DAY 10 :- EXTENT REPORT

TASK 01,02,03:-

package com.day10;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.chrome.ChromeOptions;

import org.testng.Assert;

import org.testng.ITestResult;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

import com.aventstack.extentreports.ExtentReports;

import com.aventstack.extentreports.ExtentTest;

import com.aventstack.extentreports.Status;

import com.aventstack.extentreports.reporter.ExtentHtmlReporter;

import com.aventstack.extentreports.reporter.configuration.ChartLocation;

import com.aventstack.extentreports.reporter.configuration.Theme;

import io.github.bonigarcia.wdm.WebDriverManager;

public class task1{

static ExtentTest test;

static ExtentReports extent;

ExtentHtmlReporter htmlReporter;

@BeforeClass

public void startReport() {

//ExtentSparkReporter htmlReporter = new ExtentSparkReporter("extentReport.html");

//htmlReporter = new ExtentHtmlReporter(System.getProperty("user.dir")+"/test-output/extentReport.html");

htmlReporter = new ExtentHtmlReporter("D:\\extr2.html");

extent = new ExtentReports();

extent.attachReporter(htmlReporter);

//configuration items to change the look and feel

//add content, manage tests etc

htmlReporter.config().setChartVisibilityOnOpen(true);

htmlReporter.config().setDocumentTitle("Simple Automation Report");

htmlReporter.config().setReportName("Test Report");

htmlReporter.config().setTestViewChartLocation(ChartLocation.TOP);

htmlReporter.config().setTheme(Theme.STANDARD);

// htmlReporter.config().setTimeStampFormat("EEEE, MMMM dd, yyyy, hh:mm a '('zzz')'");

}

@Test

public void Test1() {

WebDriverManager.chromedriver().setup();

WebDriver driver;

ChromeOptions co = new ChromeOptions();

co.addArguments("--remote-allow-origins=\*");

driver=new ChromeDriver(co);

driver.get("https://www.godaddy.com/en-in");

driver.manage().window().maximize();

String actualTitle="Domain Names, Websites, Hosting & Online Marketing Tools - GoDaddy IN";

String extractedTitle=driver.getTitle();

Assert.assertEquals(actualTitle, extractedTitle);

String actualURL="https://www.godaddy.com/en-in";

String extractedURL=driver.getCurrentUrl();

Assert.assertEquals(actualURL, extractedURL);

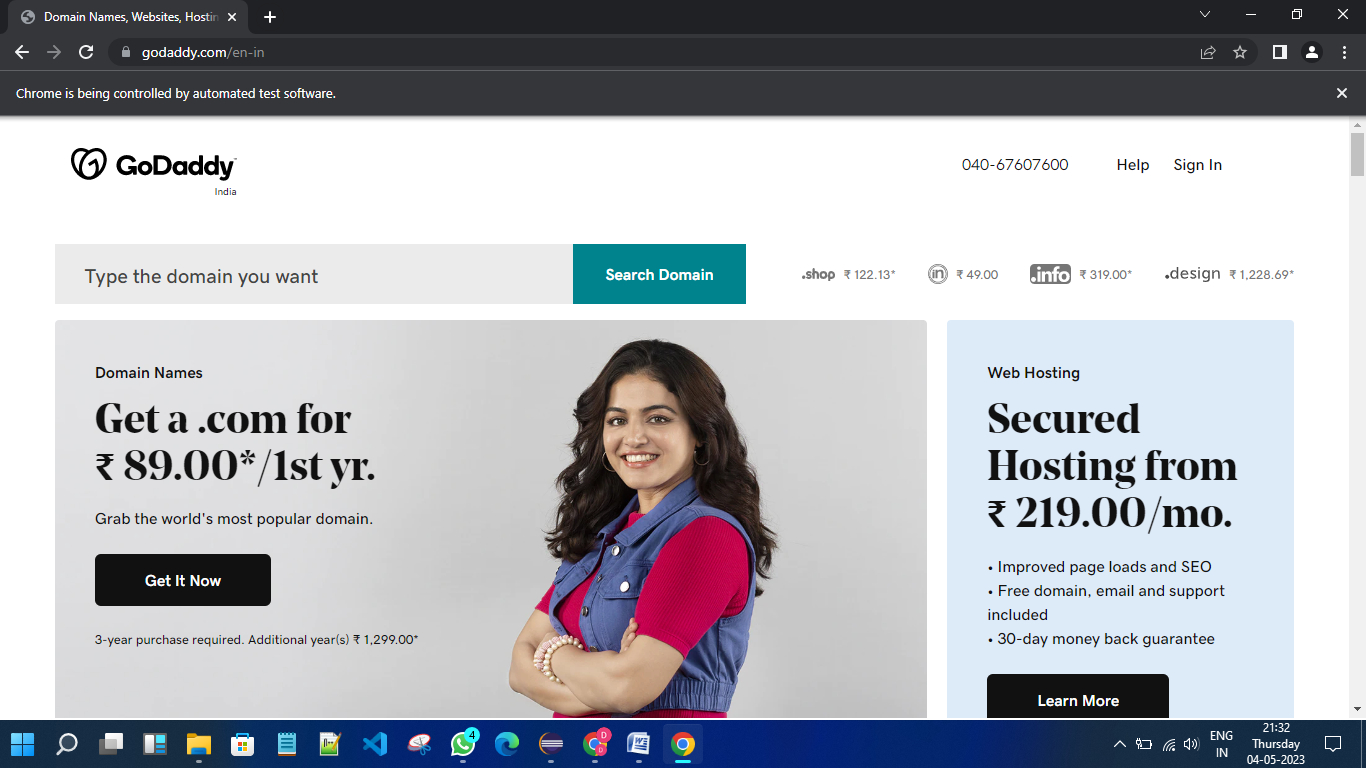
System.out.println("Smoke test");

