "FIZZ"
path 4 - the input IS NOT divisible by 3 but IS divisible by 5, return "BUZZ"

If we test to meet the requirements, we would expect to see 100% path coverage of the main section of the code with the four test cases needed to meet the requirements. We would add a fifth test case to verify that errors are handled correctly.

5 Unit Testing - Playwright

Let's write a function called ***fizzBuzz*** that we want to use in an application. We create it in the ***src*** directory. We want to test this function using Playwright to make sure it works correctly. Our function would look like this:

```
//  src/fizzbuzz.js

export function fizzbuzz (num) {
  if (num % 3 === 0 && num % 5 === 0) {
    return 'FIZZBUZZ'
  }
  else
  if (num % 3 === 0) {
    return 'FIZZ'
  }
  else
  if (num % 5 === 0) {
    return 'BUZZ'
  }
  else {
    return num
  }

  //   end     function
}
```

Figure 1: JavaScript code for the fizzBuzz function.

Our function has a single numeric input value, and we return a single result in a string. Since we have not placed any restrictions on the input, we could have thousands of possible input values. FizzBuzz only requires that we use the values from 1 to 100, so we would need at least 100 test cases to test every input value that we would expect to encounter. What happens if we input a zero? What happens if we input a value higher than 100? Would they cause an undesired result? Do we need to add code to handle inputs that are out of range? If we were to define the input as a byte, we would still have 256 possible input values (0 to 255). Assuming that we only tested the values 1 to 100, it might be a good idea to test 0 and the values greater than 100, just to be thorough. This would require at least 256 test cases.

6 Playwright Unit Test

We create our Unit Test for fizzbuzz in the ***tests/fizzbuzz*** directory. It is called ***fizzbuzz.spec.js*** . Our Unit test of fizzBuzz using Playwright would look like this:

```
//  tests/fizzbuzz/fizzbuzz.spec.js

import { test, expect } from '@playwright/test';
import { fizzbuzz } from '../../src/fizzbuzz.js';

test.describe('fizzbuzz.js', () => {

        test('fizzbuzz()', async ({ page, browserName }) => {
                console.log('Running on browser: ' + browserName );

                page.on( 'console', msg => {
                   console.log('[Browser Console]: ${msg.text()} ' );
                });


                if ( browserName == "chromium" ) {
                    console.log('*** chromium browser, turn ON coverage ***');

                    if (!page.coverage) throw new
                           Error('Could not collect coverage with browser ' + browserName );

                    console.log("*** collecting coverage ... ***");

                    await page.coverage.startJSCoverage({ resetOnNavigation: false });

                    //   end     if chromium
                }


                    console.log('*** test case 1 ***');

                // test case 1 - fizzBuzz with input of 1 returns "1"
                expect(fizzbuzz(1)).toEqual(1);


                    console.log('*** test case 2 ***');

                // test case 2 - fizzBuzz with input of 2 returns "2"
                expect(fizzbuzz(2)).toEqual(2);
```

Continued on the next page.

Figure 2a: Unit test of JavaScript function fizzBuzz written using Playwright.

```
                console.log('*** test case 3 ***');

            // test case 3 - fizzBuzz with input of 3 returns "FIZZ"
            expect(fizzbuzz(3)).toEqual('FIZZ');


                console.log('*** test case 4 ***');

            // test case 4 - fizzBuzz with input of 4 returns "4"
            expect(fizzbuzz(4)).toEqual(4);


                console.log('*** test case 5 ***');

            // test case 5 - fizzBuzz with input of 5 returns "BUZZ"
            expect(fizzbuzz(5)).toEqual('BUZZ');


                console.log('*** test case 6 ***');

            // test case 6 - fizzBuzz with input of 15 returns "FIZZBUZZ"
            expect(fizzbuzz(15)).toEqual('FIZZBUZZ');


                console.log('*** test case 7 ***');

            // test case 7 - fizzBuzz with input of 16 returns "16"
            expect(fizzbuzz(16)).toEqual(16);



        //        console.log('*** test case 8 ***');

            // test case 8 - fizzBuzz with input of 16 returns "17"
             //  expect(fizzbuzz(16)).toEqual(17);

            //  dont evaluate since this is a repeat of input value 16


            // *********************************************************

            if ( browserName == "chromium" ) {
                console.log('*** chromium browser, turn OFF coverage ***');

                const [jsCoverage] = await page.coverage.stopJSCoverage();

                //   end     if chromium
            }



            console.log('*** END OF TEST ***');


//              await browser.close();

        });

});
```

Figure 2b: Unit test of JavaScript function fizzBuzz written using Playwright.

We create 7 tests. We have an additional test case (test case 8) to verify that everything works correctly when we get an unexpected result. For now, we will leave that test case commented out.

Since we want to see the line coverage for the fizzbuzz function, we need to use ***nyc*** to run Playwright. We tell nyc to generate the line coverage information in ***lcov*** format. When we run the tests using nyc, we get the following results:

```
npx nyc --reporter=lcov --reporter=text-summary npx playwright test fizzbuzz/fizzbuzz
[MR] clean output dir ...

Running 3 tests using 1 worker

  ok 1 [chromium] › tests\fizzbuzz\fizzbuzz.spec.js:8:6 › fizzbuzz.js › fizzbuzz() (368ms)
Running on browser: chromium
*** chromium browser, turn ON coverage ***
*** collecting coverage ... ***
*** test case 1 ***
*** test case 2 ***
*** test case 3 ***
*** test case 4 ***
*** test case 5 ***
*** test case 6 ***
*** test case 7 ***
*** chromium browser, turn OFF coverage ***
*** END OF TEST ***
  ok 2 [firefox] › tests\fizzbuzz\fizzbuzz.spec.js:8:6 › fizzbuzz.js › fizzbuzz() (2.4s)
Running on browser: firefox
*** test case 1 ***
*** test case 2 ***
*** test case 3 ***
*** test case 4 ***
*** test case 5 ***
*** test case 6 ***
*** test case 7 ***
*** END OF TEST ***
  ok 3 [webkit] › tests\fizzbuzz\fizzbuzz.spec.js:8:6 › fizzbuzz.js › fizzbuzz() (553ms)
Running on browser: webkit
*** test case 1 ***
*** test case 2 ***
*** test case 3 ***
*** test case 4 ***
*** test case 5 ***
*** test case 6 ***
*** test case 7 ***
*** END OF TEST ***

  3 passed (17.8s)
[MR] generating report data ...
[MR] generating test report ...
[MR] MCR Playwright Report
```

Continued on the next page.

Figure 3a: Unit Test results for the fizzBuzz function.

```
┌──────────────────┬──────────────────────────────────────────┐
│ Tests            │ 3                                        │
│  ├ Failed        │ 0 (0.0%)                                 │
│  ├ Flaky         │ 0 (0.0%)                                 │
│  ├ Skipped       │ 0 (0.0%)                                 │
│  └ Passed        │ 3 (100.0%)                               │
│ Steps            │ 54                                       │
│ Suites           │ 3                                        │
│  ├ Projects      │ 3                                        │
│  ├ Files         │ 1                                        │
│  ├ Describes     │ 3                                        │
│  └ Shards        │ 0                                        │
│ Errors           │ 0                                        │
│ Retries          │ 0                                        │
│ Logs             │ 30                                       │
│ Attachments      │ 0                                        │
│ Artifacts        │ 0                                        │
│ Playwright       │ v1.56.1                                  │
│ Date             │ 11/22/2025, 12:05:46 PM                  │
│ Duration         │ 17.9s                                    │
└──────────────────┴──────────────────────────────────────────┘
[MR] json: test-results/mcr-report/index.json
[MR] view report: npx monocart show-report test-results/mcr-report/index.html

To open last HTML report run:

  npx playwright show-report


=============================== Coverage summary ===============================
Statements   : 100% ( 9/9 )
Branches     : 83.33% ( 10/12 )
Functions    : 100% ( 1/1 )
Lines        : 100% ( 9/9 )
================================================================================
```

Figure 3b: Unit Test results for the fizzBuzz function.

After you run the tests, there will be a directory called ***coverage*** in the project.

Change to the coverage directory.

cd coverage

You will see a directory called ***lcov-report*** .

