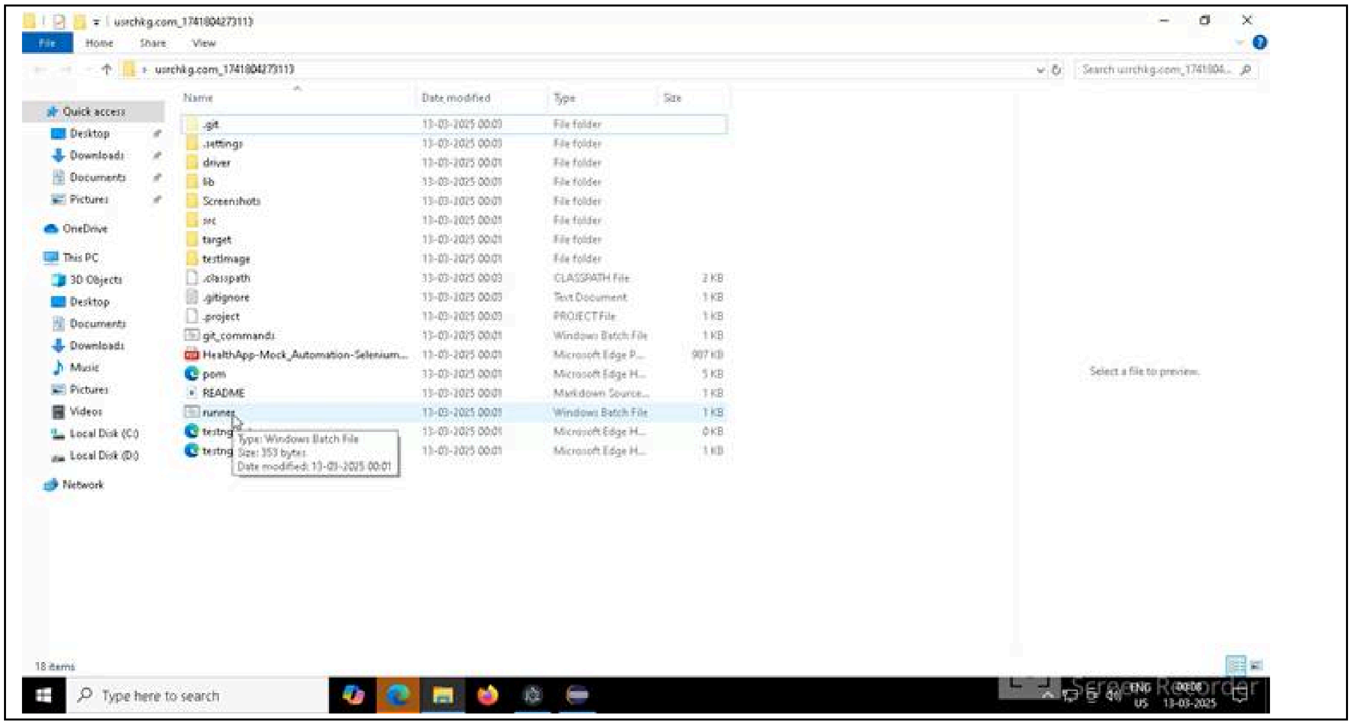
HealthApp Automation Using C# and Selenium

VM-DotNet_PL1-2 (15 TCs)

Pre-requisite:

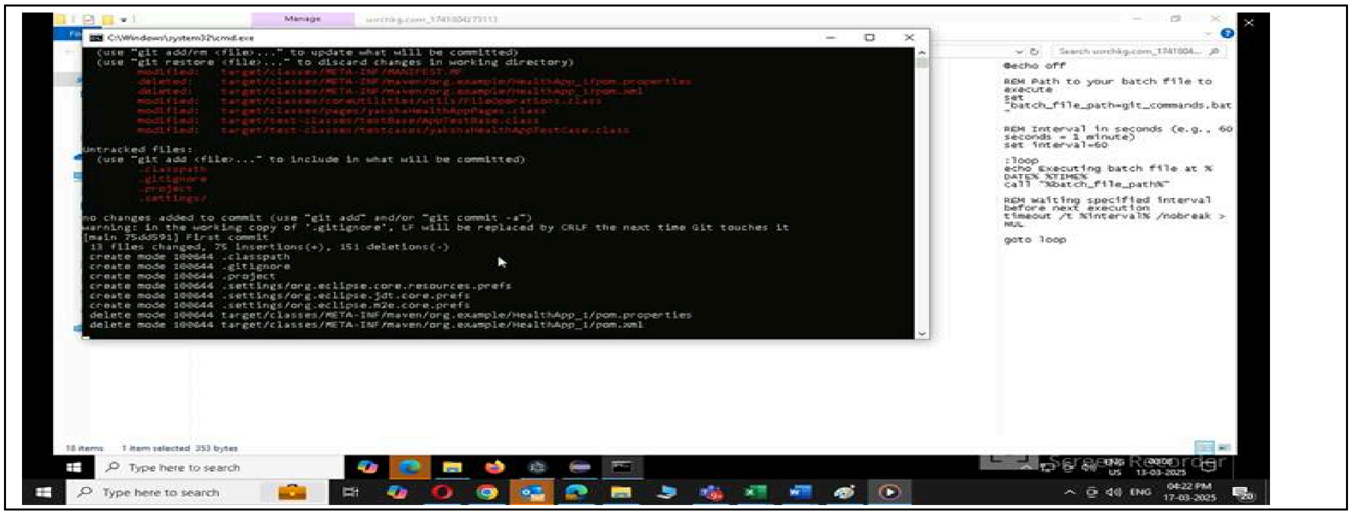
Before you start working on your project, execute the runner file present in your project folder (Simply by double clicking). This is mandatory.



Name	Date modified	Type	Size
.git	13-03-2025 00:03	File folder	
.settings	13-03-2025 00:03	File folder	
driver	13-03-2025 00:01	File folder	
lib	13-03-2025 00:01	File folder	
Screenshots	13-03-2025 00:01	File folder	
src	13-03-2025 00:01	File folder	
target	13-03-2025 00:01	File folder	
testImage	13-03-2025 00:01	File folder	
.classpath	13-03-2025 00:03	CLASSPATH File	2 KB
.gitignore	13-03-2025 00:03	Text Document	1 KB
.project	13-03-2025 00:03	PROJECT File	1 KB
git_commands	13-03-2025 00:01	Windows Batch File	1 KB
HealthApp-Mock_Automation-Selenium...	13-03-2025 00:01	Microsoft Edge P...	907 KB
pom	13-03-2025 00:01	Microsoft Edge H...	5 KB
README	13-03-2025 00:01	Markdown Source...	1 KB
runner	13-03-2025 00:01	Windows Batch File	1 KB
testng	13-03-2025 00:01	Microsoft Edge H...	0 KB
testng	13-03-2025 00:01	Microsoft Edge H...	1 KB

```
Type: Windows Batch File
Size: 353 bytes
Date modified: 13-03-2025 00:01
```

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.