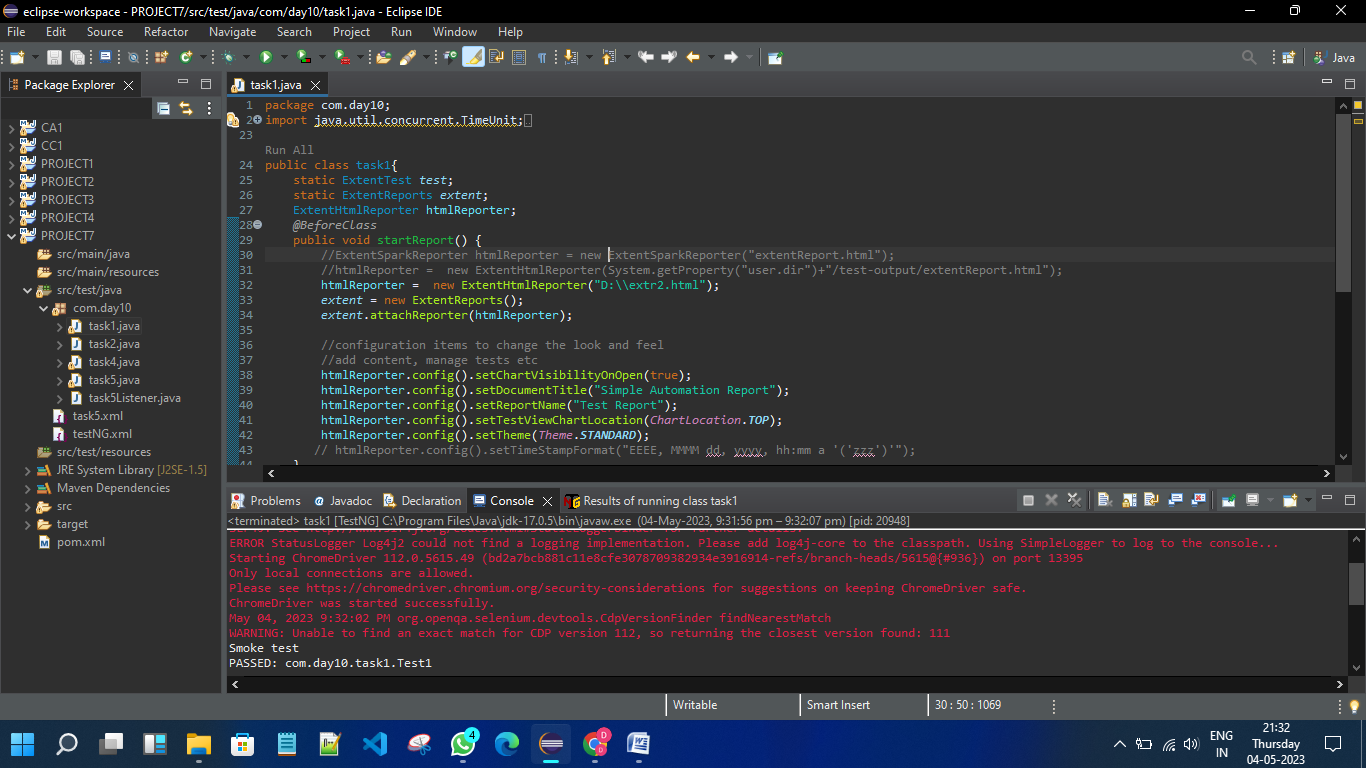
driver.quit();

test = extent.createTest("Test Case 1", "The test case 1 has passed");

