Selenium Assignment Questions

Assignment 1

1. Download and launch the "dropdown.html" file.
2. Select date 05-05-2005 from the dropdown and validate the same.
3. Fetch the year from the dropdown and validate the year in Ascending Order.
4. Objective:
   - Open dropdown.html
   - Select 05-05-2005
   - Validate values
   - Validate years in ascending order

5.
   ```java
   WebDriver driver = new ChromeDriver();
   driver.get("file:///C:/Users/Desktop/dropdown.html");
6. 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")));
7. day.selectByVisibleText("05");
   month.selectByVisibleText("May");
   year.selectByVisibleText("2005");
8. assert day.getFirstSelectedOption().getText().equals("05");
   assert month.getFirstSelectedOption().getText().equals("May");
   assert year.getFirstSelectedOption().getText().equals("2005");
9. 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);
10. driver.quit();
    ```
11. ---

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/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

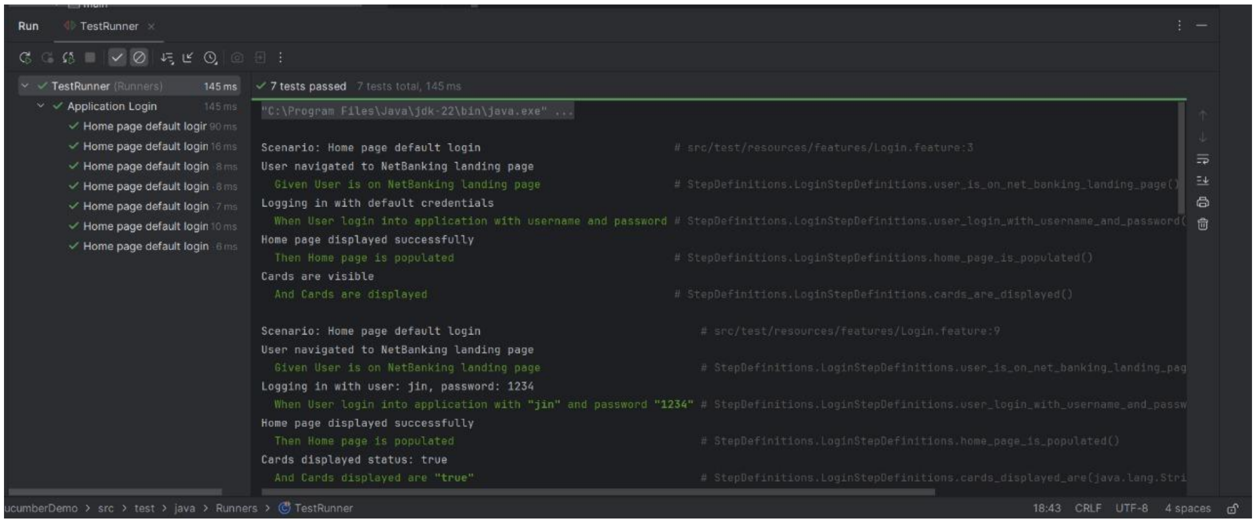
2. Create a Cucumber project

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.


Run
TestRunner
✓ TestRunner (Runners) 145 ms
✓ Application Login 145 ms
✓ Home page default logir 90 ms
✓ Home page default login 16 ms
✓ Home page default login 8 ms
✓ Home page default login 8 ms
✓ Home page default login 7 ms
✓ Home page default login 10 ms
✓ Home page default login 6 ms
✓ 7 tests passed 7 tests total, 145 ms
ucumberDemo > src > test > java > Runners > TestRunner
18:43 CRLF UTF-8 4 spaces

```
"C:\Program Files\Java\jdk-22\bin\java.exe" ...

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                   # StepDefinitions.LoginStepDefinitions.user_is_on_net_banking_landing_page()
Logging in with default credentials
  When User login into application with username and password # StepDefinitions.LoginStepDefinitions.user_login_with_username_and_password(
Home page displayed successfully
  Then Home page is populated                                # StepDefinitions.LoginStepDefinitions.home_page_is_populated()
Cards are visible
  And Cards are displayed                                    # StepDefinitions.LoginStepDefinitions.cards_are_displayed()

Scenario: Home page default login                                # src/test/resources/features/Login.feature:9
User navigated to NetBanking landing page
  Given User is on NetBanking landing page                       # StepDefinitions.LoginStepDefinitions.user_is_on_net_banking_landing_pag
Logging in with user: jin, password: 1234
  When User login into application with "jin" and password "1234" # StepDefinitions.LoginStepDefinitions.user_login_with_username_and_passw
Home page displayed successfully
  Then Home page is populated                                    # StepDefinitions.LoginStepDefinitions.home_page_is_populated()
Cards displayed status: true
  And Cards displayed are "true"                                 # StepDefinitions.LoginStepDefinitions.cards_displayed_are(java.lang.Stri
```

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.

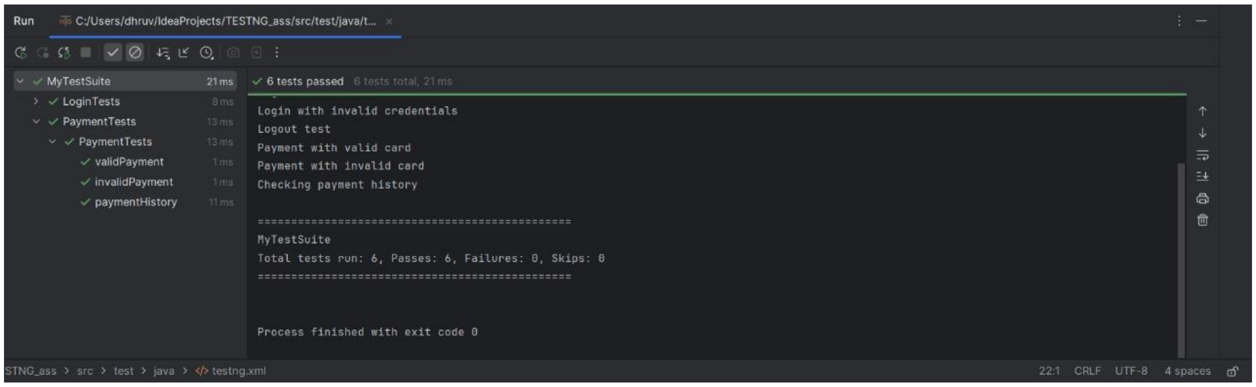
7. Assign priority to the test cases.

Once completed, please share the below files -

- testng.xml

- 2 test class files created

- Final Console Output (in a .txt file)



Run C:/Users/dhruv/IdeaProjects/TESTNG_ass/src/test/java/t...

```
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
```