

SOFTWARE TESTING AND AUTOMATION LABORATORY

Course Code : 20ISL76A

Credits : 1.5

L:T:P:S : 0:0:1.5:0

CIE Marks : 25

Exam Hours : 3

SEE Marks : 25

Course Outcomes: At the end of the Course, the Student will be able to:

CO1	Derive the test cases for a given problem using testing approaches such as decision table approach, Equivalence class testing and Boundary Value Analysis method
CO2	Derive test cases for UI of web applications.
CO3	Illustrate automated testing of web applications using selenium automation framework
CO4	Illustrate mobile app testing using APPIUM

Mapping of Course Outcomes to Program Outcomes:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	3	3								3
CO2	3	3	3	3	3							3
CO3	3	3	3	3	3							3
CO4	3	3	3	3	3							3

Mapping of Course Outcomes to Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	3	3
CO2	3	3
CO3	3	3
CO4	3	3

Program	Topics to be covered in four hours
1	Design and develop a program in a language of your choice to solve the triangle problem defined as follows: Accept three integers which are supposed to be the three sides of a triangle and determine if the three values represent an equilateral triangle, isosceles triangle, scalene triangle, or they do not form a triangle at all. Assume that the upper limit for the size of any side is 10. Derive test cases for your program based on decision table approach, execute the test cases and discuss the results.
2	Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of boundary value testing, derive different test cases, execute these test cases and discuss the test results..
3	Design, develop, code and run the program in any suitableLanguage to implement the NextDate function. Analyze it from theperspective of equivalence class value testing, derive different testcases, execute these test cases and discuss the test results.
4	Design front-end for any web application and derive the test cases as applicable. Validate the UI elements using JavaScript
5	Write a program for matrix multiplication.“Introspect the causes for its failure and write down the possible reasons”. Analyze the Positive test cases and Negative Test cases.
6	Implement test script in Selenium IDE using recording, playback/executing and saving processes. Use Selenium IDECommands Assertions and Actions to directly interact with pageelements.
7	Using Selenium IDE, create a test suite containing minimum 4test cases (for any two web sites).
8	Demonstrate selenium server installation using JAVA
9	Illustrate automated testing using selenium to perform tests on login web page.
10	Use selenium to test a program that updates 10 student records into a table from Excel file.
11	Write and test a program to provide total number of objects present on a web page using selenium.
12	Demonstrate mobile app testing using APPIUM

Program 1: Design and develop a program in a language of your choice to solve the triangle problem defined as follows. Accept 3 integers which are supposed to be 3 sides of a triangle and determine if the 3 values represent an equilateral triangle, isosceles, scalene or they do not form a triangle at all. Assume that the upper limit for the size of any side is 10. Derive testcases for your program based on Decision Table approach, execute the testcases and discuss the test results.

Program:

```
check=True
```

```
while check:
```

```
    a,b,c=map(int,input().split())
```

```
    c1=((a>=1)) and ((a<=10))
```

```
    c2=((b>=1)) and ((b<=10))
```

```
    c3=((c>=1)) and ((c<=10))
```

```
    if(not c1):
```

```
        print("a is not in range")
```

```
    if(not c2):
```

```
        print("b is not in range")
```

```
    if(not c3):
```

```
        print("c is not in range")
```

```
    check=(not c1) or (not c2) or (not c3)
```

```
    if((a<(b+c)) and (b<(a+c)) and (c<(a+b))):
```

```
        if((a==b) and (b==c)):
```

```
            print("Eqilateral triangle")
```

```
        elif((a!=b) and (b!=c) and (c!=a)):
```

```
            print("Scalene triangle")
```

```
        else:
```

```
            print("Isosceles triangle")
```

```
    else:
```

```
        print("Triangle cannot be formed")
```

Output:

<u>Decision Table</u>	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11
c1: $a < b + c$	F	T	T	T	T	T	T	T	T	T	T
c2: $b < a + c$	-	F	T	T	T	T	T	T	T	T	T
c3: $c < a + b$	-	-	F	T	T	T	T	T	T	T	T
c4: $a = b$	-	-	-	T	T	T	T	F	F	F	F
c5: $a = c$	-	-	-	T	T	F	F	T	T	F	F
c6: $b = c$	-	-	-	T	F	T	F	T	F	T	F
a1: Not a triangle	X	X	X								
a2: Scalene											X
a3: Isosceles							X		X	X	
a4: Equilateral				X							
a5: Impossible					X	X	X				

<u>Testcases</u>	Input	Output
Testcase ID	Description	
DT1	Enter a,b,c 10 2 3	Not a triangle
DT2	Enter a,b,c 2 8 3	Not a triangle
DT3	Enter a,b,c 2 3 6	Not a triangle
DT4	Enter a,b,c 5 5 5	Equilateral triangle
DT5	Enter a,b,c ? ? ?	Impossible
DT6	Enter a,b,c ? ? ?	Impossible
DT7	Enter a,b,c 5 5 4	Isosceles triangle
DT8	Enter a,b,c ? ? ?	Impossible
DT9	Enter a,b,c 5 4 5	Isosceles triangle
DT10	Enter a,b,c 4 5 5	Isosceles triangle
DT11	Enter a,b,c 2 3 4	Scalene triangle

NHCE, BANGALORE

Program 2: Design, Develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of Boundary Value Testing, derive different testcases, execute these testcases and discuss the test results.

Program:

```
#include<stdio.h>
```

```

int main()
{
int c1,c2,c3,temp;
int locks,stocks,barrels,totallocks,totalstocks,totalbarrels;
float lockprice,stockprice,barrelprice,locksales,stocksales,barrelsales,sales,com; lockprice=45.0;
stockprice=30.0;
barrelprice=25.0;
totallocks=0;
totalstocks=0;
totalbarrels=0;
printf("Enter the number of locks and to exit press-1\n");
scanf("%d",&locks);
while(locks != -1)
{
c1=(locks<=0 || locks>70);
printf("\nEnter the number of stocks and barrels\n");
scanf("%d %d",&stocks,&barrels);
c2=(stocks<=0 || stocks>80);
c3=(barrels<=0 || barrels>90);
if(c1)
printf("\nValue of locks are not in the range of 1.....70\n");
else
{
temp=totallocks+locks; if(temp>70)
printf("New total locks=%d not in the range of 1.....70\n",temp);
else
totallocks=temp;
}
}

```

```

printf("Total locks = %d",totallocks); if(c2)
printf("\n Value of stocks not in the range of 1.....80\n");
else
{
temp=totalstocks+stocks;

if(temp>80)
printf("\nNew total stocks =%d not in the range of 1..... 80",temp);
else
totalstocks=temp;
}

printf("\nTotal stocks = %d",totalstocks); if(c3)
printf("\n Value of barrels not in the range of 1. ....90\n");
else
{
temp=totalbarrels+barrels; if(temp>90)
printf("\nNew total barrels=%d not in the range of 1. ....90\n",temp);
else
totalbarrels=temp;
}

printf("\nTotal barrels=%d", totalbarrels);
printf("\nEnter the number of locks and to exit press-1\n"); scanf("%d",&locks);
}

printf("\n Total locks = %d",totallocks);
printf("\nTotal stocks = %d",totalstocks);
printf("\n Total barrels = %d",totalbarrels);

locksales=totallocks*lockprice;

```

```
stocksales=totalstocks*stockprice;  
barrelsales=totalbarrels*barrelprice;  
sales=locksales+stocksales+barrelsales;  
printf("\nTotal sales = %f",sales); if(sales>1800)  
{  
com=0.10*1000;  
com=com+(0.15*800);  
com=com+0.20*(sales-1800);  
}  
else if(sales>1000)  
{  
com=0.10*1000;  
com=com+0.15*(sales-1000);  
}  
else  
com=0.10*sales;  
printf("\nCommission = %f",com);  
return 0;  
}
```

Output:

Considering commission program, we have 3 input variables locks, stocks and barrels

Range of values for locks : 1-70, stocks = 1-80, barrels = 1-90

Variables	Mn	Mn+	Nom	Max-	Max
locks	1	2	35	69	70
stocks	1	2	40	79	80
barrels	1	2	45	89	90

Considering output variable sales we have 3 sets for calculating commission, i.e., if sales are below 1000, comm is 10%, if sales is from 1001 - 1800, then comm is 15% and if sales are greater than 1801, comm is 20%.

Sales	Mn locks, stocks barrels	Mn+ locks, stocks, barrels	Nom locks, stocks, barrels	Max- locks, stocks, barrels	Max locks, stocks, barrels
1-1000	1, 1, 1	2, 1, 1 1, 2, 1 1, 1, 2	5, 5, 5	9, 10, 10 10, 9, 10 10, 10, 9	10, 10, 10
1001-1800	11, 10, 10 10, 11, 10 10, 10, 11	12, 10, 10 10, 12, 10 10, 10, 12	14, 14, 14	17, 18, 18 18, 17, 18 18, 18, 17	18, 18, 18
1801-above	19, 18, 18 18, 19, 18 18, 18, 19	20, 18, 18 18, 20, 18 18, 18, 20	48, 48, 48	69, 80, 90 70, 79, 90 70, 80, 89	70, 80, 90

Testcase ID	Description	Input			Output	
		Lacks	Stocks	Barrels	Sales	Comm.
BVA1	Enter l,s,b	1	40	45	2370	334
BVA2	Enter l,s,b	2	40	45	2415	343
BVA3	Enter l,s,b	35	40	45	3900	640
BVA4	Enter l,s,b	69	40	45	5430	946
BVA5	Enter l,s,b	70	40	45	5475	955
BVA6	Enter l,s,b	35	1	45	2730	408
BVA7	Enter l,s,b	35	2	45	2760	412
BVA8	Enter l,s,b	35	19	45	5070	814
BVA9	Enter l,s,b	35	80	45	5100	880
BVA10	Enter l,s,b	35	40	1	2800	420
BVA11	Enter l,s,b	35	40	2	2825	425
BVA12	Enter l,s,b	35	40	89	5000	860
BVA13	Enter l,s,b	35	40	90	5025	865

Program 3: Design, Develop, code and run the program in any suitable language to implement the Next Date function. Analyze it from the perspective of Equivalence Class Testing, derive different testcases, execute these testcases and discuss the results.

Program:

```
#include<stdio.h>

int check(int day,int month)
{
    if((month==4||month==6||month==9 ||month==11) && day==31) return 1;

    else

        return 0;
}
```

```

int isleap(int year)
{
if((year%4==0 && year%100!=0) || year%400==0) return 1;
else
return 0;
}

int main()
{
int day,month,year,tomm_day,tomm_month,tomm_year; char flag;
do
{
flag='y';
printf("\nEnter the today's date in the form of dd mm yyyy\n");
scanf("%d%d%d",&day,&month,&year);
tomm_month=month; tomm_year= year; if(day<1 || day>31)
{
printf("value of day, not in the range 1...31\n"); flag='n';
}
if(month<1 || month>12)
{
printf("value of month, not in the range 1. 12\n");
flag='n';
}
else if(check(day,month))
{
printf("value of day, not in the range day<=30"); flag='n';
}
}

```

```
if(year<=1812 || year>2015)
{
    printf("value of year, not in the range 1812. 2015\n");
    flag='n';
}

if(month==2)
{
    if(isleap(year) && day>29)
    {
        printf("invalid date input for leap year"); flag='n';
    }
    else if(!(isleap(year))&& day>28)
    {
        printf("invalid date input for not a leap year"); flag='n';
    }
}

}while(flag=='n');
```

```
switch (month)
{
    case 1:
    case 3:
    case 5:
    case 7:
    case 8:
```

```
case 10:if(day<31)
tomm_day=day+1;

else
{
tomm_day=1; tomm_month=month+1;
}

break;

case 4:

case 6:

case 9:

case 11: if(day<30)
tomm_day=day+1;

else
{
tomm_day=1; tomm_month=month+1;
}

break;

case 12: if(day<31)
tomm_day=day+1;
else
{
tomm_day=1; tomm_month=1; if(year==2015)
{
printf("the next day is out of boundary value of year\n"); tomm_year=year+1;
}
}
```

```
else
tomm_year=year+1;
}
break;
case 2:

if(day<28)
tomm_day=day+1;
else if(isleap(year)&& day==28)
tomm_day=day+1; else if(day==28 || day==29)
{
tomm_day=1; tomm_month=3;
}
break;
}
printf("next day is : %d %d %d",tomm_day,tomm_month,tomm_year); return 0;
}
```

