

# 2026-002-002 What I did

To: kimberly.d.hall52.civ@us.navy.mil

Cc (carbon copy): richard.j.harris120.civ@us.navy.mil, jeffrey.d.welch.civ@us.navy.mil, peter.c.burberry.civ@us.navy.mil, peter.cullen.burberry@gmail.com

Subject: Progress Update for 2026-002-002 Monday

What I did, 2026-002-002 Monday

Dear Kimberly D. Hall, NM-04 supervisor of 54520 competency, Jeffrey D. Welch, NM-04 supervisor of 54510 competency, Richard J. Harris Jr., ND-04 Scientist of 54520 competency, and Peter Cullen Burberry, ND-03 Scientist of 54520 competency,

I would like to give you a progress report for 2026-002-002 Monday. I worked on creating a test suite today. I successfully created a test suite at [https://github.com/PeterCullenBurberry/testng-cucumber-demonstration/tree/main/Cucumber\\_TestNG\\_Demo\\_Amazon\\_with\\_working\\_screenshots\\_Test\\_Suite\\_suite\\_2026\\_001\\_029](https://github.com/PeterCullenBurberry/testng-cucumber-demonstration/tree/main/Cucumber_TestNG_Demo_Amazon_with_working_screenshots_Test_Suite_suite_2026_001_029). I had a test suite before, but it was trivial. This test suite involves 2 test cases.

Here is how it works.

```

1  2026_002_015_055_030_834203000_America_slash_New_York_2026_W006_001_2026_033_1770065730_8342
2  03000_192_168_004_098_C-CLFQPLPPJQ_C-CLFQPLPPJQ_backslash_peter_Powershell_7_5_4
3  C:\GitHub-repositories\testng-cucumber-demonstration\Cucumber_TestNG_Demo_Amazon_with_working_sc
4  reenshots_Test_Suite_suite_2026_001_029\src\test\resources\features> cat .\AmazonCart.feature
5
6  Feature: Amazon Shopping Cart Suite
7
8
9  Scenario Outline: Search and add various items to cart
10  Given User is on Amazon homepage "https://www.amazon.com"
11  And User handles the initial interstitial if present
12  When User searches for <product_name>
13  And User selects the first product from results
14  And User adds the product to the cart
15  Then User should be able to navigate to the cart page
16
17  Examples:
18    | product_name           |
19    | "Fakanhui satin pants for women"      |
20    | "EVALESS pleated v neck blouse light pink" |
21  2026_002_015_055_032_513037400_America_slash_New_York_2026_W006_001_2026_033_1770065732_5130
22  37400_192_168_004_098_C-CLFQPLPPJQ_C-CLFQPLPPJQ_backslash_peter_Powershell_7_5_4
23  C:\GitHub-repositories\testng-cucumber-demonstration\Cucumber_TestNG_Demo_Amazon_with_working_sc
24  reenshots_Test_Suite_suite_2026_001_029\src\test\resources\features>
```

This will create 2 test cases for "Fakanhui satin pants for women" and "EVALESS pleated v neck blouse light pink". I learned about the Gherkin syntax at <https://cucumber.io/docs/gherkin/reference>. For an explanation of how I went from

```

1  2026_002_016_050_033_492312400_America_slash_New_York_2026_W006_001_2026_033_1770069033_4923
2  12400_192_168_004_098_C-CLFQPLPPJQ_C-CLFQPLPPJQ_backslash_peter_Powershell_7_5_4
```

```

2 C:\GitHub-repositories\testng-cucumber-demonstration\Cucumber_TestNG_Demo_Amazon_with_working_sc
reenshots_2026_001_027\src\test\resources\features> cat .\AmazonCart.feature
3 Feature: Amazon Shopping Cart
4
5 Scenario: Search and add item to cart
6   Given User is on Amazon homepage "https://www.amazon.com"
7   And User handles the initial interstitial if present
8   When User searches for "Fakanhui satin pants for women"
9   And User selects the first product from results
10  And User adds the product to the cart
11  Then User should be able to navigate to the cart page
12 2026_002_016_050_035_346157800_America_slash_New_York_2026_W006_001_2026_033_1770069035_3461
57800_192_168_004_098_C-CLFQPLPPJQ_C-CLFQPLPPJQ_backslash_peter_Powershell_7_5_4
13 C:\GitHub-repositories\testng-cucumber-demonstration\Cucumber_TestNG_Demo_Amazon_with_working_sc
reenshots_2026_001_027\src\test\resources\features>

