Class Assignment 1

1) Generate feature file for the following applications

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

Feature: Login to Salesforce

Scenario Outline: Successful login with valid credentials

Given I open the login page

When I type "<username>" into the Username field

And I type "<password>" into the Password field

And I click the Login button

Then I should be logged in successfully

Examples:

| username | password |

| "ramesh.kumar@salesforce.com" | "Salesforce123!" |

| "suresh.menon@salesforce.com" | "Sf$Pass123" |

Scenario Outline: Unsuccessful login with invalid credentials

Given I open the login page

When I type "<username>" into the Username field

And I type "<password>" into the Password field

And I click the Login button

Then I should see an error message indicating invalid credentials

Examples:

| username | password |

| "invalid.user@salesforce.com" | "wrongpass" |

| "test.user@salesforce.com" | "incorrect" |

Scenario Outline: Remember me functionality

Given I open the login page

When I type "<username>" into the Username field

And I type "<password>" into the Password field

And I check the Remember me checkbox

And I click the Login button

Then my username should be remembered for future sessions

Examples:

| username | password |

| "ramesh.kumar@salesforce.com" | "Salesforce123!" |

| "suresh.menon@salesforce.com" | "Sf$Pass123" |

Scenario: Forgot password link

Given I open the login page

When I click the Forgot Your Password? link

Then I should be redirected to the password recovery page

Scenario Outline: Custom domain login

Given I open the login page

When I click the Use Custom Domain link

And I enter "<domain>" into the Custom Domain field

And I click the Continue button

Then I should be redirected to the custom domain login page

Examples:

| domain |

| "mycompany.my.salesforce.com" |

| "customdomain.force.com" |

2) Add a new groq model and confirm it is working

Hint: add model id and label in popup.js + close / reopen

3) Confirm 1.1 is working

Feature: Login to OpenTaps

Scenario Outline: Successful login with valid credentials

Given I open the login page

When I type "<username>" into the Username field

And I type "<password>" into the Password field

And I click the Login button

Then I should be logged in successfully

Examples:

| username | password |

| "testuser" | "testpass"|

| "admin" | "admin123"|

| "krishna" | "krishna123"|

| "suresh" | "suresh123"|

Scenario Outline: Failed login with invalid credentials

Given I open the login page

When I type "<username>" into the Username field

And I type "<password>" into the Password field

And I click the Login button

Then I should see an error message

Examples:

| username | password |

| "invaliduser" | "invalidpass"|

| "" | "testpass"|

| "testuser" | ""|

Scenario Outline: Login with custom domain

Given I open the login page

When I click the "Use Custom Domain" link

And I type "<custom\_domain>" into the Custom Domain field

And I click the Continue button

Then I should be redirected to the custom domain login page

Examples:

| custom\_domain |

| "example.com" |

| "testdomain.com"|

Scenario Outline: Login with email

Given I open the login page

When I click the "Log In with Email" link

And I type "<email>" into the Email field

And I click the Login button

Then I should be logged in successfully

Examples:

| email |

| "testuser@example.com"|

| "admin@example.com"|

| "krishna@example.com"|

1) Generate feature file for the following applications

- redbus.in (search)

2) Add a new groq model and confirm it is working

Hint: add model id and label in popup.js + close / reopen

3) Confirm 1.1 is working

Feature: Book Bus Tickets

Scenario Outline: Successful booking with valid details

Given I open the bus booking page

When I select "Bangalore" from the "From" dropdown

And I select "Mysore" from the "To" dropdown

And I select "13 Sep, 2025" from the "Date of Journey" dropdown

And I click the "Today" button

And I click the "Search buses" button

Then I should see the bus details

Examples:

| From | To | Date of Journey |

| Bangalore | Mysore | 13 Sep, 2025 |

| Chennai | Coimbatore| 13 Sep, 2025 |

| Hyderabad | Vijayawada| 13 Sep, 2025 |

Scenario Outline: Successful booking with women booking toggle

Given I open the bus booking page

When I select "Bangalore" from the "From" dropdown

And I select "Mysore" from the "To" dropdown

And I select "13 Sep, 2025" from the "Date of Journey" dropdown

And I click the "Women booking toggle" button

And I click the "Search buses" button

Then I should see the bus details

Examples:

| From | To | Date of Journey |

| Bangalore | Mysore | 13 Sep, 2025 |

| Chennai | Coimbatore| 13 Sep, 2025 |

| Hyderabad | Vijayawada| 13 Sep, 2025 |

Scenario: Cancel booking

Given I open the bus booking page

When I select "Bangalore" from the "From" dropdown

And I select "Mysore" from the "To" dropdown

And I select "13 Sep, 2025" from the "Date of Journey" dropdown

And I click the "Search buses" button

And I click the "Cancel booking" button

Then I should see the cancellation message

Assignment 2:

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

package com.testleaf.pages;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

/\*\*

\* Page Object for Salesforce Login Page

\*/

public class SalesforceLoginPage {

private WebDriver driver;