Change to the lcov-report directory.

cd lcov-report

You will see a directory called ***playwrightTest1_22nov2025*** . The date will reflect the date when you ran the test.

Change to the playwrightTest1_22nov2025 directory.

cd playwrightTest1_22nov2025

You will see a directory called ***src*** .

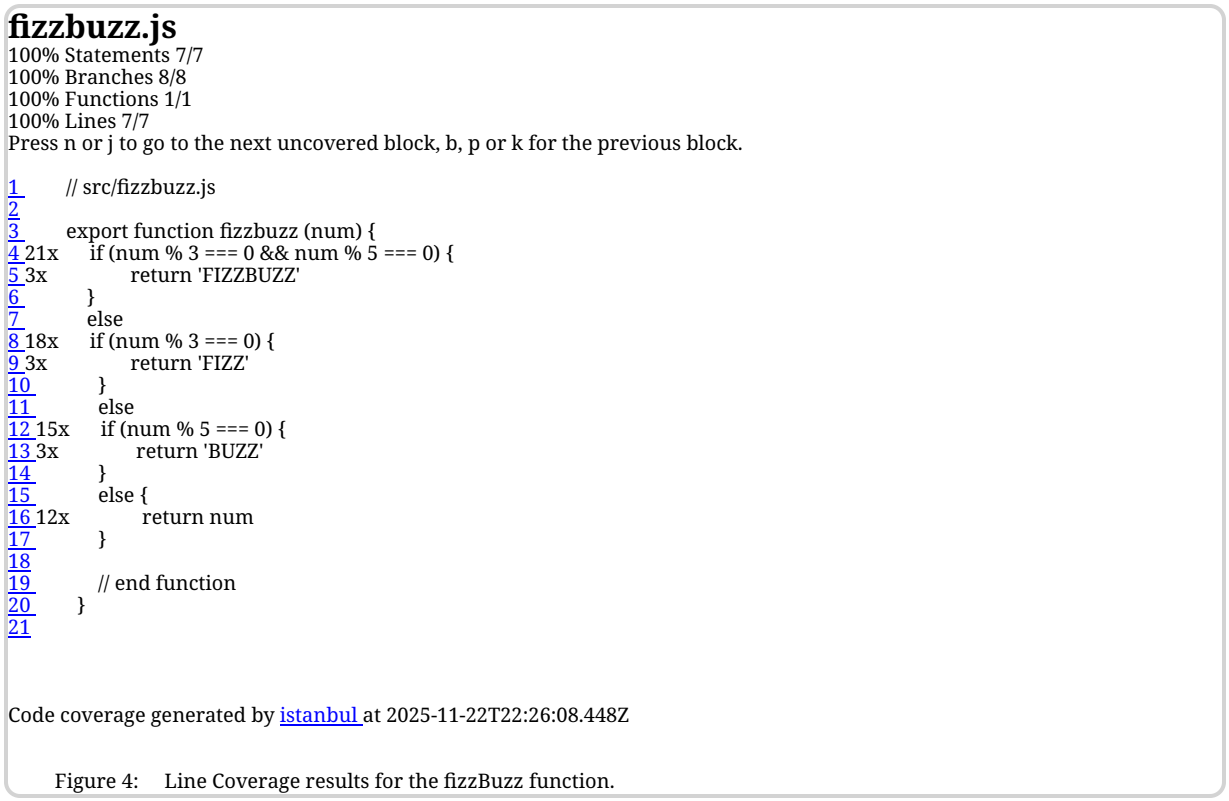
Change to the src directory.

cd src

You will see a file called ***fizzbuzz.js.html*** .

Open the file fizzbuzz.js.html with your browser.

You will see a ***Line Coverage*** report similar to the following:



fizzbuzz.js

100% Statements 7/7
100% Branches 8/8
100% Functions 1/1
100% Lines 7/7
Press n or j to go to the next uncovered block, b, p or k for the previous block.

```
1       // src/fizzbuzz.js
2
3       export function fizzbuzz (num) {
4 21x     if (num % 3 === 0 && num % 5 === 0) {
5 3x            return 'FIZZBUZZ'
6         }
7         else
8 18x     if (num % 3 === 0) {
9 3x            return 'FIZZ'
10        }
11        else
12 15x    if (num % 5 === 0) {
13 3x           return 'BUZZ'
14        }
15        else {
16 12x          return num
17        }
18
19        // end function
20      }
21
```

Code coverage generated by istanbul at 2025-11-22T22:26:08.448Z

Figure 4: Line Coverage results for the fizzBuzz function.

12 Generate the report using reportgenerator

If we have the **.NET SDK** installed on our system, we can also use the report generator tool for .NET to generate the report.

13 .NET CLI

The .NET framework includes a command line interface (CLI). The .NET command-line interface (CLI) is a cross-platform toolchain for developing, building, running, and publishing .NET applications. The .NET CLI is included with the .NET SDK. [123] After installing the SDK, CLI commands are run by opening a Command Prompt window and entering the commands at the command prompt. [122] [124] [134] Once the .NET SDK is installed, further help on using the CLI may be obtained by typing

dotnet --help or dotnet -h

To determine the version of the .NET SDK which is installed, type

dotnet --info

If we are using a **.NET SDK earlier than version 8** , we may need to add some software in order to generate coverage reports. [145] [146] [151]

dotnet new tool-manifest

dotnet add package coverlet.collector

dotnet add package coverlet.msbuild

dotnet tool install coverlet.console

dotnet tool install -g dotnet-reportgenerator-globaltool

If we are using **.NET SDK 8** , we may need to install the following software: [146] [147]

dotnet tool install --global dotnet-coverage

The report generator converts coverage reports generated by coverlet, OpenCover, dotCover, Visual Studio, NCover, Cobertura, JaCoCo, Clover, gcov or lcov into human readable reports in various formats. The reports show which lines of your source code have been covered. ReportGenerator is available under the Apache [136] License. [148]

Some of the many report types available include: [157]

Html	**A summary (index.html) and detailed reports for each class.**
Html_Light	**Html with a light theme.**
Html_Dark	**Html with a dark theme.**
HtmlSummary	**A single HTML file (summary.html) without links.**
TextSummary	**A single TXT file containing coverage information per class.**
SvgChart	**A single SVG file containing a chart with historic coverage information.**
Badges	**SVG files that show line and/or branch coverage information.**

14 Using the Reportgenerator Installed Under the Dotnet CLI

Use the dotnet reportgenerator tool to create the coverage report for fizzbuzz.js.

reportgenerator -reports:coverage/lcov.info -targetdir:coverage-report1 -reporttypes:"Html_Dark"

```
2025-11-30T13:47:53: Arguments
2025-11-30T13:47:53:  -reports:coverage/lcov.info
2025-11-30T13:47:53:  -targetdir:coverage-report1
2025-11-30T13:47:53:  -reporttypes:Html_Dark
2025-11-30T13:47:53: Writing report file 'coverage-report1\index.html'
2025-11-30T13:47:53: Report generation took 0.3 seconds
```

We wrote a coverage report for fizzbuzz.js to **coverage-report1\index.html** . We can open the file in a browser.

15 Coverage Report

The coverage report created by reportgenerator show that we have the following coverage:

File(s): fizzbuzz.js

Covered lines:	**7**
Uncovered lines:	**0**
Coverable lines:	**7**
Total lines:	**20**
Line coverage:	**100%**

The line coverage report would be similar to the following:

File(s)

src\fizzbuzz.js

#	Line	Line coverage
	1	// src/fizzbuzz.js
	2	
	3	export function fizzbuzz (num) {
21	4	if (num % 3 === 0 && num % 5 === 0) {
3	5	return 'FIZZBUZZ'
	6	}
	7	else
18	8	if (num % 3 === 0) {
3	9	return 'FIZZ'
	10	}
	11	else
15	12	if (num % 5 === 0) {
3	13	return 'BUZZ'
	14	}
	15	else {
12	16	return num
	17	}
	18	
	19	// end function
	20	}

Generated by: ReportGenerator 5.3.8.0
11/30/2025 - 1:47:53 PM
GitHub | reportgenerator.io

Figure 5: Line Coverage results for the fizzBuzz function.

This page intentionally left blank.