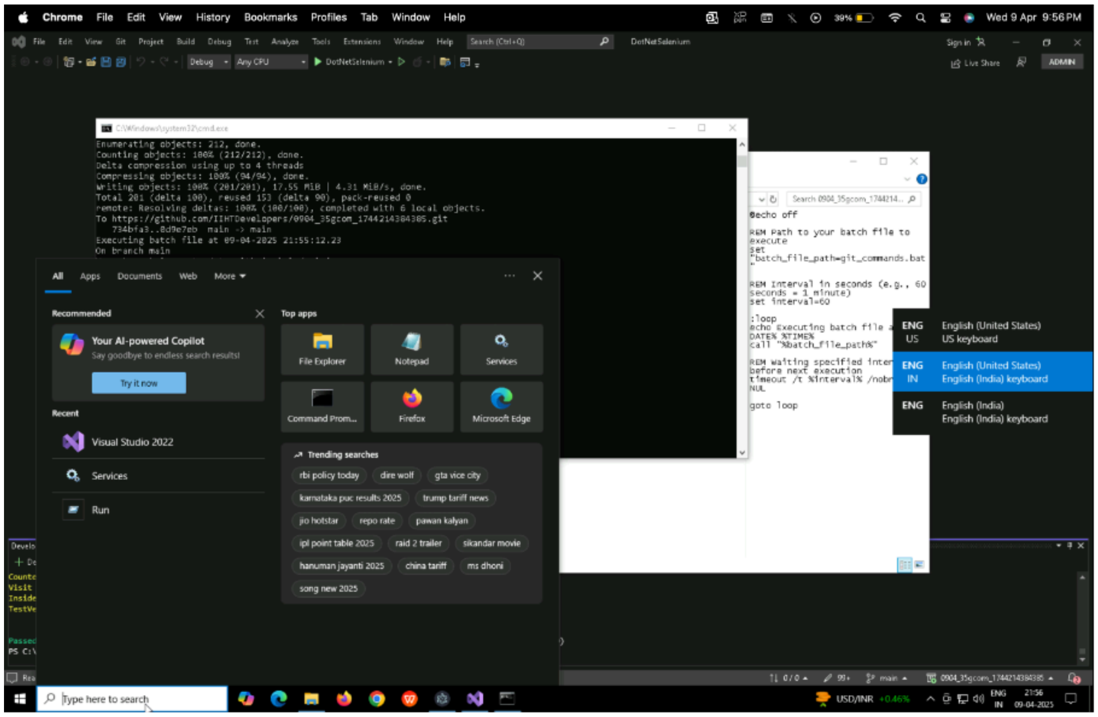


```
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   target/classes/META-INF/MANIFEST.MF
        deleted:    target/classes/META-INF/maven/org.example/HealthApp_1/pom.properties
        deleted:    target/classes/META-INF/maven/org.example/HealthApp_1/pom.xml
        modified:   target/classes/coreUtilities/utils/FileOperations.class
        modified:   target/classes/pages/yakshaHealthAppPages.class
        modified:   target/test-classes/testBase/AppTestBase.class
        modified:   target/test-classes/testcases/yakshaHealthAppTestCase.class

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .classpath
        .gitignore
        .project
        .settings/

no changes added to commit (use "git add" and/or "git commit -a")
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it
[main 75dd591] First commit
 13 files changed, 75 insertions(+), 151 deletions(-)
 create mode 100644 .classpath
 create mode 100644 .gitignore
 create mode 100644 .project
 create mode 100644 .settings/org.eclipse.core.resources.prefs
 create mode 100644 .settings/org.eclipse.jdt.core.prefs
 create mode 100644 .settings/org.eclipse.m2e.core.prefs
 delete mode 100644 target/classes/META-INF/maven/org.example/HealthApp_1/pom.properties
 delete mode 100644 target/classes/META-INF/maven/org.example/HealthApp_1/pom.xml
```

```
@echo off

REM Path to your batch file to
execute
set
"batch_file_path=git_commands.bat
"

REM Interval in seconds (e.g., 60
seconds = 1 minute)
set interval=60

:loop
echo Executing batch file at %
DATE% %TIME%
call "%batch_file_path%"

REM Waiting specified interval
before next execution
timeout /t %interval% /nobreak >
NUL

goto loop
```

