Stage 1 - C#

Stage 2 - Web Automation (Selenium & Playwright) & windows GUI (AutoIT & FLAUI) & API Automation (Restsharp)

Stage 3 - Hybrid Framework - NUnit Test Framework, Data Driven Framework, Page Object Model, Keyword Driven Framework

Stage 4 - BDD Framework

Stage 5 - CI (Git & Jenkins)

Selenium:

* Automates only web applications
* Open Source
* Language Independency - Java, C#, Python, Javascript, Ruby

Selenium

1. Selenium IDE
   1. No Programming knowledge is required.
   2. Record and playback feature
   3. Plugins - chrome, firefox, edge
   4. Use it only exploratory or simple scripting
2. Selenium RC - Depreciated
   1. Programming knowledge is required.
   2. Architecture

Source code (C#+Selenium RC)--> RC server (Turn On/OFF) → Browser

1. Selenium WebDriver
   1. Programming knowledge is required.
   2. Architecture

Source code (C#+Selenium WebDriver) → Browser

1. Selenium Grid
   1. If you want to scale by distributing and running tests on several machines and manage multiple environments from a central point.

C# Programming

1. Installation
2. Architecture

Console app

Source code (.csproj) → .exe/..dll (MSIL) → O/P

Class Library

Source code (.csproj) → .dll → O/P

..dll (MSIL) → Platform independent code

1. UpperCamelCase - MyFirstProject

lowerCamelCase - myFirstProject

1. Create a Console app

Solution → UpperCamelCase

Project 1 → UpperCamelCase

Namespace → UpperCamelCase

Class (.cs) →UpperCamelCase

Methods → UpperCamelCase

variable → lowerCamelCase

1. Datatypes
   1. Pre-defined datatypes
   2. Non-predefined datatypes - a collection of pre-defined
      1. String
      2. Array
         1. Size is fixed
      3. User-defined datatypes
2. All whole number → int

Decimal values → double

1. Debugging
   1. Terminate
   2. Resume
   3. Step into
   4. Step over
2. Methods - building block of the program
   1. Reusability
   2. Maintenance

* Static methods
  + How to create and call it?

//accessmodifier static returntype methodname(arguments)

To call it

classname.Methodname()

* Non-static methods
  + How to create and call it?
    - To call it
      * Create object
      * Use objref.MethodName()

1. Variables
   1. Static variable
   2. Non-static variable
2. Static vs Non-Static
3. Class & Object
   1. Class - A class is template or blueprint or type from which objects are created
   2. Object
      1. An object is an instance of the class
      2. Every object has its own state (non static variable) and behaviour (non-static method)
4. Object
   1. Declaration
   2. Instantiation - new
   3. Initialization
5. Accessmodifier
   1. Private - accessible within the class
   2. Internal - accessible within the assembly
   3. Protected - accessible in the child class
   4. Protected internal -
   5. Public - accessible anywhere
6. Real time example of class and object
7. Encapsulation
8. Collections (Generic vs non-generic concept)
   1. List
   2. Dictionary (key-value pair)
9. Constructor - prerequisite to the object
   1. Constructor and classname should be same
   2. It is a kind of method with no return type
   3. there will be always a default constructor which gets called whenever you create object. To load the non-static variable with default values.
   4. You can override the default constructor by creating explicit constructor
      1. Without arguments
      2. With arguments
   5. If you create a constructor explicitly,, you need to call it on object creation
10. this —>
    1. this keyword - helps to distinguish between
    2. keyword points to current instance
11. Static Ploymorphism / Compile time / early time
12. Constructor overloading
    1. Can create multiple constructor same name. By change
       1. Number of parameters
       2. Datatypes of parameter
       3. Sequence of parameters
13. Method overloading

The method to be called is resolved during the compile time.

* 1. Can create multiple methods with the same name by change in
     1. Number of parameters
     2. Datatypes of parameter
     3. Sequence of parameters

