

MONEY CONTROL AUTOMATION

YAKSHA

Time To Complete: 3 hrs

CONTENTS

1	Problem Statement	2
2	Business Requirements:	2
○		3
3	Implementation/Functional Requirements	3
3.1	Code Quality/Optimizations	3
1.	Template Code Structure	3
○	a. Package: com.iiht.automation.evaluate	3
○	b. Package: com.iiht.automation.evaluate	3
○	c. Package: com.iiht.automation.evaluate	4
4	Execution Steps to Follow	5

1 PROBLEM STATEMENT

Need to automate the following activities using Selenium + Java

2 BUSINESS REQUIREMENTS:

Navigate to <https://www.moneycontrol.com/>

1. Mouse hover on Personal Finance and click on "Tools" under Explore.
2. In Tools page, click on "Home Loan EMI Calculator".
3. Enter the load amount as "3000000".
4. Enter the load period as 20.
5. Choose "At the time of load disbursement" in "EMI Starts from" dropdown.
6. Enter interest rate as "10.5".
7. Enter Upfront charges as "10000".
8. Click on Submit.
9. Print Total payment and EMI in console
10. Under load payment schedule table, print the following details 7th year.
 - a. EMI Payment in the Year
 - b. Interest Payment in the year
 - c. Principal payment in the year.
11. Print "Outstanding Principal at end of year" value for 5th year in the table.

Expectations:

- 1) Campus Associates should write automation script using Java and selenium to automate all the steps in the above question. In other words, automation script should perform all mentioned steps.
- 2) The campus associate should not use any tools to create the xpath. They should develop the xpath/cssselector on their own.

3 IMPLEMENTATION/FUNCTIONAL REQUIREMENTS

3.1 CODE QUALITY/OPTIMIZATIONS

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

1. TEMPLATE CODE STRUCTURE

○ A. PACKAGE: COM.IIHT.AUTOMATION.EVALUATE

Resources

Class/Interface	Description	Status
App.java(class)	This represents bootstrap class i.e class with Main method, that shall contain all console interaction with user.	Implemented

○ B. PACKAGE: COM.IIHT.AUTOMATION.EVALUATE

Resources

Class/Interface	Description	Status
Activities.java(class)	This represents core activities. Following methods are there:	Not Implemented
Subject.java(class)	navigateToHomeLoanEMI()	Implement to automate the navigation to Loan Calculator page. This method must call mouseOverPersonalFinance() method from UITest class.
	calculateHomeLoanEMI()	Implement this method to automate calculation of Home Loan. This method must call following methods from UITest class: UITest.enterValueLoanAmount()

		UITest.selectEMIStartsFrom() UITest.enterInterestRate() UITest.enterValueLoanPeriod() UITest.enterValueUpfrontCharges() UITest.clickSubmit()
	getTotalPayment()	Implement this to get total payment
	getEMI()	Implement this to get EMI
	getTableDetails7thYearEMIPaymentInTheYear()	Implement this to get EMI Payment in the 7 th Year
	getTableDetails7thYearInterestPaymentInTheYear()	Implement this to get Interest Payment in the 7 th Year
	getTableDetails7thYearPrincipalPaymentInTheYear()	Implement this to get Principal Payment in the 7 th Year
	getTableDetails5thYearOutstandingPrincipalAtTheEndOfYear()	Implement this to get Outstanding Principal at the end of 5 th Year