Output:

Equivalence classes are as follows:

$$D_1 = \{ \text{Day}/DD : 1 \leq DD \leq 31 \}$$

$$M_1 = \{ \text{Month}/MM : 1 \leq MM \leq 12 \}$$

$$Y_1 = \{ \text{Year}/YY : 18/2 \leq YY \leq 2015 \}$$

Weak Normal / Strong Normal

Testcases	Description	Inputs			Output
		DD	MM	YY	
WN1/WN1	Enter dd/mm/yy	12	2	1990	12/2/1990

Weak Robust

Testcases	Description	Inputs			Output
		DD	MM	YY	
WR1	Enter dd/mm/yy	-1	6	1992	Day out of range
WR2	Enter dd/mm/yy	15	-1	1992	Month out of range
WR3	Enter dd/mm/yy	15	6	1811	Year out of range
WR4	Enter dd/mm/yy	32	6	1992	Day out of range
WR5	Enter dd/mm/yy	15	13	1992	Month out of range
WR6	Enter dd/mm/yy	15	6	2016	Year out of range
WR7	Enter dd/mm/yy	15	-1	1000	Month out of range

Strong Robust

Testcases	Description	Inputs			Output
		DD	MM	YY	
SR1	Enter dd/mm/yy	-1	-1	1992	Day, Month out of range
SR2	Enter dd/mm/yy	-1	6	1811	Day, Year out of range
SR3	Enter dd/mm/yy	15	-1	1811	Month, Year out of range
SR4	Enter dd/mm/yy	-1	-1	1811	Day, Month, year out of range
SR5	Enter dd/mm/yy	32	13	1992	Day, Month out of range
SR6	Enter dd/mm/yy	32	6	2016	Day, Year out of range
SR7	Enter dd/mm/yy	15	13	2016	Month, Year out of range

Program 4: Design Front-end for any web application and derive the testcases as applicable. Validate the UI elements using JavaScript.

Program:

1.html:

<html>

<head>

```

<title>Javascript Login Form Validation</title>
<!-- Include CSS File Here -->
<link rel="stylesheet" href="form-style.css"/>
<!-- Include JS File Here -->
<script src="login.js"></script>
</head>
<body>
<div class="container">
<div class="main">
<h2>Javascript Login Form Validation</h2>
<form id="form_id" method="post" name="myform">
<label>User Name :</label>
<input type="text" name="username" id="username"/>
<label>Password :</label>
<input type="password" name="password" id="password"/>
<input type="button" value="Login" id="submit" onclick="validate()"/>
</form>
<span><b class="note">Note : </b>For this demo use following username and password.
<br/><b class="valid">User Name : Form<br/>Password : 123</b></span>
</div>
</div>
</body>
</html>

```

login.js:

```

var attempt = 3; // Variable to count number of attempts.

// Below function Executes on click of login button.

function validate() {
    var username = document.getElementById("username").value;
    var password = document.getElementById("password").value;
}

```

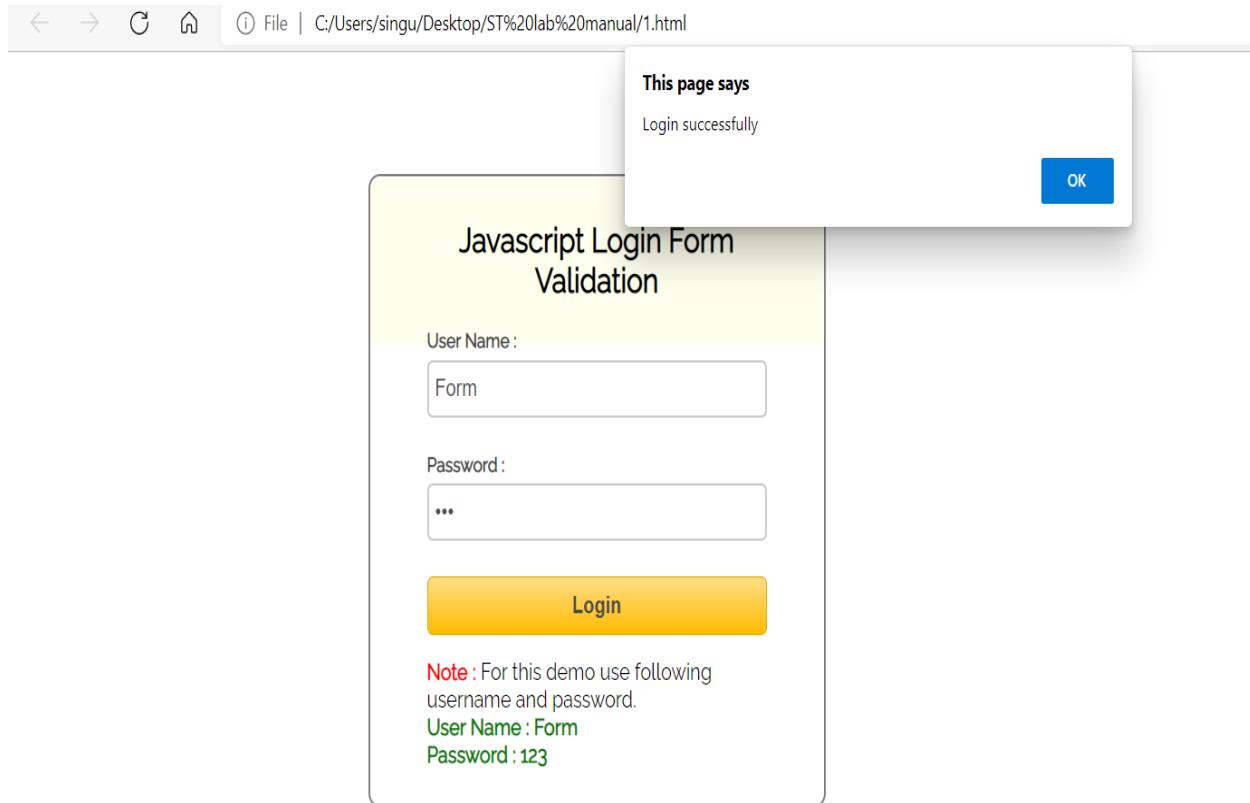
```
if ( username == "Form" && password == "123") {  
    alert ("Login successfully");  
    window.location = "success.html"; // Redirecting to other page.  
    return false;  
}  
  
else {  
    attempt --; // Decrementing by one. I  
    alert("You have left "+attempt+" attempt;");  
    // Disabling fields after 3 attempts.  
    if( attempt == 0){  
        document.getElementById("username").disabled = true;  
        document.getElementById("password").disabled = true;  
        document.getElementById("submit").disabled = true;  
        return false;  
    }  
}  
}
```

success.html:

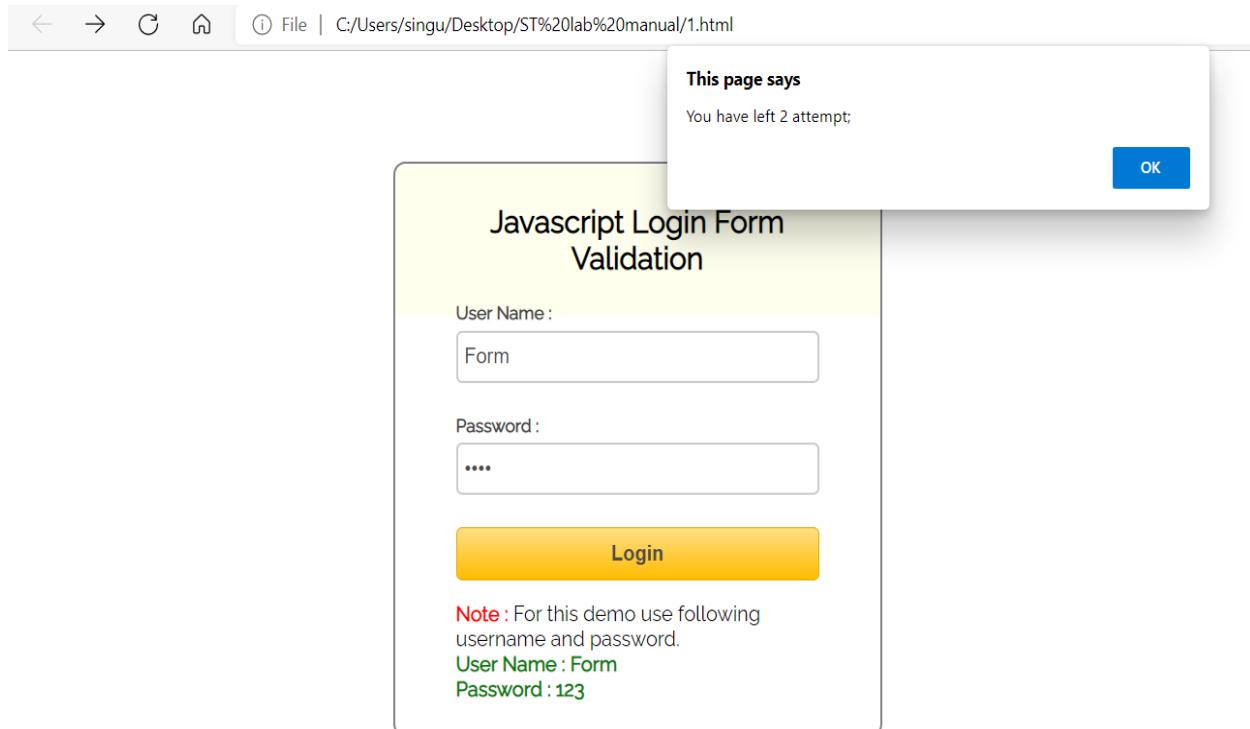
```
<html>  
<head>  
<title>Javascript Login Form Validation</title>  
</head>  
<body>  
<h2>Successful Login!</h2>  
</body>  
</html>
```

Output:

Logged in successfully.



Login failed, so the number of attempts is displayed.



Testcase Table:

Testcase ID	Description	Input		Output
		Username	Password	
1.	Enter username & password	Form	123	Login successful
2.	Enter username & password	Form1	12	2 attempts left
3.	Enter username & password	Form	12	2 attempts left
4.	Enter username & password	Form1	123	2 attempts left

Program 5: Write a program for matrix multiplication. Introspect the causes for its failure and write down the possible reasons. Analyze the positive testcases and negative testcases.

Program:

```
m=int(input("Enter M1 rows:"))
n=int(input("Enter M1 columns:"))
o=int(input("Enter M2 rows:"))
p=int(input("Enter M2 columns:"))

if(n==o):
    A=[]
    for i in range(m):
        m1=[]
        for j in range(n):
            k=int(input())
            m1.append(k)
        A.append(m1)

    print(A)
```

```

A.append(m1)

B=[]

for i in range(o):
    m1=[]
    for j in range(p):
        k=int(input())
        m1.append(k)

    B.append(m1)

result = [[0, 0, 0, 0],
           [0, 0, 0, 0],
           [0, 0, 0, 0]]


for i in range(len(A)):
    for j in range(len(B[0])):
        for k in range(len(B)):
            result[i][j] += A[i][k] * B[k][j]

for r in result:
    for b in r:
        if(b!=0):
            print(b,end=" ")
    print()

else:
    print("Matrix Multiplication is not possible")

```

Output:

Output is attached in the next page.

<u>Testcase no. 1</u> Testcase name: Matrix multiplication can't be performed.		
Input M1 rows and cols = 2 2 M2 rows and cols = 3 2	Expected output Operation can't be performed.	Actual output Fail
<u>Testcase no. 2</u> Testcase name: Out of range		
Input M1 rows and cols = 2 2 M2 rows and cols = 2 2	Expected output Out of range	Actual output Fail
1234567891 1234567891 1234567891 1234567891 54321567 54321567 54321567 54321567		
<u>Testcase no. 3</u> Testcase name: Matrix Multiplication		
Input M1 rows and cols = 2 2 M2 rows and cols = 2 2	Expected output 7 10 5 22	Actual output 7 10 5 22

Program 6: Implement test script in Selenium IDE using recording, playing back / executing and solving resources / processes. Use Selenium IDE commands, assertions and actions to directly interact with page elements.

STEPS:

Installing IDE –

Step 1: Open the Firefox browser.

Step 2: Click on the menu in the top right corner.

Step 3: Click on add-ons in the drop down box.

Step 4: Click on “Find more add-ons” and type “Selenium-IDE”.

Step 5: Click on “Add to Firefox”.