```

to

```

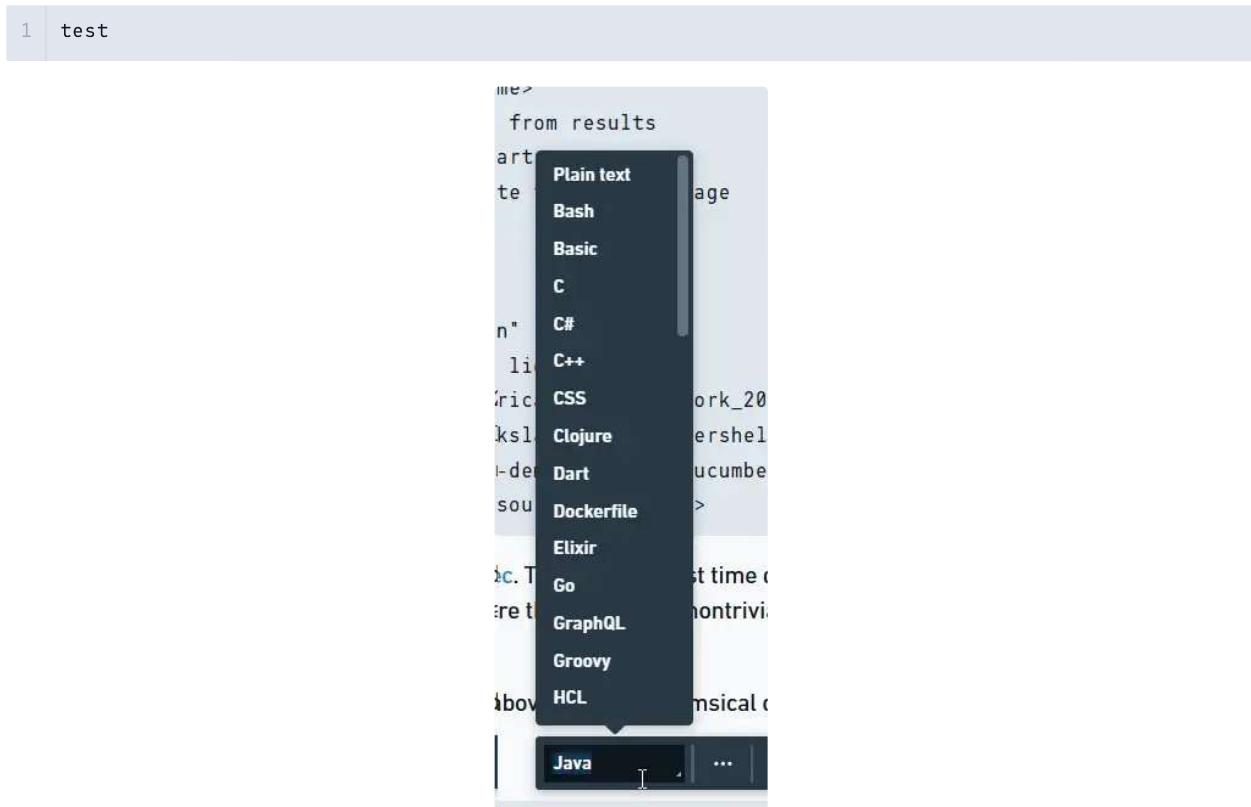
1 2026_002_015_055_030_834203000_America_slash_New_York_2026_W006_001_2026_033_1770065730_8342
03000_192_168_004_098_C-CLFQPLPPJQ_C-CLFQPLPPJQ_backslash_peter_Powershell_7_5_4
2 C:\GitHub-repositories\testng-cucumber-demonstration\Cucumber_TestNG_Demo_Amazon_with_working_sc
reenshots_Test_Suite_suite_2026_001_029\src\test\resources\features> cat .\AmazonCart.feature
3 Feature: Amazon Shopping Cart Suite
4
5 Scenario Outline: Search and add various items to cart
6   Given User is on Amazon homepage "https://www.amazon.com"
7   And User handles the initial interstitial if present
8   When User searches for <product_name>
9   And User selects the first product from results
10  And User adds the product to the cart
11  Then User should be able to navigate to the cart page
12
13 Examples:
14 | product_name           |
15 | "Fakanhui satin pants for women"      |
16 | "EVALESS pleated v neck blouse light pink" |
17 2026_002_015_055_032_513037400_America_slash_New_York_2026_W006_001_2026_033_1770065732_5130
37400_192_168_004_098_C-CLFQPLPPJQ_C-CLFQPLPPJQ_backslash_peter_Powershell_7_5_4
18 C:\GitHub-repositories\testng-cucumber-demonstration\Cucumber_TestNG_Demo_Amazon_with_working_sc
reenshots_Test_Suite_suite_2026_001_029\src\test\resources\features>

