# **Documentation:**

## **Objective**

This automation script tests different functionalities of the ServiceNow website, including verifying the homepage title, filling out forms, interacting with dropdowns, and performing login actions.

## **Technologies Used**

* **Selenium WebDriver**: Automates interactions with the web browser.
* **TestNG Framework**: Structures tests and provides features like prioritization and assertions.
* **Java**: Programming language used to create the script.
* **ChromeDriver**: WebDriver implementation for Google Chrome.

## **Code Breakdown**

### **1. Setup Method**

java

@BeforeClass

public void setup() {

driver = new ChromeDriver();

driver.manage().window().maximize();

wait = new WebDriverWait(driver, Duration.ofSeconds(10));

driver.get("https://www.servicenow.com");

System.out.println("Website opened: " + driver.getTitle());

}

**Purpose**:

* Initializes the WebDriver (Chrome).
* Maximizes the browser window.
* Opens the ServiceNow homepage.
* Sets up explicit waits for handling dynamic web elements.

### **2. Homepage Title Verification**

@Test(priority = 1)

public void verifyHomePageTitle() {

String expectedTitle = "ServiceNow – Put AI to Work";

String actualTitle = driver.getTitle();

Assert.assertEquals(actualTitle, expectedTitle, "Page title does not match!");

System.out.println("Verified the title successfully: " + actualTitle);

}

**Purpose**:

* Verifies that the homepage title matches the expected value.
* Uses TestNG assertions to compare actual and expected titles.

### **3. Get Started Form Filling**

java

@Test(priority = 2)

public void getStarted() {

driver.findElement(By.partialLinkText("Get Started")).click();

driver.findElement(By.id("field8")).sendKeys("goswamik1221@gmail.com");

driver.findElement(By.id("field6")).sendKeys("Kanav");

driver.findElement(By.id("field7")).sendKeys("Goswami");

driver.findElement(By.id("field9")).sendKeys("Masai");

driver.findElement(By.id("field13")).sendKeys("9682513658");

WebElement jobLevel = driver.findElement(By.id("field10"));

jobLevel.sendKeys("Developer/Engineer");

WebElement department = driver.findElement(By.id("field11"));

department.sendKeys("IT");

}

**Purpose**:

* Simulates form submission by entering user details (email, name, etc.).
* Navigates to the "Get Started" section and interacts with various form fields.
* Validates that dropdown values for job level and department are entered correctly.

### **4. Dropdown Interaction**

java

@Test(priority = 3)

public void dropdown() {

WebElement Products = wait.until(ExpectedConditions.visibilityOfElementLocated(By.partialLinkText("Products")));

Products.click();

driver.findElement(By.xpath("//span[text()='Platform']")).click();

}

**Purpose**:

* Locates and clicks on the "Products" dropdown menu.
* Selects the "Platform" option from the dropdown using XPath.

**Key Note**: The dropdown handling assumes that the "Products" menu is dynamically loaded. Explicit wait ensures the element is interactable before clicking.

### **5. Login Simulation**

java

@Test

public void login() {

driver.findElement(By.xpath("//i[@slot=\"fa-icon-user-circle\"]")).click();

driver.findElement(By.partialLinkText("Create an account")).click();

driver.findElement(By.id("email")).sendKeys("goswamik1221@gmail.com");

driver.findElement(By.id("text")).sendKeys("Kanav");

}

**Purpose**:

* Simulates logging in by navigating to the "Create an account" page.
* Fills out required fields (email, name) to demonstrate user input handling.

### **6. Browser Quit**

java

@AfterClass

public void quit() {

driver.quit();

System.out.println("Browser closed.");

}

**Purpose**:

* Closes the browser after all tests are executed.
* Ensures no leftover WebDriver instances are running after the test.