References

[1] G. Warren *et al.*, "Overview of .NET Framework." *Microsoft Docs*, 21-Oct-2020. [Online]. Available: https://docs.microsoft.com/en-us/dotnet/framework/get-started/overview. [Accessed: 21-Jan-2021].

[2] G. Warren *et al.*, "What is managed code?" *Microsoft Docs*, 20-Jun-2016. [Online].Available: https://docs.microsoft.com/en-us/dotnet/standard/managed-code. [Accessed: 21-Jan-2021].

[3] G. Warren *et al.*, "Common Language Runtime (CLR) overview." *Microsoft Docs*, 22-Oct-2020. [Online]. Available: https://docs.microsoft.com/en-us/dotnet/standard/clr. [Accessed: 28-Jan-2021].

[4] G. Warren, "gewarren - Overview," *GitHub*. [Online]. Available: https://github.com/gewarren. [Accessed: 22-Jan-2021].

[5] B. Wagner, "BillWagner - Overview," *GitHub*. [Online]. Available: https://github.com/BillWagner. [Accessed: 26-Jan-2021].

[6] D. Lee, "DennisLee-DennisLee - Overview," *GitHub*. [Online]. Available: https://github.com/DennisLee-DennisLee. [Accessed: 10-Feb-2021].

[7] nxtn, "nxtn - Overview," *GitHub*. [Online]. Available: https://github.com/nxtn. [Accessed: 10-Feb-2021].

[8] J. Kotas, "jkotas - Overview," *GitHub*. [Online]. Available: https://github.com/jkotas. [Accessed: 10-Feb-2021].

[9] M. Wenzel, "mairaw - Overview," *GitHub*. [Online]. Available: https://github.com/mairaw. [Accessed: 10-Feb-2021].

[10] N. Schonning, "nschonni - Overview," *GitHub*. [Online]. Available: https://github.com/nschonni. [Accessed: 10-Feb-2021].

[11] A. Aymeric, "nemrism - Overview," *GitHub*. [Online]. Available: https://github.com/nemrism. [Accessed: 10-Feb-2021].

[12] C. Maddock, "ChrisMaddock - Overview," *GitHub*. [Online]. Available: https://github.com/ChrisMaddock. [Accessed: 10-Feb-2021].

[13] xaviex, "xaviex - Overview," *GitHub*. [Online]. Available: https://github.com/xaviex. [Accessed: 10-Feb-2021].

[14] M. Jones, "Mikejo5000 - Overview," *GitHub*. [Online]. Available: https://github.com/Mikejo5000. [Accessed: 10-Feb-2021].

[15] L. Latham, "guardrex - Overview," *GitHub*. [Online]. Available: https://github.com/guardrex. [Accessed: 10-Feb-2021].

[16] tompratt-AQ, "tompratt-AQ - Overview," *GitHub*. [Online]. Available: https://github.com/tompratt-AQ. [Accessed: 10-Feb-2021].

[17] yishengjin1413, "yishengjin1413 - Overview," *GitHub*. [Online]. Available: https://github.com/yishengjin1413. [Accessed: 10-Feb-2021].

[18] S. Hoag, "stevehoag - Overview," *GitHub*. [Online]. Available: https://github.com/stevehoag. [Accessed: 10-Feb-2021].

[19] P. Onderka, "svick - Overview," *GitHub*. [Online]. Available: https://github.com/svick. [Accessed: 10-Feb-2021].

[20] T. Dykstra, "tdykstra - Overview," *GitHub*. [Online]. Available: https://github.com/tdykstra. [Accessed: 10-Feb-2021].

[21] A. De George, "Desktop Guide (Windows Forms .NET)." *Microsoft Docs*, 26-Oct-2020. [Online]. Available: https://docs.microsoft.com/en-us/dotnet/desktop/winforms/overview/. [Accessed: 19-Jan-2021].

References (continued)

[22] A. De George and M. Wenzel, "Tutorial: Create a new WinForms app (Windows Forms .NET)." ***Microsoft Docs***, 26-Oct-2020. [Online]. Available: https://docs.microsoft.com/en-us/dotnet/desktop/winforms/get-started/create-app-visual-studio. [Accessed: 28-Jan-2021].

[23] A. De George, "adegeo - Overview," ***GitHub***. [Online]. Available: https://github.com/adegeo. [Accessed: 22-Jan-2021].

[24] O. Altunyan, G. Hogenson, D. Coulter, Y. Victor, and T. Lee, "Create a Windows Forms app with C# - Visual Studio." ***Microsoft Docs***, 26-Sep-2019. [Online]. Available: https://docs.microsoft.com/en-us/visualstudio/ide/create-csharp-winform-visual-studio. [Accessed: 28-Jan-2021].

[25] O. Altunyan, "ornellaalt - Overview," ***GitHub***. [Online]. Available: https://github.com/ornellaalt. [Accessed: 22-Jan-2021].

[26] E. Nakamura, "eddynaka - Overview," ***GitHub***. [Online]. Available: https://github.com/eddynaka. [Accessed: 13-Feb-2021].

[27] D. Mabee, "damabe - Overview," ***GitHub***. [Online]. Available: https://github.com/damabe. [Accessed: 13-Feb-2021].

[28] A. Dev, "amaldevv - Overview," ***GitHub***. [Online]. Available: https://github.com/amaldevv. [Accessed: 13-Feb-2021].

[29] M. Blome, "mikeblome - Overview," ***GitHub***. [Online]. Available: https://github.com/mikeblome. [Accessed: 14-Feb-2021].

[30] A. Sal, "abdullahsalem - Overview," ***GitHub***. [Online]. Available: https://github.com/abdullahsalem. [Accessed: 14-Feb-2021].

[31] M. Koudelka, "v-makoud - Overview," ***GitHub***. [Online]. Available: https://github.com/v-makoud. [Accessed: 14-Feb-2021].

[32] B. Wagner, ".NET documentation." ***Microsoft Docs***, 2021. [Online]. Available: https://docs.microsoft.com/en-us/dotnet/. [Accessed: 14-Feb-2021].

[33] G. Hogenson, "ghogen - Overview," ***GitHub***. [Online]. Available: https://github.com/ghogen. [Accessed: 15-Feb-2021].

[34] Y. Victor, "Youssef1313 - Overview," ***GitHub***. [Online]. Available: https://github.com/Youssef1313. [Accessed: 15-Feb-2021].

[35] T. Lee, "TerryGLee - Overview," ***GitHub***. [Online]. Available: https://github.com/TerryGLee. [Accessed: 15-Feb-2021].

[36] D. Coulter, "DCtheGeek - Overview," ***GitHub***. [Online]. Available: https://github.com/DCtheGeek. [Accessed: 04-Feb-2021].

[37] "About Mono | Mono." ***Mono Project***. [Online]. Available: https://www.mono-project.com/docs/about-mono/. [Accessed: 27-Feb-2021].

[38] "Home | Mono." ***Mono Project***, 2021. [Online]. Available: https://www.mono-project.com/. [Accessed: 28-Feb-2021].

[39] .NET Foundation and Contributors, ".NET Foundation." ***.NET Foundation***, 2021. [Online]. Available: https://dotnetfoundation.org/. [Accessed: 28-Feb-2021].

[40] "Xamarin | Open-source mobile app platform for .NET," ***Microsoft***. [Online]. Available: https://dotnet.microsoft.com/apps/xamarin. [Accessed: 27-Feb-2021].

[41] "What is Xamarin? | .NET," ***Microsoft***. [Online]. Available: https://dotnet.microsoft.com/learn/xamarin/what-is-xamarin. [Accessed: 27-Feb-2021].

[42] "Cross-platform with Xamarin | .NET," ***Microsoft***. [Online]. Available: https://dotnet.microsoft.com/apps/xamarin/cross-platform. [Accessed: 27-Feb-2021].

References (continued)

[43] "ECMA-335," *Ecma International*, 2012. [Online]. Available: https://www.ecma-international.org/publications-and-standards/standards/ecma-335/. [Accessed: 28-Feb-2021].

[44] "ECMA-334," *Ecma International*, 2017. [Online]. Available: https://www.ecma-international.org/publications-and-standards/standards/ecma-334/. [Accessed: 28-Feb-2021].