/\*\*

\* Constructor to initialize the WebDriver

\* @param driver the WebDriver instance

\*/

public SalesforceLoginPage(WebDriver driver) {

this.driver = driver;

}

/\*\*

\* Gets the username input field

\* @return the username input field

\*/

public WebElement getUsernameInputField() {

return driver.findElement(By.id("username"));

}

/\*\*

\* Gets the password input field

\* @return the password input field

\*/

public WebElement getPasswordInputField() {

return driver.findElement(By.id("password"));

}

/\*\*

\* Gets the login button

\* @return the login button

\*/

public WebElement getLoginButton() {

return driver.findElement(By.id("Login"));

}

/\*\*

\* Gets the remember me checkbox

\* @return the remember me checkbox

\*/

public WebElement getRememberMeCheckbox() {

return driver.findElement(By.id("rememberUn"));

}

/\*\*

\* Gets the forgot password link

\* @return the forgot password link

\*/

public WebElement getForgotPasswordLink() {

return driver.findElement(By.id("forgot\_password\_link"));

}

/\*\*

\* Gets the use custom domain link

\* @return the use custom domain link

\*/

public WebElement getUseCustomDomainLink() {

return driver.findElement(By.id("mydomainLink"));

}

/\*\*

\* Gets the custom domain input field

\* @return the custom domain input field

\*/

public WebElement getCustomDomainInputField() {

return driver.findElement(By.id("mydomain"));

}

/\*\*

\* Gets the continue button for custom domain

\* @return the continue button for custom domain

\*/

public WebElement getContinueButtonForCustomDomain() {

return driver.findElement(By.id("mydomainContinue"));

}

/\*\*

\* Gets the back button for custom domain

\* @return the back button for custom domain

\*/

public WebElement getBackButtonForCustomDomain() {

return driver.findElement(By.id("hint\_back\_domain"));

}

/\*\*

\* Gets the log in with email button

\* @return the log in with email button

\*/

public WebElement getLogInWithEmailButton() {

return driver.findElement(By.id("gidr-email-log-in-button"));

}

}

2) 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

- Append CODE\_GENERATOR\_TYPES as well

c) chat.js

- Add logic for Playwright/typescript using getPromptKeys

3) Test the playwright typescript combination

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

/\*\*

\* Page Object for Login Page

\*/

export class LoginPage {

private readonly page: Page;

/\*\*

\* Constructor for LoginPage

\* @param page Playwright Page object

\*/

constructor(page: Page) {

this.page = page;

}

/\*\*

\* Gets the username input field

\*/

get usernameInput(): Locator {

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

}

/\*\*

\* Gets the password input field

\*/

get passwordInput(): Locator {

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

}

/\*\*

\* Gets the login button

\*/

get loginButton(): Locator {

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

}

/\*\*

\* Gets the remember me checkbox

\*/

get rememberMeCheckbox(): Locator {

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

}

/\*\*

\* Gets the forgot password link

\*/

get forgotPasswordLink(): Locator {

return this.page.locator('#forgot\_password\_link');

}

/\*\*

\* Gets the use custom domain link

\*/

get useCustomDomainLink(): Locator {

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

}

/\*\*

\* Gets the custom domain input field

\*/

get customDomainInput(): Locator {

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

}

/\*\*

\* Gets the custom domain continue button

\*/

get customDomainContinueButton(): Locator {

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

}

/\*\*

\* Gets the custom domain back button

\*/

get customDomainBackButton(): Locator {

return this.page.locator('#hint\_back\_domain');

}

/\*\*

\* Enters the username

\* @param username Username to enter

\*/

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

await this.usernameInput.fill(username);

}

/\*\*

\* Enters the password

\* @param password Password to enter

\*/

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

await this.passwordInput.fill(password);

}

/\*\*

\* Clicks the login button

\*/

async clickLoginButton(): Promise<void> {

await this.loginButton.click();

}

/\*\*

\* Checks the remember me checkbox

\*/

async checkRememberMe(): Promise<void> {

await this.rememberMeCheckbox.check();

}

/\*\*

\* Unchecks the remember me checkbox

\*/

async uncheckRememberMe(): Promise<void> {

await this.rememberMeCheckbox.uncheck();

}

/\*\*

\* Clicks the forgot password link

\*/

async clickForgotPasswordLink(): Promise<void> {

await this.forgotPasswordLink.click();

}

/\*\*

\* Clicks the use custom domain link

\*/

async clickUseCustomDomainLink(): Promise<void> {

await this.useCustomDomainLink.click();

}

/\*\*

\* Enters the custom domain

\* @param customDomain Custom domain to enter

\*/

async enterCustomDomain(customDomain: string): Promise<void> {

await this.customDomainInput.fill(customDomain);

}

/\*\*

\* Clicks the custom domain continue button

\*/

async clickCustomDomainContinueButton(): Promise<void> {

await this.customDomainContinueButton.click();

}

/\*\*

\* Clicks the custom domain back button

\*/

async clickCustomDomainBackButton(): Promise<void> {

await this.customDomainBackButton.click();

}

}

Home Assignment:

