

UNDERSTANDING



**SOFTWARE DEVELOPMENT
LIFE CYCLE**

TESTING AUTOMATION TOOLS

By :- Parth Tyagi

July 4, 2025

Playwright Test Case



File Edit Selection View Go Run Terminal Help

my-playwright-tests

EXPLORER

tests > JS example.spec.js > ...

```
1 // @ts-check
2 import { test, expect } from '@playwright/test';
3
4 test('has title', async ({ page }) => {
5   await page.goto('https://playwright.dev/');
6
7   // Expect a title "to contain" a substring.
8   await expect(page).toHaveTitle(/Playwright/);
9 });
10
11 test('get started link', async ({ page }) => {
12   await page.goto('https://playwright.dev/');
13
14   // Click the get started link.
15   await page.getByRole('link', { name: 'Get started' }).click();
16
17   // Expects page to have a heading with the name of Installation.
18   await expect(page.getByRole('heading', { name: 'Installation' })).toBeVisible();
19 });
20
```

Browser	has title	get started link
Chromium	✗ Failed	✓ Passed
Firefox	✓ Passed	✓ Passed
WebKit	✓ Passed	✓ Passed



Test Report Breakdown

example.spec.js, contains 2 tests:

- has title
- get started link

Playwright executed these 2 tests across 3 different browsers:

- Chromium (Google Chrome-like)
- Firefox
- WebKit (Safari-like)



```
Windows PowerShell
npx playwright codegen
Auto generate tests with Codegen.

We suggest that you begin by typing:

npx playwright test

And check out the following files:
- .\tests\example.spec.js - Example end-to-end test
- .\tests-examples\demo-todo-app.spec.js - Demo Todo App end-to-end tests
- .\playwright.config.js - Playwright Test configuration

Visit https://playwright.dev/docs/intro for more information. ✨

Happy hacking! 🐼
PS C:\Users\hp\Desktop\my-playwright-tests> npx playwright test

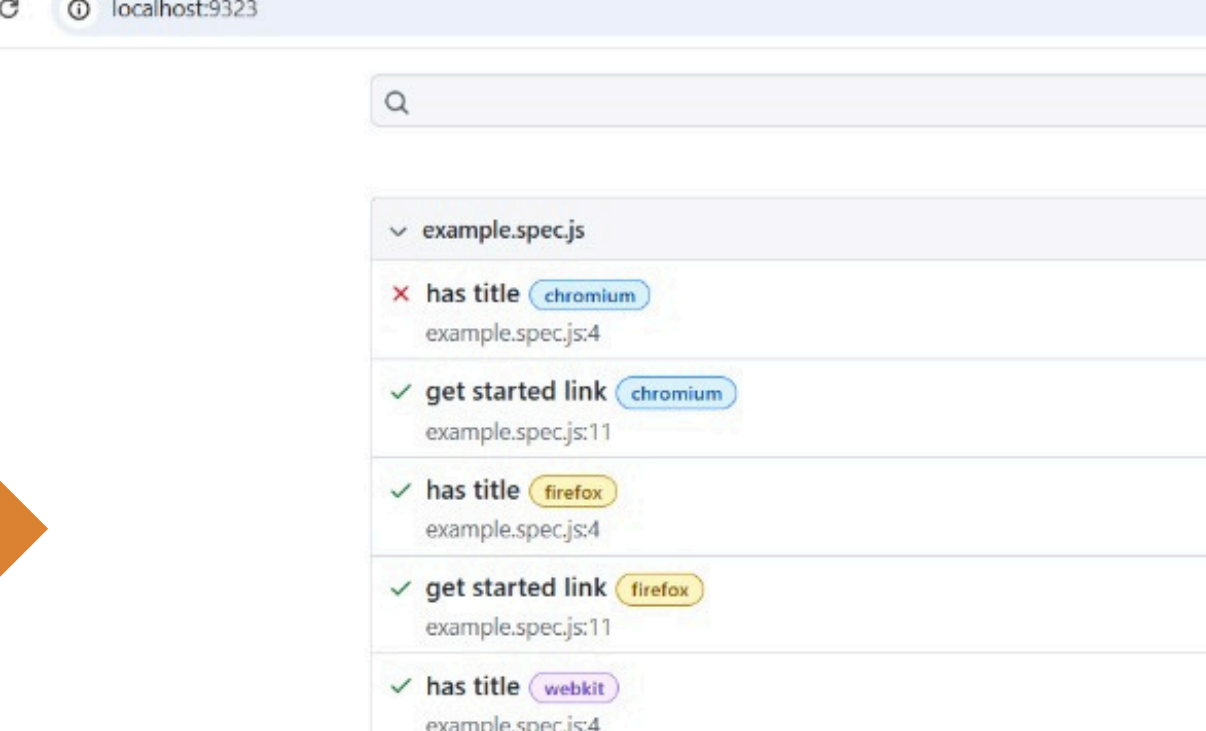
Running 6 tests using 6 workers
1) [chromium] › tests\example.spec.js:4:5 › has title -----

Error: page.goto: net::ERR_CONNECTION_RESET at https://playwright.dev/
Call log:
  - navigating to "https://playwright.dev/", waiting until "load"

3 |
4 | test('has title', async ({ page }) => {
> 5 |   await page.goto('https://playwright.dev/');
    |               ^
6 |
7 |   // Expect a title "to contain" a substring.
8 |   await expect(page).toHaveTitle(/Playwright/);
   at C:\Users\hp\Desktop\my-playwright-tests\tests\example.spec.js:5:14

1 failed
[chromium] › tests\example.spec.js:4:5 › has title -----
5 passed (15.8s)

Serving HTML report at http://localhost:9323. Press Ctrl+C to quit.
```