Once installed, the selenium IDE icon appears on the top right corner of the browser. Once you click on it , a welcome message appears.

Recording a test –

Step 1: Provide a name for your project

Step 2: Before recording, specify a valid URL. The recording begins once the browser navigates to this URL.

Step 3: Clicking on ‘Start recording’ will redirect to the specified URL and start recording the user interactions.

The user is at liberty to stop the recording. All user actions are recorded and converted into a script.

Save the work –

Step 1: Click the save icon in the top right corner of the IDE.

Step 2: It will prompt for name and a location of where to save the project. The result is a single file with a side extension.

Playback –

In browser, the tests can be played back in the Selenium automation testing IDE by selecting the test to play and by clicking on the play button.

Output:

Output screenshot is attached in the next page.

Project: st

Executing: st_test

https://www.google.com

Command	Target	Value
1 open	/	
2 set window size	1366x728	
3 run script	window.scrollTo(0,0)	
4 type	id=lst-ib	software testing
5 send keys	id=lst-ib	\$(KEY_ENTER)
6 assert title	software testing - Google Search	
7 run script	window.scrollTo(0,209)	
8 click	css=h3.LC20lb	

Command: //

Target:

Value:

Description:

Runs: 1 Failures: 0

Program 7:

Using Selenium IDE, create a test suite containing minimum 4 test cases (for any two web sites).

	Command	Target	Value
1	open	/	
2	set window size	1366x728	
3	click	id=lst-ib	
4	type	id=lst-ib	seleniumhq
5	send keys	id=lst-ib	\$(KEY_ENTER)
6	click	css=h3.LC20lb	
7	click	id=q	
8	type	id=q	web driver
9	submit	id=searchbox_016909259827549404702:hzru01fdsm	

	Command	Target	Value
1	open	/	
2	set window size	1366x728	
3	send keys	id=lst-ib	\${KEY_DOWN}
4	type	id=lst-ib	wikipedia
5	send keys	id=lst-ib	\${KEY_ENTER}
6	click	id=nqsbq	
7	type	id=nqsbq	selenium
8	click	css=img.J0mlqc	
9	click	css=h3.LC20lb	

	Command	Target	Value
1	open	/	
2	set window size	1366x728	
3	type	id=lst-ib	wikipedia
4	send keys	id=lst-ib	\${KEY_ENTER}
5	click	id=nqsbq	
6	type	id=nqsbq	joshua
7	send keys	id=nqsbq	\${KEY_ENTER}
8	click	css=h3.LC20lb	

	Command	Target	Value
1	open	/	
2	set window size	1366x728	
3	run script	window.scrollTo(0,0)	
4	type	id=lst-ib	software testing
5	send keys	id=lst-ib	\${KEY_ENTER}
6	assert title	software testing - Google Search	
7	run script	window.scrollTo(0,209)	
8	click	css=h3.LC20lb	

Program 8: Demonstrate selenium server installation using JAVA.

```
//http://localhost:4444/selenium-server/driver/?cmd=getLogMessages  
//java -jar selenium-server-standalone.jar -role hub  
//http://localhost:4444/grid/console  
//http://192.168.1.103:5555/wd/hub/static/resource/hub.html
```

cmd prompt 1

```
cd C:\Latha_Drive\SeleniumJars  
java -jar selenium-server-standalone-3.5.3.jar -role hub
```

cmd prompt 2

```
cd C:\Latha_Drive\SeleniumJars  
java -jar selenium-server-standalone-3.5.3.jar -role node -hub http://localhost:4444External files  
to be added:
```

1. Selenium-server-standalone-3.5.3.jar

The above JAR files have to be added under the Classpath section in Build Path Configuration function.

Program:

```
import static org.junit.Assert.*;  
import java.util.*;  
import java.util.concurrent.TimeUnit;  
  
import org.openqa.selenium.*;  
import org.openqa.selenium.chrome.ChromeDriver;  
import org.openqa.selenium.support.ui.ExpectedConditions;  
import org.openqa.selenium.support.ui.WebDriverWait;  
  
import java.time.Duration;  
  
class server_p {  
    static WebDriver driver;  
    public static void main(String[] args) {  
  
        server_p c=new server_p();  
  
        driver=c.Launch("http://demo.guru99.com/test/newtours/");
```

```

driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
System.out.println("waited for 10 sec");

WebElement objTextBox=driver.findElement(By.name("userName"));
objTextBox.sendKeys("userName");
driver.findElement(By.name("password")).sendKeys("password")// enter password
System.out.println("waiting");
objTextBox.submit();
WebDriverWait wait=new WebDriverWait(driver, 10);
System.out.println("waiting over");
wait.until(ExpectedConditions.titleContains("Welcome: Mercury Tours"));
System.out.println("check");

System.out.println(driver.getTitle());
System.out.println("all test case pass");

assertEquals("Welcome: Mercury Tours",driver.getTitle());

driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
c.Close();  }
public server_p()
{
System.setProperty("webdriver.chrome.driver","C:\\\\Latha_Drive\\\\SeleniumJars\\\\Chrome
Driver\\\\chromedriver.exe");
driver=new ChromeDriver();
driver.manage().window().maximize();
System.out.println("Launching Chrome1");
}
public WebDriver Launch(String url){
driver.get(url);
System.out.println("Opened URL in Chrome:"+url);
return driver;
}
public void Close()
{
driver.quit();
System.out.println("Closed Chrome");
}  }

```

```
55+     public void Close()
56+     {
57+     }
58+
59+ }
```

<terminated> selenium2 [Java Application] C:\Users\latha\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64.17.0.0.v20211012-1059\jre\bin\javaw.exe (Jan 1, 2022, 10:00:32 AM – 10:00:41 AM)
Starting ChromeDriver 96.0.4664.45 (76e4c1bb2ab4671b8beba3444e61c0f17584b2fc-refs/branch-head Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver was started successfully.
Jan 01, 2022 10:00:36 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Jan 01, 2022 10:00:36 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 96
Launching Chrome1
Opened URL in Chrome: http://demo.guru99.com/test/newtours/
waited for 10 sec
waiting
waiting over
check
Welcome: Mercury Tours
all test case pass|
Jan 01, 2022 10:00:40 AM org.openqa.selenium.remote.http.WebSocket\$Listener onError
WARNING: Connection reset

Program 9: Illustrate automated testing using selenium to perform tests on login web page

External files to be added:

1. JAR file obtained from selenium-java-4.1.0 folder and sub-folder lib.
2. Chromedriver.exe file in a folder named driver

The above JAR files have to be added under the Classpath section in Build Path Configuration function.

Program:

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class GuruWebpageLogin {
    public static void main(String[] args) throws InterruptedException {
        //System.setProperty("webdriver.gecko.driver","C:\\Latha_Drive\\SeleniumJars\\geckodriver-v0.30.0-win64\\geckodriver.exe");
    }
}
```

```

System.setProperty("webdriver.chrome.driver"," C:\\Latha_Drive\\SeleniumJars\\geckodriver-v0.30.0-win64\\driver\\chromedriver.exe");

WebDriver driver=new ChromeDriver();

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

System.out.println("Launching Chrome");

//WebDriver driver = new FirefoxDriver();
//driver.get("https://accounts.google.com/signin/v2/identifier?continue=https%3A%2F%2Fmail.google.com%2Fmail%2F&service=mail&sacu=1&rip=1&flowName=GlifWebSignIn&flowEntry=ServiceLogin");

driver.get("http://demo.guru99.com/test/newtours/");

//driver.findElement(By.id("identifierId")).sendKeys(""); // enter username
driver.findElement(By.name("userName")).sendKeys("userName"); // enter username
//driver.findElement(By.id("identifierNext")).click();

Thread.sleep(5000);

// driver.findElement(By.name("password")).sendKeys(""); // enter password
driver.findElement(By.name("password")).sendKeys("password"); // enter password
// driver.findElement(By.id("passwordNext")).click();

Thread.sleep(5000);

WebElement ob = driver.findElement(By.name("submit"));

ob.click();

String title = driver.getTitle();

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

if(title.equals("Login: Mercury Tours"))

{

    System.out.println("LOGIN SUCCESSFUL... ");

    System.out.println("all test case pass");

