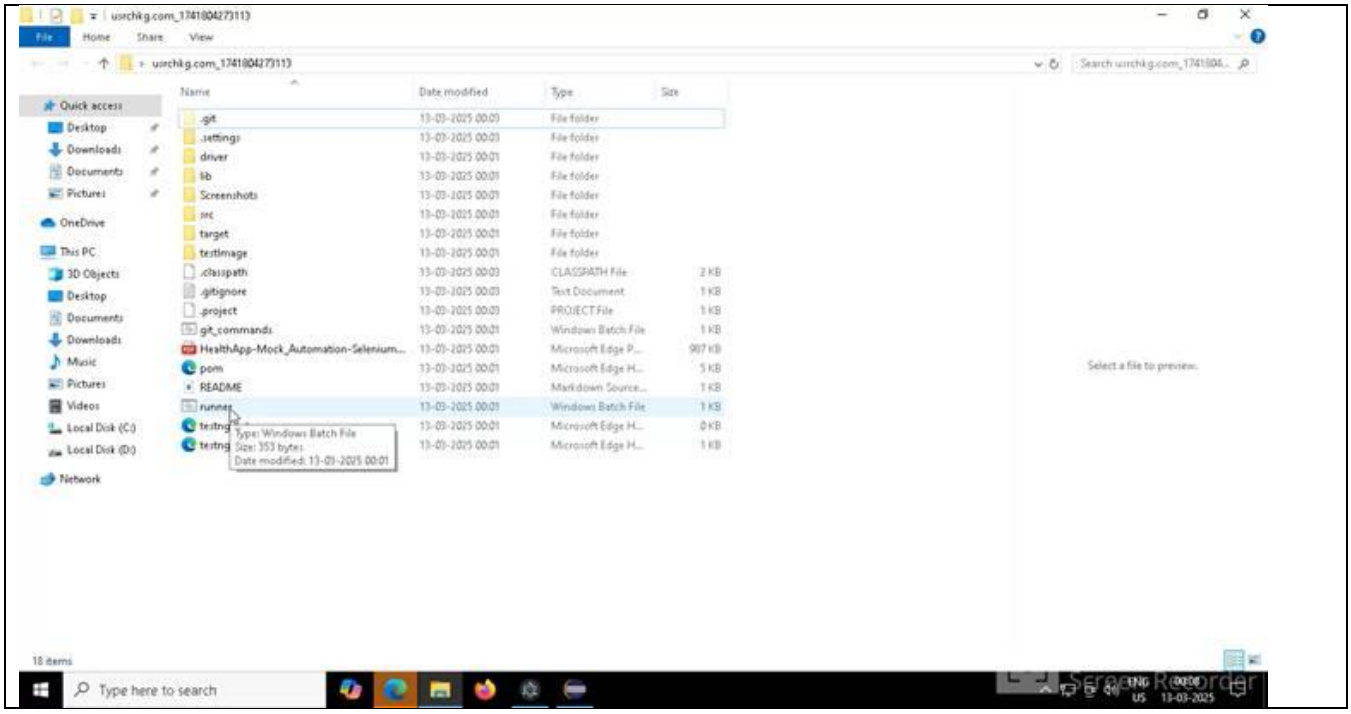
HealthApp Automation Using C# and Selenium VM-DotNet_PL1-Set3(6 TCs)