[45] "Microsoft Announces Visual Studio 97, A Comprehensive Suite of Microsoft Visual Development Tools," *Stories*, 28-Jan-1997. [Online]. Available: https://news.microsoft.com/1997/01/28/microsoft-announces-visual-studio-97-a-comprehensive-suite-of-microsoft-visual-development-tools/. [Accessed: 05-Mar-2021].

[46] "WineHQ - Visual Studio 2002 (7.0)." *WINE^{HQ}*. [Online]. Available: https://appdb.winehq.org/objectManager.php?sClass=version&iId=15422. [Accessed: 05-Mar-2021].

[47] P. Morlion, "The Death of WinForms Has Been Greatly Exaggerated - SubMain Blog," *Software Quality Blog - SubMain Software*, 28-Aug-2018. [Online]. Available: https://blog.submain.com/death-winforms-greatly-exaggerated/. [Accessed: 05-Mar-2021].

[48] T. Segal, "Profit Center: Characteristics vs. a Cost Center, With Examples," *Investopedia*, 07-Dec-2020. [Online]. Available: https://www.investopedia.com/terms/p/profitcentre.asp. [Accessed: 31-May-2024].

[49] "Test Coverage Tutorial: Comprehensive Guide With Best Practices." *lambdatest.com*. [Online]. Available: https://www.lambdatest.com/learning-hub/test-coverage. [Accessed: 31-May-2024].

[50] H. Shah, "A Detailed Guide on Test Coverage," *Simform - Product Engineering Company*, 28-Jun-2021. [Online]. Available: https://www.simform.com/blog/test-coverage/. [Accessed: 31-May-2024].

[51] T. Hamilton, "Test Coverage in Software Testing," *Guru99*, 13-Apr-2024. [Online]. Available: https://www.guru99.com/test-coverage-in-software-testing.html. [Accessed: 31-May-2024].

[52] "What is DO-178B?," *AdaCore*. [Online]. Available: https://www.adacore.com/industries/avionics/what-is-do-178b. [Accessed: 31-May-2024].

[53] D. Goswami, "Why does your Avionics Software Need DO-178B/C Compliance?," *Qualitest*, 24-Dec-2020. [Online]. Available: https://medium.com/qualitest/why-does-your-avionics-software-need-do-178b-c-compliance-4d1cba8c94a0. [Accessed: 31-May-2024].

[54] V. Nanda, "Test Coverage in Software Testing." *tutorialspoint.com*. [Online]. Available: https://www.tutorialspoint.com/test-coverage-in-software-testing. [Accessed: 31-May-2024].

[55] "Manual vs Automated Testing | Cypress Testing Tools." *cypress.io*. [Online]. Available: https://learn.cypress.io/testing-foundations/manual-vs-automated-testing. [Accessed: 30-May-2024].

[56] "End-to-End Testing: Your First Test with Cypress | Cypress Documentation," *Cypress Documentation*, 09-May-2024. [Online]. Available: https://docs.cypress.io/guides/end-to-end-testing/writing-your-first-end-to-end-test. [Accessed: 30-May-2024].

[57] "From Manual Testing to Automated Testing With Cypress - Our Story | Cogworks Blog." *Cogworks Blog*. [Online]. Available: https://www.wearecogworks.com/innerworks/cogworks-blog-archive/from-manual-to-automated-testing-with-cypress-our-story. [Accessed: 30-May-2024].

[58] "Best Practices | Cypress Documentation," *Cypress Documentation*, 11-Apr-2024. [Online]. Available: https://docs.cypress.io/guides/references/best-practices. [Accessed: 30-May-2024].

[59] "Testing Frameworks for Javascript | Write, Run, Debug | Cypress." *cypress.io*. [Online]. Available: https://www.cypress.io/. [Accessed: 30-May-2024].

[60] "Comprehensive Cypress Test Automation Guide | Cypress Documentation," *Cypress Documentation*, 11-Apr-2024. [Online]. Available: https://docs.cypress.io/guides/overview/why-cypress. [Accessed: 30-May-2024].

References (continued)

[61] "Key Differences | Cypress Documentation," *Cypress Documentation*, 14-Dec-2022. [Online]. Available: https://docs.cypress.io/guides/overview/key-differences. [Accessed: 30-May-2024].

[62] "How Cypress Works | End to end and component testing tools." *cypress.io*. [Online]. Available: https://www.cypress.io/how-it-works. [Accessed: 30-May-2024].

[63] G. Hegde, "Getting Started with Cypress Test Automation : Tutorial," *BrowserStack*. [Online]. Available: https://browserstack.wpengine.com/guide/cypress-automation-tutorial/. [Accessed: 30-May-2024].

[64] G. Hegde and P. Bhat, "How to perform Component Testing using Cypress," *BrowserStack*, 03-Oct-2022. [Online]. Available: https://browserstack.wpengine.com/guide/cypress-component-testing/. [Accessed: 30-May-2024].

[65] K. Pathak, "How to Perform Component Testing using Cypress," *Medium*, 29-Jan-2023. [Online]. Available: https://kailash-pathak.medium.com/how-to-perform-component-testing-using-cypress-3bbe74af7492. [Accessed: 30-May-2024].

[66] F. Hric, "Component testing in Cypress: What is it and why it's important," *Automated Visual Testing | Applitools*, 15-Dec-2022. [Online]. Available: https://applitools.com/blog/component-testing-in-cypress-what-is-it-and-why-its-important/. [Accessed: 30-May-2024].

[67] Technocrat, "How to write Test Case in Cypress: (with testing example)," *BrowserStack*. [Online]. Available: https://browserstack.wpengine.com/guide/how-to-write-test-case-in-cypress/. [Accessed: 30-May-2024].

[68] D. Greene, "Wow! Cypress can run unit tests! 😻," *DEV Community*, 13-Nov-2020. [Online]. Available: https://dev.to/dgreene1/wow-cypress-can-run-unit-tests-15l5. [Accessed: 29-May-2024].

[69] "Recipes | Cypress Documentation," *Cypress Documentation*, 11-Apr-2024. [Online]. Available: https://docs.cypress.io/examples/recipes. [Accessed: 29-May-2024].

[70] "Code Coverage | Cypress Documentation," *Cypress Documentation*, 25-Mar-2024. [Online]. Available: https://docs.cypress.io/guides/tooling/code-coverage. [Accessed: 29-May-2024].

[71] G. Bahmutov, "Readable Cypress.io tests," *Better world by better software*, 01-Apr-2019. [Online]. Available: https://glebbahmutov.com/blog/readable-tests/index.html. [Accessed: 29-May-2024].

[72] "cypress-example-recipes/examples/unit-testing__application-code at master · cypress-io/cypress-example-recipes," *GitHub*. [Online]. Available: https://github.com/cypress-io/cypress-example-recipes/tree/master/examples/unit-testing__application-code. [Accessed: 29-May-2024].

[73] "Cypress Testing Framework Tutorial With Examples | Sauce Labs." *Sauce Labs*. [Online]. Available: https://saucelabs.com/resources/blog/getting-started-with-cypress. [Accessed: 29-May-2024].

[74] Morrismoses, "Cypress For Test Automation: A Step-by-Step Guide," *Medium*, 22-Apr-2024.[Online]. Available: https://medium.com/@morrismoses149/cypress-for-test-automation-a-step-by-step-guide-93cb47163d05. [Accessed: 29-May-2024].

[75] S. Nadeesha, "Making a simple Test Automation Framework with Cypress.io." *LinkedIn*. [Online]. Available: https://www.linkedin.com/pulse/making-simple-test-automation-framework-cypressio-supun-nadeesha. [Accessed: 29-May-2024].

[76] "Selenium vs Cypress vs alternatives: choosing the right framework for web testing." *zebrunner.com*. [Online]. Available: https://zebrunner.com/blog-posts/what-is-cypress-and-is-it-a-real-alternative-to-selenium. [Accessed: 29-May-2024].

[77] "How to use Cypress Testing Framework?," *Testbytes Softwares*. [Online]. Available: https://www.testbytes.net/blog/cypress-testing/. [Accessed: 29-May-2024].

[78] "Reporters | Cypress Documentation," 25-Mar-2024. *Cypress Documentation*. [Online]. Available: https://docs.cypress.io/guides/tooling/reporters. [Accessed: 29-May-2024].