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

1. Open the Visual Studio code.

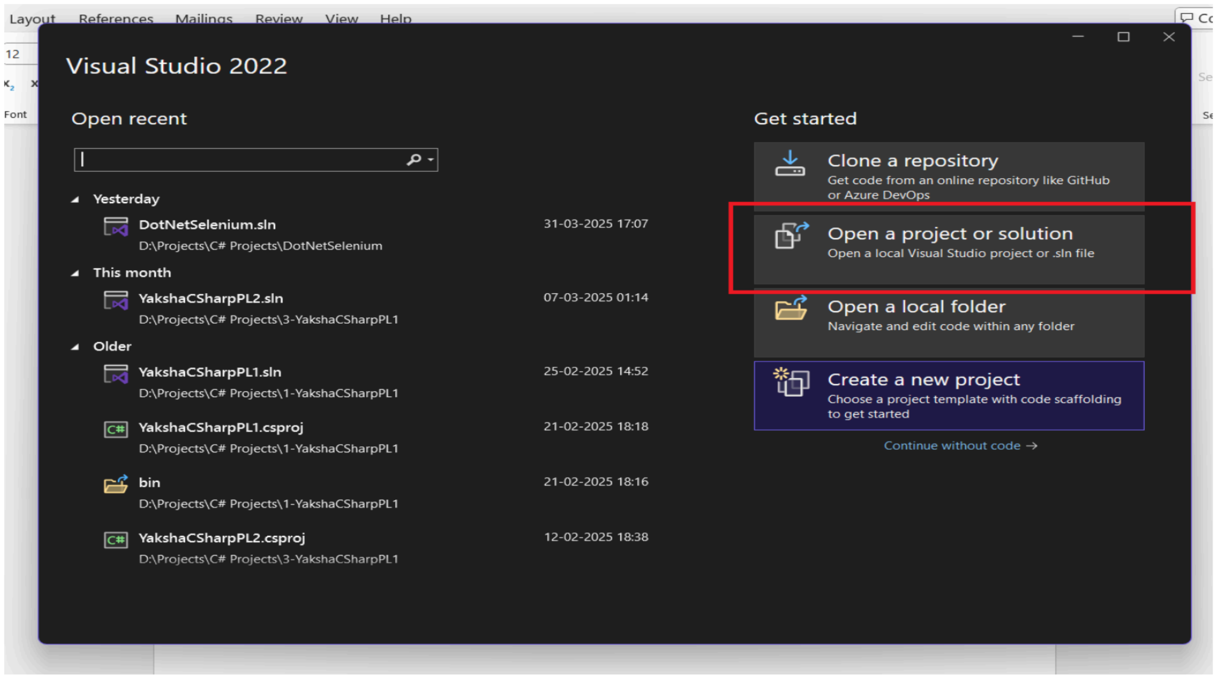


```
Enumerating objects: 212, done.
Counting objects: 100% (212/212), done.
Delta compression using up to 4 threads
Compressing objects: 100% (94/94), done.
Writing objects: 100% (201/201), 17.55 MiB | 4.31 MiB/s, done.
Total 201 (delta 100), reused 153 (delta 90), pack-reused 0
remote: Resolving deltas: 100% (100/100), completed with 6 local objects.
To https://github.com/IIHTDevelopers/0904_35gcom_1744214384385.git
   734bfa3..8d9e7eb  main -> main
Executing batch file at 09-04-2025 21:55:12.23
On branch main
```

```
@echo off

REM Path to your batch file to
execute
set
"batch_file_path=git_commands.bat
"

REM Interval in seconds (e.g., 60
seconds = 1 minute)
set interval=60

:loop
echo Executing batch file a
DATE% %TIME%
call "%batch_file_path%"

REM Waiting specified inter
before next execution
timeout /t %interval% /nobr
NUL

goto loop
```

Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed 9 Apr 9:56PM
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
DotNetSelenium
Debug
Any CPU
All Apps Documents Web More
Recommended
Your AI-powered Copilot
Say goodbye to endless search results!
Try it now
Top apps
File Explorer
Notepad
Services
Command Prom...
Firefox
Microsoft Edge
Recent
Visual Studio 2022
Services
Run
Trending searches
ENG US English (United States) US keyboard
ENG IN English (United States) English (India) keyboard
ENG English (India) English (India) keyboard
Type here to search