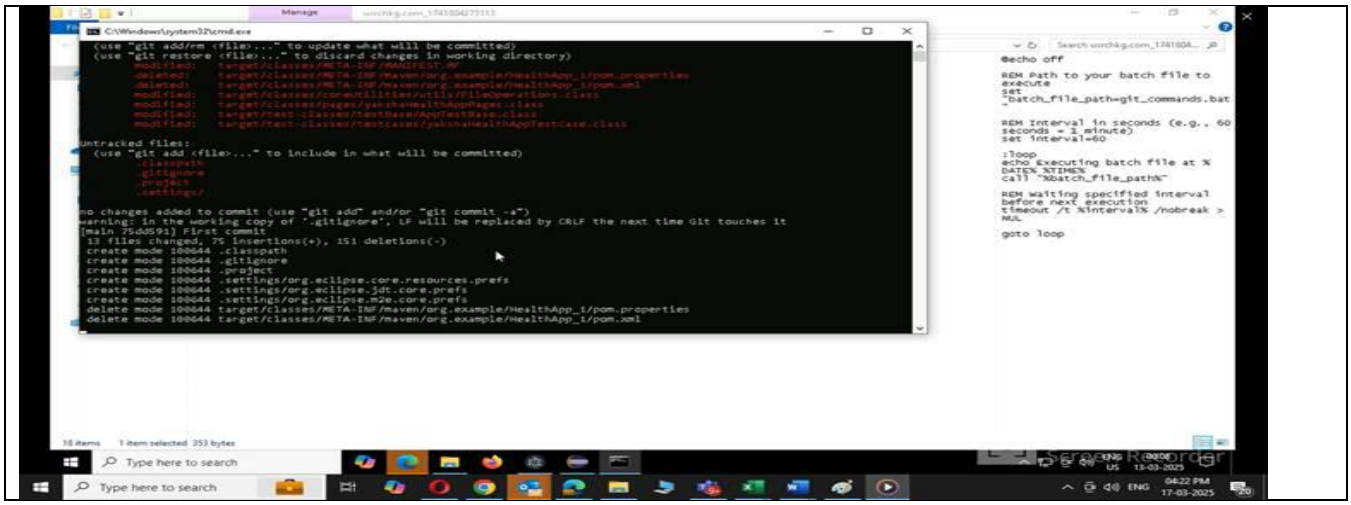
Table of Contents

Pre-requisite:

Before you start working on your project, you must execute the runner file present in your project folder (simply by double-clicking). This is mandatory.

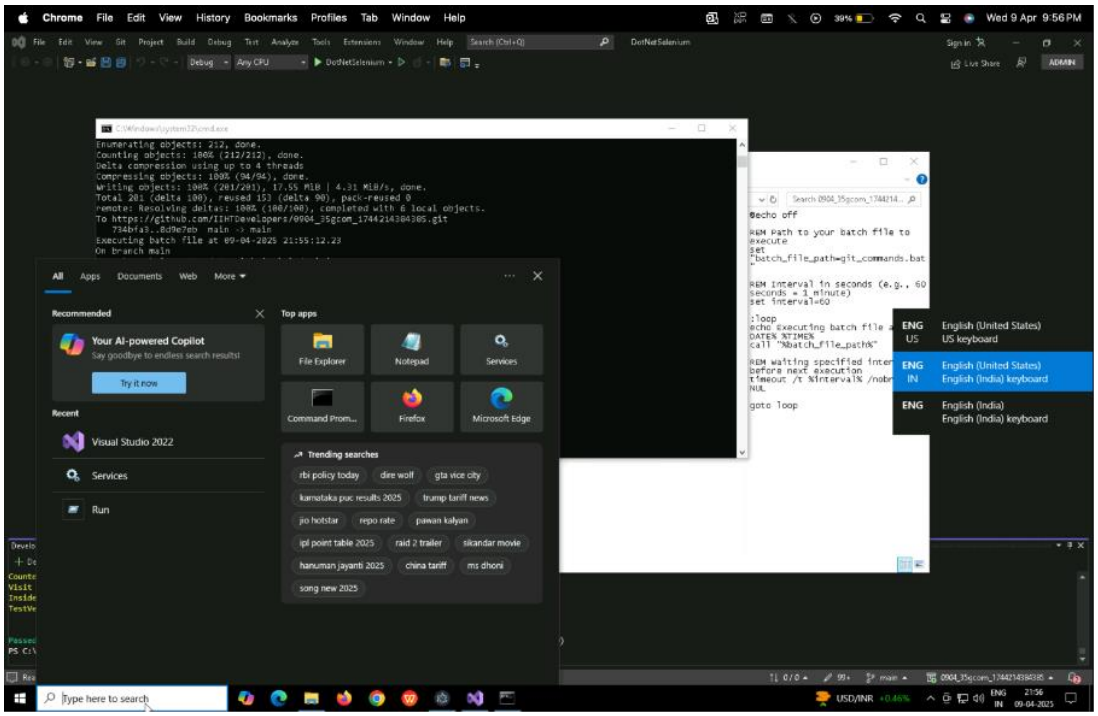


This will launch a command terminal for you where it will keep on pushing your updated code to GIT on regular intervals. Keep that command terminal open at backend and you can continue working on your project.