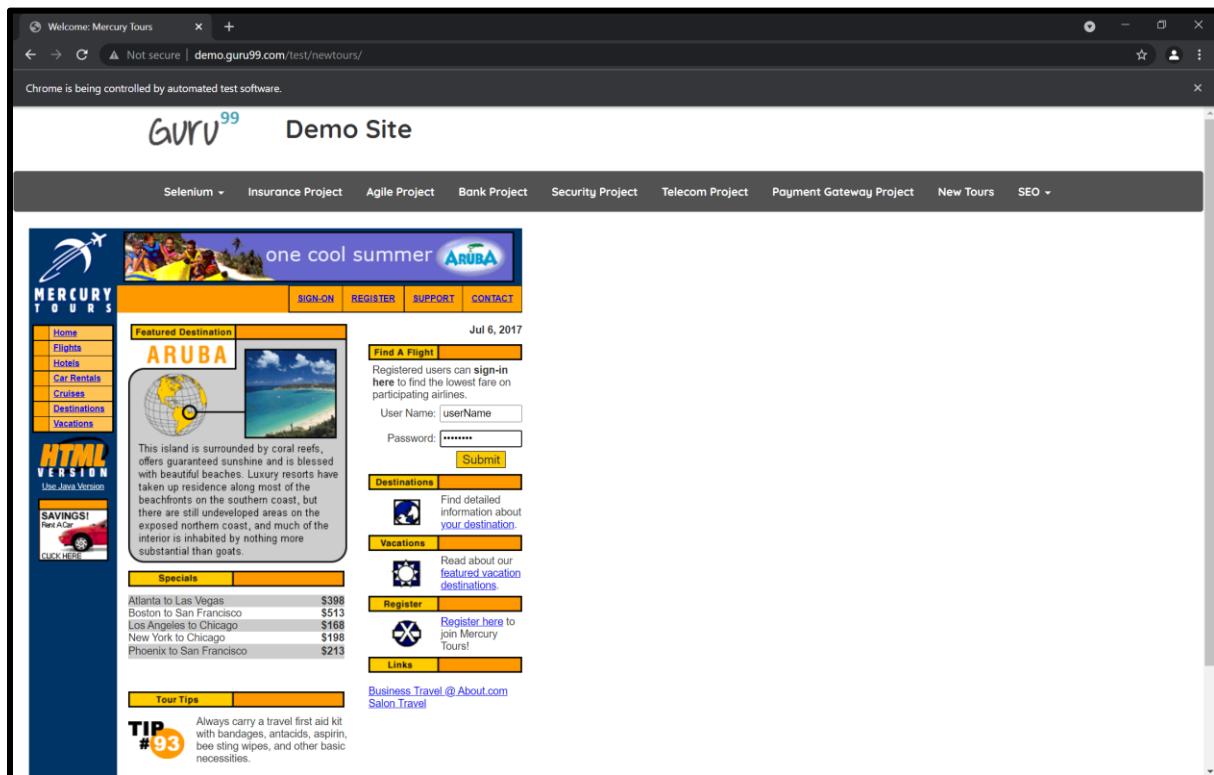
}

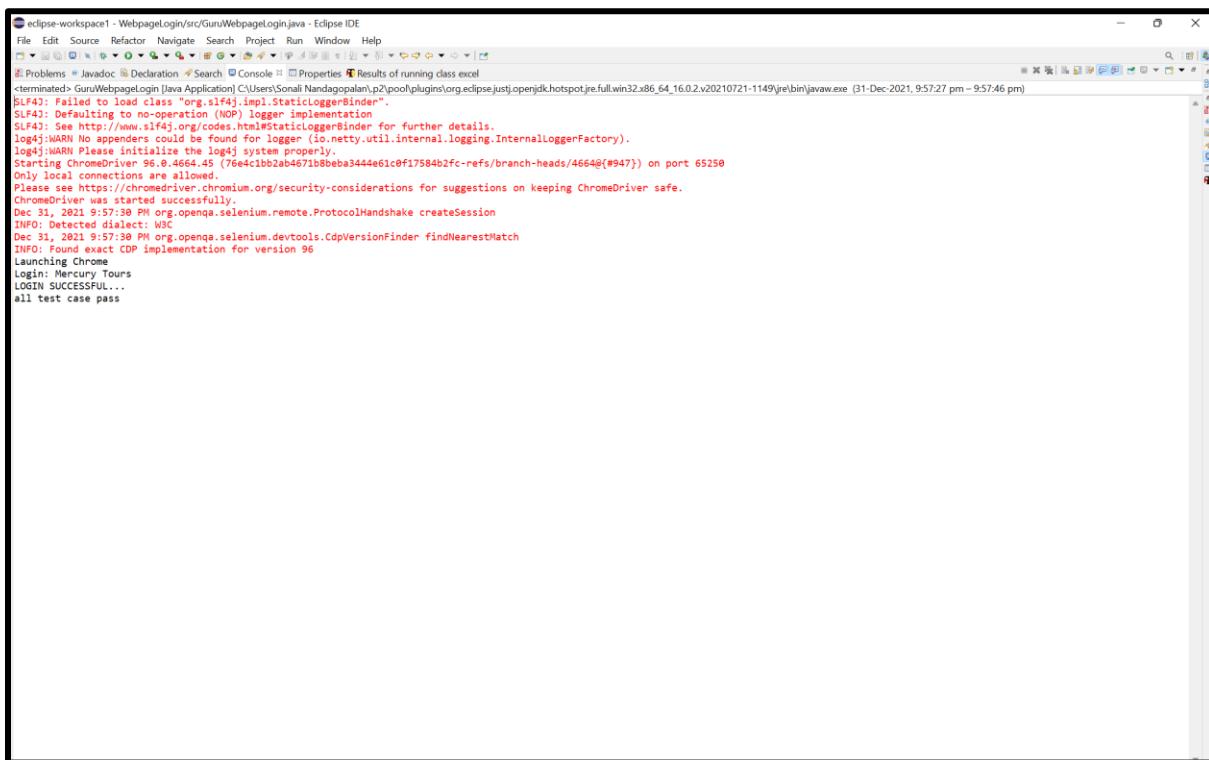
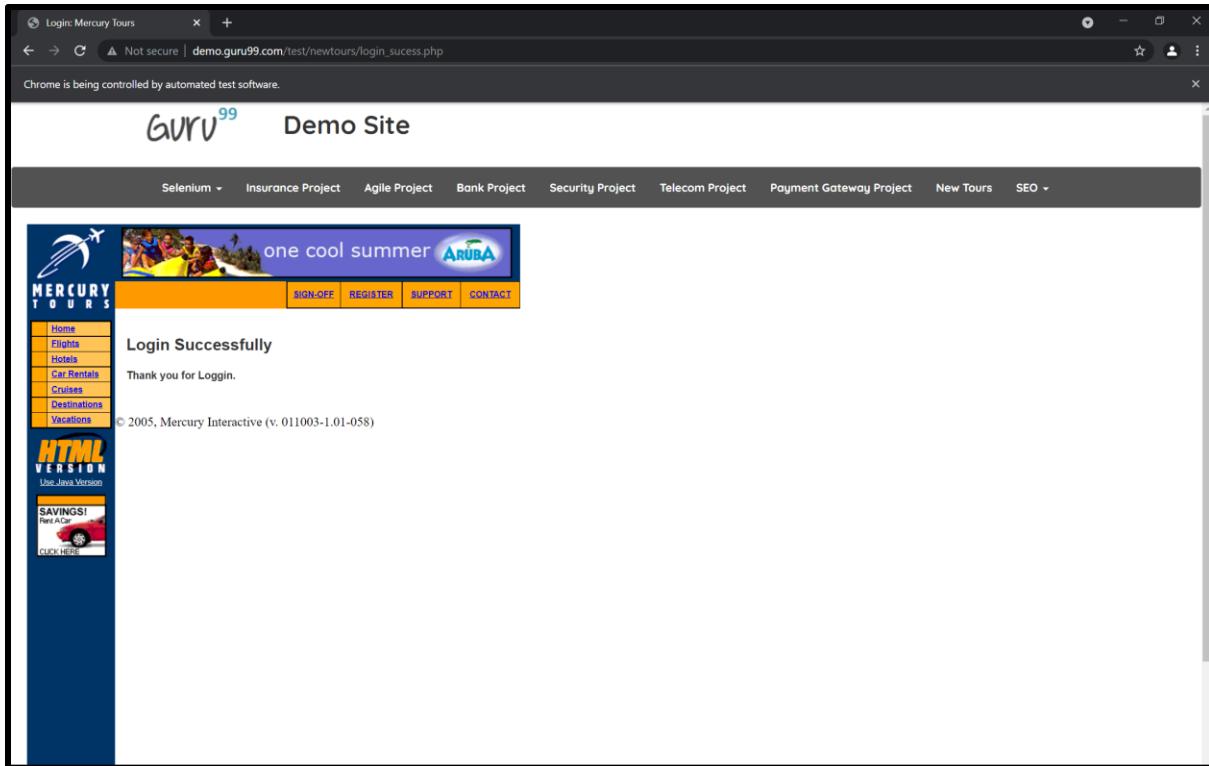
else

```

```
{
    System.out.println("LOGIN FAILED");
}
}
}
```

Output:





Program 10: Use selenium to test a program that updates 10 student records into a table from Excel file.

Prerequisite :

1) jxl jar file ,Downloaded from <https://sourceforge.net/projects/jexc...>

2) An excel file with student records present in it.

(In my case "Students.xls").

3) Install TestNG Plugin for eclipse ide from the given Link below.

<https://www.techbeamers.com/install-t...>

I have already installed it....

Steps :

1) Open eclipse create a new java project - add a class - then add the jar file using Build path as shown in my previous video.

Dont select any method here...

2) Import all the packages.

```
import java.io.FileInputStream;  
import java.io.FileOutputStream;  
import jxl.Sheet;  
import jxl.Workbook;  
import jxl.write.Label;  
import jxl.write.WritableSheet;  
import jxl.write.WritableWorkbook;  
import org.testng.annotations.*;
```

3) Write down the code given in the description.

Link to download code : <https://drive.google.com/open?id=1F8Q...>

Run the program and see the output on the specified location in my case it is Desktop....

Run the program as TestNG suite...

External files to be added:

1. log4j-1.2.14.jar
2. jcommander-1.78.jar
3. jquery-3.5.1.jar
4. testng-7.4.0.jar

The above JAR files have to be added under the Classpath section in Build Path Configuration function.

The following excel file (.xls format only) has to be created and the path for the file has to be provided in the program.

A screenshot of Microsoft Excel showing a spreadsheet titled "Students.xls". The spreadsheet is in Protected View, indicated by a yellow bar at the top. The data is organized into columns: RollNo, Name, DBMS, JAVA, LINUX, and TOTAL. The TOTAL column shows the sum of DBMS, JAVA, and LINUX marks. The data is as follows:

RollNo	Name	DBMS	JAVA	LINUX	TOTAL
1	abc	65	67	78	210
2	def	66	68	79	213
3	ghi	67	69	80	216
4	lmn	68	70	32	170
5	xyz	32	71	22	125
6	pqr	70	72	83	225
7	rst	71	73	84	228
8	uvw	72	74	85	231
9	abc1	73	22	86	181
10	abc2	74	76	87	237

Program:

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import jxl.Sheet;
import jxl.Workbook;
import jxl.write.Label;
import jxl.write.WritableSheet;
import jxl.write.WritableWorkbook;
import org.testng.annotations.*;
public class excel {
    @BeforeClass // @BeforeClass runs once before the entire test.
    public void setUp() throws Exception { }
```

```

@Test
public void testImportexport1() throws Exception {
    FileInputStream fi = new FileInputStream("C:\\Users\\ST AND AUTO
LAB\\excel\\Students.xls");
    Workbook w = Workbook.getWorkbook(fi);
    Sheet s = w.getSheet(0);
    String a[][] = new String[s.getRows()][s.getColumns()];
    FileOutputStream fo = new FileOutputStream("C:\\Users \\ST AND AUTO
LAB\\excel\\Result.xls");
    WritableWorkbook wwb = Workbook.createWorkbook(fo);
    WritableSheet ws = wwb.createSheet("result1", 0);
    for (int i = 0; i < s.getRows(); i++)
        for (int j = 0; j < s.getColumns(); j++)
    {
        a[i][j] = s.getCell(j, i).getContents();
        Label l2 = new Label(j, i, a[i][j]);
        ws.addCell(l2);
        Label l1 = new Label(6, 0, "Result");
        ws.addCell(l1);
    }
    for (int i = 1; i < s.getRows(); i++) {
        for (int j = 2; j < s.getColumns(); j++)
    {
        a[i][j] = s.getCell(j, i).getContents();
        int x=Integer.parseInt(a[i][j]);
        if(x > 35)
    {
        Label l1 = new Label(6, i, "pass");
        ws.addCell(l1);
    }
}
}

```

```
    }
    else
    {
        Label l1 = new Label(6, i, "fail");
        ws.addCell(l1);
        break;
    }
}

System.out.println("Records sucessfully updated ");
}

wwb.write();
wwb.close();
}

}
```

Output:

RollNo	Name	DBMS	JAVA	LINUX	TOTAL	Result
1	abc	65	67	78	210	pass
2	def	66	68	79	213	pass
3	ghi	67	69	80	216	pass
4	lmn	68	70	32	170	fail
5	xyz	32	71	22	125	fail
6	pqr	70	72	83	225	pass
7	rst	71	73	84	228	pass
8	uvw	72	74	85	231	pass
9	abc1	73	22	86	181	fail
10	abc2	74	76	87	237	pass
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						
31						
32						
33						
34						
35						
36						
37						

Program 11: Write and test a program to provide total number of objects present on a web page using selenium.

External files to be added:

- a) JAR file obtained from selenium-java-4.1.0 folder and sub-folder lib.
- b) Chromedriver.exe file in a folder named driver

The above JAR files have to be added under the Classpath section in Build Path Configuration function.

Program:

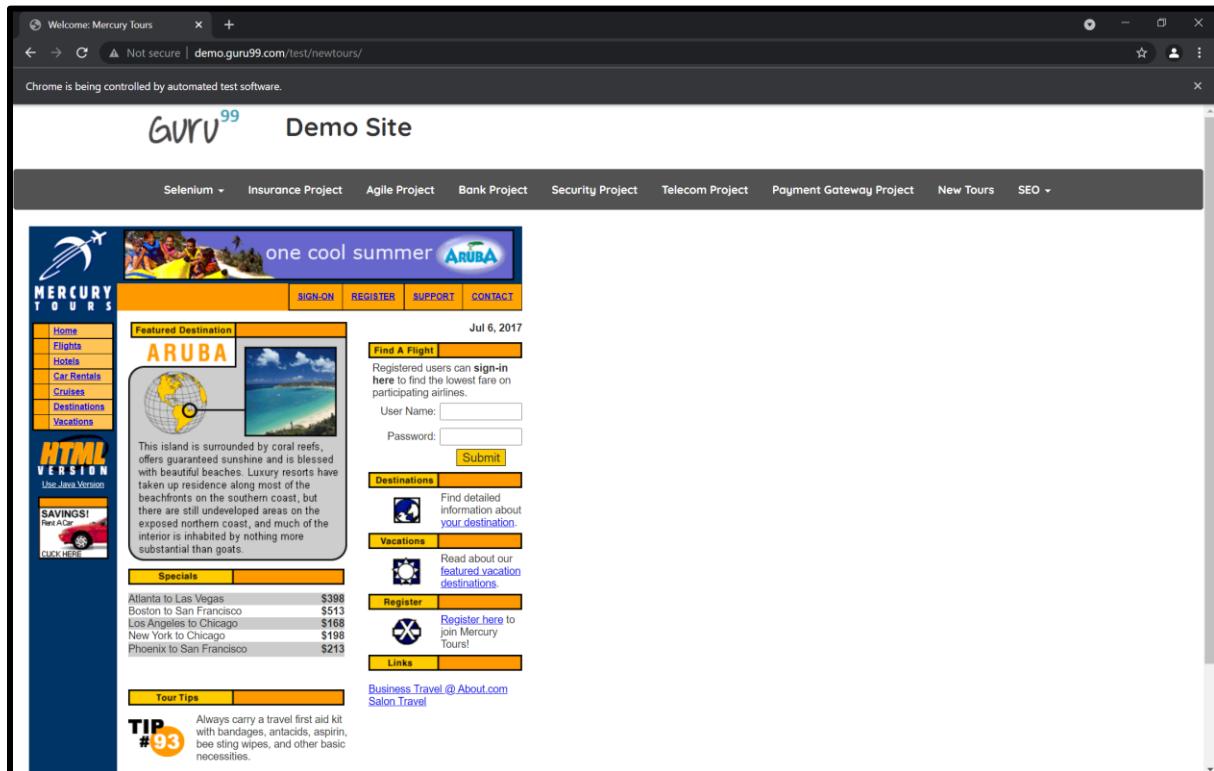
```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import java.util.List;
```

```

public class program_11 {
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\\\driver\\\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window( ).maximize();
        System.out.println("Launching Chrome");
        //driver.get("http://demo.guru99.com/test/newtours/");
        driver.get("https://mail.google.com/mail/");
        List< WebElement> mylist=driver.findElements (By.xpath("//a"));
        System.out.println("Number of links =" +mylist.size());
    }
}

```

Output:



```
eclipse-workspace1 - WebpageLogin/src/program_11.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Problems Javadoc Declaration Search Console Properties Results of running class excel
<terminated> program_11 [Java Application] C:\Users\Sonal Nandagopalan.p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64_16.0.v20210721-1149\jre\bin\javaw.exe (31-Dec-2021, 10:09:04 pm - 10:09:10 pm)
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
log4j:WARN No appenders could be found for logger (io.netty.util.internal.logging.InternalLoggerFactory).
log4j:WARN Please initialize the log4j system properly.
Starting ChromeDriver 96.0.4664.45 (76e4c1bb2ab4671b8beba3444e61cf17584b2fc-refs/branch-heads/4664@{#947}) on port 62644
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Dec 31, 2021 10:09:06 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected browser version W3C
Dec 31, 2021 10:09:06 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 96
INFO: Launching Chrome
Number of links =53
```

Program 12: Demonstrate mobile app testing using APPIUM.