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



Visual Studio 2022

Open recent

Yesterday

DotNetSelenium.sln — D:\Projects\C# Projects\DotNetSelenium — 31-03-2025 17:07

This month

YakshaCSharpPL2.sln — D:\Projects\C# Projects\3-YakshaCSharpPL1 — 07-03-2025 01:14

Older

YakshaCSharpPL1.sln — D:\Projects\C# Projects\1-YakshaCSharpPL1 — 25-02-2025 14:52

YakshaCSharpPL1.csproj — D:\Projects\C# Projects\1-YakshaCSharpPL1 — 21-02-2025 18:18

bin — D:\Projects\C# Projects\1-YakshaCSharpPL1 — 21-02-2025 18:16

YakshaCSharpPL2.csproj — D:\Projects\C# Projects\3-YakshaCSharpPL1 — 12-02-2025 18:38

Get started

Clone a repository
Get code from an online repository like GitHub or Azure DevOps

Open a project or solution
Open a local Visual Studio project or .sln file

Open a local folder
Navigate and edit code within any folder

Create a new project
Choose a project template with code scaffolding to get started

Continue without code →

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	DispensaryPage.cs DoctorPage.cs LoginPage.cs MaternityPage.cs MedicalRecordsPage.cs PharmacyPage.cs RadiologyPage.cs SettingsPage.cs SubStorePage.cs (Each 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 below "Key Activities") 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.
/TestData	Doctor.json Maternity.json MedicalRecord.json PatientName.json Pharmacy.json Radiology.json Settings.json SubStore.json ValidLogin.json	It contains data for login and filling in the form.

PROBLEM STATEMENT

We need to automate the following activities using Selenium + C#.

Before the execution of any test steps, the user must be logged in to the health app.

Key Activities:

#	Summary	Action	Expected Result
1	PharmacyPage.cs ================== To validate that alerts are triggered and properly handled during the process of printing a Good Receipt without entering required fields like Supplier and Invoice Number.	1. Navigate to Pharmacy → Order. 2. Click Add New Good Receipt. 3. Attempt to print the receipt without filling mandatory fields. 4. Handle alerts for missing fields.	Alert messages such as "Please select supplier" and "Please enter Invoice no." should appear and be handled correctly.
2	PharmacyPage.cs ================== To ensure the successful creation and printing of a Good Receipt by adding item details and supplier information, and verifying the confirmation message.	1. Navigate to Pharmacy → Order. 2. Click Add New Good Receipt. 3. Add an item (name, batch number, quantity, rate). 4. Enter Supplier Name and Invoice Number. 5. Click Print Receipt. 6. Validate the success message.	A success message: "Goods Receipt is Generated and Saved." should be displayed.
3	MedicalRecordsPage.cs =================== To verify that filtering outpatient medical records by gender and date works correctly.	1. Load From Date and Gender from MedicalRecord.json. 2. Navigate to Medical Records → Outpatient List. 3. Apply date filter and click OK. 4. Enter Gender in the search bar and press Enter. 5. Validate that each record's Gender matches the input.	All displayed records should match the selected gender. If any mismatch is found, the test will fail.
4	MedicalRecordsPage.cs =================== To verify that filtering outpatient medical records by doctor name and date range works correctly and returns only relevant results.	1. Load Doctor Name and From Date from MedicalRecord.json. 2. Navigate to Medical Records → Outpatient List. 3. Enter Doctor Name and From Date, then click OK. 4. Validate that each result lists the correct Doctor Name.	All displayed records must be associated with the selected doctor. Any mismatches should trigger a failure.
5	SettingsPage.cs ============ To verify the functionality for creating a dynamic template in the Settings module.	1. Load template data from settings.json. 2. Navigate to Settings → Dynamic Templates. 3. Click Add New Template. 4. Select Template Type. 5. Fill in Template Name, Template Code, and Text Field content. 6. Switch to the Rich Text Editor iframe and enter text. 7. Click Add to save the template.	The template should be successfully created and saved without any errors.
6	SubStorePage.cs ================= To verify that a user can successfully create an inventory requisition in the Ward Supply → Substore module.	1. Load test data from SubStore.json. 2. Navigate to Ward Supply → Accounts → Inventory Requisition. 3. Click Create Requisition. 4. Select Target Inventory and enter Item Name. 5. Submit the requisition.	A success message should confirm the requisition was generated and saved successfully.
7	MaternityPage.cs ================ To verify that the Maternity Allowance Report loads correctly after entering a valid "From Date" and clicking Show Report.	1. Load the From Date from Maternity.json. 2. Navigate to the Maternity module. 3. Open the Reports section. 4. Select Maternity Allowance Report. 5. Enter the From Date. 6. Click Show Report. 7. Wait for the report data grid to appear.	The data grid (report content) should be visible, confirming that the report has been generated based on the input date.

