

Technical Task

Manual Testing

Task 1:

- Validate that text fields are blue
- Validate that lists are green
- Validate that the list data is correct
- Validate that all fields are required when searching for a school
- Validate that all required data in the Student's information must be filled
- Validate on optional data
- Validate duplicate IDs are not allowed
- Validate that the format of each field is correct
 - For example: Email is in email format
 - Birth date
 - Field language
- Validate that drop down lists are all with the correct data
- Validate that birthdates that is newer than today won't be accepted
- Validate that the user could upload photo from device
- Validate that the user could take a photo
- Validate that Information source is disabled by default
- Validate on the condition that makes Information source drop down list becomes active
- Validate that the data gets saved by pressing +
- Validate that by not pressing +, data won't be saved
- Validate that only 1 student could be saved at a time

Task 2:

- Added a new column to calculate the length of the National IDs
- Query used: =QUERY(Student!A:Z,"Select * where (A Starts with 1234) and (L = 10) and ((K = 1 or K = 3 or K = 4) and H = 'M') or ((K = 2 or K = 5 or K = 3) and H = 'F')",1)

Code Assessment Task

Code 1:

Best code practise:

```
if(Card.Type == "Master"){  
  
    Accept Transaction;  
  
}else{ return false; }
```

Code 2:

```
n=5;
```

The assignment of n inside the while loop causes infinite loop

API Manual Testing Task

API test cases:

1. Validate that response matches the API contract when a simple get request with a valid city name
 - a. Response code is 200
 - b. Response body matches the contract
2. Validate that response matches the API contract when a simple get request with a valid city name with space
 - a. Response code is 200
 - b. Response body matches the contract
3. Validate that response matches the API contract when a simple get request with an invalid city name
 - a. Response code is 400

Automation Testing

1. Selenium Locators:
 - a. ID
 - b. Name
 - c. Class name
 - d. CSS Selector

- e. XPath
 - f. Tag name
 - g. Link text
 - h. Partial link text
2. Different Selenium Drivers:
- a. ChromeDriver
 - b. FirefoxDriver
 - c. EdgeDriver
 - d. SafariDriver
 - e. Headless Browsers
 - f. Remote WebDriver
3. Different waits in Selenium
- a. **Implicit Wait:** Use for basic waiting needs when dealing with elements that generally load within a predictable timeframe.
 - b. **Explicit Wait:** Preferred for waiting on specific elements or conditions, especially for asynchronous content or dynamic elements.
 - c. **Fluent Wait:** Use when you need more control over retry logic, polling frequency, and exception handling during the waiting period.
4. Difference between driver.quit() and driver.close()
- a. driver.quit()
 - i. Closes all browser windows and tabs
 - ii. Terminates the session
 - b. driver.close()
 - i. Closes the currently focused browser window
 - ii. Session remains active

Github URL: <https://github.com/Abdelrahman-Wagdy/IT-worx.git>