Install Appium and Android Studio And AVD manager, Set env variable.

<https://github.com/appium/appium-desktop/releases/tag/v1.17.0>

Appium-windows-1.17.0

v1.17.0

v1.17.0

▼ Assets (11)

⌚ Appium-1.17.0-mac.zip	163 MB
⌚ Appium-linux-1.17.0.AppImage	119 MB
⌚ Appium-mac-1.17.0.dmg	137 MB
⌚ Appium-mac-1.17.0.dmg.blockmap	148 KB
⌚ Appium-windows-1.17.0.exe	109 MB
⌚ Appium-windows-1.17.0.exe.blockmap	102 KB
⌚ latest-linux.yml	379 Bytes
⌚ latest-mac.yml	517 Bytes
⌚ latest.yml	348 Bytes
⌚ Source code (.zip)	
⌚ Source code (.tar.gz)	

https://github.com/appium/appium-desktop/releases/download/v1.17.0/Appium-windows-1.17.0.exe

Windows Type here to search

18°C ENG 7:20 AM IN 1/6/2022

[Download Android Studio and SDK tools | Android Developers](#)

<https://developer.android.com/studio/>

android-studio-2020.3.1.26-windows

developers

Platform

Android Studio

Google Play

Jetpack

More

Search

English

android studio

Android Studio provides the fastest tools for building apps on every type of Android device.

Download Android Studio

2020.3.1 for Windows 64-bit (914 MB)

Download options

Release notes

AndroidStudioProject Application src main res

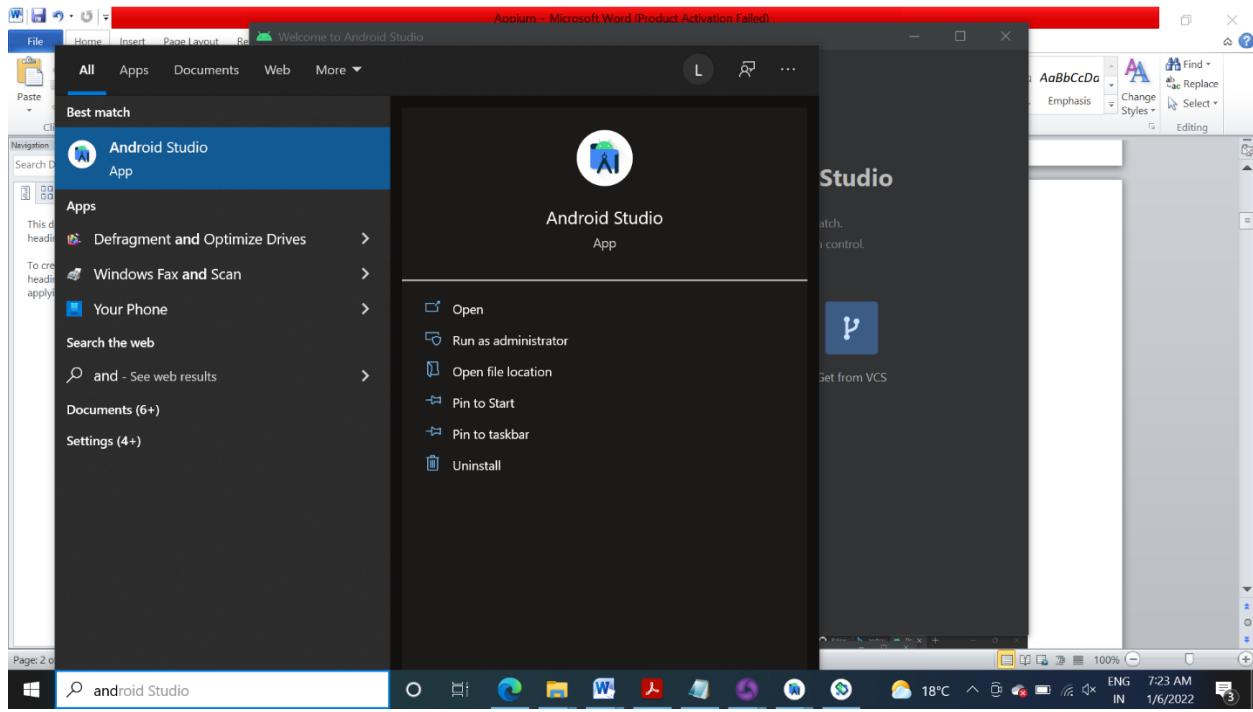
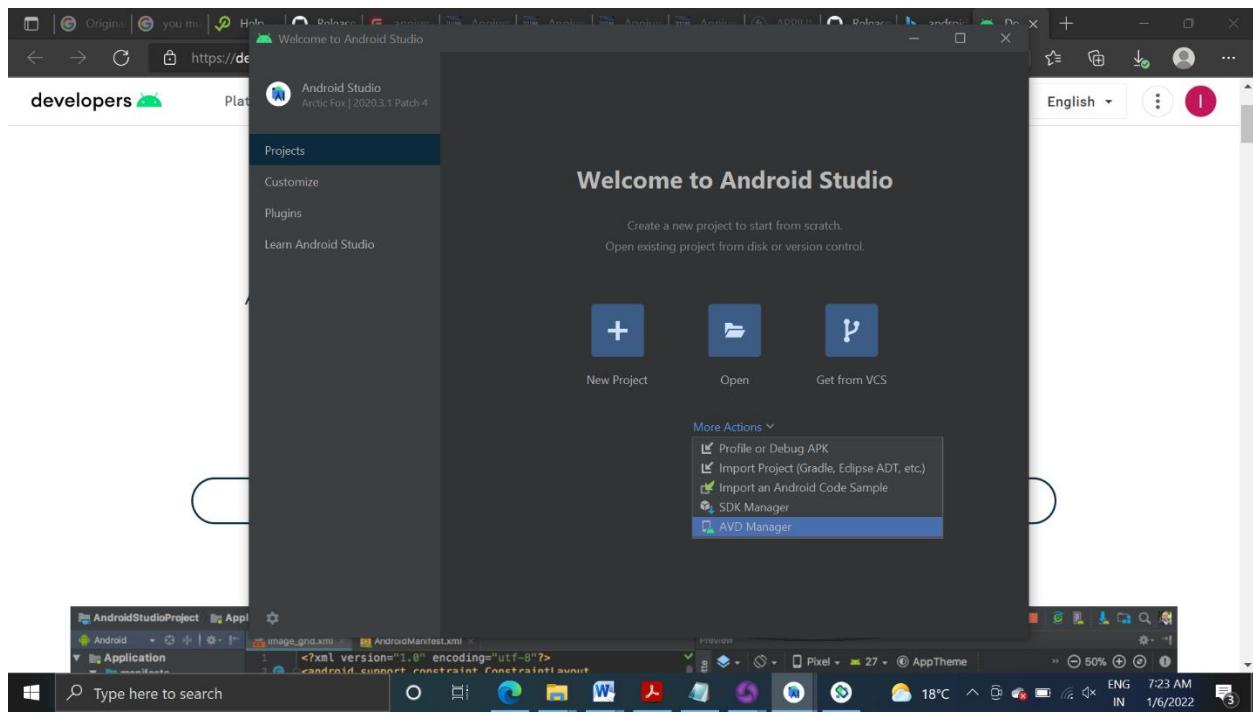
Android image_grid.xml AndroidManifest.xml

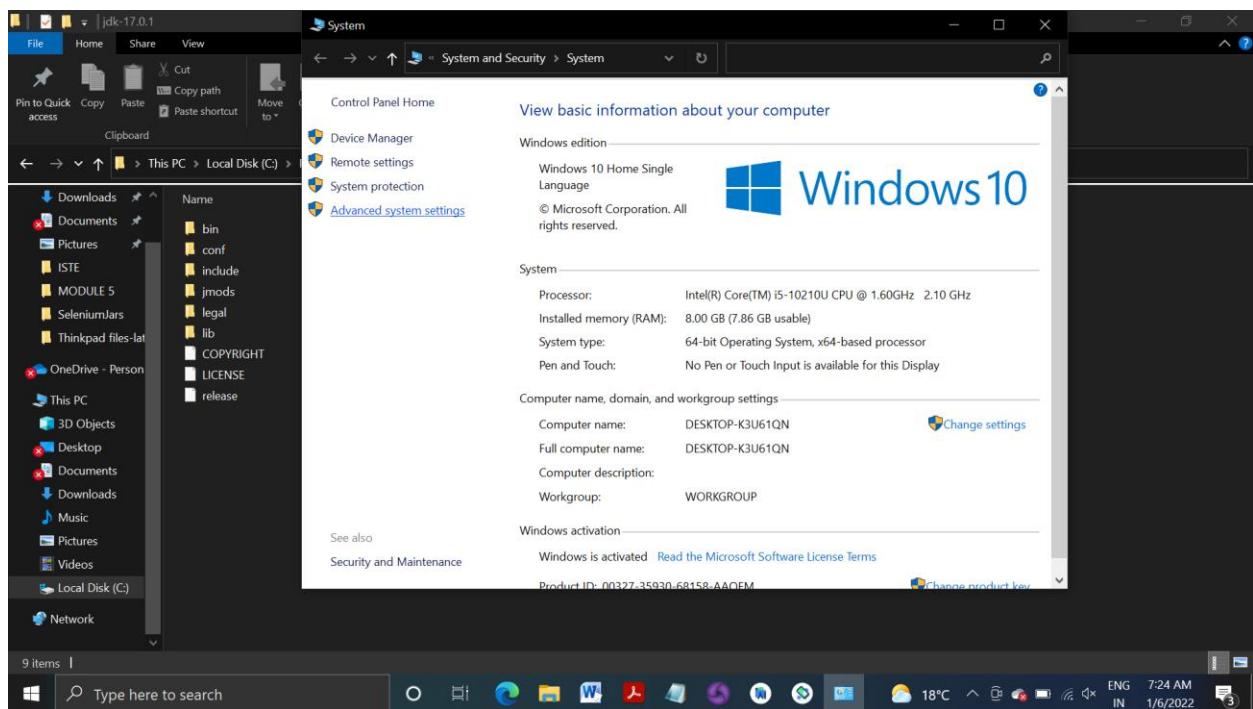
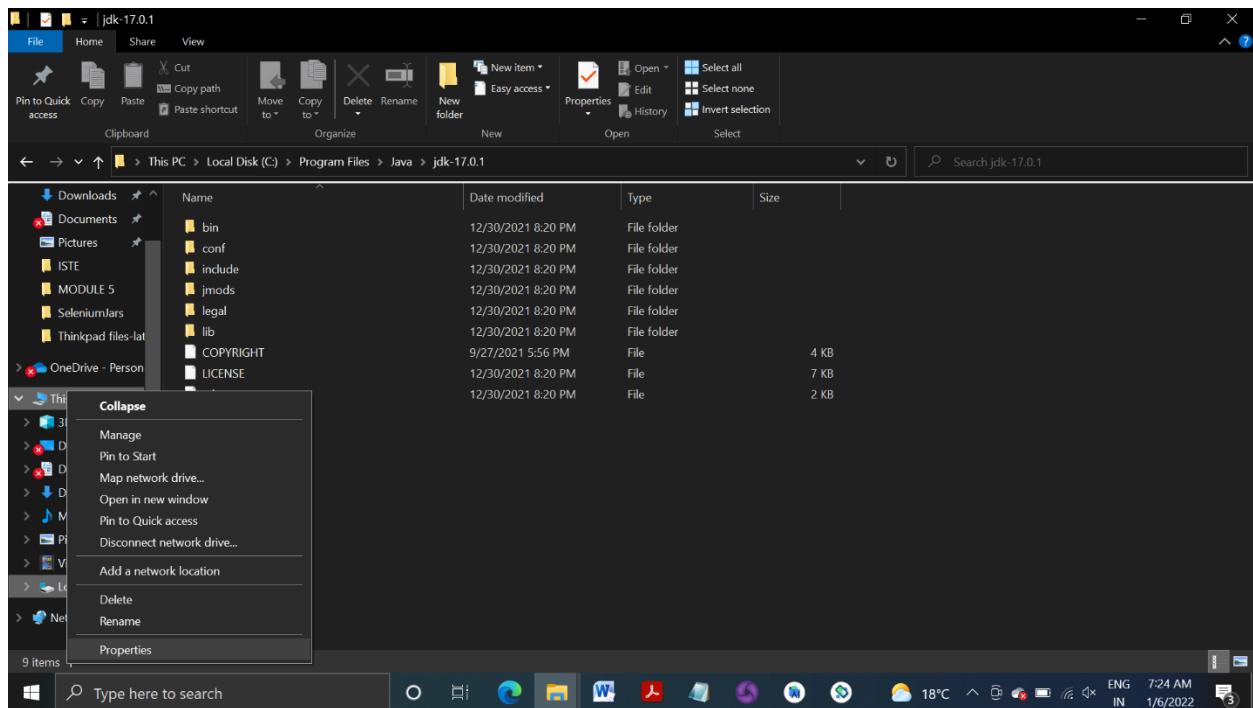
```
<?xml version="1.0" encoding="utf-8"?>
<manifest ...>
```