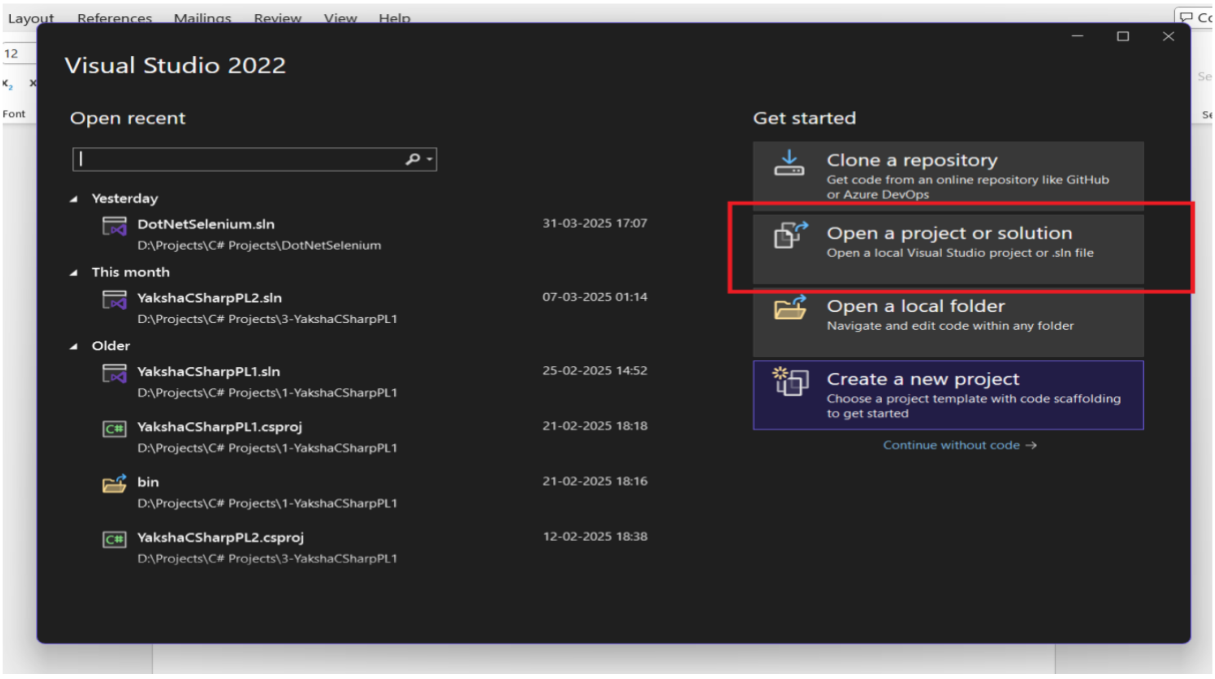


As soon as you import the project into Visual Studio 2022, Please follow the below steps:

1. Open the Visual Studio code.



2. Open Your C# Selenium Project
    a. Launch Visual Studio.
    b. Open your Selenium automation project (.sln file).



Template Code Structure:

a. The packages and files you will be required to work on are listed in the table below.

b. Other Files and packages you can ignore.

c. In other Files and packages, do not make any changes. It would affect your evaluation.

d. You are not required to work in the "TestCases" Folder. Files there are non-editable. Editing those files and trying to save them will throw errors and would affect your evaluation.

Package	Class/File	Description
/DotNetSelenium/Utilities/	JsonReader.cs	**The method for reading data as input from an JSON file has already been implemented here.** **This method is used to fetch the required data from JSON including the URL for navigation.**
/DotNetSelenium/PageObjects	SubstorePage.cs (File has the Testcase function in templated form here, You need to write the required logic and xpath into it.)	1. All core activities (mentioned in the list above) are to be performed here. 2. The comments associated with each templated method here describe the expectation. 3. You can define locators and xpath here. 4. Declare any variable/object you need to share data/status between different methods. 5. Do not modify the signature of methods declared here. 6. You can create additional supportive common methods if required.
/DotNetSelenium/TestData	LoginData.json PatientName.json	It contains data for login and filling in the form.

PROBLEM STATEMENT

Need to automate the following activities using Selenium + Java.