Playwright Test Report

localhost:9323

Search

All 6

example.spec.js

- ✗ has title chromium
example.spec.js:4
- ✓ get started link chromium
example.spec.js:11
- ✓ has title firefox
example.spec.js:4
- ✓ get started link firefox
example.spec.js:11
- ✓ has title webkit
example.spec.js:4
- ✓ get started link webkit
example.spec.js:11

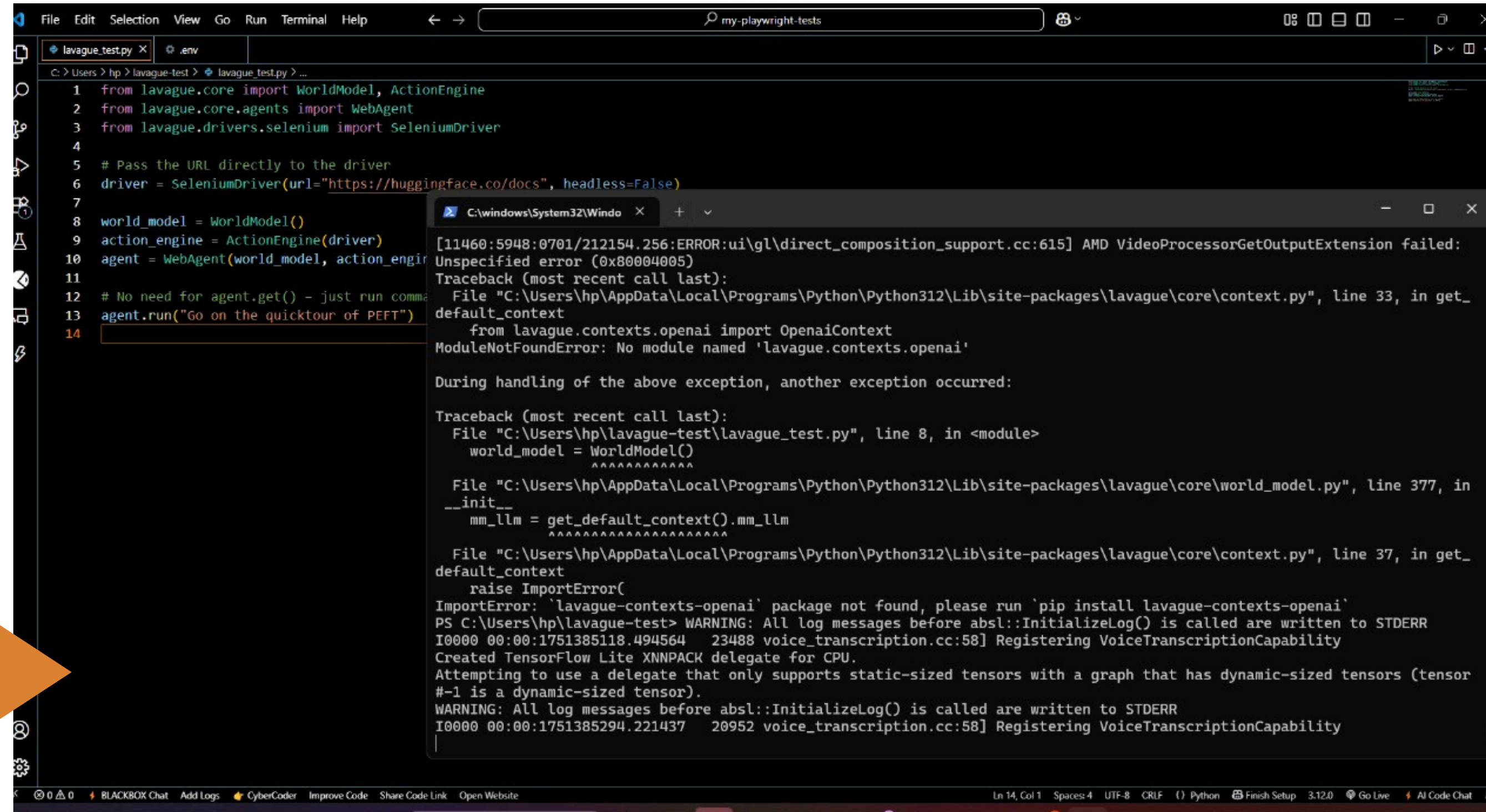
AI Agent Controlling Chrome

What Happens:

- Chrome is launched via Selenium.
- Ollama's LLaMA3 interprets your command.
- Agent interacts with the web page.

Success: HuggingFace Docs opened via Agent.