8	DoctorPage.cs ============== To verify that an inpatient imaging order can be successfully placed from the Doctors module, and that the process completes with a confirmation message.	1. Load test data from Doctor.json for patient name, dropdown option, and search order item. 2. Navigate to the Doctors module. 3. Click the Inpatient Department tab. 4. Search for a specific patient. 5. Click on the Imaging action. 6. Select the Order Type from the dropdown. 7. Enter and search the Order Item. 8. Click Proceed. 9. Click Sign. 10. Verify the success message is displayed.	A success message: "Imaging and lab order added successfully" should appear, confirming the imaging order was placed successfully.
9	RadiologyPage.cs =============== To confirm that the **radiology request dates** shown in the system fall within the **last 3 months**.	1. Navigate to Radiology → List Requests. 2. Open the date range filter dropdown. 3. Select Last 3 Months and click OK. 4. Validate that each date in the results grid: – Matches the format yyyy-MM-dd – Falls within the last 90 days.	All dates should be correctly formatted and lie between today and 90 days prior. If not, the test should throw a detailed error.
10	DispensaryPage.cs =================== To verify that the User Collection Report in the Dispensary module can be exported successfully and downloaded with the expected file name.	1. Navigate to the Dispensary module by clicking the relevant link. 2. Click on the Reports tab. 3. Select the User Collection Report option. 4. Enter the From Date (e.g., "01-01-2020"). 5. Click on Show Report to generate the report. 6. Click on the Export to Excel button. 7. Wait and verify if the file gets downloaded in the Downloads folder.	A file named (or containing) "PharmacyUserwiseCollectionReport_2025" should be successfully downloaded into the user's Downloads directory.
11	SubStorePage.cs =============== To confirm that a new consumption entry can be created in the Ward Supply module.	1. Load the Item Name from SubStore.json. 2. Navigate to Ward Supply → Accounts → Inventory Consumption. 3. Click New Consumption. 4. Add the item and save. 5. Verify the success message.	A success message: "Consumption completed" should be visible after submitting the form.
12	SubStorePage.cs ================= To verify that reports in the Ward Supply module are correctly generated based on a selected Subcategory Item and date filter.	**1. Load Item Name from SubStore.json.** **2. Navigate to Ward Supply → Accounts → Reports.** **3. Click Consumption Report.** **4. Set the From Date and select Item from the subcategory dropdown.** **5. Click Show Report.** **6. Validate that the item name appears in the report grid.**	The selected Item Name should appear in the report results grid under the SubCategoryName column.
13	PharmacyPage.cs ================== To verify that entering a Supplier Name in the Order section filters the results correctly and displays the expected supplier in the result grid.	1. Navigate to Pharmacy → Order. 2. Input a Supplier Name in the filter. 3. Click Show Details. 4. Verify that the supplier appears in the grid under SupplierName.	The specified supplier name should be present in the result grid under the SupplierName column.
14	RadiologyPage.cs =============== To ensure that the Radiology List Requests can be successfully filtered by a custom date range and a specific imaging type.	**1. Load Filter Option, From Date, and To Date from Radiology.json.** **2. Navigate to Radiology → List Requests.** **3. Select the filter option and input the dates.** **4. Click OK.** **5. Check that all results have matching imaging type.**	Only records with the specified Imaging Type should be displayed in the results grid. If not, the test throws an error.
15	LoginPage.cs ============ To verify that a user can successfully log in using valid credentials from the ValidLogin.json file.	**1. Load valid username and password from JSON.** **2. Enter the credentials in the login form.** **3. Click the Login button.** **4. Verify the Admin element is visible.**	User is logged in successfully and the Admin dropdown becomes visible.