}

}





package com.day10;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.chrome.ChromeOptions;

import org.testng.ITestResult;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

import com.aventstack.extentreports.ExtentReports;

import com.aventstack.extentreports.ExtentTest;

import com.aventstack.extentreports.Status;

import com.aventstack.extentreports.reporter.ExtentHtmlReporter;

import com.aventstack.extentreports.reporter.configuration.ChartLocation;

import com.aventstack.extentreports.reporter.configuration.Theme;

import io.github.bonigarcia.wdm.WebDriverManager;

public class task2 {

static ExtentTest test;

static ExtentReports extent;

ExtentHtmlReporter htmlReporter;

@BeforeClass

public void startReport() {

//ExtentSparkReporter htmlReporter = new ExtentSparkReporter("extentReport.html");

//htmlReporter = new ExtentHtmlReporter(System.getProperty("user.dir")+"/test-output/extentReport.html");

htmlReporter = new ExtentHtmlReporter("D:\\extr2.html");

extent = new ExtentReports();

extent.attachReporter(htmlReporter);

//configuration items to change the look and feel

//add content, manage tests etc

htmlReporter.config().setChartVisibilityOnOpen(true);

htmlReporter.config().setDocumentTitle("Simple Automation Report");

htmlReporter.config().setReportName("Test Report");

htmlReporter.config().setTestViewChartLocation(ChartLocation.TOP);

htmlReporter.config().setTheme(Theme.STANDARD);

// htmlReporter.config().setTimeStampFormat("EEEE, MMMM dd, yyyy, hh:mm a '('zzz')'");

}

@Test

public void Test2() throws Exception {

WebDriverManager.chromedriver().setup();

WebDriver driver;

ChromeOptions co = new ChromeOptions();

co.addArguments("--remote-allow-origins=\*");

driver=new ChromeDriver(co);

driver.get("https://www.godaddy.com/en-in");

driver.manage().window().maximize();

driver.navigate().refresh();

//driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);

//driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS);

Thread.sleep(10);

driver.findElement(By.xpath("//\*[@id=\"id-e6aaf13d-272e-44df-903a-e07e539d57c7\"]/span[1]")).click();

//driver.findElement(By.linkText("Domain Name Search")).click();

System.out.println(driver.getTitle());

System.out.println("Regression test");

test = extent.createTest("Test Case 2", "The test case 2 has passed");

//driver.quit();

}

@AfterMethod public void getResult(ITestResult result) {

if(result.getStatus() == ITestResult.FAILURE) {

test.log(Status.FAIL,result.getThrowable()); } else if(result.getStatus() ==

ITestResult.SUCCESS) { test.log(Status.PASS, result.getTestName()); } else {

test.log(Status.SKIP, result.getTestName()); } }

@AfterTest public void tearDown()

{

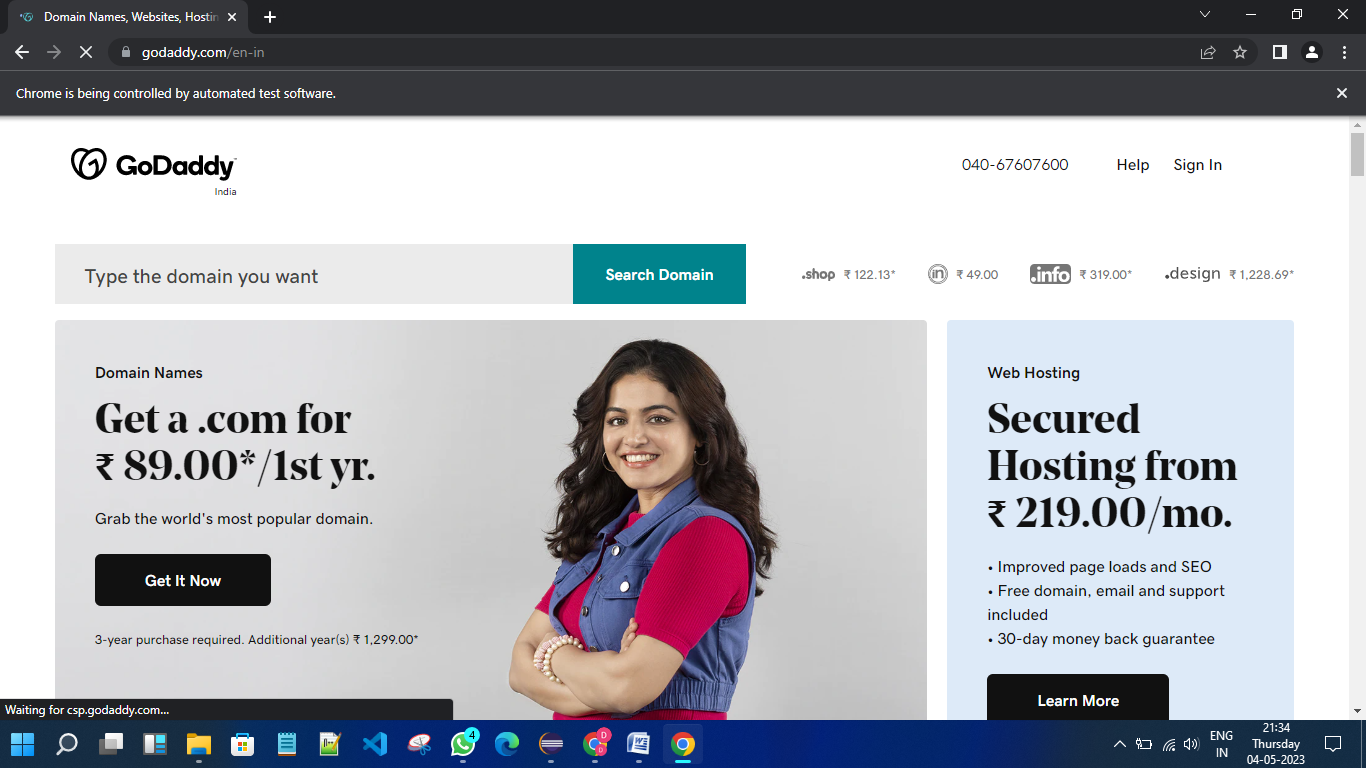
//to write or update test information to