Input

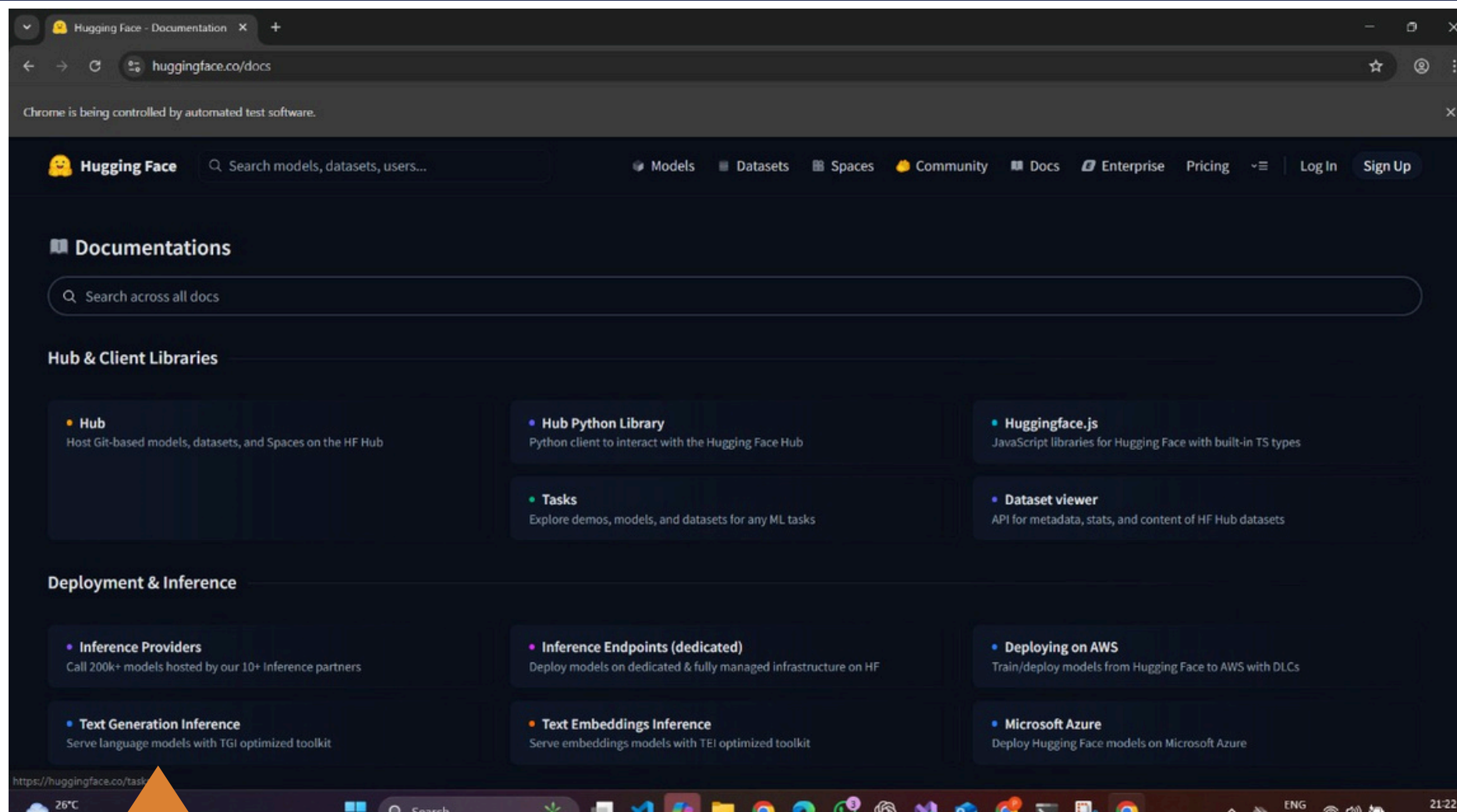


```
File Edit Selection View Go Run Terminal Help
lavague_test.py .env
C:\Users\hp> cd lavague-test & cd lavague_test.py > ...
1 from lavague.core import WorldModel, ActionEngine
2 from lavague.core.agents import WebAgent
3 from lavague.drivers.selenium import SeleniumDriver
4
5 # Pass the URL directly to the driver
6 driver = SeleniumDriver(url="https://huggingface.co/docs", headless=False)
7
8 world_model = WorldModel()
9 action_engine = ActionEngine(driver)
10 agent = WebAgent(world_model, action_engine)
11
12 # No need for agent.get() - just run command
13 agent.run("Go on the quicktour of PEFT")
14

C:\windows\System32\WindowsPowerShell\v1.0\
[11460:5948:0701/212154.256:ERROR:ui\gl\direct_composition_support.cc:615] AMD VideoProcessorGetOutputExtension failed:
Unspecified error (0x80004005)
Traceback (most recent call last):
  File "C:\Users\hp\AppData\Local\Programs\Python\Python312\Lib\site-packages\lavague\core\context.py", line 33, in get_
default_context
    from lavague.contexts.openai import OpenaiContext
ModuleNotFoundError: No module named 'lavague.contexts.openai'


During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "C:\Users\hp\lavague-test\lavague_test.py", line 8, in <module>
    world_model = WorldModel()
                  ^^^^^^^^^^^
  File "C:\Users\hp\AppData\Local\Programs\Python\Python312\Lib\site-packages\lavague\core\world_model.py", line 377, in
__init__
    mm_llm = get_default_context().mm_llm
             ^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\hp\AppData\Local\Programs\Python\Python312\Lib\site-packages\lavague\core\context.py", line 37, in get_
default_context
    raise ImportError(
ImportError: 'lavague-contexts-openai' package not found, please run 'pip install lavague-contexts-openai'
PS C:\Users\hp\lavague-test> WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
I0000 00:00:1751385118.494564    23488 voice_transcription.cc:58] Registering VoiceTranscriptionCapability
Created TensorFlow Lite XNNPACK delegate for CPU.
Attempting to use a delegate that only supports static-sized tensors with a graph that has dynamic-sized tensors (tensor
#-1 is a dynamic-sized tensor).
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
I0000 00:00:1751385294.221437    20952 voice_transcription.cc:58] Registering VoiceTranscriptionCapability
```