```

see <https://gemini.google.com/share/617faa38feec>. That was my first time doing a non-one-test-case test suite. I call a test suite with one test a trivial test suite and a test suite with more than one test a nontrivial test suite. That was my first time doing a non-trivial test suite.

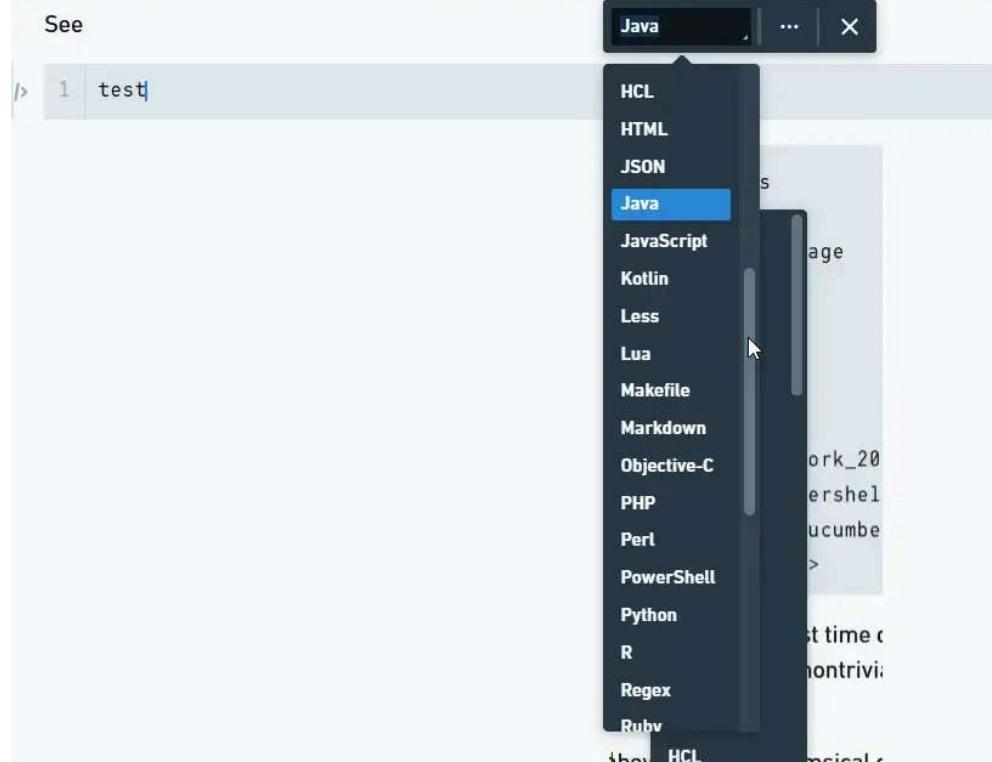
Note: There is not syntax highlighting for Gherkin above because Whimsical does not offer syntax highlight for Gherkin syntax.