○ C. **PACKAGE: COM.IIHT.AUTOMATION.EVALUATE**

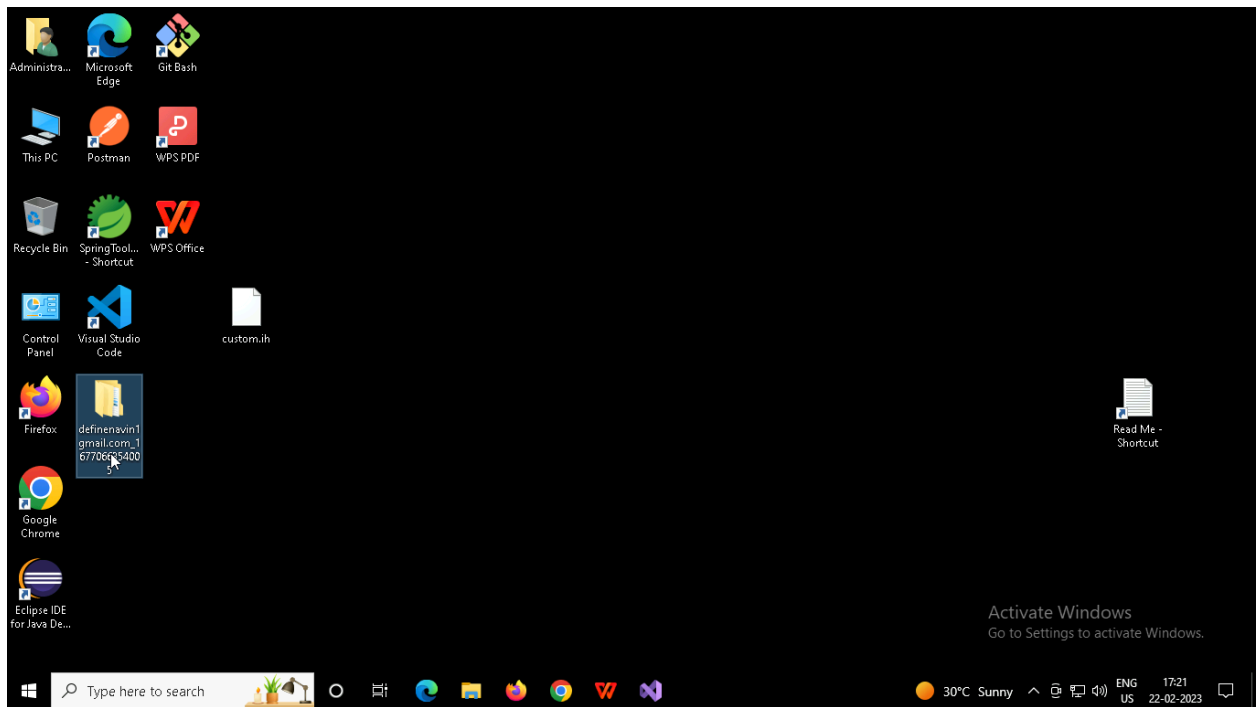
Resources

Class/Interface	Description	Status
UITest	This is a class containing helper methods. Following methods needs to be implemented	Not Implemented
EntityDaoImpl.java	mouseOverPersonalFinance()	Implement this method to automate mouseover on Personal Finance menu. It must call clickTools() method of UITest
	clickTools()	Implement this method to automate clicking on Tools option. It must call clickOnHomeLoanEMIC

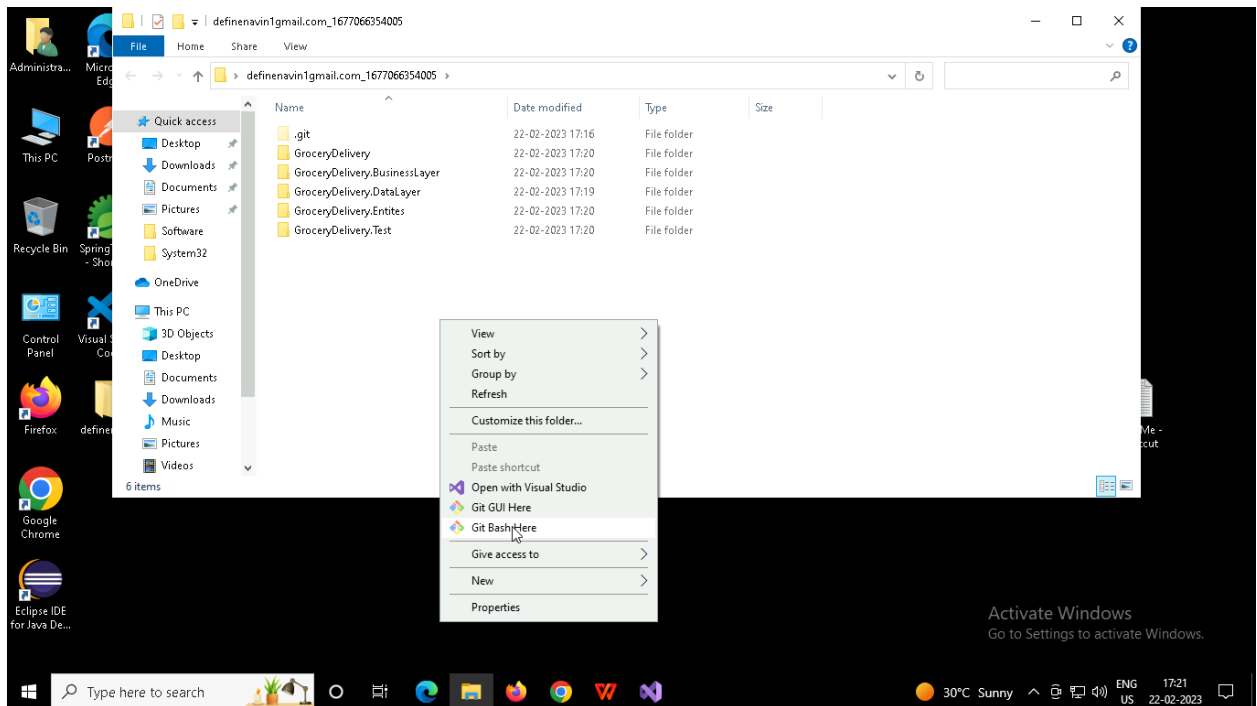
		calculator() method of UITest
	clickOnHomeLoanEMICalculator()	Implement this method to automate activating the EMI calculator.
	enterValueLoanAmount()	Implement this method to automate entering Loan Amount
	enterValueLoanPeriod()	Implement this method to automate entering Loan Period
	selectEMISStartsFrom()	Implement this method to automate entering EMI start option
	enterInterestRate()	Implement this method to automate entering interest rate
	enterValueUpfrontCharges()	Implement this method to automate entering upfront value
	clickSubmit()	Implement this method to automate clicking submit button

