

Assignment Submission

Course: GenAI Core Essentials for QA Engineers [AI-Powered Testing Mastery]

Topic: Building Apps using LLM - 1 [Live Session Date: 21st Feb 2026]

Name: Sachin Patil

Role: Software Test Engineer

Home Assignment

Q. 1: Use 'Bug-Magnet' and 'Fake filler' extension for form filling

Q. 2: Use 'Grammarly' extension to correct the mistakes in your content

Q. 3: Vibe Coding Generate Playwright Typescript Or (any tool with language binding in which you are working)

What Is This Assignment Really Testing?

As a QA Engineer learning GenAI, this Assignment verifies if we can:

SN	Skills	What is being Evaluated
1	AI-assisted testing	Using Bug-Magnet & Fake Filler
2	AI-assisted writing	Using Grammarly
3	AI-powered coding	Generating Playwright using LLM
4	Practical implementation	Not just theory

Q. 1: Use 'Bug-Magnet' and 'Fake filler' extension for form filling

Execution Steps:

1. Installing browser extensions:
 - o **Fake Filler**
 - o **Bug Magnet**
2. Using them to:
 - o Auto-fill forms
 - o Generate boundary values
 - o Test edge cases in web forms

 **Purpose:** To simulate real-world test data injection and boundary testing.

Auto-filling the form by **Fake Filler**

The screenshot shows the Salesforce login page at login.salesforce.com. A context menu is open over the 'Username' input field, with the 'Fake Filler' option highlighted. The menu also includes options like 'Fill all inputs', 'Fill this form', and 'Fill this input'. To the right of the login form, there is a promotional banner for free Salesforce usage.

This screenshot is similar to the one above, but the context menu is now open over the 'Password' input field. The 'Fake Filler' option is still visible in the list. The rest of the page, including the promotional banner, remains the same.

This screenshot shows the final result of using the 'Fake Filler' extension. The 'Username' field now contains the value 'fygicedac@mailinator.com'. The context menu is no longer visible, and the rest of the page, including the promotional banner, remains consistent with the previous screenshots.

Auto-filling the form by Bug Magnet

The screenshot shows the Salesforce login page. A context menu is open over the password input field. The menu includes standard options like Cut, Copy, Paste, and Undo, along with a "Bug Magnet" option under a "Fake Filler" heading. The "Bug Magnet" option is highlighted with a red arrow. To the right of the menu, a promotional banner for Salesforce free trials is visible.

The screenshot shows the same Salesforce login page after the password field has been filled by the "Bug Magnet" feature. The password field now contains "TabsAndNewlines". The rest of the page, including the promotional banner, remains unchanged.

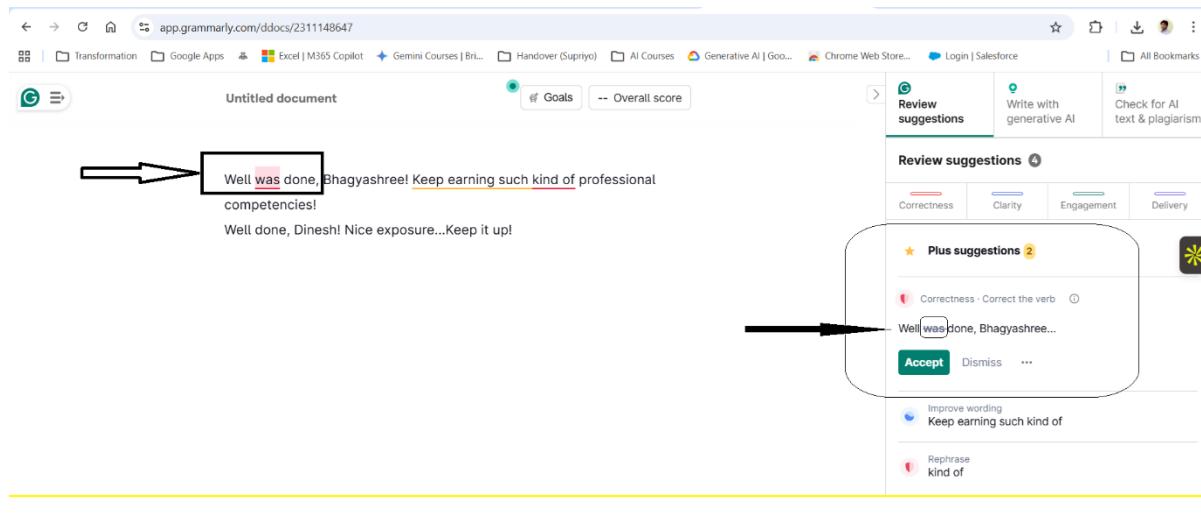
The screenshot shows the Salesforce login page again. This time, the password field is empty, and an error message is displayed above the password input field: "Error: Please check your username and password. If you still can't log in, contact your Salesforce administrator." An arrow points from the left towards this error message. The rest of the page, including the promotional banner, is identical to the previous screenshots.