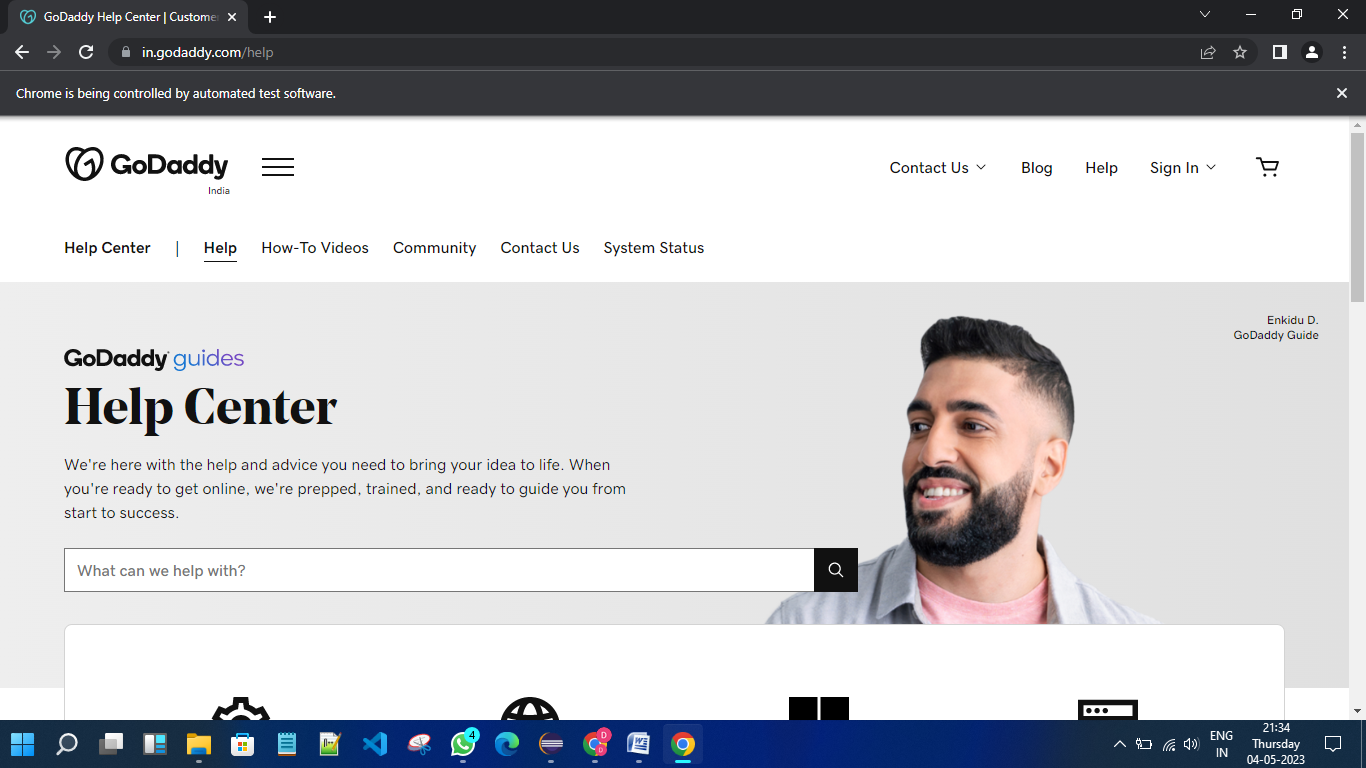
//reporter

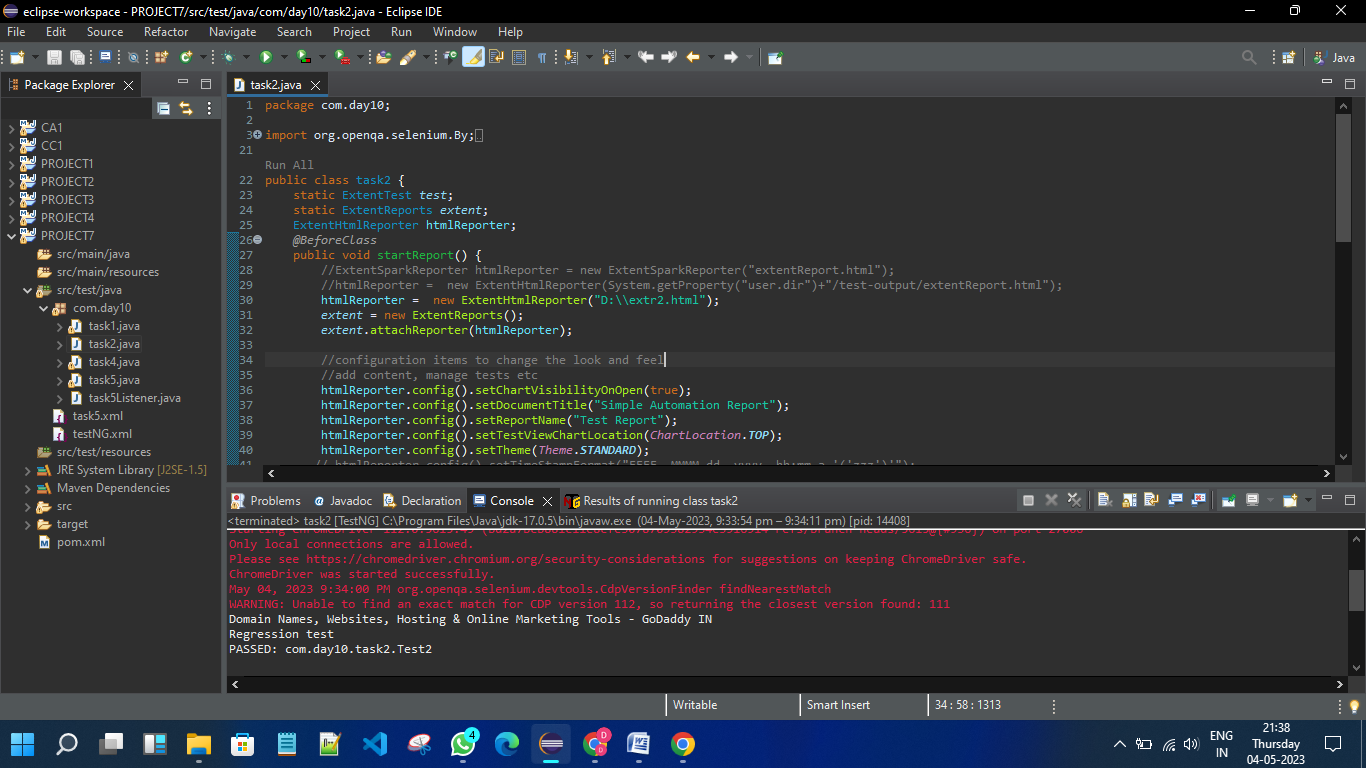
extent.flush();

}

}







**TASK 04:-**

package com.day10;

import java.awt.AWTException;

import java.awt.Dimension;

import java.awt.Rectangle;

import java.awt.Robot;

import java.awt.Toolkit;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.IOException;

import javax.imageio.ImageIO;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.chrome.ChromeOptions;

import org.testng.Assert;

import org.testng.ITestResult;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeSuite;

import org.testng.annotations.Test;

import com.aventstack.extentreports.ExtentReports;

import com.aventstack.extentreports.ExtentTest;

import com.aventstack.extentreports.Status;

import com.aventstack.extentreports.reporter.ExtentHtmlReporter;

import com.aventstack.extentreports.reporter.configuration.ChartLocation;

import com.aventstack.extentreports.reporter.configuration.Theme;

import io.github.bonigarcia.wdm.WebDriverManager;

public class task4 {

WebDriver driver;

static ExtentTest test;

static ExtentReports extent;

ExtentHtmlReporter htmlReporter;

String sstloc="D://CE4.png";

@BeforeSuite

void init()

{

WebDriverManager.chromedriver().setup();