Output

 **Tool Used: Selenium WebDriver**

 **Problem Statement:** Automate browser actions using AI prompt and natural language.

 **Test Flow:**

- Open browser
- Navigate to HuggingFace Docs
- Locate 'PEFT' Quick Tour link
- Interact with the page

Powered By: LaVague + Selenium (no API key)

Context: **Ollama LLM context used locally**

Selenium WebDriver with Node.js

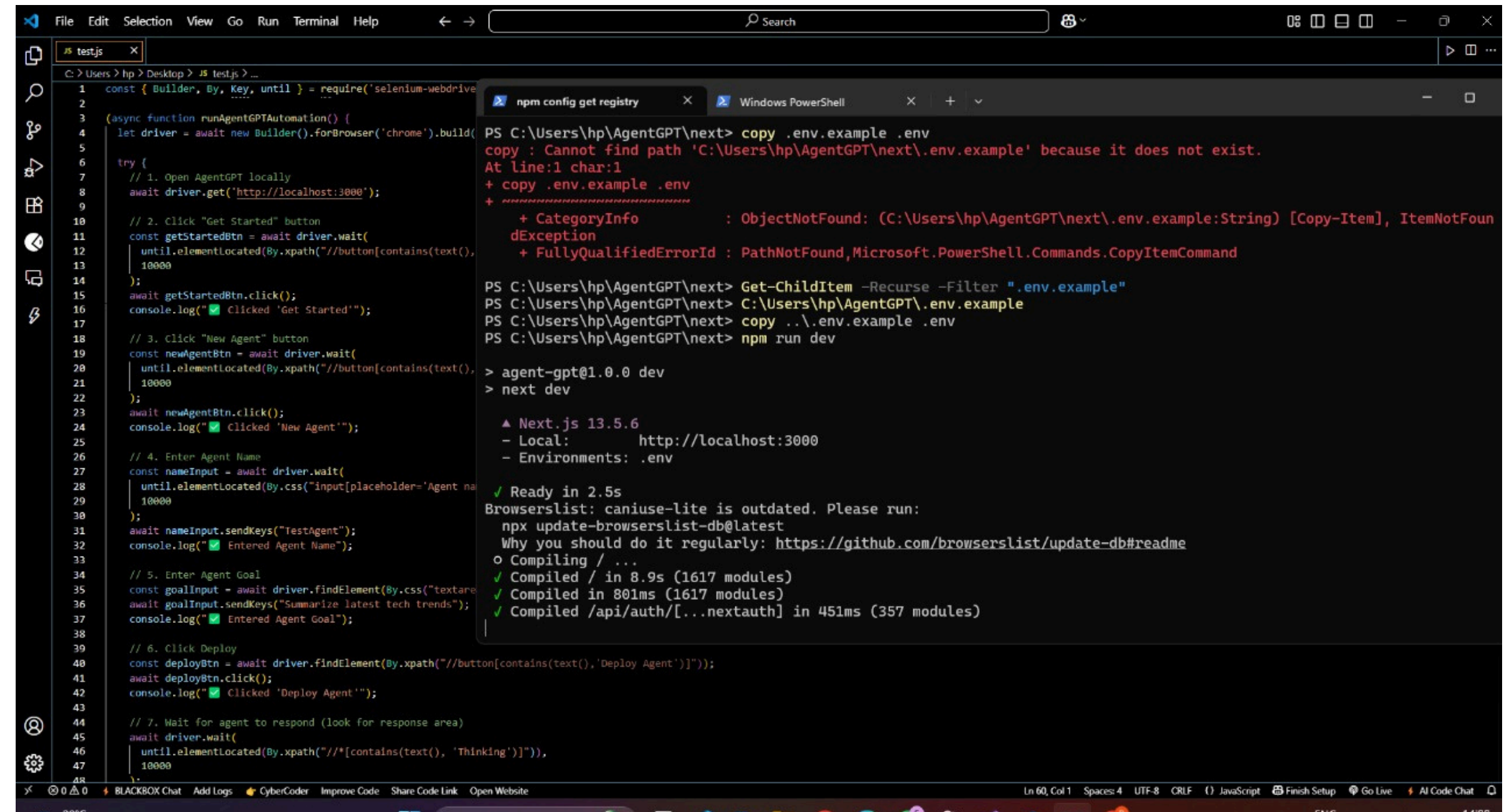
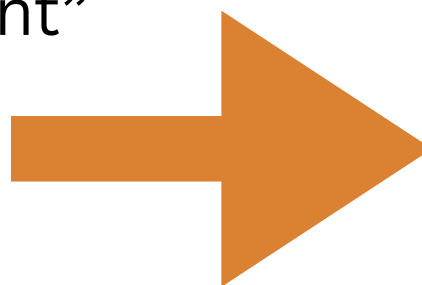
Goal:

Automate the UI flow of creating a new Agent in AgentGPT

Steps Automated:

1. Launch AgentGPT on localhost
2. Click “Get Started”
3. Click “New Agent”
4. Enter Agent Name
5. Enter Agent Goal
6. Click “Deploy Agent”