See

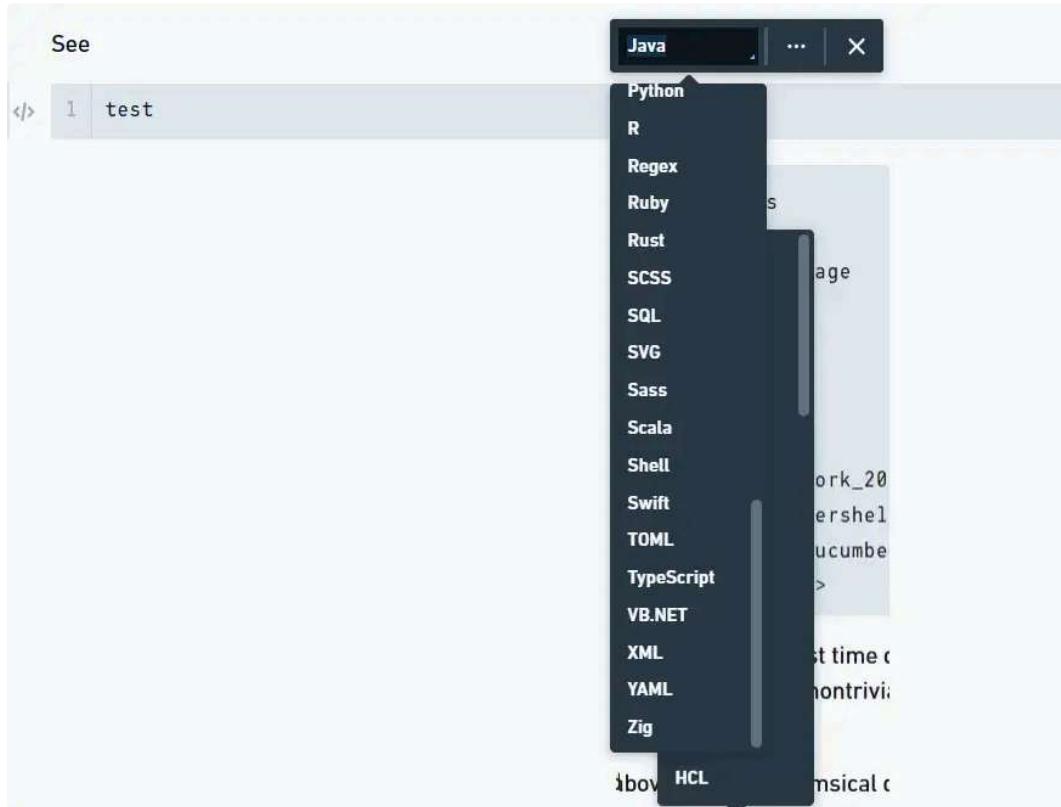


and

Note: There is no syntax highlighting for Gherkin above because Whimsical does not offer syntax highlighting for it.



and



Notice that Gherkin is not included among syntax highlighting. For syntax highlighting, you have to use Eclipse.

```

AmazonCart.feature ×
1@feature: Amazon Shopping Cart Suite
2
3@ Scenario Outline: Search and add various items to cart
4  Given User is on Amazon homepage "https://www.amazon.com"
5  And User handles the initial interstitial if present
6  When User searches for <product_name>
7  And User selects the first product from results
8  And User adds the product to the cart
9  Then User should be able to navigate to the cart page
10
11@ Examples:
12  | product_name |
13  | "Fakanhui satin pants for women" |
14  | "EVALESS pleated v neck blouse light pink" |

```

VS Code may support syntax highlighting for Gherkin files but I haven't gotten around to it. I am not using IntelliJ because I ran into Java problems in IntelliJ. IntelliJ probably has a way to add syntax for Gherkin files. I could file a feature request for Gherkin syntax highlighting in Whimsical, but I don't think they will get around to it any time soon. I did submit a support request for Powershell syntax highlighting and they added that in a matter of weeks, but Gherkin is not as common as Powershell so I do not think they would add that to Whimsical right away.