ChromeOptions co = new ChromeOptions();

co.addArguments("--remote-allow-origins=\*");

driver=new ChromeDriver(co);

htmlReporter = new ExtentHtmlReporter("D://extr4.html");

extent = new ExtentReports();

extent.attachReporter(htmlReporter);

//configuration items to change the look and feel

//add content, manage tests etc

htmlReporter.config().setChartVisibilityOnOpen(true);

htmlReporter.config().setDocumentTitle("Simple Automation Report");

htmlReporter.config().setReportName("Test Report");

htmlReporter.config().setTestViewChartLocation(ChartLocation.TOP);

htmlReporter.config().setTheme(Theme.STANDARD);

}

@Test

void Test1() throws InterruptedException

{

test = extent.createTest("Test Case 1", "The test case 1 has passed");

String url="https://opensource-demo.orangehrmlive.com/web/index.php/auth/login";

driver.get(url);

Thread.sleep(3000);

WebElement uname=driver.findElement(By.name("username"));

WebElement pwd=driver.findElement(By.name("password"));

WebElement submit=driver.findElement(By.xpath("//\*[@class='oxd-button oxd-button--medium oxd-button--main orangehrm-login-button']"));

uname.sendKeys("DEEPIKA");

pwd.sendKeys("12345");

submit.click();

String url1="https://opensource-demo.orangehrmlive.com/web/index.php/dashboard/index";

}

@Test()

void Test2() throws InterruptedException

{

//WebElement logout=driver.findElement(By.linkText("Logout"));

//logout.click();

driver.close();

}

@AfterMethod

public void getResult(ITestResult result) throws AWTException, IOException {

if(result.getStatus() == ITestResult.FAILURE) {

screencapture(sstloc);

test.log(Status.FAIL,result.getThrowable());

test.addScreenCaptureFromPath(sstloc);

}

else if(result.getStatus() == ITestResult.SUCCESS) {

test.log(Status.PASS, result.getTestName());

}

else {

test.log(Status.SKIP, result.getTestName());

}

}

@AfterTest

public void tearDown() {

//to write or update test information to reporter

extent.flush();

}

void screencapture(String sstloc1) throws AWTException, IOException

{

Robot robot=new Robot();

Dimension scr\_size=Toolkit.getDefaultToolkit().getScreenSize();

Rectangle rect=new Rectangle(scr\_size);

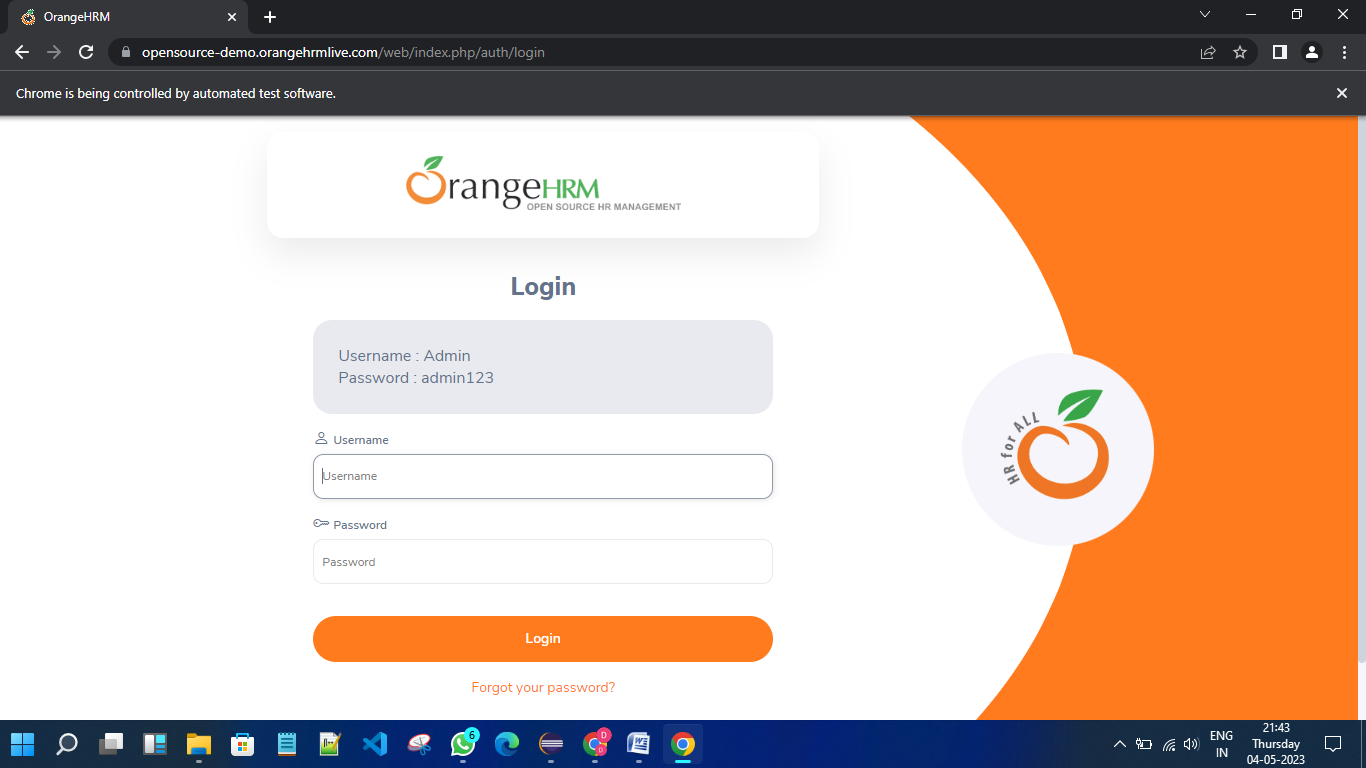
BufferedImage scrshot=robot.createScreenCapture(rect);

