Assignment 2:

1. Generate Page class for salesforce login page & confirm all looks fine

package com.salesforce.pages;

/\*\*

\* Page Object for Salesforce Login Page

\*/

public class SalesforceLoginPage {

private org.openqa.selenium.WebDriver driver;

private org.openqa.selenium.WebElement usernameInput;

private org.openqa.selenium.WebElement passwordInput;

private org.openqa.selenium.WebElement loginButton;

private org.openqa.selenium.WebElement rememberMeCheckbox;

private org.openqa.selenium.WebElement forgotPasswordLink;

private org.openqa.selenium.WebElement customDomainLink;

private org.openqa.selenium.WebElement customDomainInput;

private org.openqa.selenium.WebElement customDomainContinueButton;

private org.openqa.selenium.WebElement chooserError;

private org.openqa.selenium.WebElement editList;

private org.openqa.selenium.WebElement saveEditButton;

private org.openqa.selenium.WebElement cancelEditButton;

public SalesforceLoginPage(org.openqa.selenium.WebDriver driver) {

this.driver = driver;

org.openqa.selenium.support.PageFactory.initElements(driver, this);

}

/\*\*

\* Enters the username in the username input field.

\* @param username the username to be entered

\*/

public void enterUsername(String username) {

usernameInput.sendKeys(username);

}

/\*\*

\* Enters the password in the password input field.

\* @param password the password to be entered

\*/

public void enterPassword(String password) {

passwordInput.sendKeys(password);

}

/\*\*

\* Clicks the login button.

\*/

public void clickLogin() {

loginButton.click();

}

/\*\*

\* Checks if the remember me checkbox is selected.

\* @return true if the checkbox is selected, false otherwise

\*/

public boolean isRememberMeSelected() {

return rememberMeCheckbox.isSelected();

}

/\*\*

\* Clicks the forgot password link.

\*/

public void clickForgotPassword() {

forgotPasswordLink.click();

}

/\*\*

\* Clicks the custom domain link.

\*/

public void clickCustomDomain() {

customDomainLink.click();

}

/\*\*

\* Enters the custom domain in the custom domain input field.

\* @param domain the custom domain to be entered

\*/

public void enterCustomDomain(String domain) {

customDomainInput.sendKeys(domain);

}

/\*\*

\* Clicks the custom domain continue button.

\*/

public void clickCustomDomainContinue() {

customDomainContinueButton.click();

}

/\*\*

\* Gets the chooser error message.

\* @return the chooser error message

\*/

public String getChooserError() {

return chooserError.getText();

}

/\*\*

\* Gets the edit list elements.

\* @return the edit list elements

\*/

public java.util.List<org.openqa.selenium.WebElement> getEditList() {

return editList.findElements(org.openqa.selenium.By.tagName("li"));

}

/\*\*

\* Clicks the save edit button.

\*/

public void clickSaveEdit() {

saveEditButton.click();

}

/\*\*

\* Clicks the cancel edit button.

\*/

public void clickCancelEdit() {

cancelEditButton.click();

}

}

1. **Configure code for playwright / typescript instead of the selenium / java**

**a) No change in the panel.html**

**b) prompt.js**

- Get the same code template for playwright typescript

- Add to the prompt

 PLAYWRIGHT\_TS\_PAGE\_ONLY: `

Instructions:

- Generate ONLY a Playwright TypeScript Page Object Class (no test code).

- Use Playwright v1.35+ compatible syntax.

- Include JSDoc-style comments for all methods & class.

- Use meaningful, readable method names.

- Do NOT include explanations or test scripts.

Context:

DOM:

\`\`\`html

\${domContent}

\`\`\`

Example:

\`\`\`ts

import { Page, Locator } from '@playwright/test';

/\*\*

 \* Page Object for Component Page

 \*/

export class ComponentPage {

  readonly page: Page;

  constructor(page: Page) {

    this.page = page;

  }

  // Add methods and locators as per the DOM

}

\`\`\`

Persona:

- Audience: Automation engineer focused on Playwright-based POM structure.

Output Format:

- A single TypeScript class inside a \`\`\`ts\`\`\` block.

