**Assignment 1**

1.Download and Launch the "dropdown.html" file.

2.Select date 05-05-2005 from the dropdown and validate the same.

3. Fetch the year from the dropdown and validate the year is in Ascending Order.

**Solution**

package assignment;

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.support.ui.Select;

import java.time.Duration;

public class Assignment2 {

public static void main(String[] args) {

//giving path to where chrome driver is downloaded

System.setProperty("webdriver.chrome.driver", "C:\\Users\\Pallavi.Arora\\Downloads\\chromedriver\_win32\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

//maximizing the window which opens

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

//deleting all the cookies and cache

driver.manage().deleteAllCookies();

//adding pageLoadTimeout of 30secs so that the page is fully loaded before performing any action

driver.manage().timeouts().pageLoadTimeout(Duration.ofSeconds(30));

//opening dropdown.html

driver.get("file:///C:/Users/Pallavi.Arora/Downloads/Dropdown.html");

//selecting value of date

Select day = new Select(driver.findElement(By.name("dob-day")));

day.selectByValue("05");

//selecting value of month

Select month = new Select(driver.findElement(By.name("dob-month")));

month.selectByValue("05");

//selecting value of year

Select year = new Select(driver.findElement(By.name("dob-year")));

year.selectByValue("2005");

//storing date in strings

String finalDate = driver.findElement(By.xpath("//\*[@id='dob-day']/option[7]")).getText();

String finalMonth = driver.findElement(By.xpath("//\*[@id='dob-month']/option[7]")).getText();

String finalYear = driver.findElement(By.xpath("//\*[@id='dob-year']/option[10]")).getText();

//validation

if ((finalDate+"-"+finalMonth+"-"+finalYear).contains("05-May-2005"))

{

System.out.println("Date Validated");

}

else {

System.out.println("Validation failed");

}

//taking the first two values in dropdown list of years

String y1 = driver.findElement(By.xpath("//\*[@id='dob-year']/option[3]")).getText();

String y2 = driver.findElement(By.xpath("//\*[@id='dob-year']/option[4]")).getText();

int year1 = Integer.parseInt(y1);

int year2 = Integer.parseInt(y2);

//comparing to see if they are in ascending order or not

if(year1>year2) {

System.out.println("Not in ascending order");