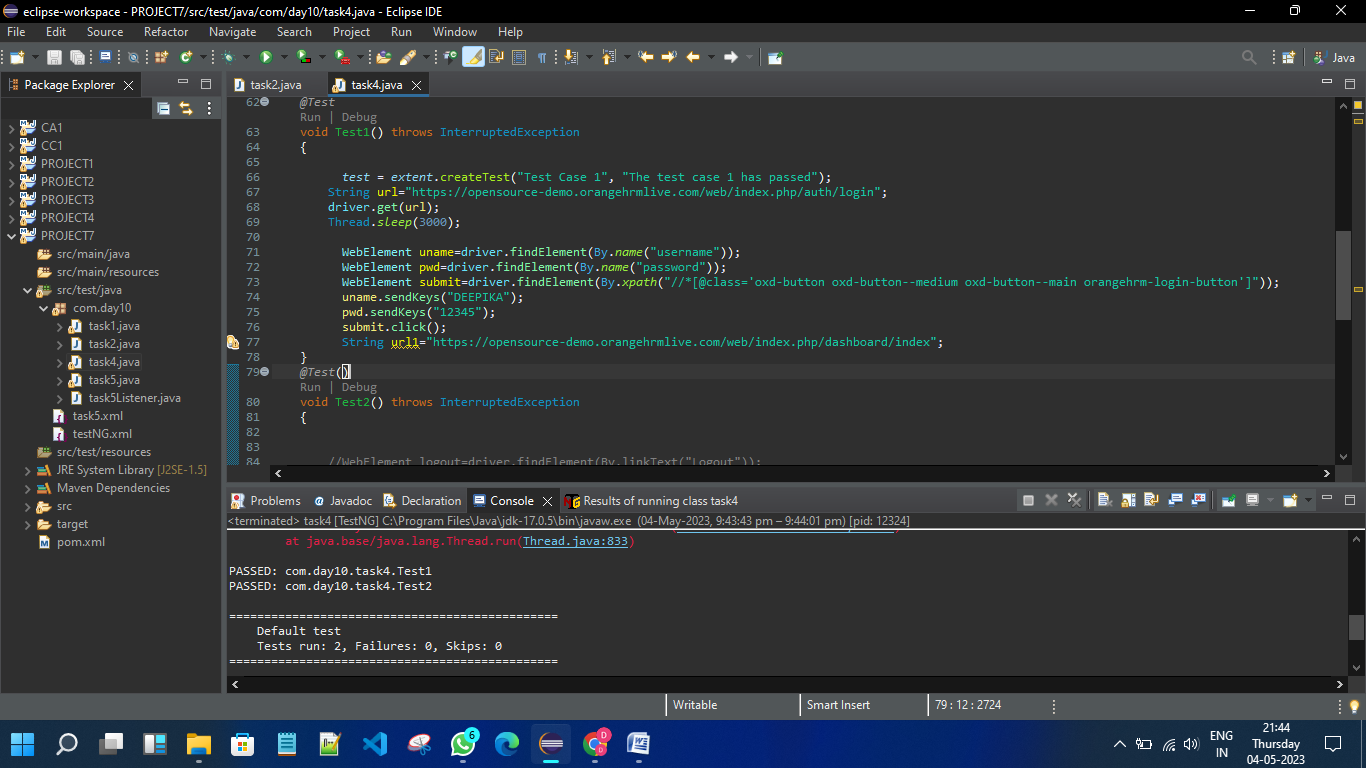
File dest=new File(sstloc1);

ImageIO.write(scrshot,"png",dest);

}

}





**TASK 05:-**

Java class :

package com.day10;

import org.testng.ITestContext;

import org.testng.ITestListener;

import org.testng.ITestResult;

public class task5Listener implements ITestListener {

public void onTestStart(ITestResult result) {

// TODO Auto-generated method stub

System.out.println("on Test Start"+result.getName());