4 EXECUTION STEPS TO FOLLOW

1. You are mandatory required to run test cases for both the applications before final submission. Without which project evaluation will not happen
2. You can run the Junit test cases using Eclipse menu options
3. Before final submission, you are also required to push your code to GIT. Following are the steps to follow:

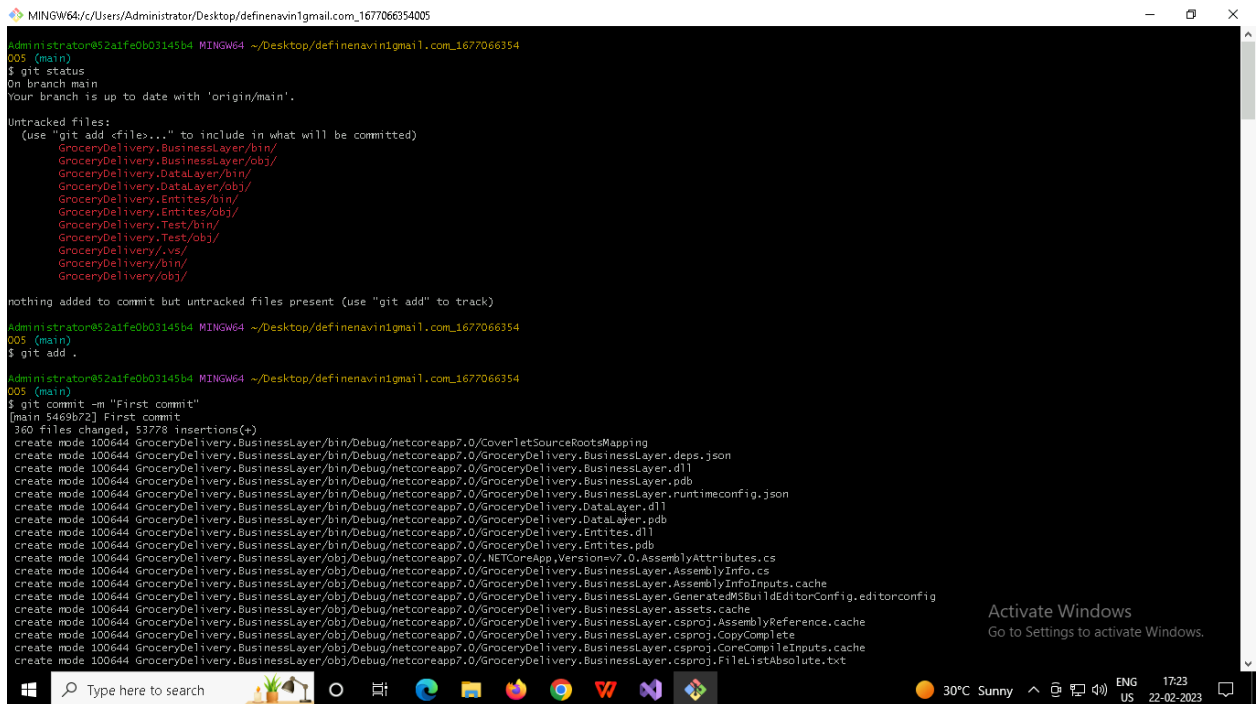


Right click in folder and open Git Bash



In Git bash terminal, run following commands

- git status
- git add .
- git commit -m "First commit"
(You can provide any message every time you commit)
- git push



```
Administrator@52a1fe0b03145b4 MINGW64 ~/Desktop/definavin@gmail.com_1677066354005 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    GroceryDelivery.BusinessLayer/bin/
    GroceryDelivery.BusinessLayer/obj/
    GroceryDelivery.DataLayer/bin/
    GroceryDelivery.DataLayer/obj/
    GroceryDelivery.Entities/bin/
    GroceryDelivery.Entities/obj/
    GroceryDelivery.Test/bin/
    GroceryDelivery.Test/obj/
    GroceryDelivery/.vs/
    GroceryDelivery/bin/
    GroceryDelivery/obj/

nothing added to commit but untracked files present (use "git add" to track)
Administrator@52a1fe0b03145b4 MINGW64 ~/Desktop/definavin@gmail.com_1677066354005 (main)
$ git add .
Administrator@52a1fe0b03145b4 MINGW64 ~/Desktop/definavin@gmail.com_1677066354005 (main)
$ git commit -m "First commit"
[main 5469b72] First commit
360 files changed, 53778 insertions(+)
create mode 100644 GroceryDelivery.BusinessLayer/bin/Debug/netcoreapp7.0/CoverletSourceRootsMapping
create mode 100644 GroceryDelivery.BusinessLayer/bin/Debug/netcoreapp7.0/GroceryDelivery.BusinessLayer.deps.json
create mode 100644 GroceryDelivery.BusinessLayer/bin/Debug/netcoreapp7.0/GroceryDelivery.BusinessLayer.dll