Key Activities to implement:

Sl No.	Summary	Action	Expected Result
1	Verify the SubStore module is present or not	1. go to URL : https://healthapp.yaksha.com/ 2. login as valid credential (username : admin , password : pass123) and click on "Sign in" Button 3. Scroll down menu till SubStore 4. Click on the SubStore	Return the current URL to ensure that SubStore module should be present.
2	To ensure that the "Select Your Substore" heading is displayed and all expected sub-module cards/tiles are also displayed.	Pre-condition: User should be logged in and “Substore” module must be already selected. **Steps:** **1.** Locate and click on the 4th 'Substore' tile.	Upon clicking the 'Substore' module, the "Select Your Substore" heading should appear with the correct heading. All specified sub-module cards/tiles should be present and displayed correctly.
3	Ensure that the tooltip text on the substore switch button accurately displays the correct information when hovered over in the "Account" substore.	**Preconditions**: The user must be logged in to the application. The user is already on the "Substore" module page. **Steps**: 1. Click on the “Account” option within the substore module. 2. Move the cursor to hover over the substore switch button.	Verify that the tooltip text contains the following message: “You are currently in Accounts sub store. To change, you can always click here.”
4	Ensure that all expected sub-modules are displayed correctly.	**Preconditions**: The User must be logged into the HealthApp application. The user is already on the SubStore module. **Test Steps**: 1. Select 'Inventory' Sub-Module. 2. Select 'Pharmacy' Sub-Module.	All sub-modules should be displayed correctly. Expected submodules are: Pharmacy, Inventory
5	To verify that all sub-modules under the Inventory module are present and visible in the user interface.	**Preconditions**: The user must be logged into the system. The user should be on the ”Inventory” submodule of the "SubStore" module page. **Test Steps**: 1. NA (Observe the sub-modules that appear under this module.)	All sections should be displayed correctly. Expected Sub modules are: Stock, Inventory Requisition, Consumption, Reports, Patient Consumption, Return
6	To manually verify that navigation between different submodules within the "Inventory" module updates the URL correctly, reflecting the content of the newly navigated submodule.	**Preconditions**: The user must be logged into the application and on the "Inventory" module and its respective sub-module. **Test Steps**: 1. Navigate to the 'Inventory' Submodule. 2. Navigate to 'Stock' Submodule. 3. Navigate to 'Inventory Requisition' Submodule. 4. Move to 'Consumption' Submodule. 5. Proceed to 'Reports' Submodule. 6. Navigate to 'Patient Consumption' Submodule. 7. Go to 'Return' Submodule. 8. Return to 'Stock' Submodule.	Each click should lead to the correct submodule, and the URL should update accordingly to reflect the navigation accurately.

NOTE: "Please do not delete any file in the project folder. But you are free to add any other files if required".

Expectations:

1) Learners should write automation scripts using C# and Selenium to automate all the steps in the above question. In other words, the automation script should perform all the mentioned steps.
2) Learners should not use any tools to create the xpath. They should develop the xpath/cssselector on their own.