References (continued)

[79] "Mocha - the fun, simple, flexible JavaScript test framework." ***mochajs.org***. [Online]. Available: https://mochajs.org/. [Accessed: 29-May-2024].

[80] "Jest vs Mocha: Which one Should You Choose?," ***GeeksforGeeks***, 27-Feb-2024. [Online]. Available: https://www.geeksforgeeks.org/jest-vs-mocha-which-one-should-you-choose/. [Accessed: 29-May-2024].

[81] "Bundled Libraries | Cypress Documentation," ***Cypress Documentation***, 15-Dec-2022. [Online]. Available: https://docs.cypress.io/guides/references/bundled-libraries. [Accessed: 29-May-2024].

[82] "Chai." ***chaijs.com***. [Online]. Available: https://www.chaijs.com/. [Accessed: 29-May-2024].

[83] D. Vasudevan, "How can you use Jest or Cypress to ensure front-end code quality?" ***LinkedIn***. [Online]. Available: https://www.linkedin.com/advice/3/how-can-you-use-jest-cypress-ensure-front-end-code-k6bnf. [Accessed: 29-May-2024].

[84] "Cypress vs Jest | Top 15 Key Differences," ***Testsigma Blog***, 02-Apr-2024. [Online]. Available: https://testsigma.com/blog/cypress-vs-jest/. [Accessed: 29-May-2024].

[85] "Testing Types | Cypress Documentation," ***Cypress Documentation***, 30-Jan-2023. [Online]. Available: https://docs.cypress.io/guides/core-concepts/testing-types. [Accessed: 29-May-2024].

[86] "Karma - Spectacular Test Runner for Javascript." [Online]. Available: https://karma-runner.github.io/latest/index.html. [Accessed: 29-May-2024].

[87] K. Choksi, "How to write unit tests with Jasmine & Karma?," ***Simform Engineering***, 01-Jun-2023. [Online]. Available: https://medium.com/simform-engineering/how-to-write-unit-tests-with-jasmine-karma-f1908bdeb617. [Accessed: 29-May-2024].

[88] Testim, "Karma JS Testing: What, Why, and How to Get Going Right Now," ***AI-driven E2E automation with code-like flexibility for your most resilient tests***, 11-Dec-2021. [Online]. Available: https://www.testim.io/blog/karma-js-testing-what-why-and-how-to-get-going-right-now/. [Accessed: 29-May-2024].

[89] "How to Install Cypress for Test Automation," ***BrowserStack***. [Online]. Available: https://browserstack.wpengine.com/guide/cypress-installation-for-test-automation/. [Accessed: 30-May-2024].

[90] "Code Coverage | Cypress Documentation," ***Cypress Documentation***, 25-Mar-2024. [Online]. Available: https://docs.cypress.io/guides/tooling/code-coverage. [Accessed: 30-May-2024].

[91] "Open the App with Cypress: Step-by-Step Guide | Cypress Documentation," ***Cypress Documentation***, 19-Apr-2024. [Online]. Available: https://docs.cypress.io/guides/getting-started/opening-the-app. [Accessed: 30-May-2024].

[92] "End-to-End Testing: Your First Test with Cypress | Cypress Documentation," ***Cypress Documentation***, 09-May-2024. [Online]. Available: https://docs.cypress.io/guides/end-to-end-testing/writing-your-first-end-to-end-test. [Accessed: 30-May-2024].

[93] "Installing Cypress and writing your first test | Cypress Testing Tools." [Online]. Available: https://learn.cypress.io/testing-your-first-application/installing-cypress-and-writing-your-first-test. [Accessed: 30-May-2024].

[94] admin, "Cypress Code Coverage," ***testomat.io***, 07-Feb-2024. [Online]. Available: https://testomat.io/blog/cypress-code-coverage/. [Accessed: 29-May-2024].

[95] "Setting up Cypress Code Coverage," ***BrowserStack***. [Online]. Available: https://browserstack.wpengine.com/guide/cypress-code-coverage/. [Accessed: 29-May-2024].

[96] "Node.js — Run JavaScript Everywhere." ***nodejs.org***. [Online]. Available: https://nodejs.org/en. [Accessed: 31-May-2024].

[97] "Node.js Tutorial." ***W3Schools***. [Online]. Available: https://www.w3schools.com/nodejs/. [Accessed: 31-May-2024].

References (continued)

[98] "Node.js — Introduction to Node.js." ***nodejs.org***. [Online]. Available: https://nodejs.org/en/learn/getting-started/introduction-to-nodejs. [Accessed: 31-May-2024].

[99] M. Wankhede, "Automation testing with Cypress, Mocha, and JavaScript," ***TO THE NEW BLOG***, 17-Feb-2023. [Online]. Available: https://www.tothenew.com/blog/automation-testing-with-cypress-mocha-and-javascript/. [Accessed: 06-Jun-2024].

[100] C. Adams, "Cypress Limitations you should be aware of," ***Codoid***, 12-Jan-2023. [Online]. Available: https://codoid.com/automation-testing/cypress-limitations-you-should-be-aware-of/. [Accessed: 06-Jun-2024].

[101] "Mocha vs Cypress.io comparison of testing frameworks." ***Knapsack Pro***. [Online]. Available: https://knapsackpro.com/testing_frameworks/difference_between/mochajs/vs/cypress-io. [Accessed: 06-Jun-2024].

[102] K. Halder, "Automated Testing with Cypress," ***The Startup***, 03-May-2020. [Online]. Available: https://medium.com/swlh/automated-testing-with-cypress-17bf74bfd97d. [Accessed: 06-Jun-2024].

[103] "Comparing cypress vs. jasmine vs. jest vs. karma vs. mocha," ***NPMCompare***. [Online]. Available: https://npmcompare.com/compare/cypress,jasmine,jest,karma,mocha. [Accessed: 06-Jun-2024].

[104] "Comparing cucumber vs. cypress vs. jasmine vs. mocha vs. nightwatch," ***NPMCompare***. [Online]. Available: https://npmcompare.com/compare/cucumber,cypress,jasmine,mocha,nightwatch. [Accessed: 06-Jun-2024].

[105] R. Kumari, "Cypress vs Cucumber - Which One to Choose?," ***Testsigma Blog***, 03-Feb-2024. [Online]. Available: https://testsigma.com/blog/cypress-vs-cucumber/. [Accessed: 06-Jun-2024].

[106] "Cucumber vs Cypress.io comparison of testing frameworks." ***Knapsack Pro***. [Online]. Available: https://knapsackpro.com/testing_frameworks/difference_between/cucumber/vs/cypress-io. [Accessed: 06-Jun-2024].

[107] "Selenium," ***Selenium***. [Online]. Available: https://www.selenium.dev/. [Accessed: 09-Jun-2024].

[108] "WebDriver," ***Selenium***, 29-Mar-2024. [Online]. Available: https://www.selenium.dev/documentation/webdriver/. [Accessed: 09-Jun-2024].

[109] "WebDriver." ***W3C***, 2018. [Online]. Available: https://www.w3.org/TR/webdriver1/. [Accessed: 09-Jun-2024].

[110] "Getting started," ***Selenium***. [Online]. Available: https://www.selenium.dev/documentation/webdriver/getting_started/. [Accessed: 09-Jun-2024].

[111] "Write your first Selenium script," ***Selenium***. [Online]. Available: https://www.selenium.dev/documentation/webdriver/getting_started/first_script/. [Accessed: 09-Jun-2024].

[112] "Supported Browsers," ***Selenium***, 20-Sep-2022. [Online]. Available: https://www.selenium.dev/documentation/webdriver/browsers/. [Accessed: 09-Jun-2024].

[113] "Downloads," ***Selenium***. [Online]. Available: https://www.selenium.dev/downloads/. [Accessed: 09-Jun-2024].

[114] "Selenium WebDriver Tutorial - javatpoint," ***www.javatpoint.com***. [Online]. Available: https://www.javatpoint.com/selenium-webdriver. [Accessed: 09-Jun-2024].

[115] "WebDriver For Mobile Browsers," ***Selenium***, 12-Jan-2022. [Online]. Available: https://www.selenium.dev/documentation/legacy/selenium_2/mobile/. [Accessed: 09-Jun-2024].

