Selenium Assignment Questions

Assignment 1

1. Download and launch the "dropdown.html" file.

Dropdown.html
Objective:

- Open dropdown.html
- Select 05-05-2005
- Validate values
- Validate years in ascending order

```java
WebDriver driver = new ChromeDriver();
driver.get("C:\Users\HP\Downloads\Dropdown.html");
Select day = new Select(driver.findElement(By.id("day")));
Select month = new Select(driver.findElement(By.id("month")));
Select year = new Select(driver.findElement(By.id("year")));
day.selectByVisibleText("05");
month.selectByVisibleText("May");
year.selectByVisibleText("2005");
assert day.getFirstSelectedOption().getText().equals("05");
assert month.getFirstSelectedOption().getText().equals("May");
assert year.getFirstSelectedOption().getText().equals("2005");
List<WebElement> yearOptions = year.getOptions();
List<String> years =
yearOptions.stream().map(WebElement::getText).collect(Collectors.toList());
List<String> sorted = new ArrayList<>(years);
Collections.sort(sorted);
assert years.equals(sorted);
driver.quit();
```

---

2. Select date 05-05-2005 from the dropdown and validate the same.

Objective: Load Assignment.html and extract unique rows.

```java
WebDriver driver = new ChromeDriver();
driver.get("C:\Users\HP\Downloads\Assignment.html");
List<WebElement> rows = driver.findElements(By.xpath("//table/tbody/tr"));
Set<String> uniqueRows = new LinkedHashSet<>();
for (WebElement row : rows) {
    uniqueRows.add(row.getText());
}
for (String row : uniqueRows) {
    System.out.println(row);
}
driver.quit();
```

---

3. Fetch the year from the dropdown and validate the year in Ascending Order.

Objective:
- Extract data from state-wise election pages
- Dump into Excel
- Derive metrics

```python
from selenium import webdriver
from selenium.webdriver.common.by import By
import pandas as pd
driver = webdriver.Chrome()
data = []
state_urls = {
```

~~~
    "Goa": "https://results.eci.gov.in/ResultAcGenMar2022/ConstituencywiseS0510.htm?ac=10",
}
for state, url in state_urls.items():
    driver.get(url)
    constituency = driver.find_element(By.XPATH, "//span[@id='ctl00_ContentPlaceHolder1_lblACName']").text
    rows = driver.find_elements(By.XPATH, "//table[2]//tr")
for row in rows[1:]:
        cells = row.find_elements(By.TAG_NAME, "td")
        if len(cells) >= 4:
            data.append({
                "State": state,
                "Constituency": constituency,
                "Candidate": cells[0].text,
                "Party": cells[1].text,
                "Votes": int(cells[2].text.replace(",", "")),
                "Percentage": float(cells[3].text.replace('%', '')) if cells[3].text else 0.0
            })
driver.quit()
df = pd.DataFrame(data)
df.to_excel("election_data.xlsx", index=False)
```
**Post-processing:**
```python
max_vote_per_state = df.loc[df.groupby('State')['Votes'].idxmax()]
max_pct_per_state = df.loc[df.groupby('State')['Percentage'].idxmax()]
~~~

Dropdown.html

Assignment 2

1. Download and launch the "Assignment.html" file.
2. Launch the file.
3. Read the table and find the unique rows from the table.

Assignment.html

Objective: Load Assignment.html and extract unique rows.

```java
WebDriver driver = new ChromeDriver();
driver.get("file:///C:/Users/Chahat/Desktop/Assignment.html");

List<WebElement> rows = driver.findElements(By.xpath("//table/tbody/tr"));
Set<String> uniqueRows = new LinkedHashSet<>();

for (WebElement row : rows) {
    uniqueRows.add(row.getText());
}

for (String row : uniqueRows) {
    System.out.println(row);
}

driver.quit();
```

Assignment 3

2022 Elections

https://results.eci.gov.in/ResultAcGenMar2022/ConstituencywiseS0510.htm?ac=10

Each question is state wise

1. Output should be name of constituency, candidate name, and vote number/percentage or whatever is the deciding factor, dump all the data in excel with column (all column+state+constituency name).

2. get the candidate which has got the maximum vote in each state with their constituency name.

3. get the candidate which has got the maximum percentage of vote in each state with their constituency name. (percentage)
4. candidate who won with maximum vote difference.
5. candidate who won with maximum vote percentage difference.
6. candidate who won with the minimum vote.
7. candidate who won with minimum vote percentage.
8. total count of candidate who have got less vote than nota.
9. total count of candidates who have gotten greater than 50% vote.
10. name of candidate who has got minimum vote in each state.