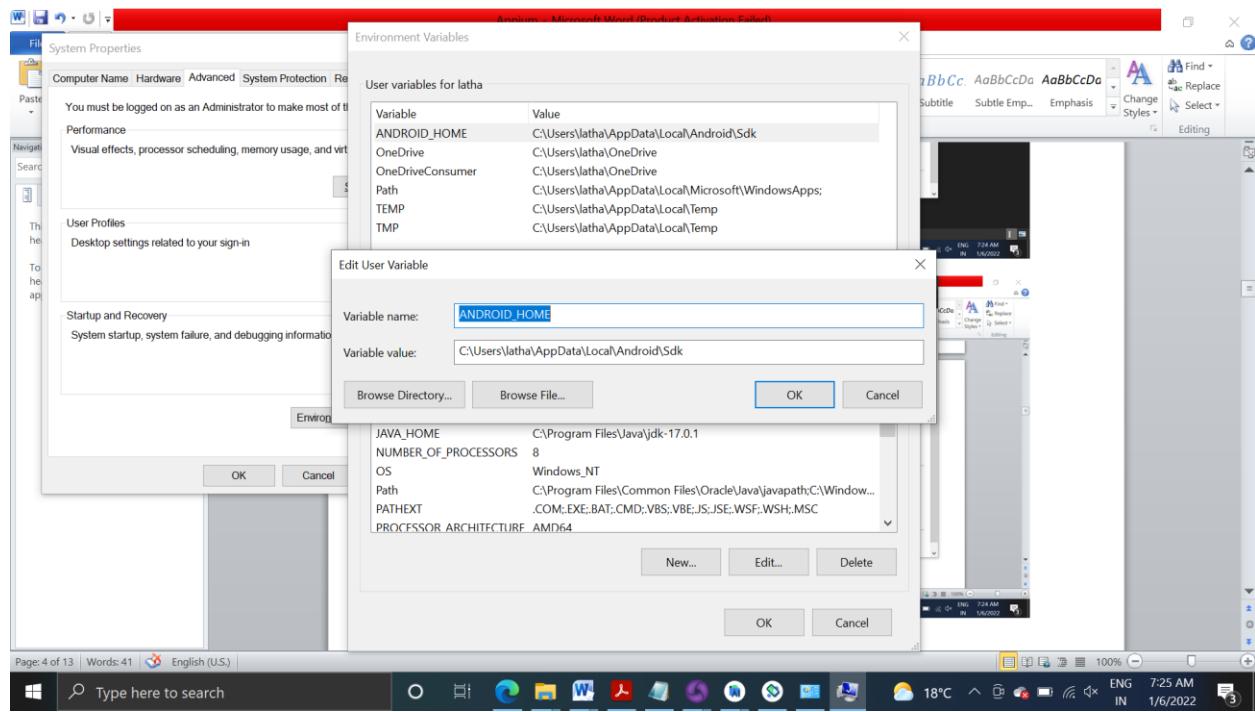
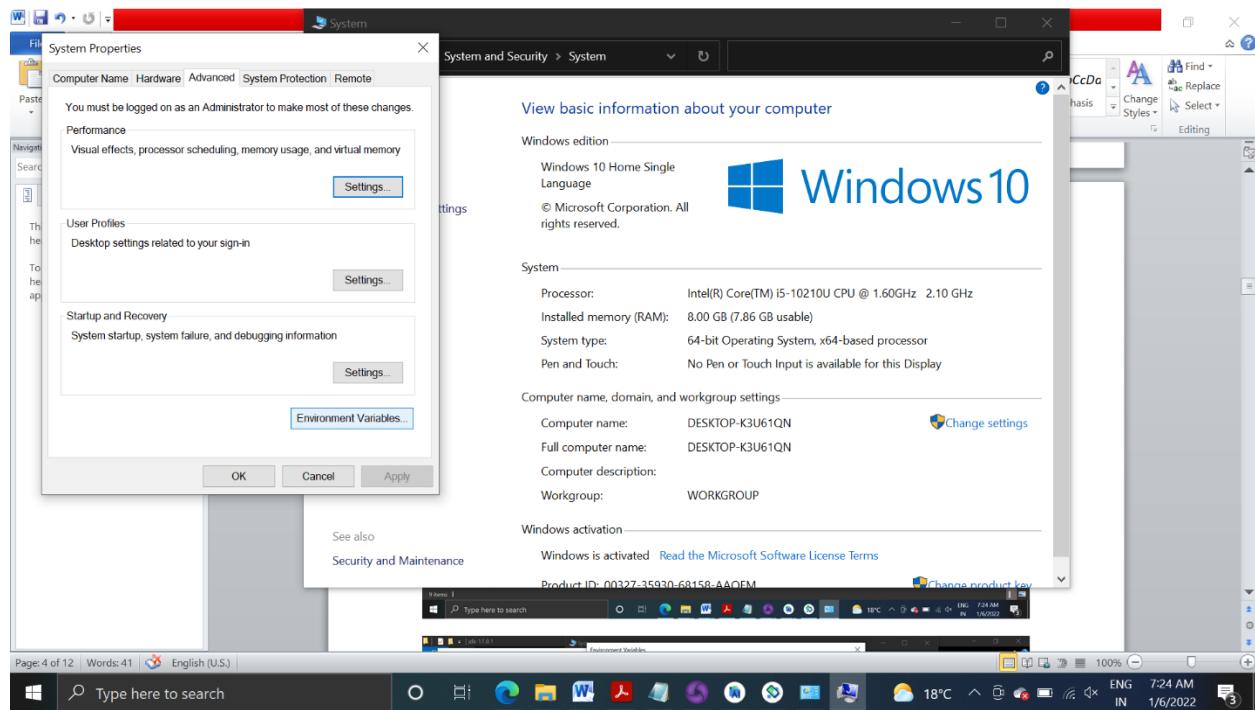
Preview

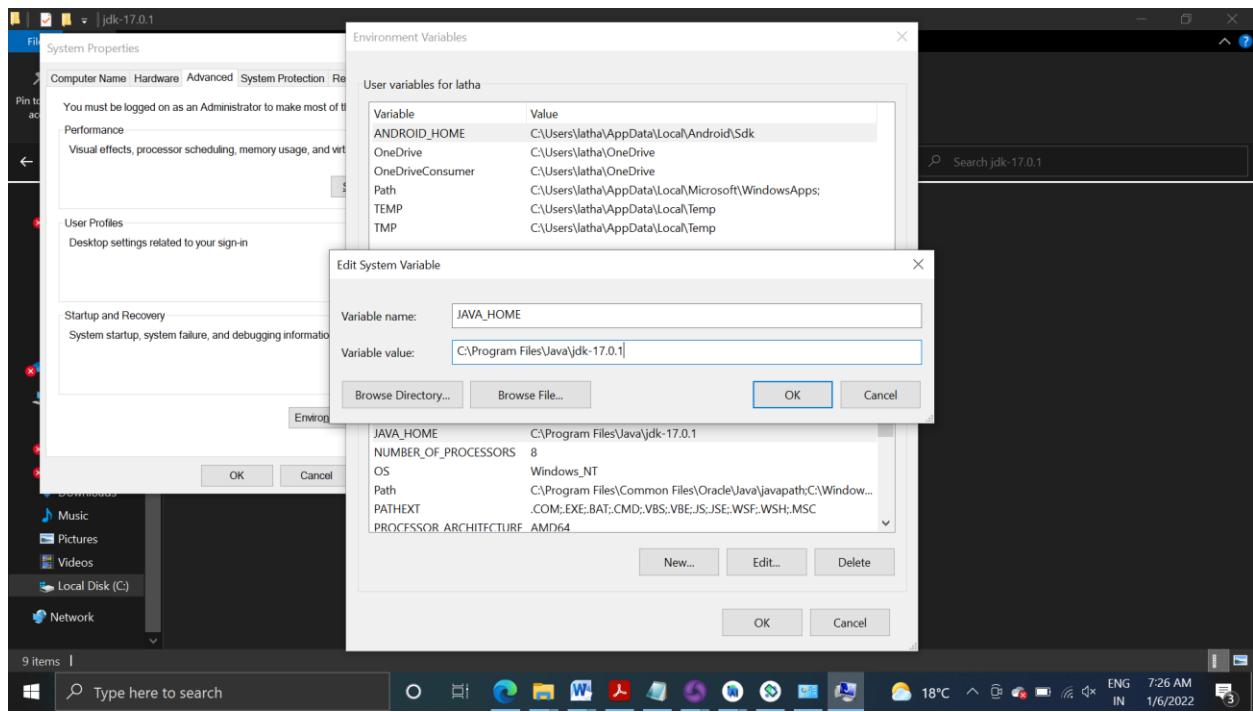
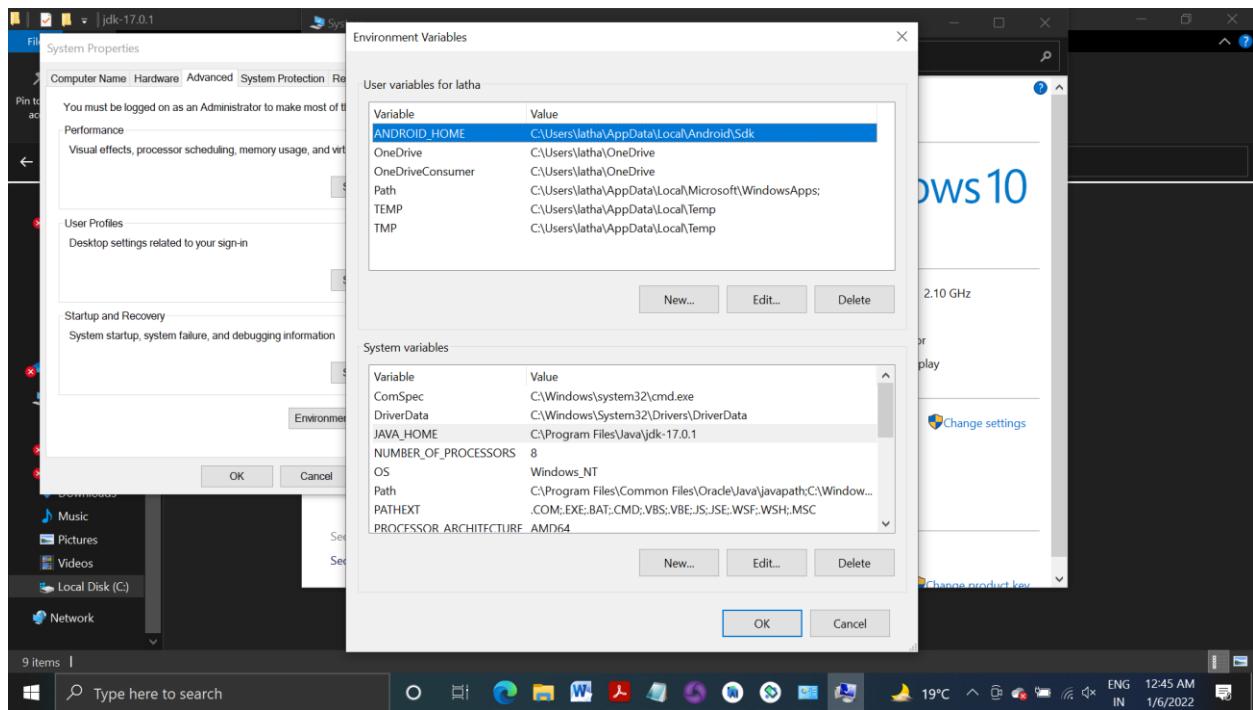
Type here to search