Input



```
File Edit Selection View Go Run Terminal Help
testjs
C:\Users\hp\Desktop> node testjs
1 const { Builder, By, Key, until } = require('selenium-webdriver');
2
3 (async function runAgentGPTAutomation() {
4   let driver = await new Builder().forBrowser('chrome').build();
5
6   try {
7     // 1. Open AgentGPT locally
8     await driver.get('http://localhost:3000');
9
10    // 2. Click "Get Started" button
11    const getStartedBtn = await driver.wait(
12      until.elementLocated(By.xpath("//button[contains(text(), 'Get Started')]")),
13      10000
14    );
15    await getStartedBtn.click();
16    console.log("Clicked 'Get Started'");
17
18    // 3. Click "New Agent" button
19    const newAgentBtn = await driver.wait(
20      until.elementLocated(By.xpath("//button[contains(text(), 'New Agent')]")),
21      10000
22    );
23    await newAgentBtn.click();
24    console.log("Clicked 'New Agent'");
25
26    // 4. Enter Agent Name
27    const nameInput = await driver.wait(
28      until.elementLocated(By.css("input[placeholder='Agent Name']")),
29      10000
30    );
31    await nameInput.sendKeys("TestAgent");
32    console.log("Entered Agent Name");
33
34    // 5. Enter Agent Goal
35    const goalInput = await driver.findElement(By.css("textarea"));
36    await goalInput.sendKeys("Summarize latest tech trends");
37    console.log("Entered Agent Goal");
38
39    // 6. Click Deploy
40    const deployBtn = await driver.findElement(By.xpath("//button[contains(text(), 'Deploy Agent')]"));
41    await deployBtn.click();
42    console.log("Clicked 'Deploy Agent'");
43
44    // 7. Wait for agent to respond (look for response area)
45    await driver.wait(
46      until.elementLocated(By.xpath("//div[contains(text(), 'Thinking')]")),
47      10000
48    );
49  } catch (error) {
50    console.error(error);
51  }
52 }());
```

```
npm config get registry
Windows PowerShell
PS C:\Users\hp\AgentGPT\next> copy .env.example .env
copy : Cannot find path 'C:\Users\hp\AgentGPT\next\.env.example' because it does not exist.
At line:1 char:1
+ copy .env.example .env
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (C:\Users\hp\AgentGPT\next\.env.example:String) [Copy-Item], ItemNotFoundEx
+ FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.CopyItemCommand

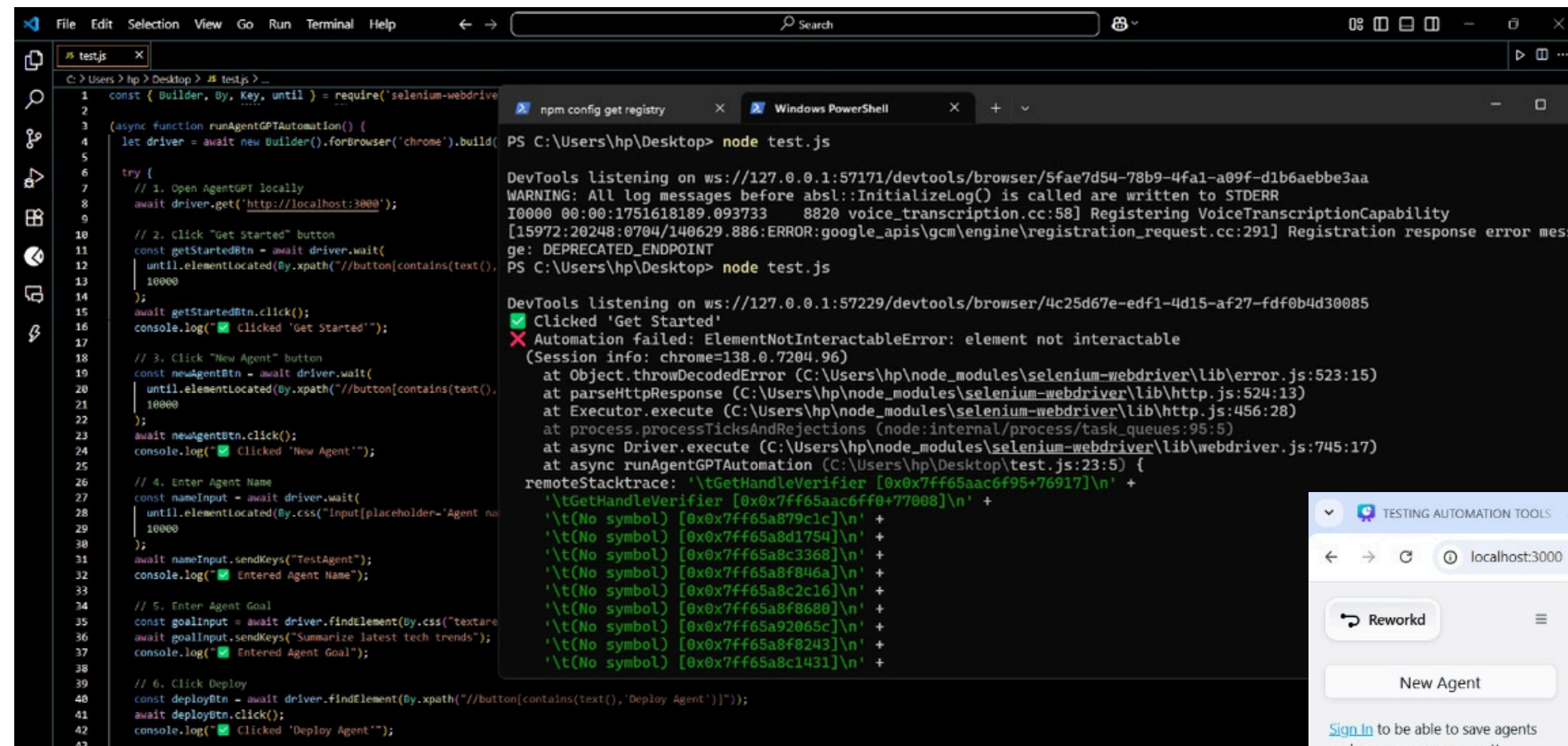
PS C:\Users\hp\AgentGPT\next> Get-ChildItem -Recurse -Filter ".env.example"
PS C:\Users\hp\AgentGPT\next> C:\Users\hp\AgentGPT\next> copy ..\.env.example .env
PS C:\Users\hp\AgentGPT\next> copy ..\.env.example .env
PS C:\Users\hp\AgentGPT\next> npm run dev

> agent-gpt@1.0.0 dev
> next dev

  ▲ Next.js 13.5.6
  - Local:      http://localhost:3000
  - Environments: .env

  ✓ Ready in 2.5s
  Browserslist: caniuse-lite is outdated. Please run:
    npx update-browserslist-db@latest
  Why you should do it regularly: https://github.com/browserslist/update-db#readme
  o Compiling / ...
  ✓ Compiled / in 8.9s (1617 modules)
  ✓ Compiled in 801ms (1617 modules)
  ✓ Compiled /api/auth/[...nextauth] in 451ms (357 modules)
```