}

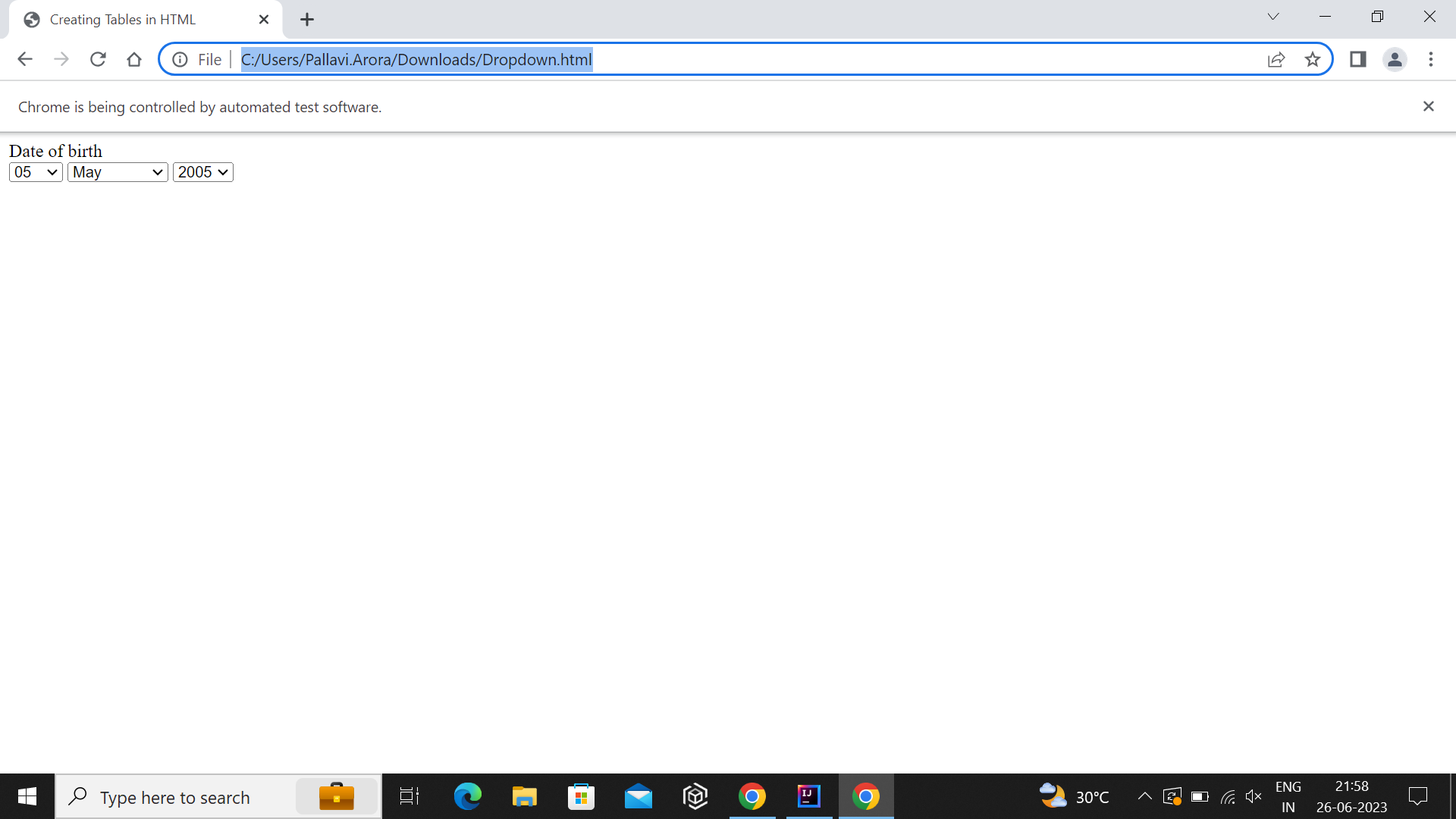
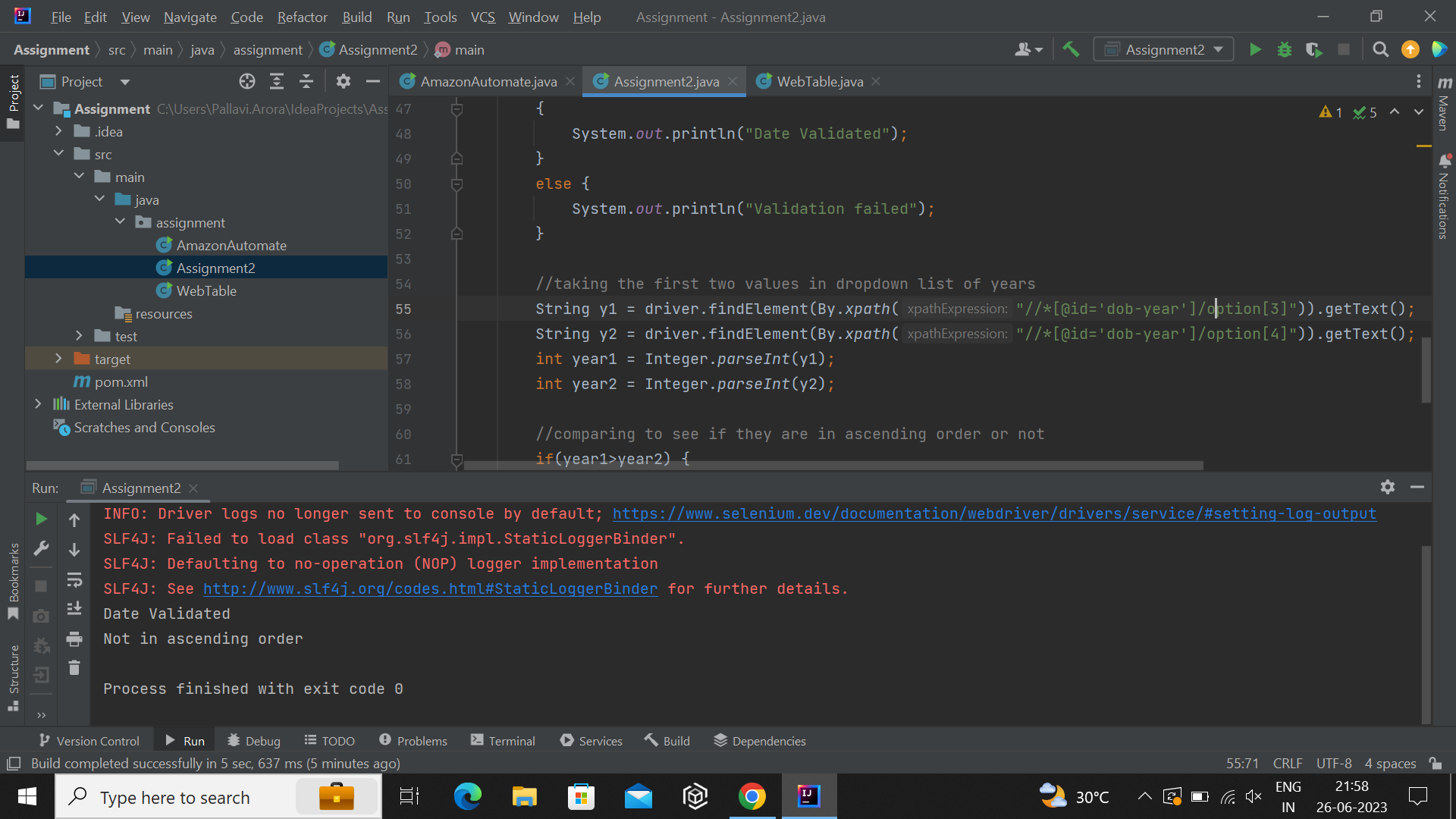
else {