Tone:

- Clean, modular, enterprise-ready.

`,

- Append CODE\_GENERATOR\_TYPES as well

export const CODE\_GENERATOR\_TYPES = {

  SELENIUM\_JAVA\_PAGE\_ONLY: 'Selenium-Java-Page-Only',

  CUCUMBER\_ONLY: 'Cucumber-Only',

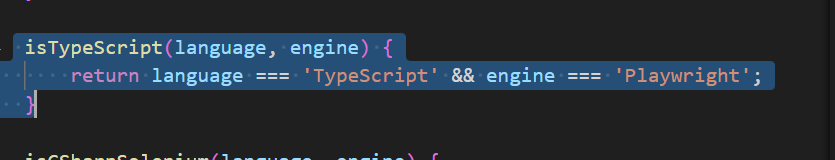
  PLAYWRIGHT\_TS\_PAGE\_ONLY:'Playwright-TS-Page-Only',

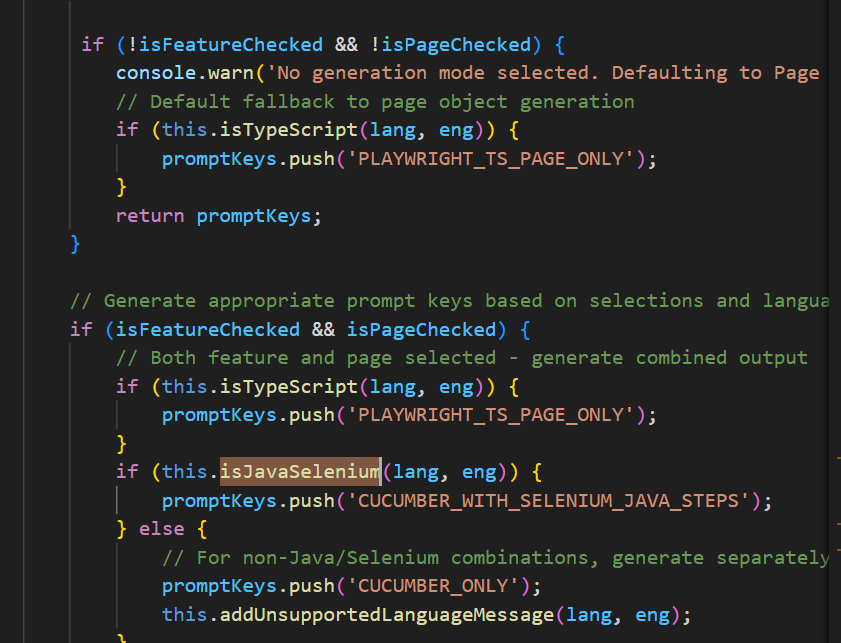
  CUCUMBER\_WITH\_SELENIUM\_JAVA\_STEPS: 'Cucumber-With-Selenium-Java-Steps',