AgentGPT Using Selenium



```

1 const { Builder, By, Key, until } = require('selenium-webdriver');
2
3 (async function runAgentGPTAutomation() {
4   let driver = await new Builder().forBrowser('chrome').build();
5
6   try {
7     // 1. Open AgentGPT locally
8     await driver.get('http://localhost:3000');
9
10    // 2. Click "Get Started" button
11    const getStartedBtn = await driver.wait(
12      until.elementLocated(By.xpath("//button[contains(text(), 'Get Started')]")),
13      10000
14    );
15    await getStartedBtn.click();
16    console.log("✅ Clicked 'Get Started'");
17
18    // 3. Click "New Agent" button
19    const newAgentBtn = await driver.wait(
20      until.elementLocated(By.xpath("//button[contains(text(), 'New Agent')]")),
21      10000
22    );
23    await newAgentBtn.click();
24    console.log("✅ Clicked 'New Agent'");
25
26    // 4. Enter Agent Name
27    const nameInput = await driver.wait(
28      until.elementLocated(By.css("input[placeholder='Agent name']")),
29      10000
30    );
31    await nameInput.sendKeys("TestAgent");
32    console.log("✅ Entered Agent Name");
33
34    // 5. Enter Agent Goal
35    const goalInput = await driver.findElement(By.css("textare"));
36    await goalInput.sendKeys("Summarize latest tech trends");
37    console.log("✅ Entered Agent Goal");
38
39    // 6. Click Deploy
40    const deployBtn = await driver.findElement(By.xpath("//button[contains(text(), 'Deploy Agent')]"));
41    await deployBtn.click();
42    console.log("✅ Clicked 'Deploy Agent'");
43  }
44 }
45
46 runAgentGPTAutomation().catch(console.error);

```

```

PS C:\Users\hp\Desktop> node test.js
DevTools listening on ws://127.0.0.1:57171/devtools/browser/5fae7d54-78b9-4fa1-a09f-d1b6aebbe3aa
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
I0000 00:00:1751618189.093733    8820 voice_transcription.cc:58] Registering VoiceTranscriptionCapability
[15972:20248:0704/140629.886:ERROR:google_apis\gcm\engine\registration_request.cc:291] Registration response error message: DEPRECATED_ENDPOINT
PS C:\Users\hp\Desktop> node test.js
DevTools listening on ws://127.0.0.1:57229/devtools/browser/4c25d67e-edf1-4d15-af27-fdf0b4d30085
✅ Clicked 'Get Started'
❌ Automation failed: ElementNotInteractableError: element not interactable
(Session info: chrome=138.0.7204.96)
at Object.throwDecodedError (C:\Users\hp\node_modules\selenium-webdriver\lib\error.js:523:15)
at parseHttpResponse (C:\Users\hp\node_modules\selenium-webdriver\lib\http.js:524:13)
at Executor.execute (C:\Users\hp\node_modules\selenium-webdriver\lib\http.js:456:28)
at process.processTicksAndRejections (node:internal/process/task_queues:95:5)
at async Driver.execute (C:\Users\hp\node_modules\selenium-webdriver\lib\webdriver.js:745:17)
at async runAgentGPTAutomation (C:\Users\hp\Desktop\test.js:23:5) {
  remoteStackTrace: '\tGetHandleVerifier [0x0x7ff65aac6ff0+77008]\n' +
    '\t(No symbol) [0x0x7ff65a879c1c]\n' +
    '\t(No symbol) [0x0x7ff65a8d1754]\n' +
    '\t(No symbol) [0x0x7ff65a8c3368]\n' +
    '\t(No symbol) [0x0x7ff65a8f846a]\n' +
    '\t(No symbol) [0x0x7ff65a8c2c16]\n' +
    '\t(No symbol) [0x0x7ff65a8f8680]\n' +
    '\t(No symbol) [0x0x7ff65a92065c]\n' +
    '\t(No symbol) [0x0x7ff65a8f8243]\n' +
    '\t(No symbol) [0x0x7ff65a8c1431]\n' +

```

✅ **Success up to Step 2:**

- "Clicked 'Get Started'" is printed — so Selenium successfully located and interacted with that button.