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

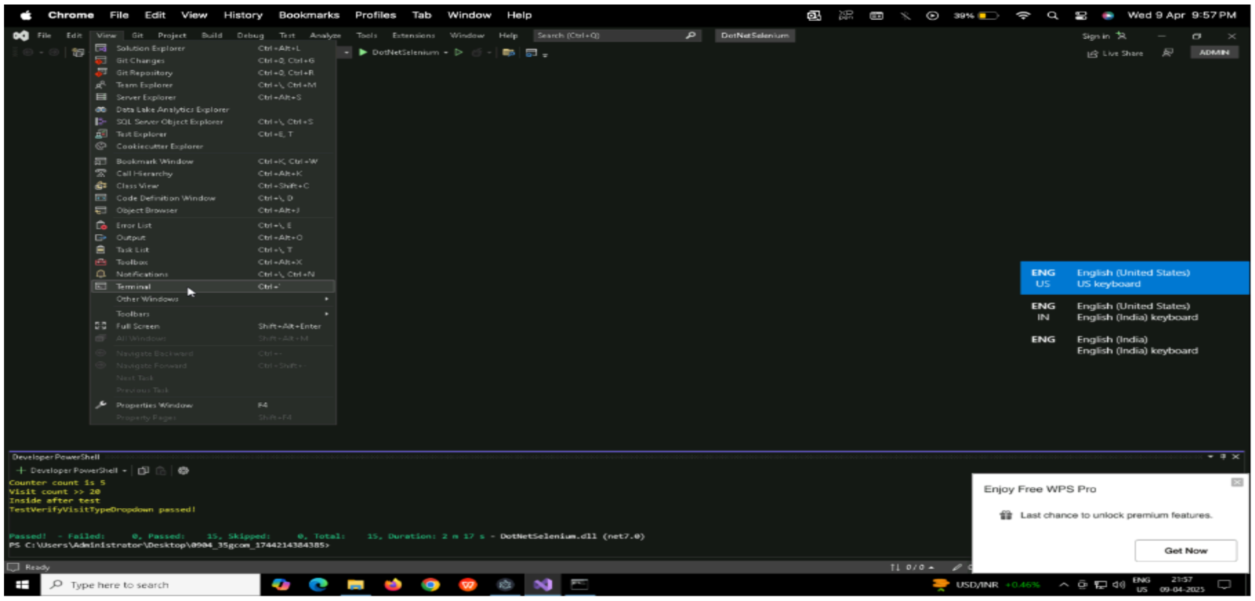
1.1 CODE QUALITY/OPTIMIZATIONS

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

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

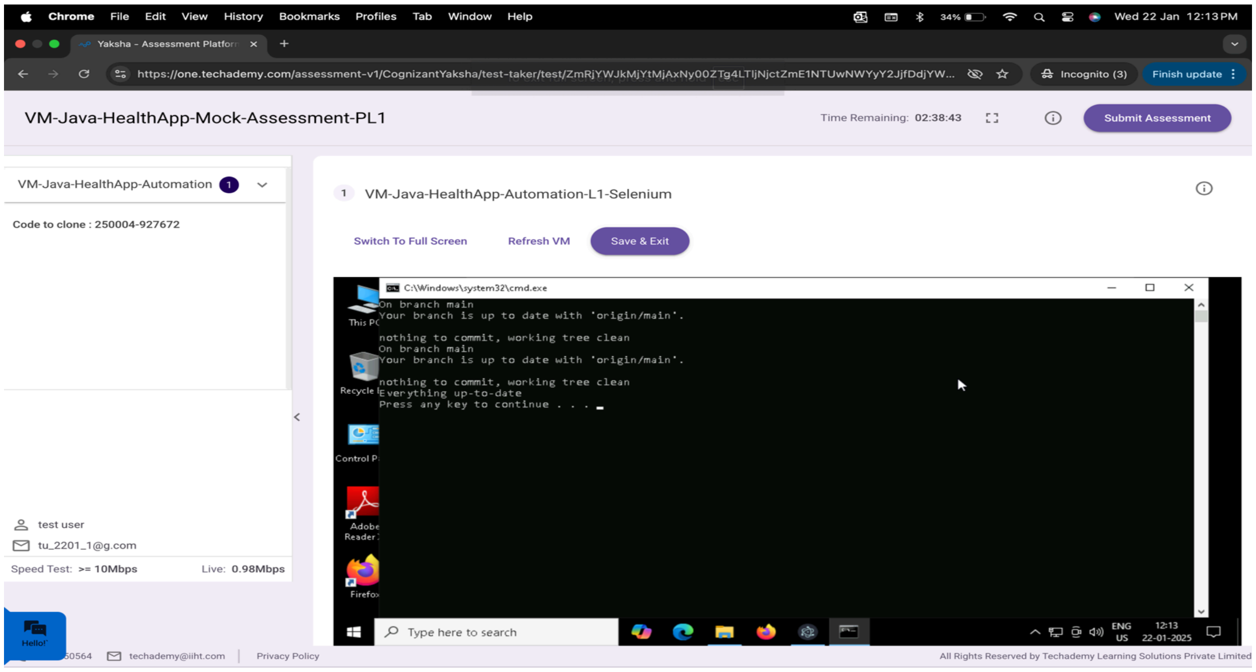


```
Counter count is 5
Visit count >> 20
Inside after test
TestVerifyVisitTypeDropdown passed!

Passed!  - Failed:     0, Passed:    15, Skipped:     0, Total:    15, Duration: 2 m 17 s - DotNetSelenium.dll (net7.0)
PS C:\Users\Administrator\Desktop\0904_35gcom_1744214384385>
```

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.**



Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed 22 Jan 12:13 PM
Yaksha - Assessment Platform
https://one.techademy.com/assessment-v1/CognizantYaksha/test-taker/test/ZmRjYWJkMjYtMjAxNy00ZTg4LTljNjctZmE1NTUwNWYyY2JjfDdjYW...
Incognito (3)
Finish update
VM-Java-HealthApp-Mock-Assessment-PL1
Time Remaining: 02:38:43
Submit Assessment
VM-Java-HealthApp-Automation 1
Code to clone : 250004-927672
1 VM-Java-HealthApp-Automation-L1-Selenium
Switch To Full Screen
Refresh VM
Save & Exit
C:\Windows\system32\cmd.exe
This PC
Recycle Bin
Control Panel
Adobe Reader
Firefox
test user
tu_2201_1@g.com
Speed Test: >= 10Mbps
Live: 0.98Mbps
Hello!
Type here to search
ENG US
12:13 22-01-2025
techademy@iiht.com
Privacy Policy
All Rights Reserved by Techademy Learning Solutions Private Limited

```
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
Everything up-to-date
Press any key to continue . . .
```

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

===

All the Best