[116] C. Tozzi , "Can Selenium Be Used for Mobile Testing?" ***Sauce Labs Inc.***, 04-Feb-2023. [Online]. Available: https://saucelabs.com/resources/blog/can-selenium-be-used-for-mobile-testing. [Accessed: 09-Jun-2024].

[117] "Welcome - Appium Documentation." ***Appium Documentation***. [Online]. Available: https://appium.io/docs/en/latest/. [Accessed: 10-Jun-2024].

References (continued)

[118] "Appium in a Nutshell - Appium Documentation." ***Appium Documentation***. [Online]. Available: https://appium.io/docs/en/latest/intro/. [Accessed: 10-Jun-2024].

[119] "Selendroid: Selenium for Android." ***selendroid.io***. [Online]. Available: http://selendroid.io/. [Accessed: 10-Jun-2024].

[120] "Mobile Testing - Selendroid Framework." ***www.tutorialspoint.com***. [Online]. Available: https://www.tutorialspoint.com/mobile_testing/mobile_testing_selendroid_framework.htm#:~:text=Selendroid%20is%20a%20test%20automation,the%20Selenium%20Webdriver%20client%20API. [Accessed: 10-Jun-2024].

[121] "Appium vs Selendroid," ***GeeksforGeeks***, 10-Dec-2023. [Online]. Available: https://www.geeksforgeeks.org/appium-vs-selendroid/. [Accessed: 10-Jun-2024].

[122] adegeo, "Install .NET on Windows - .NET," ***Microsoft Learn***, 20-Jun-2024. [Online]. Available: https://learn.microsoft.com/en-us/dotnet/core/install/windows. [Accessed: 29-Aug-2024].

[123] tdykstra, ".NET CLI," 20-Jun-2024. ***Microsoft Learn***. [Online]. Available: https://learn.microsoft.com/en-us/dotnet/core/tools/. [Accessed: 29-Aug-2024].

[124] tdykstra, "dotnet command - .NET CLI," ***Microsoft Learn***, 11-Oct-2022. [Online]. Available: https://learn.microsoft.com/en-us/dotnet/core/tools/dotnet. [Accessed: 29-Aug-2024].

[125] tdykstra, "dotnet test command - .NET CLI," ***Microsoft Learn***, 27-Mar-2024. [Online]. Available: https://learn.microsoft.com/en-us/dotnet/core/tools/dotnet-test. [Accessed: 29-Aug-2024].

[126] "Using xUnit to Test your C# Code," ***Auth0 - Blog***. [Online]. Available: https://auth0.com/blog/xunit-to-test-csharp-code/. [Accessed: 29-Aug-2024].

[127] B. Code, "Unit Testing in C# with xUnit: Complete guide," ***Medium***, 11-Apr-2024. [Online]. Available: https://medium.com/@codebob75/unit-testing-in-c-with-xunit-complete-guide-18ee2b919b05. [Accessed: 29-Aug-2024].

[128] "Getting started: .NET Core with command line," ***xUnit.net***. [Online]. Available: https://xunit.net/docs/getting-started/v2/netcore/cmdline. [Accessed: 29-Aug-2024].

[129] "Getting started: .NET Framework with command line," ***xUnit.net***. [Online]. Available: https://xunit.net/docs/getting-started/v2/netfx/cmdline. [Accessed: 29-Aug-2024].

[130] "Capturing Output," ***xUnit.net***. [Online]. Available: https://xunit.net/docs/capturing-output. [Online]. [Accessed: 29-Aug-2024].

[131] erikdietrich, "The history of C#," ***Microsoft Learn***, 07-Mar-2024. [Online]. Available: https://learn.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-version-history. [Accessed: 29-Aug-2024].

[132] "Download - Stable | Mono." [Online]. Available: https://www.mono-project.com/download/stable/. [Accessed: 29-Aug-2024].

[133] "Microsoft Offloads The Mono Project To Wine." [Online]. Available: https://www.phoronix.com/news/Microsoft-Gives-Mono-To-Wine. [Accessed: 29-Aug-2024].

[134] "How YOU can get started with .NET Core and C# in VS Code." [Online]. Available: https://softchris.github.io/pages/dotnet-core.html#resources. [Accessed: 30-Aug-2024].

[135] "Home," ***xUnit.net***. [Online]. Available: https://xunit.net/. [Accessed: 03-Sept-2024].

[136] "Apache License, Version 2.0," ***Open Source Initiative***. [Online]. Available: https://opensource.org/license/apache-2-0. [Accessed: 03-Sept-2024].

[137] "Why Did We Build xUnit 1.0?," ***xUnit.net***. [Online]. Available: https://xunit.net/docs/why-did-we-build-xunit-1.0. [Accessed: 03-Sept-2024].

References (continued)

[138] "Using xUnit to Test your C# Code," ***Auth0 - Blog***. [Online]. Available: https://auth0.com/blog/xunit-to-test-csharp-code/. [Accessed: 03-Sept-2024].

[139] "Getting Started: .NET Framework with Visual Studio," ***xUnit.net***. [Online]. Available: https://xunit.net/docs/getting-started/v2/netfx/visual-studio. [Accessed: 03-Sept-2024].

[140] S. Dhakne, "Why Should You Use xUnit? A Unit Testing Framework For .NET." [Online]. Available: https://www.clariontech.com/blog/why-should-you-use-xunit-a-unit-testing-framework-for-.net. [Accessed: 03-Sept-2024].

[141] "NUnit vs xUnit vs MSTest: .NET Unit Testing Framework Comparison." [Online]. Available: https://daily.dev/blog/nunit-vs-xunit-vs-mstest-net-unit-testing-framework-comparison. [Accessed: 03-Sept-2024].

[142] A. Kumar, "Unit Testing with xUnit." [Online]. Available: https://www.c-sharpcorner.com/article/unit-testing-with-xunit-in-net/. [Accessed: 03-Sept-2024].

[143] "Testing with the xUnit Framework - Overview (2 of 12)," ***Microsoft Learn***, 29-Nov-2022. [Online]. Available: https://learn.microsoft.com/en-us/shows/visual-studio-toolbox/testing-with-the-xunit-framework-overview-2-of-12-automated-software-testing. [Accessed: 03-Sept-2024].

[144] "Testing with the xUnit Framework - Overview (2 of 12)," ***Microsoft Learn***, 29-Nov-2022. [Online]. Available: https://learn.microsoft.com/en-us/shows/visual-studio-toolbox/testing-with-the-xunit-framework-overview-2-of-12-automated-software-testing. [Accessed: 03-Sept-2024].

[145] V. Magalhães, "Test Coverage Analysis with Coverlet in .NET," ***Medium***, 23-Sept-2023. [Online]. Available: https://victormagalhaes-dev.medium.com/test-coverage-analysis-with-coverlet-in-net-2e38df3c6ed7. [Accessed: 03-Sept-2024].

[146] G. Barré, "Computing code coverage for a .NET project - Gérald Barré," ***Meziantou's blog***, 15-Apr-2024. [Online]. Available: https://www.meziantou.net/computing-code-coverage-for-a-dotnet-project.htm. [Accessed: 03-Sept-2024].

[147] gewarren, "dotnet-coverage code coverage tool - .NET CLI - .NET", 24-Feb-2023. [Online]. Available: https://learn.microsoft.com/en-us/dotnet/core/additional-tools/dotnet-coverage. [Accessed: 04-Sept-2024].

[148] "ReportGenerator - Code coverage reports." [Online]. Available: https://reportgenerator.io/. [Accessed: 04-Sept-2024].

[149] E. Dahl, "Code Coverage Reports for .NET Projects." [Online]. Available: https://knowyourtoolset.com/2024/01/coverage-reports/. [Accessed: 04-Sept-2024].

[150] "Code Coverage in .NET," ***my tech ramblings***, 10-Jul-2024. [Online]. Available: https://www.mytechramblings.com/posts/code-coverage-in-dotnet/. [Accessed: 04-Sept-2024].

[151] C. Allen, "Generating Code Coverage Metrics for .NET Framework Applications," ***Coding with Calvin - Calvin Allen***, 18-Aug-2022. [Online]. Available: https://www.codingwithcalvin.net/generating-code-coverage-metrics-for-net-framework-applications/. [Accessed: 04-Sept-2024].