18°C ENG 7:23 AM IN 1/6/2022

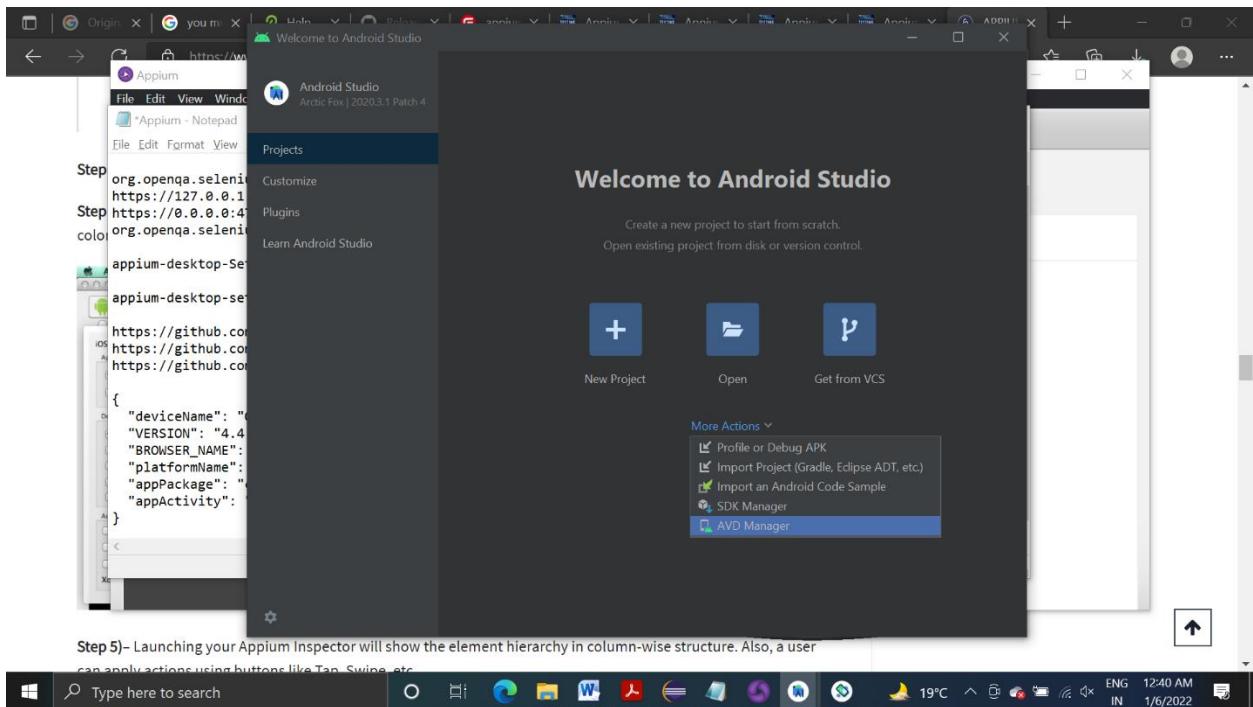


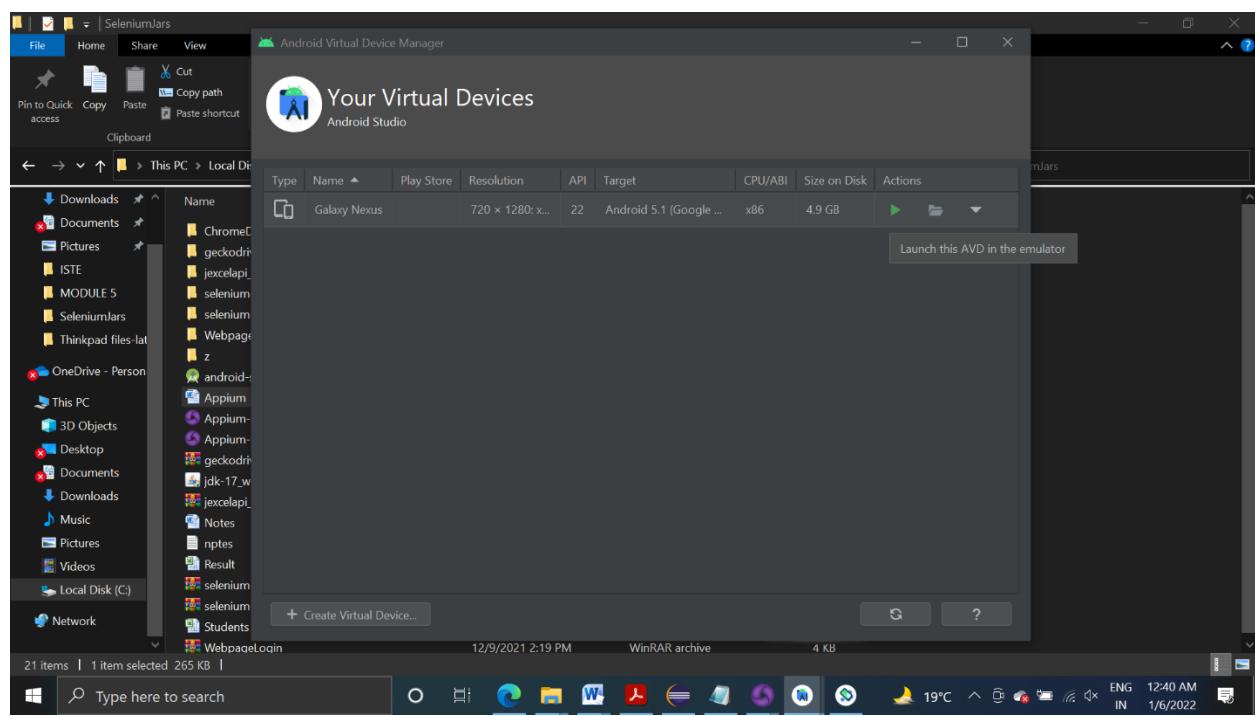
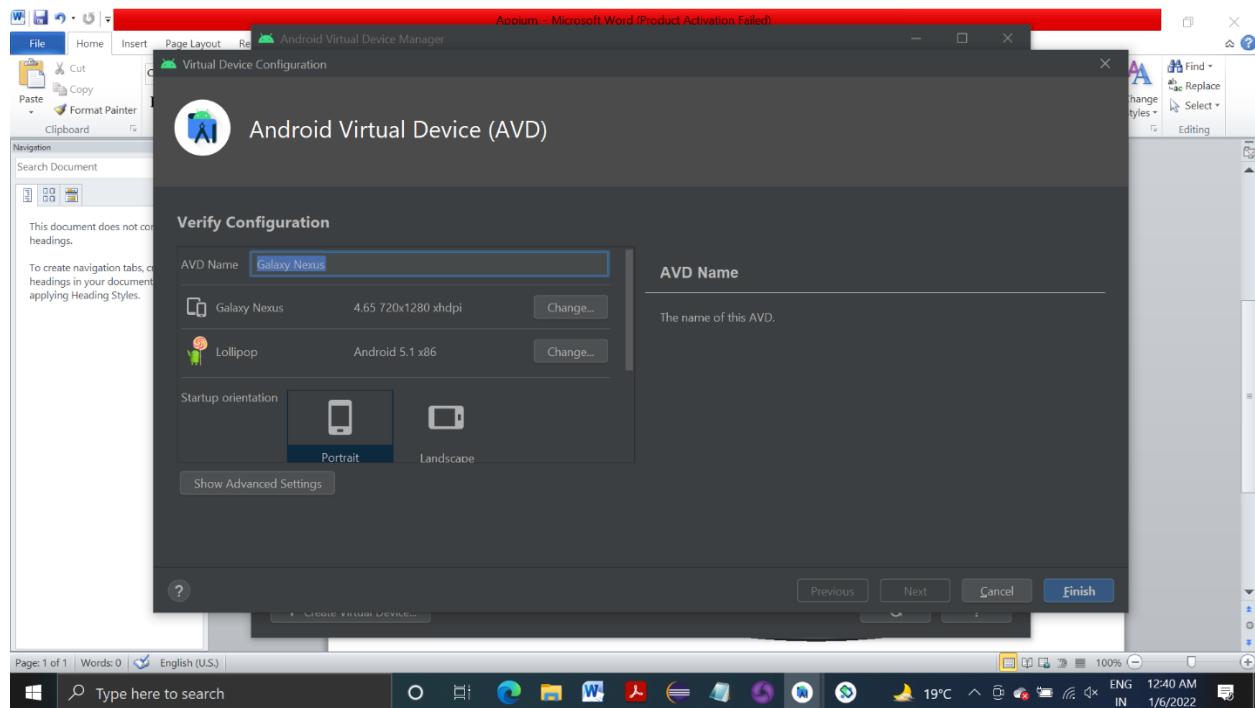