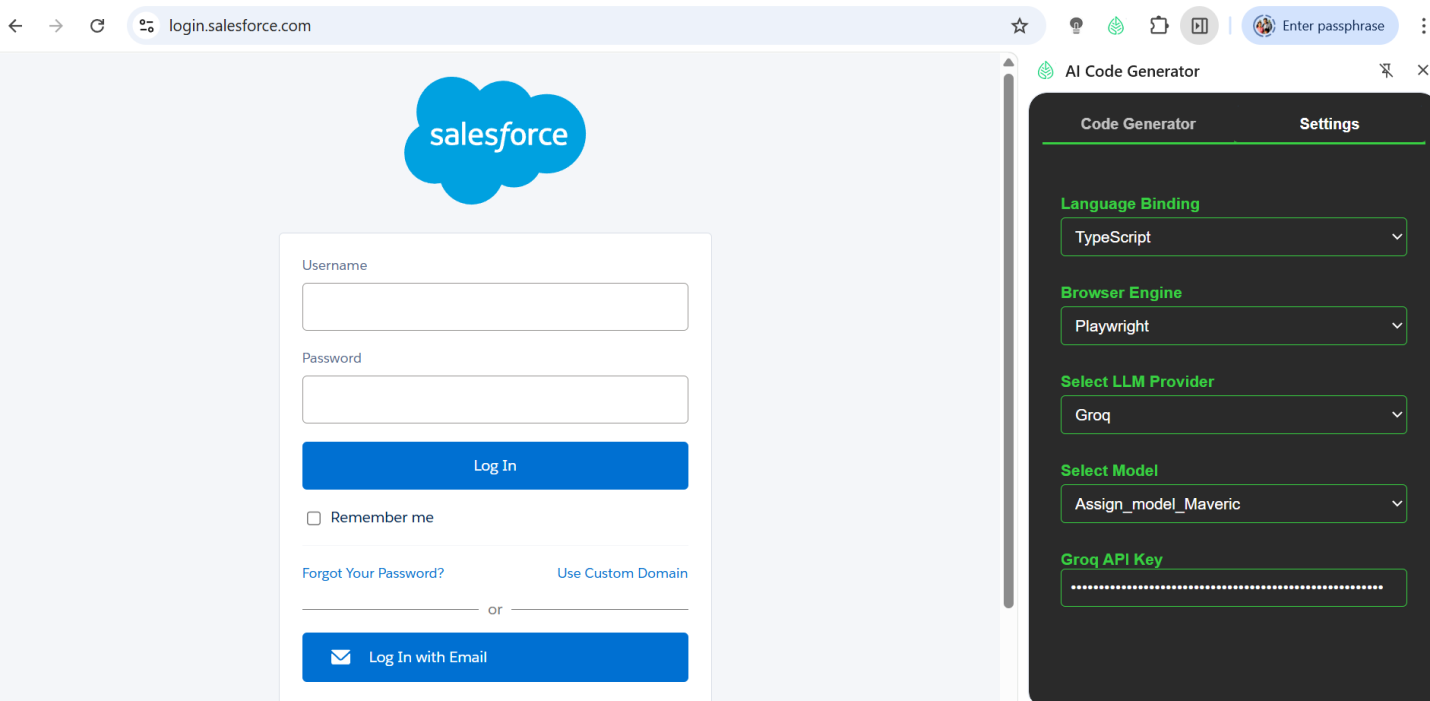
};

c) chat.js

- Add logic for Playwright/typescript using getPromptKeys







1. Test the playwright typescript combination

import { Page, Locator } from '@playwright/test';

/\*\*

\* Page Object for Salesforce Login Page

\*/

export class SalesforceLoginPage {

readonly page: Page;

/\*\*

\* Constructor for SalesforceLoginPage

\* @param page - The Playwright page object

\*/

constructor(page: Page) {

this.page = page;

}

/\*\*

\* Locator for the username input field

\*/

get usernameInput(): Locator {

return this.page.locator('#username');

}

/\*\*

\* Locator for the password input field

\*/

get passwordInput(): Locator {

return this.page.locator('#password');

}

/\*\*

\* Locator for the login button

\*/

get loginButton(): Locator {

return this.page.locator('#Login');

}

/\*\*

\* Locator for the remember me checkbox

\*/

get rememberMeCheckbox(): Locator {

return this.page.locator('#rememberUn');

}

/\*\*

\* Locator for the remember me label

\*/

get rememberMeLabel(): Locator {

return this.page.locator('label[for="rememberUn"]');

}

/\*\*

\* Enter the username

\* @param username - The username to enter

\*/

async enterUsername(username: string): Promise<void> {

await this.usernameInput.fill(username);

}

/\*\*

\* Enter the password

\* @param password - The password to enter

\*/

async enterPassword(password: string): Promise<void> {

await this.passwordInput.fill(password);

}

/\*\*

\* Click the login button

\*/

async clickLoginButton(): Promise<void> {

await this.loginButton.click();

}

/\*\*

\* Check the remember me checkbox

\*/

async checkRememberMe(): Promise<void> {

await this.rememberMeCheckbox.check();

}

/\*\*

\* Uncheck the remember me checkbox

\*/

async uncheckRememberMe(): Promise<void> {

await this.rememberMeCheckbox.uncheck();

}

}