Anyways, I will get back to what I was saying.

So the way it works is in `AmazonPageDefinitions.java`, which contains

```
1 package com.exampledefinitions;
2
3 import io.cucumber.java.After;
4 import io.cucumber.java.Before;
5 import io.cucumber.java.en.Given;
6 import io.cucumber.java.en.Then;
7 import io.cucumber.java.en.When;
8 import org.openqa.selenium.*;
9 import org.openqa.selenium.chrome.ChromeDriver;
10 import org.openqa.selenium.chrome.ChromeOptions;
11 import org.openqa.selenium.support.ui.ExpectedConditions;
12 import org.openqa.selenium.support.ui.WebDriverWait;
13 import org.testng.Assert;
14 import io.cucumber.java.AfterStep;
15 import io.cucumber.java.Scenario;
16 import org.openqa.selenium.OutputType;
17 import org.openqa.selenium.TakesScreenshot;
18
19 import java.time.Duration;
20
21 public class AmazonPageDefinitions {
22     private static WebDriver driver;
23     private final static int timeout_seconds = 15;
24
25     @Before
26     public void set_up() {
27         ChromeOptions options = new ChromeOptions();
28         options.addArguments("--start-maximized");
29         options.addArguments("--remote-allow-origins=*");
30         driver = new ChromeDriver(options);
31     }
32
33     @Given("User is on Amazon homepage {string}")
34     public void open_amazon_homepage(String url) {
35         driver.get(url);
36     }
37
38     @Given("User handles the initial interstitial if present")
39     public void handle_interstitial() {
40         try {
41             WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(5));
42             WebElement continue_button = wait.until(
43                 ExpectedConditions.elementToBeClickable(By.xpath("//button[contains(., 'Continue shopping')]")));
44             continue_button.click();
45         } catch (TimeoutException ignored) {
46             // Interstitial not found, proceed normally
47         }
48     }
}
```

```
49
50     @When("User searches for {string}")
51     public void search_for_item(String product_name) {
52         WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(timeout_seconds));
53         WebElement search_box = wait.until(ExpectedConditions.visibilityOfElementLocated(By.id
54             ("twotabsearchtextbox")));
55         search_box.clear();
56         search_box.sendKeys(product_name);
57         driver.findElement(By.id("nav-search-submit-button")).click();
58     }
59
60     @When("User selects the first product from results")
61     public void select_first_product() {
62         WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(timeout_seconds));
63
64         // This XPath specifically looks for a search result that is NOT a sponsored
65         // 'sparkle' ad
66         // It targets the first organic product link in the search grid
67         By organic_product_xpath = By.xpath(
68             "//div[@data-component-type='s-search-result']" + // Main result container
69             "[not(.//span[contains(text(),'Sponsored')])]" + // Exclude sponsored la
70             "labels" + // The product link
71             "//a[contains(@class,'a-link-normal') and contains(@href,'/dp/')]"); // Th
72
73         WebElement first_link = wait.until(ExpectedConditions.elementToBeClickable(organic_produ
74
75             // Scroll it into view just to be safe before clicking
76             ((JavascriptExecutor) driver).executeScript("arguments[0].scrollIntoView({block: 'cen
77             r'});", first_link);
78
79             // Use Javascript click to avoid any other potential interceptions
80             ((JavascriptExecutor) driver).executeScript("arguments[0].click()", first_link);
81     }
82
83     @When("User adds the product to the cart")
84     public void add_to_cart() {
85         WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(timeout_seconds));
86         WebElement add_button = wait.until(ExpectedConditions.elementToBeClickable(By.id("add-to
87             -cart-button")));
88         add_button.click();
89     }
90
91     @Then("User should be able to navigate to the cart page")
92     public void verify_cart_navigation() {
93         WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(timeout_seconds));
```

```

92         // Multi-locator strategy for "Go to Cart" button as seen in your original
93         // script
94         By go_to_cart_selector = By.cssSelector("#sw-gtc, #attach-sidesheet-view-cart-button, [h
ref*='/cart']");
95         WebElement go_to_cart_btn = wait.until(ExpectedConditions.elementToBeClickable(go_to_car
t_selector));
96         go_to_cart_btn.click();
97
98         boolean is_at_cart = wait.until(ExpectedConditions.or(
99             ExpectedConditions.titleContains("Shopping Cart"),
100            ExpectedConditions.urlContains("/cart")));
101
102         Assert.assertTrue(is_at_cart, "Failed to navigate to the shopping cart page.");
103     }
104
105     @After
106     public void teardown() {
107         // Keep browser open if needed for debugging, otherwise use driver.quit()
108         // if (driver != null) driver.quit();
109     }
110
111     @AfterStep
112     public void add_screenshot(Scenario scenario) {
113         // validate if driver is initialized before taking screenshot
114         if (driver != null) {
115             final byte[] screenshot = ((TakesScreenshot) driver).getScreenshotAs(OutputType.BYTE
S);
116             // Attach the screenshot to the Cucumber report
117             scenario.attach(screenshot, "image/png", "image_at_step_" + scenario.getLine());
118         }
119     }
120 }
```