IMPLEMENTATION/FUNCTIONAL REQUIREMENT

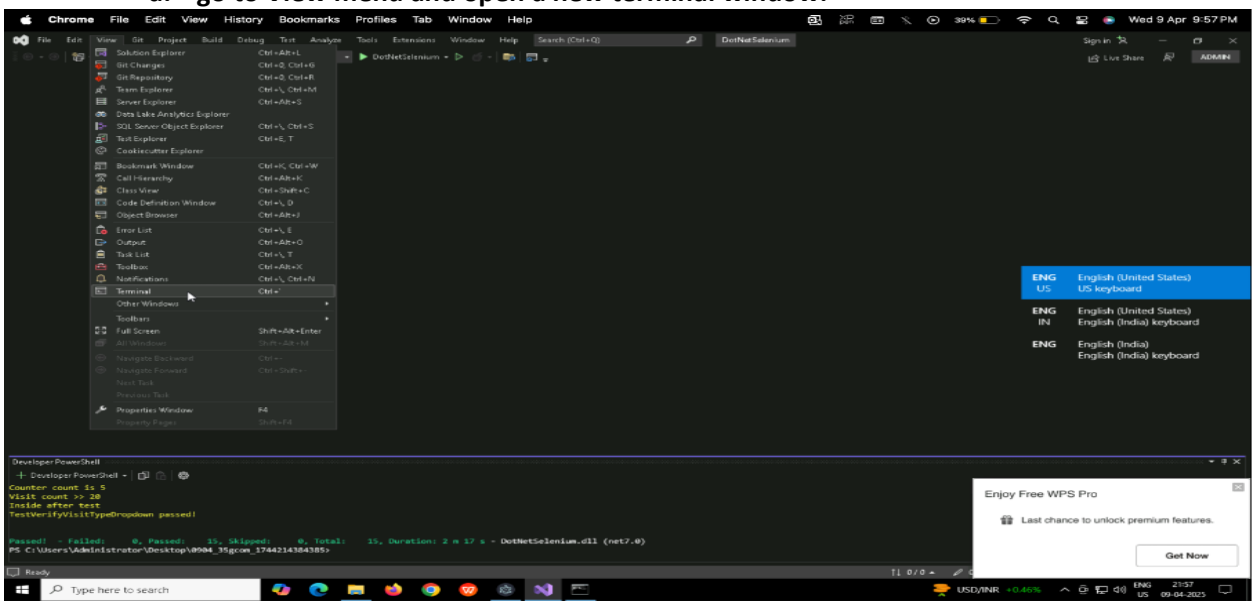
1.1 CODE QUALITY/OPTIMIZATIONS

1. Associates should have written clean code that is readable.
2. Associates need to follow SOLID programming principles.

EXECUTION STEPS TO FOLLOW

1. **You are required to run test cases for applications before final submission. Without this, the project evaluation will not happen.**

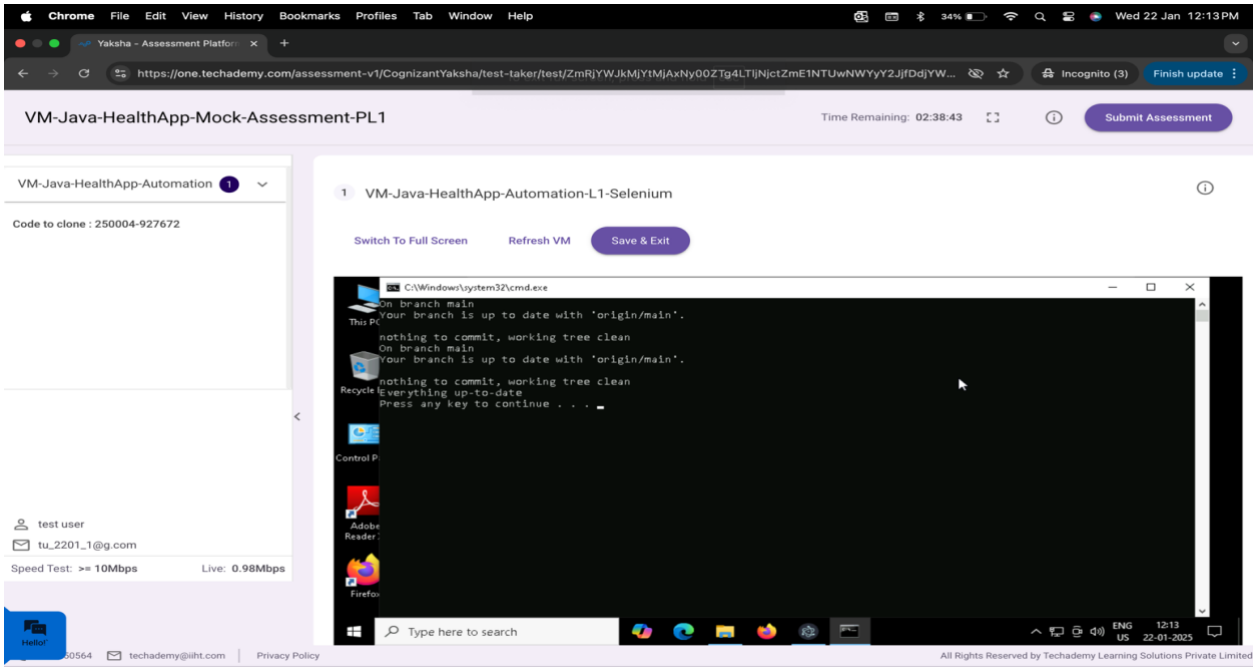
2. **You can launch test cases at any time as follows:**
    a. **go to View menu and open a new terminal window.**



3. **Use the following commands :**
    a. **Dotnet clean (to clean the project)**
    b. **Dotnet build (to build the project with latest changes)**

c. **Dotnet test (to execute the test cases)**

4. **To do the final submission of the assessment :**
    a. **Press escape to come out of Fullscreen mode.**
    b. **Submit the assessment.**



After the successful submission of the assessment, you will get a confirmation message displayed on your screen.

==========

All the Best