Q. 2: Use 'Grammarly' extension to correct the mistakes in your content

Execution Steps:

1. Write some documentation/content (probably about our app or testing approach)
2. Use Grammarly to:
 - o Correct grammar
 - o Improve sentence clarity
 - o Fix spelling mistakes

👉 Purpose: To demonstrate professional documentation skills using AI assistance.



Q. 3: Vibe Coding Generate Playwright Typescript Or (any tool with language binding in which you are working)

Execution Steps:

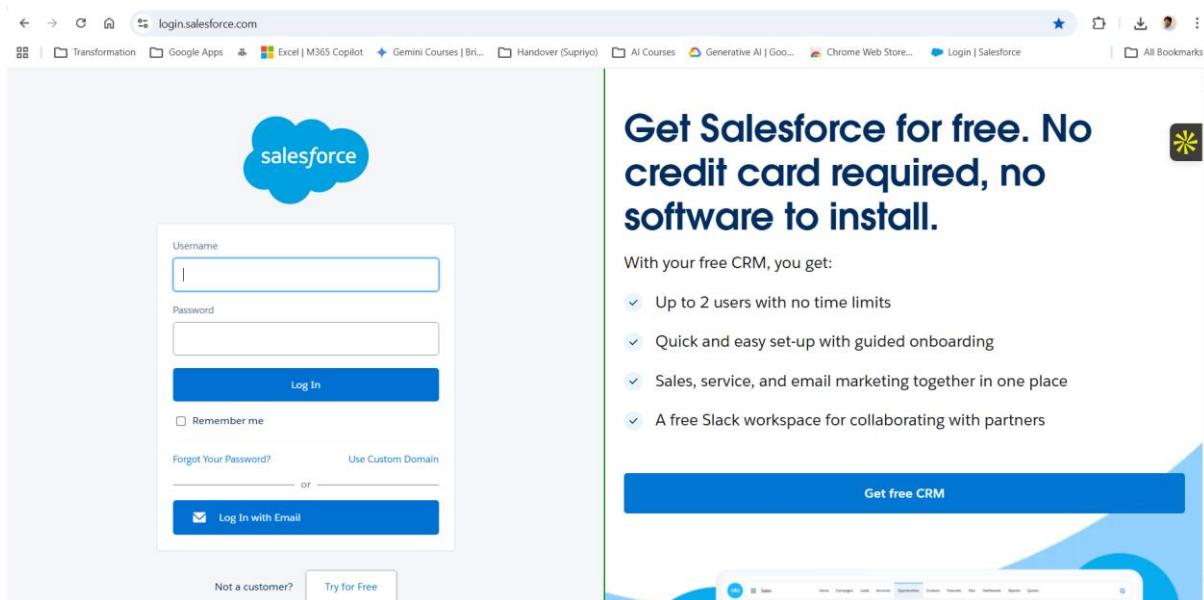
1. Use an LLM (ChatGPT / Claude / etc.)
2. Generate **Playwright automation code (TypeScript)**
OR
3. Use any tool with your preferred language binding

👉 Purpose:

To show that you can:

- Build automation using AI assistance
- Understand and refine generated code
- Apply LLM practically in QA

<https://login.salesforce.com/>



By using a Browser Extension '**Automated Gest Script Generator**'

This screenshot shows the Chrome DevTools interface with the 'Elements' tab selected. On the left, the 'Recorded Steps' panel lists four actions: 'Goto: https://login.salesforce.com/' (Delete), 'Click #username' (Delete), 'Click #password' (Delete), and 'Click #Login' (Delete). Below this is a dropdown set to 'Playwright'. A 'Generate Test' button is visible. The main content area shows the generated TypeScript code for an automated test. The code starts with `const { test, expect } = require('@playwright/test');` and then defines a test function that navigates to the Salesforce login page, inputs credentials, and clicks the login button. The generated code uses the Playwright API to perform these actions. The right side of the DevTools shows the browser's styles and computed styles for the current page elements.

```
const { test, expect } = require('@playwright/test');

test('Automated test', async ({ page }) => {
  await page.goto('https://login.salesforce.com/');
  await page.click('#username');
  await page.click('#password');
  await page.click('#Login');
});
```

By using a Browser Extension 'pH - Playwright Helper'

The screenshot shows the Microsoft Edge Add-ons page for the 'pH - Playwright Helper' extension. The extension is listed with a rating of 0 stars from 1408 users. It is categorized under 'Developer tools'. A 'Remove' button is visible, indicating it is already installed. The extension's version is 1.0, updated on September 22, 2022, and is available in one language. The developer, Monirul Sk, has created more add-ons. A 'Report abuse' button is also present.

Description

Find Playwright Test locators such as - id, name and custom attributes highlighted in different colors;Generates Playwright Page Objects

User reviews

Sort by Filter by region

Most recent All regions Add a review Sign in to add a review

No one has reviewed this add-on yet. Be the first to add a review.

The screenshot shows a Salesforce login page with the pH - Playwright Helper extension active. The extension highlights specific DOM elements in different colors (e.g., red, green, blue) to identify them as Playwright locators. The highlighted elements include various IDs, classes, and attributes used in the login form's structure. The browser's developer tools are open, showing the highlighted elements in the Elements tab.