Now, what you'll notice is that we have

```

1  @When("User searches for {string}")
2  public void search_for_item(String product_name) {
3      WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(timeout_seconds));
4      WebElement search_box = wait.until(ExpectedConditions.visibilityOfElementLocated(By.id
("twotabsearchtextbox")));
5      search_box.clear();
6      search_box.sendKeys(product_name);
7      driver.findElement(By.id("nav-search-submit-button")).click();
8  }
```

So what happens is when it gets to

```
1 When User searches for <product_name>
```

in AmazonCart.feature, it replaces product name based on what is in the table.

```

1 Examples:
2   | product_name           |
3   | "Fakanhui satin pants for women"    |
4   | "EVALESS pleated v neck blouse light pink" |

```

The table has to be named product\_name or it will not work. Anyways, it searches for that product and goes through the results.

```

1 2026_002_002_015_055_030_834203000_America_slash_New_York_2026_W006_001_2026_033_1770065730_8342
2 03000_192_168_004_098_C-CLFQPLPPJQ_C-CLFQPLPPJQ_backslash_peter_Powershell_7_5_4
C:\GitHub-repositories\testng-cucumber-demonstration\Cucumber_TestNG_Demo_Amazon_with_working_sc
3 reenshots_Test_Suite_suite_2026_001_029\src\test\resources\features> cat .\AmazonCart.feature
Feature: Amazon Shopping Cart Suite
4
5 Scenario Outline: Search and add various items to cart
6   Given User is on Amazon homepage "https://www.amazon.com"
7   And User handles the initial interstitial if present
8   When User searches for <product_name>
9   And User selects the first product from results
10  And User adds the product to the cart
11  Then User should be able to navigate to the cart page
12
13 Examples:
14   | product_name           |
15   | "Fakanhui satin pants for women"    |
16   | "EVALESS pleated v neck blouse light pink" |
17 2026_002_002_015_055_032_513037400_America_slash_New_York_2026_W006_001_2026_033_1770065732_5130
37400_192_168_004_098_C-CLFQPLPPJQ_C-CLFQPLPPJQ_backslash_peter_Powershell_7_5_4
18 C:\GitHub-repositories\testng-cucumber-demonstration\Cucumber_TestNG_Demo_Amazon_with_working_sc
reenshots_Test_Suite_suite_2026_001_029\src\test\resources\features>

```

Anyways, so there's that. That's a test suite. That's a suite works. That's that.

Okay, now the next step is to work on the Amazon document.

I have started working on that, but I don't have anything for now. I am making progress.

Sincerely,

Peter Burberry

P.S. Remember I have sick leave tomorrow 2026-002-003 so I am taking leave tomorrow.