[152] J. Carracedo, "Code coverage in .Net Core," ***Medium***, 09-May-2021. [Online]. Available: https://jke94.medium.com/code-coverage-in-net-core-840bcd96b1e9. [Accessed: 04-Sept-2024].

[153] "NET Core Code Coverage," ***DEV Community***, 07-Mar-2021. [Online]. Available: https://dev.to/eduardstefanescu/net-core-code-coverage-1dfn. [Accessed: 04-Sept-2024].

[154] L. Bennett, "9 Best Code Coverage Tools for Java, Python, C, C++, C#, .NET," 13-Aug-2024. [Online]. Available: https://www.guru99.com/code-coverage-tools.html. [Accessed: 04-Sept-2024].

[155] "Cobertura." [Online]. Available: https://cobertura.github.io/cobertura/. [Accessed: 04-Sept-2024].

[156] R. Singh, "What is Cobertura and use cases of Cobertura ?," ***DevOpsSchool.com***, 15-Feb-2024. [Online]. Available: https://www.devopsschool.com/blog/what-is-cobertura-and-use-cases-of-cobertura/. [Accessed: 04-Sept-2024].

References (continued)

[157] "ReportGenerator - Code coverage reports." [Online]. Available: https://reportgenerator.io/usage. [Accessed: 04-Sept-2024].

[158] DeDiv-VR, "Visual Studio 2022 System Requirements," *Microsoft Learn*, 09-Apr-2024. [Online]. Available: https://learn.microsoft.com/en-us/visualstudio/releases/2022/system-requirements. [Accessed: 30-Sept-2024].

[159] "Download Visual Studio Tools - Install Free for Windows, Mac, Linux," *Visual Studio*. [Online]. Available: https://visualstudio.microsoft.com/downloads/. [Accessed: 30-Sept-2024].

[160] DeDiv-VR, "Visual Studio 2019 System Requirements," *Microsoft Learn*, 22-Jan-2024. [Online]. Available: https://learn.microsoft.com/en-us/visualstudio/releases/2019/system-requirements. [Accessed: 30-Sept-2024].

[161] DeDiv-VR, "Visual Studio 2017 System Requirements," 22-Jan-2024. [Online]. Available: https://learn.microsoft.com/en-us/visualstudio/releases/2017/vs2017-system-requirements-vs. [Accessed: 30-Sept-2024].

[162] "Requirements for Visual Studio Code." [Online]. Available: https://code.visualstudio.com/docs/supporting/requirements. [Accessed: 30-Sept-2024].

[163] K. Buzdar, "How to Install Visual Studio Code on Windows," *Ultahost Knowledge Base*, 25-May-2024. [Online]. Available: https://ultahost.com/knowledge-base/install-visual-studio-code-windows/. [Accessed: 30-Sept-2024].

[164] "Configure Visual Studio Code for Microsoft C++." [Online]. Available: https://code.visualstudio.com/docs/cpp/config-msvc. [Accessed: 30-Sept-2024].

[165] TylerMSFT, "Use the Microsoft C++ toolset from the command line," 02-Mar-2023. [Online]. Available: https://learn.microsoft.com/en-us/cpp/build/building-on-the-command-line?view=msvc-170. [Accessed: 30-Sept-2024].

[166] "Developer Community." [Online]. Available: https://developercommunity.visualstudio.com/t/always-show-it-looks-like-you-dont-have-enough-dis/26279. [Accessed: 30-Sept-2024].

[167] "C/C++ - Visual Studio Marketplace." [Online]. Available: https://marketplace.visualstudio.com/items?itemName=ms-vscode.cpptools. [Accessed: 30-Sept-2024].

[168] "Get Started with C++ and Windows Subsystem for Linux in Visual Studio Code." [Online]. Available: https://code.visualstudio.com/docs/cpp/config-wsl. [Accessed: 30-Sept-2024].

[169] "WSL - Visual Studio Marketplace." [Online]. Available: https://marketplace.visualstudio.com/items?itemName=ms-vscode-remote.remote-wsl. [Accessed: 30-Sept-2024].

[170] craigloewen-msft, "Install WSL," 28-Aug-2023. [Online]. Available: https://learn.microsoft.com/en-us/windows/wsl/install. [Accessed: 30-Sept-2024].

[171] "C++ programming with Visual Studio Code." [Online]. Available: https://code.visualstudio.com/docs/languages/cpp. [Accessed: 30-Sept-2024].

[172] "Download Visual Studio Code - Mac, Linux, Windows." [Online]. Available: https://code.visualstudio.com/Download. [Accessed: 30-Sept-2024].

[173] "MSYS2." [Online]. Available: https://www.msys2.org/. [Accessed: 02-Oct-2024].

[174] "MSYS2 - Updating MSYS2." [Online]. Available: https://www.msys2.org/docs/updating/. [Accessed: 02-Oct-2024].

[175] "MSYS2 - Package Management." [Online]. Available: https://www.msys2.org/docs/package-management/. [Accessed: 02-Oct-2024].

[176] "pacman/Tips and tricks - ArchWiki." [Online]. Available: https://wiki.archlinux.org/title/Pacman/Tips_and_tricks. [Accessed: 02-Oct-2024].

References (continued)

[177] S. Ifti, "How to Install GCC and GDB on Windows Using MSYS2 — Tutorial," ***Medium***, 23-Nov-2023. [Online]. Available: https://sajidifti.medium.com/how-to-install-gcc-and-gdb-on-windows-using-msys2-tutorial-0fceb7e66454. [Accessed: 02-Oct-2024].

[178] "Compiling with g++," ***GeeksforGeeks***, 19-Feb-2019. [Online]. Available: https://www.geeksforgeeks.org/compiling-with-g-plus-plus/. [Accessed: 07-Dec-2024].

[179] "An Overview of Functions ." [Online]. Available: https://www.umsl.edu/~lawtonb/224/functions4.html. [Accessed: 07-Dec-2024].

[180] P. Shen, "Code coverage testing of C/C++ projects using Gcov and LCOV," ***Medium***, 28-Jan-2022. [Online]. Available: https://medium.com/@xianpeng.shen/use-gcov-and-lcov-to-perform-code-coverage-testing-for-c-c-projects-c85708b91c78. [Accessed: 07-Dec-2024].

[181] "Code coverage with lcov — The Vector Packet Processor v22.10-0-g07e0c05e6 documentation." [Online]. Available: https://s3-docs.fd.io/vpp/22.10/developer/extras/lcov.html. [Accessed: 07-Dec-2024].

[182] Naveenkhasyap, "Generating Code Coverage Report Using GNU Gcov & Lcov.," ***DEV Community***, 16-Feb-2024. [Online]. Available: https://dev.to/naveenkhasyap/generating-code-coverage-report-using-gnu-gcov-lcov-59p. [Accessed: 07-Dec-2024].

[183] S. Jobing, "Code Coverage Analysis with GCC." ***LinkedIn***. [Online]. Available: https://www.linkedin.com/pulse/code-coverage-analysis-sander-jobing. [Accessed: 07-Dec-2024].

[184] "GoogleTest User's Guide," ***GoogleTest***. [Online]. Available: http://google.github.io/googletest/. [Accessed: 20-Dec-2024].

[185] "GoogleTest Primer," ***GoogleTest***. [Online]. Available: http://google.github.io/googletest/primer.html. [Accessed: 20-Dec-2024].

[186] "Assertions Reference," ***GoogleTest***. [Online]. Available: http://google.github.io/googletest/reference/assertions.html. [Accessed: 20-Dec-2024].

[187] "GTest Framework," ***GeeksforGeeks***, 25-Jan-2021. [Online]. Available: https://www.geeksforgeeks.org/gtest-framework/. [Accessed: 20-Dec-2024].

[188] C. Sethi, "Introduction to Unit Testing with Google Test (GTest) in C++," ***Medium***, 28-May-2023. [Online]. Available: https://medium.com/@chittaranjansethi/introduction-to-unit-testing-with-google-test-gtest-in-c-344a89b8eb4. [Accessed: 20-Dec-2024].