Objective:
- Extract data from state-wise election pages
- Dump into Excel
- Derive metrics

```python
from selenium import webdriver
from selenium.webdriver.common.by import By
import pandas as pd

driver = webdriver.Chrome()
data = []

state_urls = {
    "Goa": "https://results.eci.gov.in/ResultAcGenMar2022/ConstituencywiseS0510.htm?ac=10",
}

for state, url in state_urls.items():
    driver.get(url)
    constituency = driver.find_element(By.XPATH, "//span[@id='ctl00_ContentPlaceHolder1_lblACName']").text
    rows = driver.find_elements(By.XPATH, "//table[2]//tr")

for row in rows[1:]:
        cells = row.find_elements(By.TAG_NAME, "td")
        if len(cells) >= 4:
            data.append({
                "State": state,
                "Constituency": constituency,
                "Candidate": cells[0].text,
                "Party": cells[1].text,
            "Votes": int(cells[2].text.replace(",", "")),
            "Percentage": float(cells[3].text.replace('%', '')) if cells[3].text else 0.0
        })

driver.quit()
df = pd.DataFrame(data)
df.to_excel("election_data.xlsx", index=False)
```

Post-processing:
```python
max_vote_per_state = df.loc[df.groupby('State')['Votes'].idxmax()]
max_pct_per_state = df.loc[df.groupby('State')['Percentage'].idxmax()]
```

Assignment 4

Please do the following assignment for cucumber framework -

1. Install Cucumber

2. Create a Cucumber project

3. Use the attached feature file and implement the stepDefinitions for all the scenarios in the feature file. (You can use dummy code in the stepDefinition methods)

Login.feature

4. Execute TestRunner.

5. Assign tags to specific scenarios in the feature file and execute TestRunner for those particular tags.

TestRunner.java

```
package Runners;


import org.junit.runner.RunWith;

import io.cucumber.junit.Cucumber;

import io.cucumber.junit.CucumberOptions;
```

```
@RunWith(Cucumber.class)
@CucumberOptions(
        features = "src/test/resources/Features",
        glue = {"StepDefinations"},
        plugin = {"pretty", "html:target/cucumber.html"}
)
public class TestRunner {
}
```

LoginStepDefinitions.java

```
package StepDefinations;

import io.cucumber.java.en.*;

public class LoginStepDefinations {

    @Given("User is on NetBanking landing page")
    public void user_is_on_net_banking_landing_page() {
        System.out.println("User navigated to NetBanking landing page");
    }

    @When("User login into application with username and password")
    public void user_login_with_username_and_password() {
        System.out.println("Logging in with default credentials");
    }

    @Then("Home page is populated")
    public void home_page_is_populated() {
        System.out.println("Home page displayed successfully");
    }

    @And("Cards are displayed")
    public void cards_are_displayed() {
        System.out.println("Cards are visible");
    }

    @When("User login into application with {string} and password {string}")
    public void user_login_with_username_and_password_param(String username, String password) {
        System.out.println("Logging in with user: " + username + ", password: " + password);
    }

    @And("Cards displayed are {string}")
    public void cards_displayed_are(String status) {
        System.out.println("Cards displayed status: " + status);
    }

    @When("^User login in to application with (.+) and password (.+)$")
    public void user_login_in_to_application_with_and_password(String username, String password) {
        System.out.println("Scenario Outline - Username: " + username + ", Password: " + password);
    }
```

```
}
```

login.feature

```
Feature: Application Login

Scenario: Home page default login
Given User is on NetBanking landing page
When User login into application with username and password
Then Home page is populated
And Cards are displayed

Scenario: Home page default login
Given User is on NetBanking landing page
When User login into application with "jin" and password "1234"
Then Home page is populated
And Cards displayed are "true"

Scenario: Home page default login
Given User is on NetBanking landing page
When User login into application with "john" and password "4321"
Then Home page is populated
And Cards displayed are "false"

Scenario Outline: Home page default login
Given User is on NetBanking landing page
When User login in to application with <Username> and password <password>
```

Then Home page is populated

And Cards displayed are "true"

Examples:

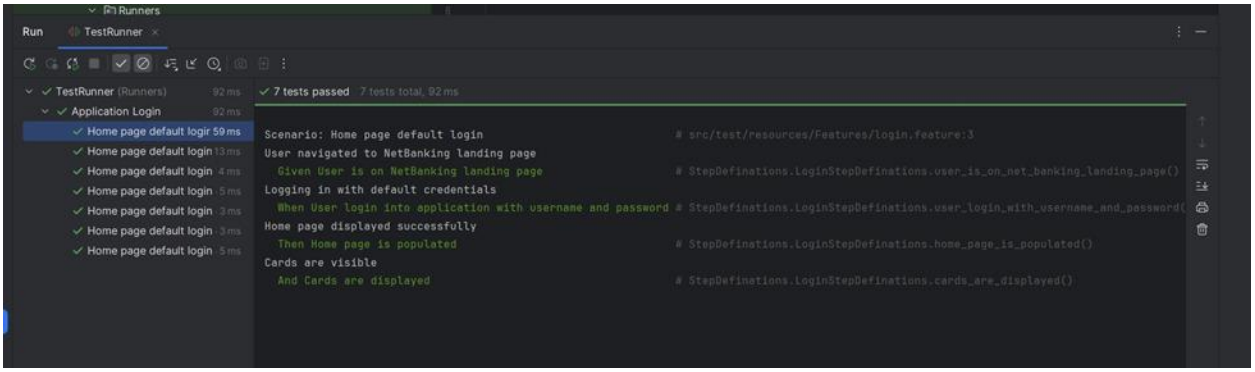
|Username |password|

|user1 |pass1 |

|user2 |pass2 |

|user3 |pass3 |

|user4 |pass4 |



```
✓ TestRunner (Runners)             92 ms
  ✓ Application Login              92 ms
    ✓ Home page default logir      59 ms
    ✓ Home page default login      13 ms
    ✓ Home page default login       4 ms
    ✓ Home page default login       5 ms
    ✓ Home page default login       3 ms
    ✓ Home page default login       3 ms
    ✓ Home page default login       5 ms
```

```
✓ 7 tests passed  7 tests total, 92 ms

Scenario: Home page default login                        # src/test/resources/Features/login.feature:3
User navigated to NetBanking landing page
  Given User is on NetBanking landing page               # StepDefinations.LoginStepDefinations.user_is_on_net_banking_landing_page()
Logging in with default credentials
  When User login into application with username and password # StepDefinations.LoginStepDefinations.user_login_with_username_and_password(
Home page displayed successfully
  Then Home page is populated                            # StepDefinations.LoginStepDefinations.home_page_is_populated()
Cards are visible
  And Cards are displayed                                # StepDefinations.LoginStepDefinations.cards_are_displayed()
```

Assignment 5

Please do the following assignment for TestNG framework -

1. Install TestNG

2. Create a TestNG Project

3. Create 2 test classes (with 3 test cases each).

4. Keep the 2 test classes in 2 different <test> tags in testng.xml

5. Execute the tests above using testng.xml

6. Assign a group to a few test cases and update testng.xml to run test cases belonging to the group.

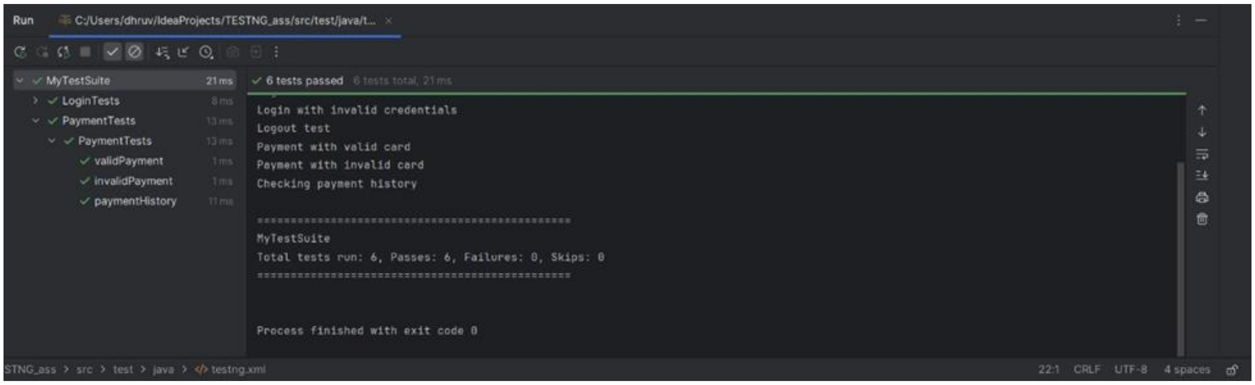
7. Assign priority to the test cases.

Once completed, please share the below files -

- testng.xml

- 2 test class files created

- Final Console Output (in a .txt file)



```
Login with invalid credentials
Logout test
Payment with valid card
Payment with invalid card
Checking payment history

===============================================
MyTestSuite
Total tests run: 6, Passes: 6, Failures: 0, Skips: 0
===============================================


Process finished with exit code 0
```