❌ **Failure at Step 6:**

- **ElementNotInteractableError** occurred while trying to click on the Deploy Agent button.

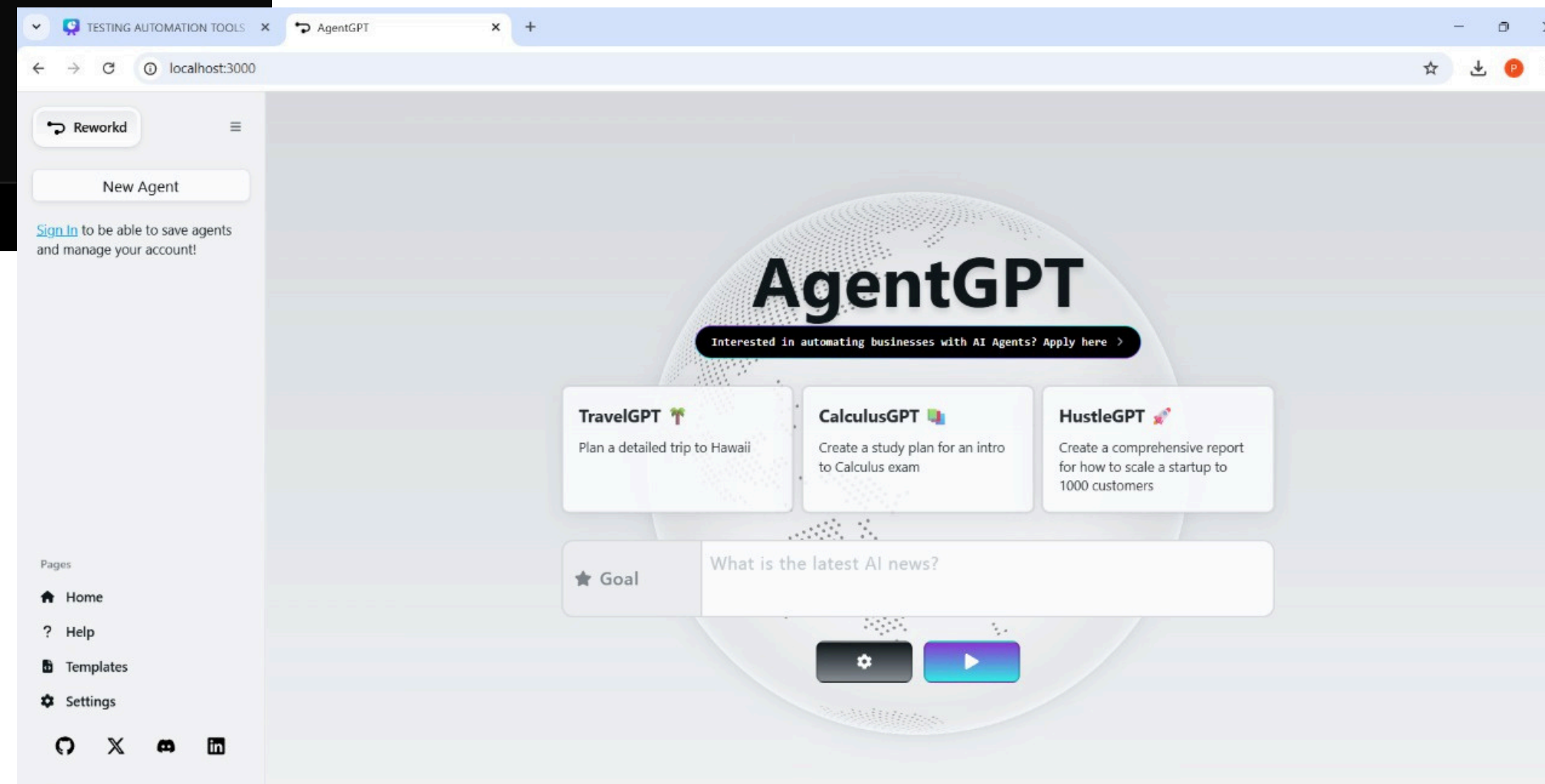
Automation Script

```

await driver.get("http://localhost:3000");

const getStartedBtn = await driver.wait(
  until.elementLocated(By.xpath("//button[contains(text(), 'Get Started')]")),
  10000
);
await getStartedBtn.click();

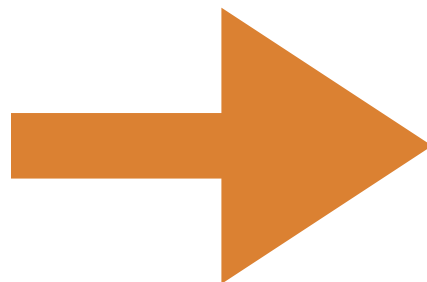
```