}

public void onTestSuccess(ITestResult result) {

// TODO Auto-generated method stub

System.out.println("on Test Success"+result.getName());

}

public void onTestFailure(ITestResult result) {

// TODO Auto-generated method stub

System.out.println("on Test Failure"+result.getName());

}

public void onTestSkipped(ITestResult result) {

// TODO Auto-generated method stub

System.out.println("on Test Skipped"+result.getName());

}

public void onTestFailedButWithinSuccessPercentage(ITestResult result) {

// TODO Auto-generated method stub

System.out.println("on Test FailedButWithinSuccessPercentage"+result.getName());

}

public void onTestFailedWithTimeout(ITestResult result) {

// TODO Auto-generated method stub

System.out.println("on Test FailedWithTimeout"+result.getName());

}

public void onStart(ITestContext context) {

// TODO Auto-generated method stub

System.out.println("on Start"+context.getName());

}

public void onFinish(ITestContext context) {

// TODO Auto-generated method stub

System.out.println("on Finish"+context.getName());

}

}

testNG class:

package com.day10;

import java.awt.AWTException;

import java.awt.Dimension;

import java.awt.Rectangle;

import java.awt.Robot;

import java.awt.Toolkit;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.IOException;

import javax.imageio.ImageIO;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.chrome.ChromeOptions;

import org.testng.Assert;

import org.testng.ITestResult;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeSuite;

import org.testng.annotations.Test;

import com.aventstack.extentreports.ExtentReports;

import com.aventstack.extentreports.ExtentTest;

import com.aventstack.extentreports.Status;

import com.aventstack.extentreports.reporter.ExtentHtmlReporter;

import com.aventstack.extentreports.reporter.configuration.ChartLocation;

import com.aventstack.extentreports.reporter.configuration.Theme;

import io.github.bonigarcia.wdm.WebDriverManager;

public class task5 {

WebDriver driver;

static ExtentTest test;

static ExtentReports extent;

ExtentHtmlReporter htmlReporter;

String sstloc="D://CE4.png";

@BeforeSuite

void init()

{

WebDriverManager.chromedriver().setup();

ChromeOptions co = new ChromeOptions();

co.addArguments("--remote-allow-origins=\*");

driver=new ChromeDriver(co);

htmlReporter = new ExtentHtmlReporter("D://extr4.html");

extent = new ExtentReports();

extent.attachReporter(htmlReporter);

//configuration items to change the look and feel

//add content, manage tests etc

htmlReporter.config().setChartVisibilityOnOpen(true);

htmlReporter.config().setDocumentTitle("Simple Automation Report");

htmlReporter.config().setReportName("Test Report");

htmlReporter.config().setTestViewChartLocation(ChartLocation.TOP);

htmlReporter.config().setTheme(Theme.STANDARD);

}

@Test

void Test1() throws InterruptedException

{

test = extent.createTest("Test Case 1", "The test case 1 has passed");

String url="https://opensource-demo.orangehrmlive.com/web/index.php/auth/login";

driver.get(url);

Thread.sleep(3000);

WebElement uname=driver.findElement(By.name("username"));

WebElement pwd=driver.findElement(By.name("password"));

WebElement submit=driver.findElement(By.xpath("//\*[@class='oxd-button oxd-button--medium oxd-button--main orangehrm-login-button']"));

uname.sendKeys("Suvitha");

pwd.sendKeys("12345");

submit.click();

String url1="https://opensource-demo.orangehrmlive.com/web/index.php/dashboard/index";

}

@Test()

void Test2() throws InterruptedException

{

//WebElement logout=driver.findElement(By.linkText("Logout"));

//logout.click();

driver.close();

}

@AfterMethod

public void getResult(ITestResult result) throws AWTException, IOException {

if(result.getStatus() == ITestResult.FAILURE) {

screencapture(sstloc);

test.log(Status.FAIL,result.getThrowable());

test.addScreenCaptureFromPath(sstloc);

}

else if(result.getStatus() == ITestResult.SUCCESS) {

test.log(Status.PASS, result.getTestName());

}

else {

test.log(Status.SKIP, result.getTestName());

}

}

@AfterTest

public void tearDown() {

//to write or update test information to reporter

extent.flush();

}

void screencapture(String sstloc1) throws AWTException, IOException

{

Robot robot=new Robot();

Dimension scr\_size=Toolkit.getDefaultToolkit().getScreenSize();

Rectangle rect=new Rectangle(scr\_size);

BufferedImage scrshot=robot.createScreenCapture(rect);

File dest=new File(sstloc1);

ImageIO.write(scrshot,"png",dest);

}

}