1. Inhertiance
   1. Reuse methods, variables, properties
   2. Maintenance

Refrenece

1 GB - 1024 MB

1 MB - 1024 KB

1 KB - 1024 B

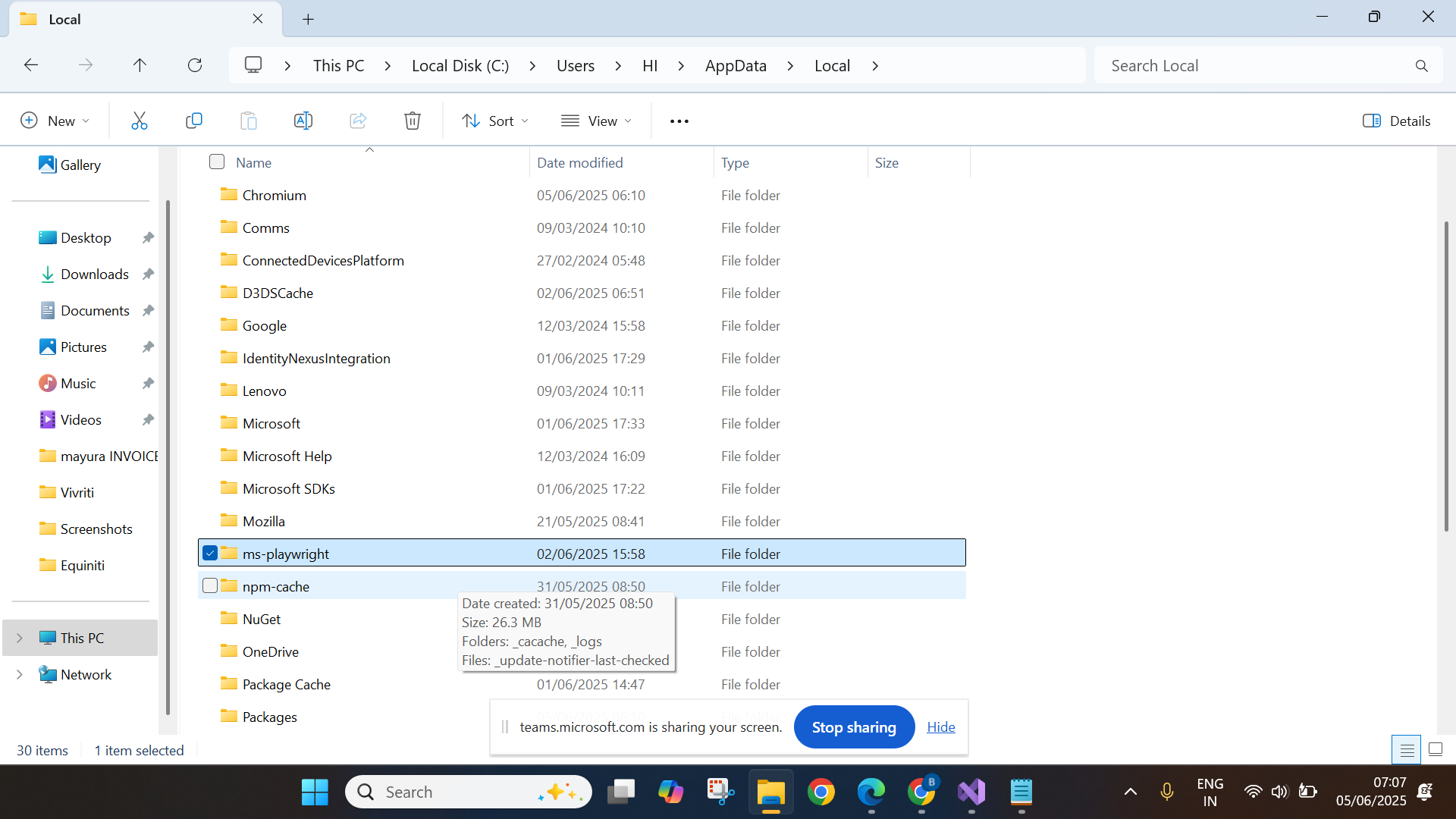
1 Byte - 8 bits

Working with playwright

1. Create Nunit test project with Playwright

Verify or add .dll

1. Nunit
2. Nunit adapter
3. Microsoft.Playwright.nunit
4. Microsoft.Playwright
5. Setup the browser for playwright
   1. pwsh bin/Debug/netX/**playwright.ps1** install



1. Create test methods
2. Browser launch
   1. Create playwright instance
   2. Browser instance
   3. Context
   4. Page (tab)
3. Navigate to url
4. Get the url, title
5. Click, type, select
6. Inspect → tagname, attribute, text or not
7. Using Locator method - supports
   1. Xpath
   2. Css
8. Wait for page load until networkidle
9. Dropdown
   1. With select tag
      1. Label
      2. Value
      3. Index -> 0
   2. Without select tag
10. Locator method using xpath, css and PageLocatoroption

await page.Locator("css=#SubscriptionAgreement").ClickAsync();

await page.Locator("xpath=(//div[@class='checkbox-ui'])[2]").ClickAsync();

//click on submit

await page.Locator("button", new PageLocatorOptions() { HasText = "Submit" }).ClickAsync();

1. Class in the arguments of FillAsync(), ClickAsync(), SelectOptionAsync(), Locator()
2. Multiple tabs/windows, Frame, Alert
3. Mutliple tabs/windows
   1. Handling the newly opened tab
   2. Handle using page.Context.Pages
   3. RunAndWaitForPopupAsync
4. Frame
   1. Even though locator is correct, we get no locator present

System.TimeoutException : Timeout 30000ms exceeded.

Call log:

- waiting for Locator("//input[@name='fldLoginUserId']")

* 1. Check for tagname - frame or iframe
  2. Get frame control using - (anyone option)
     1. Frame(“name”)
     2. FrameByUrl(“url”)
     3. FrameLocator(“selector”) (xpath or css)
     4. Using page.Frames (List)
     5. FrameByUrl(Regex)

1. Alert
2. Upload the file
   1. Direct method
   2. Event handler
   3. RunwithFileChooser
3. Shadowroot element (xpath does not work)
4. Dynamic Tables
5. Get locators (optional)
6. Collection of elements

BDD - Behaviour Driven Development

BDD – Main focus - understanding the requirement

ATDD – Acceptance Test Driven Development - Main Focus - Writing the acceptance testing

Where to implement actual BDD?

1. Everyone knows about the requirement
2. Some of the team members knows about the requirement
3. Only top-level managements know the requirement
4. Some other organization done it, it is new to your organizationn
5. Need to conduct R&D and then need to develop.

Implement the BDD Framework

1. C# - Specflow (ReqnRoll)
2. Java/Javascript - Cucumber
3. Python - jbehave

Scenario:

Given - Pre-requisite for that scenario

When - actions/operation

Then - Assertion/validation

Architecture of C# Reqnroll

Feature (.feature) → Step definition (.cs)

Steps to set up the framework

1. Create a project in a new solution
2. Add/Update Dependencies
   1. Reqnroll
   2. Playwright
3. Create Feature file
   1. Add Feature header
   2. Add Feature description (optional)
      1. Unformatted description
      2. Formatted description

In order [goal]

As a [role]

I would like to [visible change in application]

1. Scenario
   1. Add Scenario title
   2. Add at least one given, one when, one then.
2. Step parameterization
   1. Code reuse
   2. Reduce duplicate step definition
3. Scenario outline
   1. One scenario - run with multiple set of test data
   2. Reduce duplicates in the feature file
4. Background
   1. Use only when repeated given for all scenarios in a single file
5. Hooks
   1. BeforeScenario
   2. AfterScenario
6. Minimum one assertion should be there in the assertion. It decides whether the scenario or test method is pass or fail.
7. Datatable
8. Scenario Outline with Datatable
9. Dependency injection - Featuer Context and Scenario context
10. Page Object Model
    1. Reusablity
    2. Maintenance
    3. Reaability

To implement page object model

1. For each web page, we need to create a class - Page class
2. Operations/actions happens through the method - Page Methods
3. Collecting the object repository at class level or different file d
4. Base class for page object (keywords)
5. Non-static for all the field to support parallel execution.
6. DB connection

Important Concepts - To improve maintenance/readability

1. Step parameterization- code reuse
2. Scenario Outline - one scenario-multiple set of test data
3. Background - repeated given
4. Datatable - sending tabular data

—

Git - Git is a free and open-source **distributed version control system**

Architecture of git:

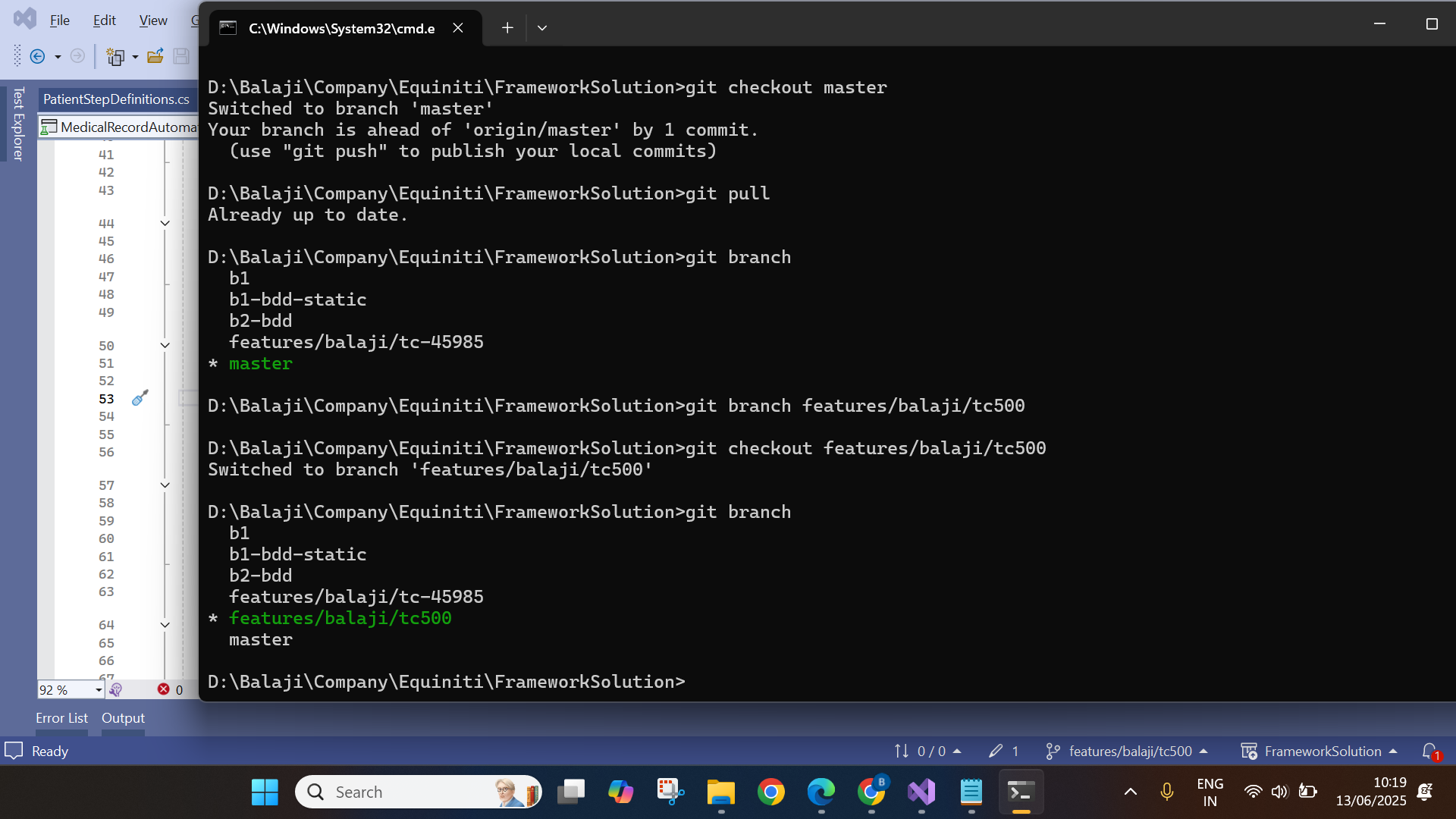
Project (local machine) → Local repository (local machine - for every project) → remote repo (github, bitbucket, aws code commit)

Git Concept

Modified - staged - commit

Steps to move to remote repor

1. git init - initialize the local repo
2. git add . → staging
3. git commit -m "first commit" – update the local repo
4. git remote add origin1 <https://github.com/balaji-githubstore/csharp-concepts-equiniti-jun-2025.git> -> register the remote url with name origin1
5. git push -u origin master



Playwight Reference:-

1. Direct methods
2. RunWith methods
3. Event handler

<https://www.gonitro.com/pdf-to-word?srsltid=AfmBOopca0uIxKiJKPkf3HHm6qPQbcdWLeRscNIY9uo563LuK2YSxeel>

Assignments

​​Contents - Assignments

***​***

​

***​***

​

​

​

***​***

​

​

​

**​​**

***Day 1***

Task 1

1. Create console app project
2. Create a class with name as “**Volume**” and namespace as CompanyName.Formulae
3. Create static method for below formula (any 4 methods)

| [Volume Of Sphere](https://byjus.com/maths/volume-of-sphere/) | [Volume Of A Cylinder](https://byjus.com/maths/volume-of-a-cylinder/) |
| --- | --- |
| [Volume Of A Pyramid](https://byjus.com/maths/volume-of-a-pyramid/) | [Volume Of Cone](https://byjus.com/maths/volume-of-cone/) |
| [Volume Of Cuboid](https://byjus.com/maths/volume-of-cuboid/) | [Volume Of Hemisphere](https://byjus.com/maths/volume-of-hemisphere/) |