Sample

- fixture: Defines the test suite
- .page: Specifies URL
- Selector: Used to select HTML elements
- expect: Performs assertions

Input



```
import { Selector } from 'testcafe';

fixture `My First Test`
  .page `https://example.com`;

test('Check Example Domain text', async t => {
  await t.expect(Selector('h1').innerText).eq('Example Domain');
});
```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

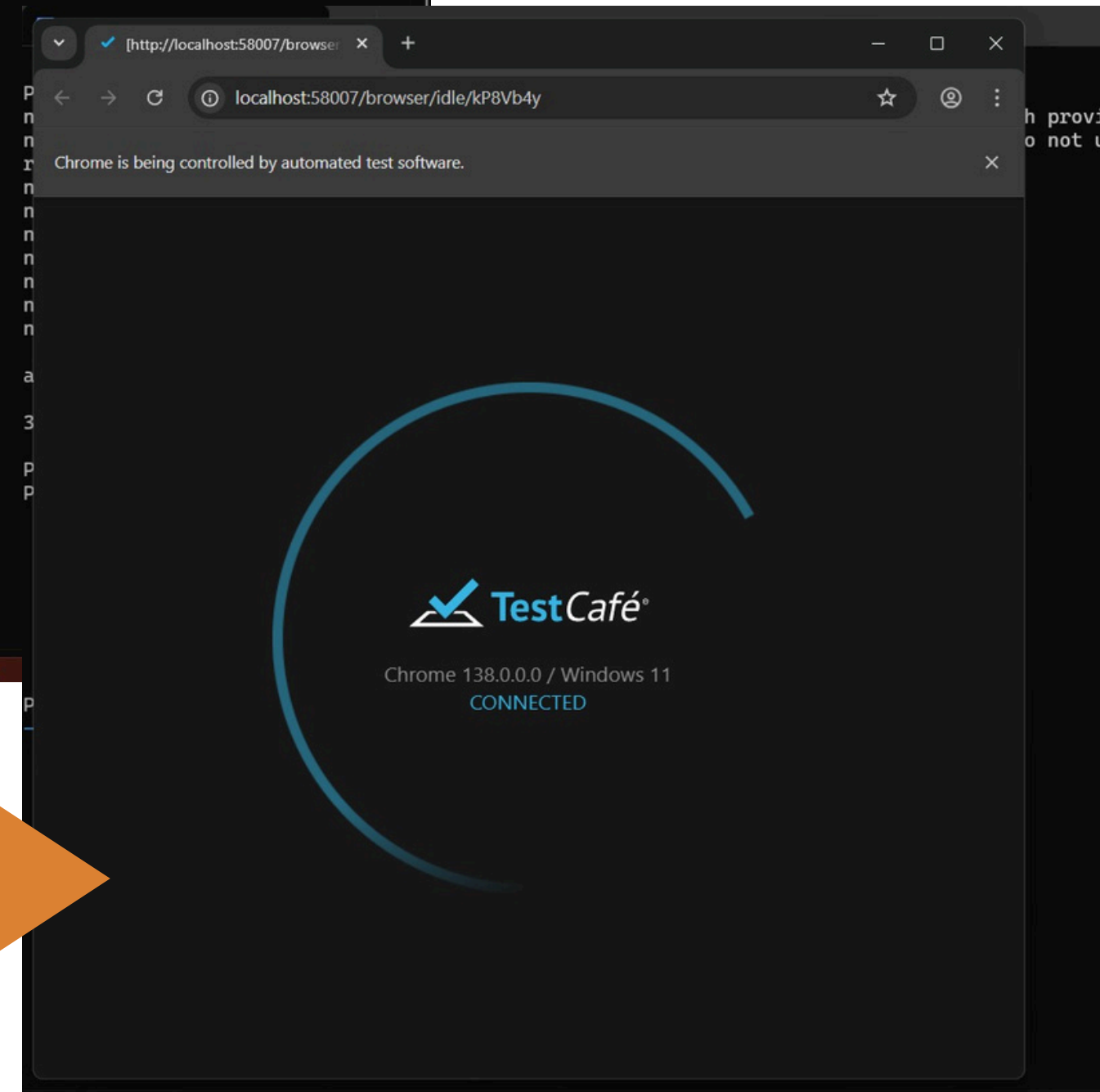
PS C:\Users\hp> npm install -g testcafe
npm WARN deprecated @types/minimatch@6.0.0: This is a stub types definition. minimatch provides its own type definitions, so you do not need this installed.
npm WARN deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerful.
npm WARN deprecated rimraf@3.0.2: Rimraf versions prior to v4 are no longer supported
npm WARN deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm WARN deprecated glob@8.1.0: Glob versions prior to v9 are no longer supported
npm WARN deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm WARN deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm WARN deprecated rimraf@2.7.1: Rimraf versions prior to v4 are no longer supported

added 451 packages in 21s

39 packages are looking for funding
  run 'npm fund' for details
PS C:\Users\hp> notepad test.js
PS C:\Users\hp> testcafe chrome test.js
Running tests in:
- Chrome 138.0.0.0 / Windows 11

My First Test
✓ Check Example Domain text

1 passed (2s)
PS C:\Users\hp> |
```



Output



What Happens:

- Launches Chrome browser automatically
- Displays message: “Chrome is being controlled by automated test software”

• Shows: TestCafe
Connected

Console displays:

1 passed (2s)

**Thank
you**