**Case study**

* **Prerequisites:**
* [Download and Install](http://toolsqa.com/selenium-webdriver/c-sharp/download-and-install-visual-studio/) Python 3.7 and above version
* [Set Up Selenium Web Driver](http://toolsqa.com/selenium-webdriver/c-sharp/set-up-selenium-webdriver-with-visual-studio-in-c/)
* **Python Fundamentals and Unit testing**

**1. Create a file name in Python and find the number of lines, characters and words present on it.**

* Write a python program to Create a file named "sample.txt" for writing.
* Enter few lines of text to this file
* Close the file and save it
* Open the File in read mode
* Put the logic and find the below requirements

Tests :

* Calculate the number of lines present inside the file.
* Calculate the number of Characters present inside the file.
* Calculate the number of words present inside the file.
* Calculate the number of spaces present inside the file.

**2. Implement Python Object Oriented concept, Inheritance for getting the details of Student**

* Create a base class named Person with instance member's as (name, age, address) and define a constructor to initialize the data members.
* Create a derived class named Student from the base class Person with instance member's (roll no and percentage) and define a constructor to initialize the data members.
* Include method overriding with the method named "display" to display the details of instance members.

Tests :

Once you have the class and interface structure completed, write the following tests to show your code working.

* Using a Student Object, display all the details of a student (roll no, name, age, and percentage). If the mark is > 60%, print First Class. If the mark is > 50% and < 60%, print Second Class, rest is Third class
* Count number of student objects/instances have been created.

**3. Install and make the Setup for Selenium. Use locators for automation (you can choose either chrome or Firefox web driver)**

Selenium Web driver

Selenium Web driver Chrome driver/Firefox

Nugget package

* Navigate to this URL - [**http://demo.automationtesting.in/Register.html**](http://demo.automationtesting.in/Register.html)
* Automate the process of Filling Registration form with submission.
* For filling form Interact with Web Elements like (Textbox, List box, Radio Buttons, Check box, buttons etc.) With different locators like ID, Name, Xpath, CSS selector, Class, Tag etc.
* Use Waits while loading Elements. Upload Photo with Windows based alert window Handling.
* Highlight & click Submit Button . Maximize Browser Window while automating Registration process.

**4.Perform keyboard and mouse events Using Actions class**

* Navigate to given URL - [**https://jqueryui.com/droppable/**](https://jqueryui.com/droppable/)

Switch to Frame. Drag source & drop it on target

* Navigate to given URL - [**https://www.spicejet.com/**](https://www.spicejet.com/)

Perform Mouse hover action on add-on menu then Click on visa services from coming list.

* Navigate to given URL - [**https://demoqa.com/text-box**](https://demoqa.com/text-box)

Enter Username & current address. Copy Current address & paste it to Permanent address (with keyboard events with actions class). click on submit button & accept alert.

**5. Build Automation Framework & Automate the complete step for logging into gmail.com account.**

Create a new project with Page object Model. Develop separate page class for automating below scenarios 1. Login 2. Search Item 3. Add to Cart 4. Checkout & order receipt.

Create object repository, Base class, common Utility class as per requirement.

Use below actions & Run the project &

* Navigate to you gmail id
* Verify the title
* Click on login button
* Enter username & password
* Click on Submit Button
* Verify that message is displayed “Login successfully”.
* Compose a new mail and sent it. Make sure mail sent successfully.
* Very that mail is available in sent items list.

**Miscellaneous**

* Create Project structure.
* In the code handle Exceptions. Use Assertions & Run tests.
* In case of failures take screenshots.
* Install NLog & generate Logs.
* Generate Extent Reports.
* Install AWS Devops & create VM, install Jenkin server. Configure it with GIT plugins
* Data drive your tests on five different configurations in parallel.
* Configure the project in pipeline for Continuous integration & achieve End to End automation.
* Configure Reports on Execution.