[189] TylerMSFT, "Create and run unit tests with Google Test for C++ - Visual Studio (Windows)," ***Microsoft Learn***, 12-Jan-2024. [Online]. Available: https://learn.microsoft.com/en-us/visualstudio/test/how-to-use-google-test-for-cpp?view=vs-2022. [Accessed: 20-Dec-2024].

[190] O. S, "Introduction to GoogleTest (gtest)," ***Medium***, 14-Dec-2022. [Online]. Available: https://levelup.gitconnected.com/introduction-to-googletest-gtest-1ece21f55d0e. [Accessed: 20-Dec-2024].

[191] B. Stroustrup, "Bjarne Stroustrup's Homepage." [Online]. Available: https://www.stroustrup.com/. [Accessed: 22-Mar-2025].

[192] B. Stroustrup, "Bjarne Stroustrup's FAQ," ***Bjarne Stroustrup's FAQ***, 26-May-2024. [Online]. Available: https://www.stroustrup.com/bs_faq.html#invention. [Accessed: 22-Mar-2025].

[193] B. Stroustrup, "Thriving in a crowded and changing world: C++ 2006–2020," ***Proc. ACM Program. Lang.***, vol. 4, no. HOPL, pp. 1–168, Jun. 2020, doi: 10.1145/3386320. [Online]. Available: https://dl.acm.org/doi/10.1145/3386320. [Accessed: 22-Mar-2025].

[194] "C++ Introduction." ***www.w3schools.com***. [Online]. Available: https://www.w3schools.com/cpp/cpp_intro.asp. [Accessed: 22-Mar-2025].

References (continued)

[195] TylerMSFT, “Microsoft C/C++ Documentation." ***Microsoft Learn***. [Online]. Available: https://learn.microsoft.com/en-us/cpp/?view=msvc-170. [Accessed: 22-Mar-2025].

[196] “Fast and reliable end-to-end testing for modern web apps | Playwright.” [Online]. Available: https://playwright.dev/. [Accessed: 04-Dec-2025] .

[197] O. Sheremeta, “Playwright Reporting: Full Guide and Best Practices,” ***testomat.io***, 16-Jul-2025. [Online]. Available: https://testomat.io/blog/playwright-reporting-generation/. [Accessed: 04-Dec-2025] .

[198] “Installation | Playwright.” [Online]. Available: https://playwright.dev/docs/intro. [Accessed: 04-Dec-2025] .

[199] “Writing tests | Playwright.” [Online]. Available: https://playwright.dev/docs/writing-tests. [Accessed: 04-Dec-2025] .

[200] “Running and debugging tests | Playwright.” [Online]. Available: https://playwright.dev/docs/running-tests. [Accessed: 04-Dec-2025] .

[201] “Setting up CI | Playwright.” [Online]. Available: https://playwright.dev/docs/ci-intro. [Accessed: 04-Dec-2025] .

[202] “Using Playwright Test to run Unit Tests,” 17-Jun-2022. [Online]. Available: https://patricktree.me/blog/using-playwright-to-run-unit-tests. [Accessed: 04-Dec-2025] .

[203] “How to use Playwright for unit testing,” 18-Aug-2025. [Online]. Available: https://gomakethings.com/how-to-use-playwright-for-unit-testing/. [Accessed: 04-Dec-2025] .

[204] “MIT License.” [Online]. Available: https://gomakethings.com/mit/. [Accessed: 04-Dec-2025] .

[205] M. Street, “Run unit tests in Playwright - Mike Street - Lead Developer and CTO,” ***Mike Street***, 08-May-2024. [Online]. Available: https://www.mikestreety.co.uk/blog/run-unit-tests-in-playwright/. [Accessed: 04-Dec-2025] .

[206] L. Davis, “How to Write Unit tests for Your Playwright Code?,” ***Metapress***, 07-Nov-2022. [Online]. Available: https://metapress.com/how-to-write-unit-tests-for-your-playwright-code/. [Accessed: 04-Dec-2025] .

[207] N. Victory, “Getting started with Playwright component testing,” ***LogRocket Blog***, 14-Sept-2022. [Online]. Available: https://blog.logrocket.com/getting-started-playwright-component-testing/. [Accessed: 04-Dec-2025] .

[208] E. Uchenna, “Testing with Playwright: A Tutorial Guide With Examples." 11-Oct-2023. [Online]. Available: https://blog.openreplay.com/testing-with-playwright/. [Accessed: 04-Dec-2025] .

[209] “Assertions | Playwright.” [Online]. Available: https://playwright.dev/docs/test-assertions. [Accessed: 04-Dec-2025] .

[210] “Coverage | Playwright.” [Online]. Available: https://playwright.dev/docs/api/class-coverage. [Accessed: 04-Dec-2025] .

[211] P. Patel, “Improving Playwright Test Coverage: Best Practices + Strategies.” 07-Jul-2025. [Online]. Available: https://www.alphabin.co/blog/playwright-test-coverage. [Accessed: 04-Dec-2025] .

[212] “Code coverage for a Next.js app using Playwright tests,” ***DEV Community***, 05-Apr-2022. [Online]. Available: https://dev.to/anishkny/code-coverage-for-a-nextjs-app-using-playwright-tests-18n7. [Accessed: 04-Dec-2025] .

[213] “istanbuljs/nyc.” ***Istanbul Code Coverage***, 29-Nov-2025. [Online]. Available: https://github.com/istanbuljs/nyc. [Accessed: 04-Dec-2025] .

[214] Schmitt, “mxschmitt/playwright-test-coverage.” 24-Oct-2025. [Online]. Available: https://github.com/mxschmitt/playwright-test-coverage. [Accessed: 04-Dec-2025] .

References (continued)

[215] anishkny, "[Feature] Enable Istanbul code coverage collection with Playwright Test · Issue #8734 · microsoft/playwright," *GitHub*, 07-Sept-2021. [Online]. Available: https://github.com/microsoft/playwright/issues/8734. [Accessed: 04-Dec-2025] .

[216] "Reporters | Playwright." [Online]. Available: https://playwright.dev/docs/test-reporters. [Accessed: 04-Dec-2025] .

[217] "Turn Playwright Test Reports into Insights." [Online]. Available: https://www.alphabin.co/blog/turn-playwright-test-reports-into-insights. [Accessed: 04-Dec-2025] .

[218] "Optimising Playwright Reporting Through Monocart | Insights from the Kablamo Team." [Online]. Available: https://engineering.kablamo.com.au/posts/optimising-playwright-reporting-through-monocart/. [Accessed: 04-Dec-2025] .

[219] CenFun, "cenfun/monocart-coverage-reports." 25-Nov-2025. [Online]. Available: https://github.com/cenfun/monocart-coverage-reports. [Accessed: 04-Dec-2025] .

[220] CenFun, "cenfun/monocart-reporter." 25-Nov-2025. [Online]. Available: https://github.com/cenfun/monocart-reporter. [Accessed: 04-Dec-2025] .

[221] "What are the steps to create a TypeScript code coverage report using Playwright and Monocart?," 30-Sept-2023. [Online]. Available: https://ray.run/questions/what-are-the-steps-to-create-a-typescript-code-coverage-report-using-playwright-and-monocart. [Accessed: 04-Dec-2025] .

[222] Q. A. Team, "Reporting with Playwright: Enhance Your Test Results," *JigNect Technologies Pvt Ltd*, 24-Dec-2024. [Online]. Available: https://jignect.tech/reporting-with-playwright-make-your-test-results-shine/. [Accessed: 04-Dec-2025] .

[223] E. Yortuçboylu, "Test coverage with Istanbul.js." 19-Dec-2017. [Online]. Available: https://medium.com/@yortuc/test-coverage-with-istanbul-js-6170726d43c4. [Accessed: 04-Dec-2025] .

[224] L. Nový, "Code coverage of E2E tests with Playwright." 19-Sept-2020. [Online]. Available: https://medium.com/@novyludek/code-coverage-of-e2e-tests-with-playwright-6f8b4c0b56e1. [Accessed: 04-Dec-2025] .

[225] "Modern E2E Testing for Angular Apps with Playwright." 13-May-2025. [Online]. Available: https://angular.love/modern-e2e-testing-for-angular-apps-with-playwright. [Accessed: 04-Dec-2025] .

This page intentionally left blank.

End of document