System.out.println("Ascending order");

}

}

}

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

**Assignment 2**

1. Download and Launch the "Assignment.html" file.

2. Launch the file.

3. Read the table and find the unique rows from the table.

**Solution**

package assignment;  
  
import org.openqa.selenium.By;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.WebElement;  
import org.openqa.selenium.chrome.ChromeDriver;  
import java.time.Duration;  
import java.util.HashSet;  
import java.util.List;  
  
public class WebTable {  
 public static void main(String[] args) {  
 //giving path to where chrome driver is downloaded  
 System.*setProperty*("webdriver.chrome.driver", "C:\\Users\\Pallavi.Arora\\Downloads\\chromedriver\_win32\\chromedriver.exe");  
 WebDriver driver = new ChromeDriver();  
  
 //maximizing the window which opens  
 driver.manage().window().maximize();  
 //deleting all the cookies and cache  
 driver.manage().deleteAllCookies();  
 //adding pageLoadTimeout of 30secs so that the page is fully loaded before performing any action  
 driver.manage().timeouts().pageLoadTimeout(Duration.*ofSeconds*(30));  
  
 //opening assignment.html  
 driver.get("C:\\Users\\Pallavi.Arora\\Downloads\\Assignment.html");  
 //creating a web element for table  
 WebElement table = driver.findElement(By.*xpath*("/html/body/table"));  
 //storing all the rows of the table in a list  
 List<WebElement> rows = table.findElements(By.*tagName*("tr"));  
 //creating hashset because hashset only contains unique elements  
 HashSet<String> uniqueRows = new HashSet<>();  
 //traversing the list of rows  
 for(WebElement row:rows) {  
 //creating a list of table columns  
 List<WebElement> cells = row.findElements(By.*tagName*("td"));  
 StringBuilder data = new StringBuilder();  
 //traversing the columns  
 for(WebElement cell : cells){  
 //storing data in StringBuilder  
 data.append(cell.getText()).append(" ");  
 }  
 //adding data to set so only unique data gets added  
 uniqueRows.add(data.toString());  
 }  
 //printing the unique rows  
 for(String row:uniqueRows) {  
 System.*out*.println(row);  
 }  
 }  
}