create mode 100644 GroceryDelivery.BusinessLayer/bin/Debug/netcoreapp7.0/GroceryDelivery.BusinessLayer.pdb
create mode 100644 GroceryDelivery.BusinessLayer/bin/Debug/netcoreapp7.0/GroceryDelivery.BusinessLayer.runtimeconfig.json
create mode 100644 GroceryDelivery.BusinessLayer/bin/Debug/netcoreapp7.0/GroceryDelivery.DataLayer.dll
create mode 100644 GroceryDelivery.BusinessLayer/bin/Debug/netcoreapp7.0/GroceryDelivery.DataLayer.pdb
create mode 100644 GroceryDelivery.BusinessLayer/bin/Debug/netcoreapp7.0/GroceryDelivery.Entities.dll
create mode 100644 GroceryDelivery.BusinessLayer/bin/Debug/netcoreapp7.0/GroceryDelivery.Entities.pdb
create mode 100644 GroceryDelivery.BusinessLayer/obj/Debug/netcoreapp7.0/.NETCoreApp,Version=v7.0.AssemblyAttributes.cs
create mode 100644 GroceryDelivery.BusinessLayer/obj/Debug/netcoreapp7.0/GroceryDelivery.BusinessLayer.AssemblyInfo.cs
create mode 100644 GroceryDelivery.BusinessLayer/obj/Debug/netcoreapp7.0/GroceryDelivery.BusinessLayer.AssemblyInfoInputs.cache
create mode 100644 GroceryDelivery.BusinessLayer/obj/Debug/netcoreapp7.0/GroceryDelivery.BusinessLayer.GeneratedMSBuildEditorConfig.editorconfig
create mode 100644 GroceryDelivery.BusinessLayer/obj/Debug/netcoreapp7.0/GroceryDelivery.BusinessLayer.assets.cache
create mode 100644 GroceryDelivery.BusinessLayer/obj/Debug/netcoreapp7.0/GroceryDelivery.BusinessLayer.csproj.AssemblyReference.cache
create mode 100644 GroceryDelivery.BusinessLayer/obj/Debug/netcoreapp7.0/GroceryDelivery.BusinessLayer.csproj.CopyComplete
create mode 100644 GroceryDelivery.BusinessLayer/obj/Debug/netcoreapp7.0/GroceryDelivery.BusinessLayer.csproj.CoreCompileInputs.cache
create mode 100644 GroceryDelivery.BusinessLayer/obj/Debug/netcoreapp7.0/GroceryDelivery.BusinessLayer.csproj.FileListAbsolute.txt
```