1. Create a class with name as “**Runner**” and namespace as CompanyName.Formulae

Task 2

1. Create Student type under student package with below details (decide between static and non-static variable)

Student - attributes/state

•   studentRollno

•   studentName

•   studentMailid

•   studentPercentage

•   schoolName

•   schoolAddress

1. create a method to printStudentDetails
2. Create stud\_runner module under student\_runner package

Create 3 different instance for storing below values

1001,"jack",jack@gmail.com,45.2, Global school, chennai

1002,"peter",peter@gmail.com,85.2, Global school, chennai

1003,"mark",mark@gmail.com,56.5, Global school, chennai

Task 3

1. Create the Item class in the package amazon\_package

2. Declare public fields for ID (int), descr (String),

    quantity (int), price (double).

3. Create ShoppingCart class.

4. Declare and instantiate 2 objects of type Item. Initialize only

     the descr field in each, using different values for each.

5. Print the description for each item and run the code.

6. (Optional) Above the code that prints the descriptions,

     assign item2 to item1.  Run it again.

7. create a method in item class to print discounted price.

    If qty = 2 --> print the final price by providing 10% discount of price

    If qty = 3 to 5--> print the final price by providing 15% discount of price

    If qty >5 --> print the final price by providing 25% discount of price

    Example:

    price = 100

    qty = 2

    if qty==2

        final\_price = (price – (price\*10/100))\*qty

        print(final\_price)

***Day 2***

Task 1 (Important)

1. Navigate onto <https://www.salesforce.com/in/form/signup/freetrial-sales/>
2. Enter first name as “John”
3. Enter last name as “wick”
4. Enter work email as “john@gmail.com”
5. Select Job title as “IT Manager”
6. Select Employees as “101-500 employees”
7. Select country as “United Kingdom”
8. Do not fill the phone number
9. Click on check box
10. Click on start my free trial
11. Get the error message displayed “Enter a valid phone number”

Task 2 (Important) - Multiple tabs

1. Navigate onto https://www.online.citibank.co.in/
2. Close if any pop up comes
3. Click on My Account
4. Click on Banking with citi
5. In new tab
6. Enter userid as john123
7. Click on signup
8. Get the error displayed for password

Task 3 (Important)

1. Navigate onto https://www.medibuddy.in/
2. Close if any popup and Click on Login
3. Click on I have Corporate Account
4. Click on Learn More
5. Click on Skip
6. Click on Login using Username & Password
7. Enter username as john
8. Enter password as john123
9. Click on show password
10. Click log in
11. Get the error message shown and print it in terminal

Task 4

1. Navigate onto https://nasscom.in/
2. Click on Login and then click on register
3. Enter First name as admin
4. Enter Last name as pass
5. Enter email address as admin@gmail.com
6. Enter company name as Google
7. Select IT Consulting from dropdown
8. No need to automate CAPTCHA
9. Click on Register

Task 4

1. Navigate to<https://secure1.inmotionhosting.com/index/login>
2. Click on visit our support center
3. Click on “No Thanks” if any popup
4. Search for “diskspace”
5. Get the text “Search Results for: diskspace” and print it

--------------------------------------

Task 3 (Important)

1. Navigate onto http://demo.openemr.io/b/openemr/
2. Update username as admin
3. Update password as pass
4. Select language as English (Indian)
5. Click on the login button
6. Click on Patient Click New Search
7. Add the first name, last name
8. Update DOB as today's date

driver.findElement(By.id("form\_DOB")).sendKeys("2024-01-12");

1. Update the gender
2. . Click on the create new patient button above the form
3. . Click on confirm create new patient button.
4. . Print the text from alert box (if any error before handling alert add 5 sec wait)
5. . Handle alert
6. Close the Happy Birthday popup
7. Get the added patient name and print in the console.

Task 3 (Important) - shadowroot element

1. Navigate onto https://www.royalcaribbean.com/account/signin
2. Click Create an account
3. First Name as Dennis
4. Last Name as Rich
5. Select Month as April
6. Day as 4
7. Year as 1990
8. Country as India
9. Any email address and password
10. Select as “What was your first car's make or model?”
11. Type answer
12. Accept the terms and condition
13. Click done

Task 4

1. Navigate to <https://www.gonitro.com/pdf-to-word?srsltid=AfmBOoqdJQhPEuoros_GD8EhntByoW-opgz9DeVR5a60mrvMeR3z8byW>
2. Upload pdf using FileChooser
3. Download the word