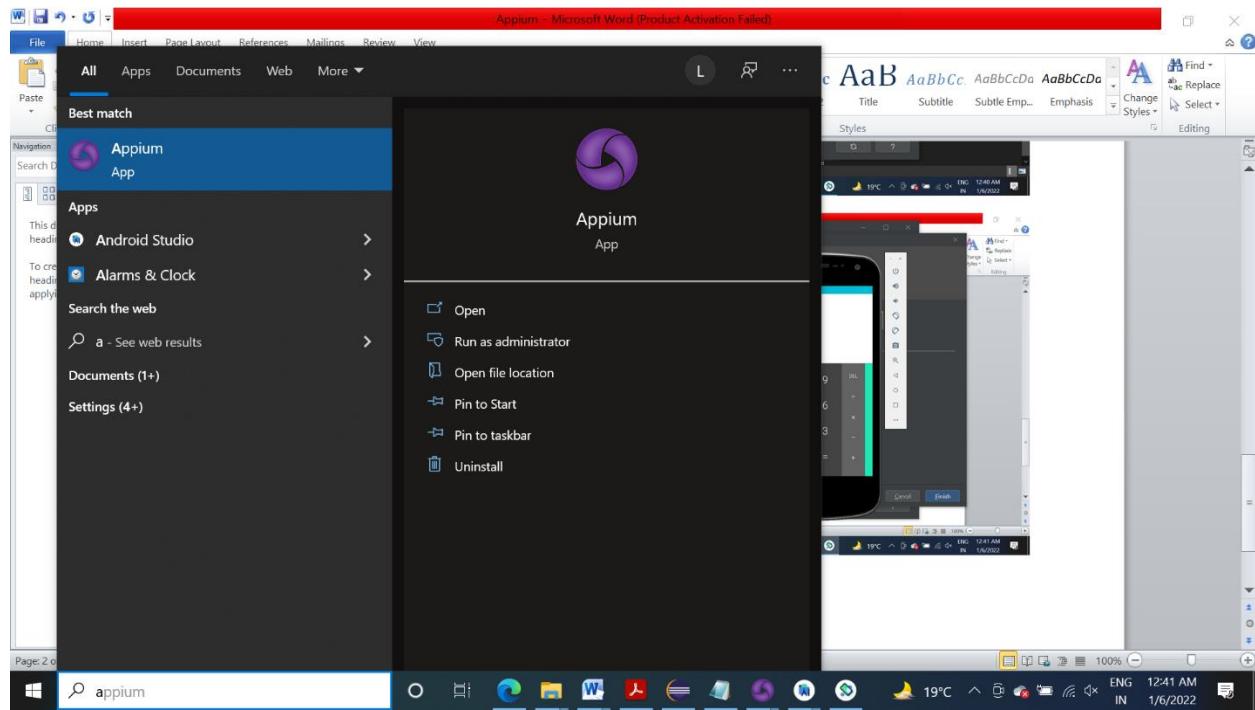
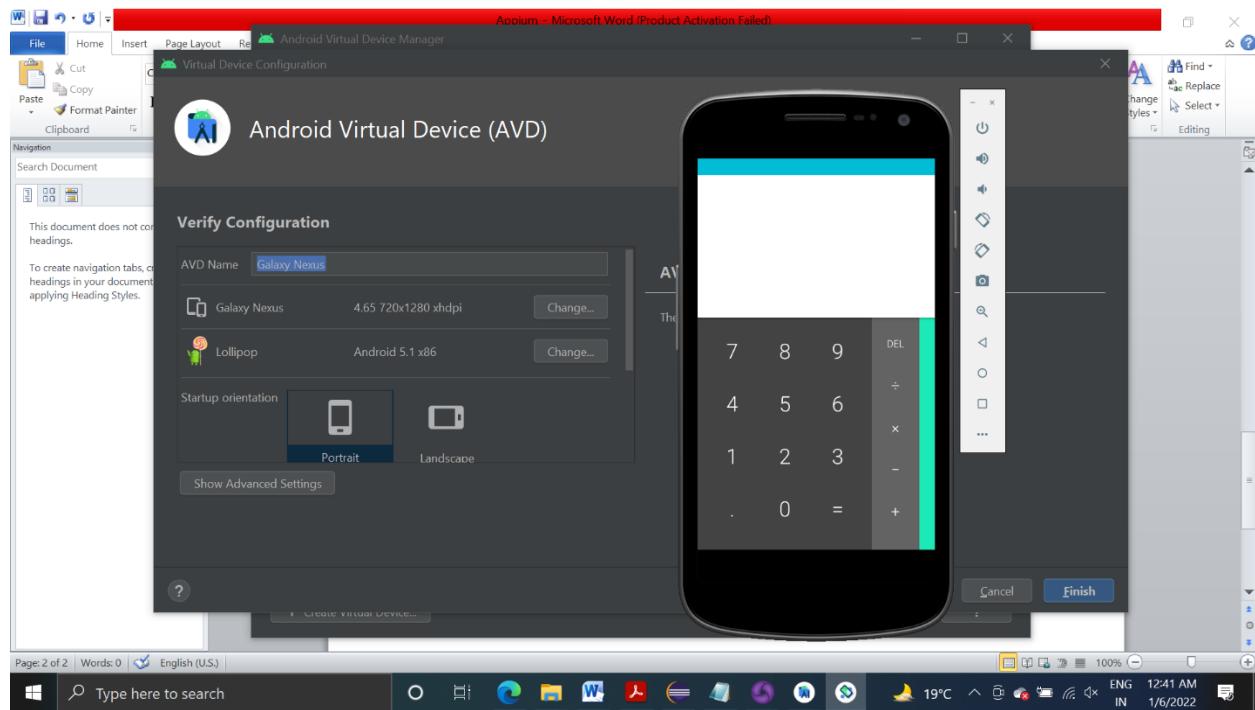


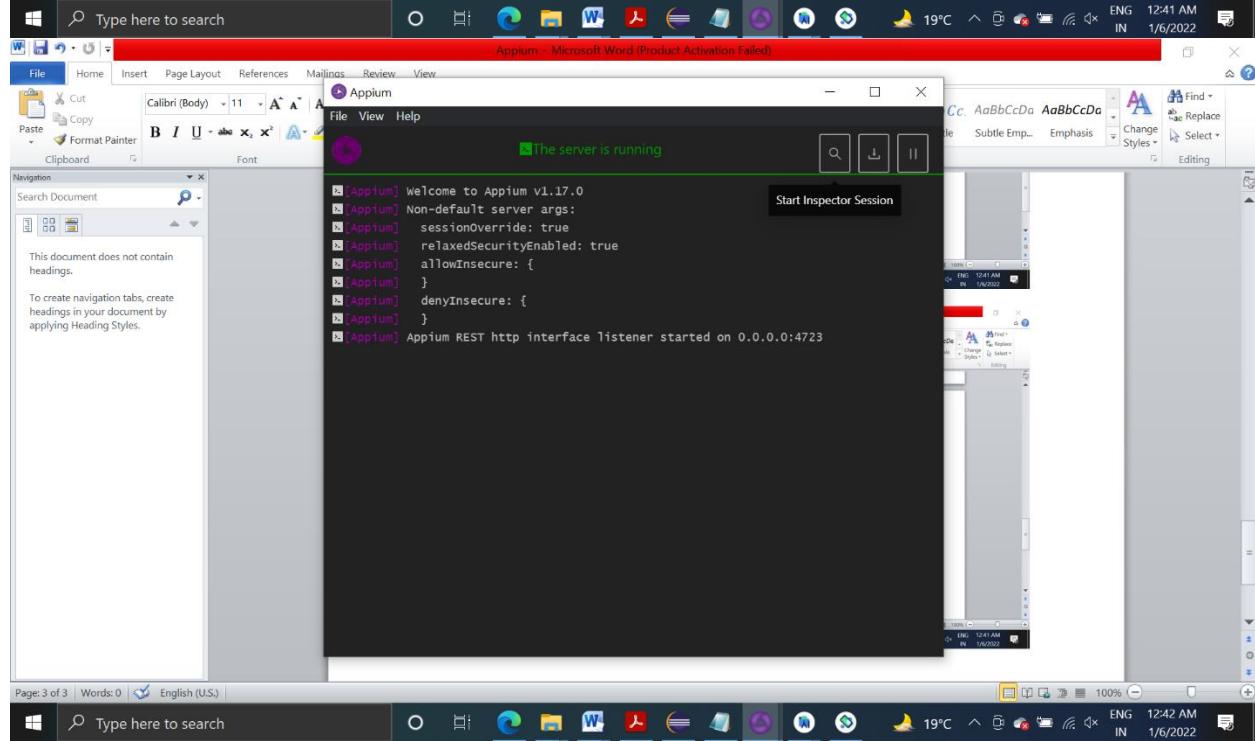
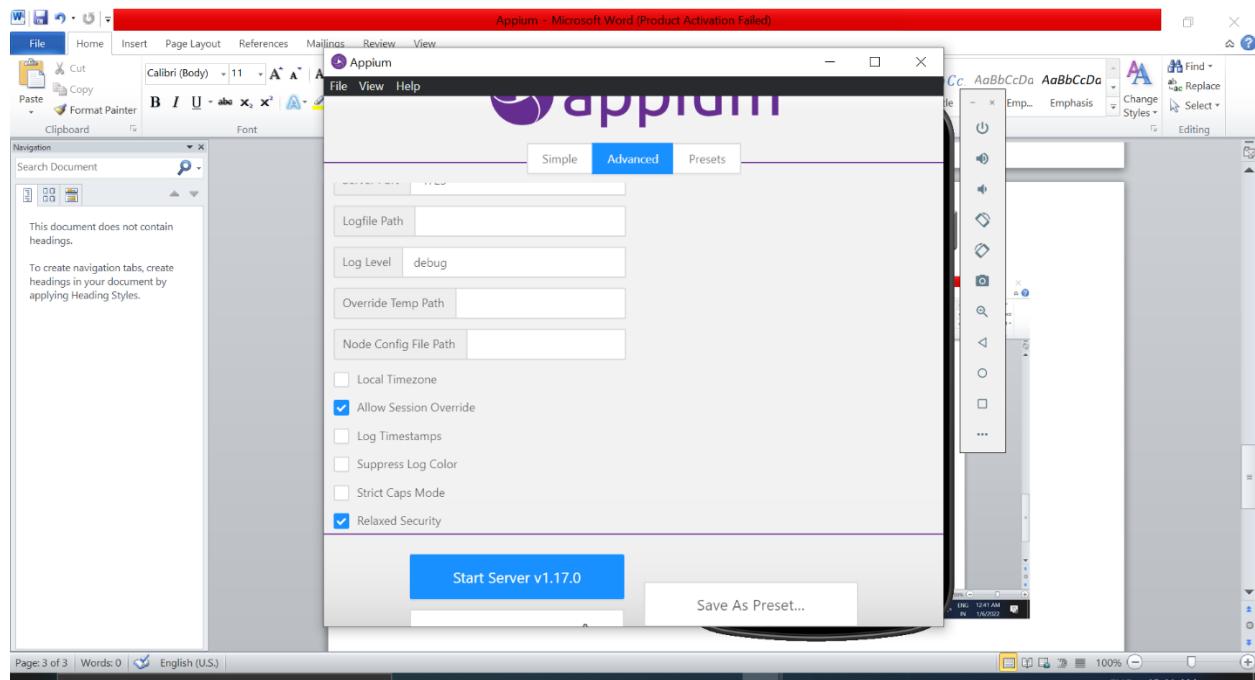


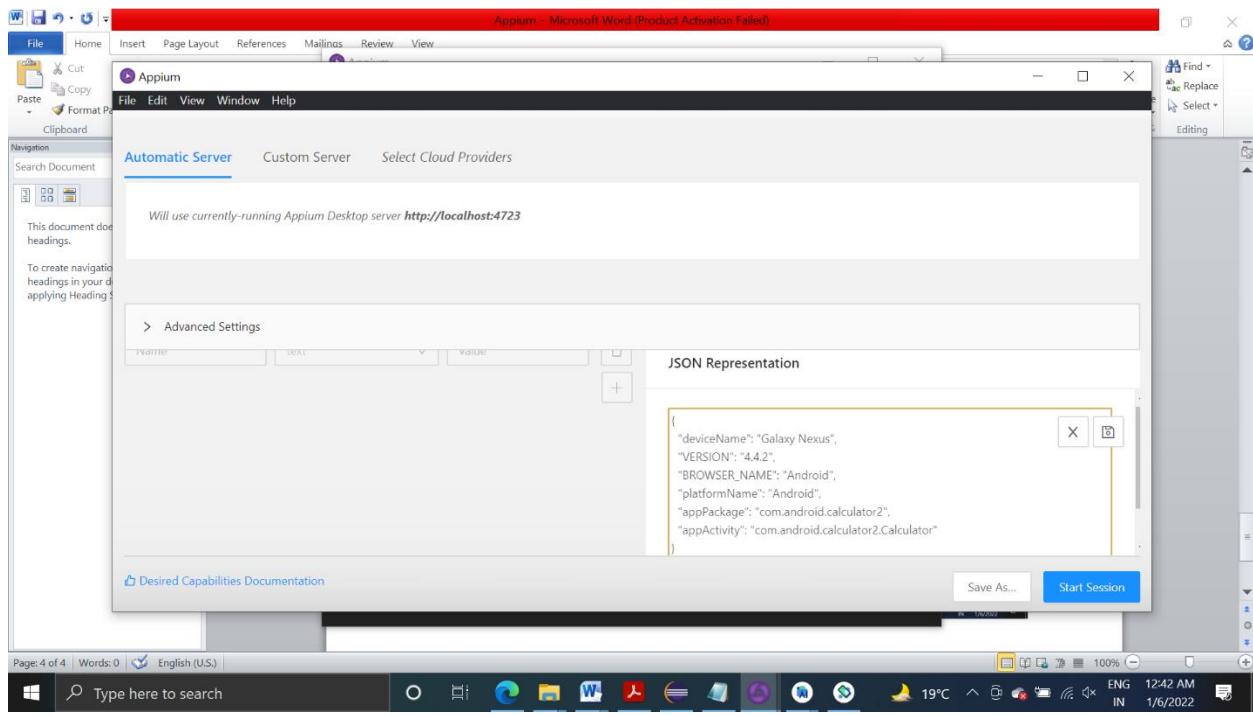
Start AVD











Page: 4 of 4 | Words: 0 | Type here to search | 100% | 19°C | ENG 12:42 AM IN 1/6/2022

\

{

```
"deviceName": "Galaxy Nexus",
"VERSION": "4.4.2",
"BROWSER_NAME": "Android",
"platformName": "Android",
"appPackage": "com.android.calculator2",
"appActivity": "com.android.calculator2.Calculator"}
```

