LAB 4 REPORT

Name: Osama Ayman Mokhtar Amin

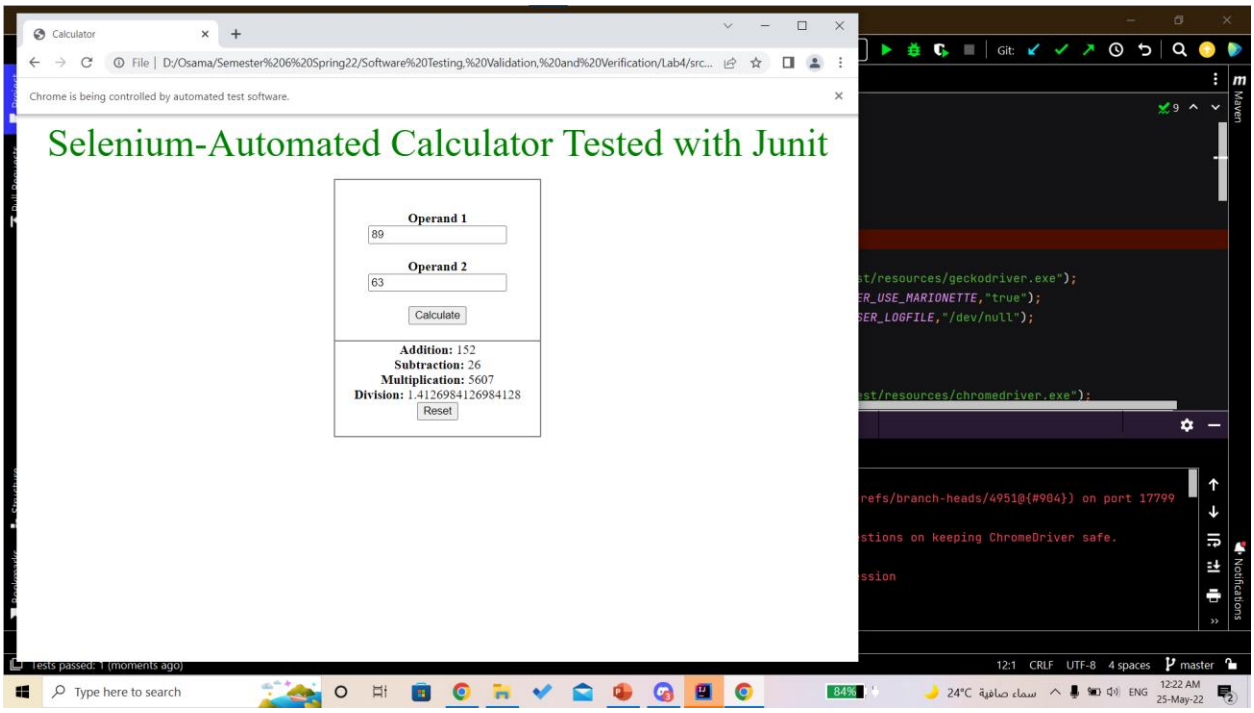
ID: 19P1609

Course Name: Software Testing, Validation, and Verification

Course Code: CSE 338

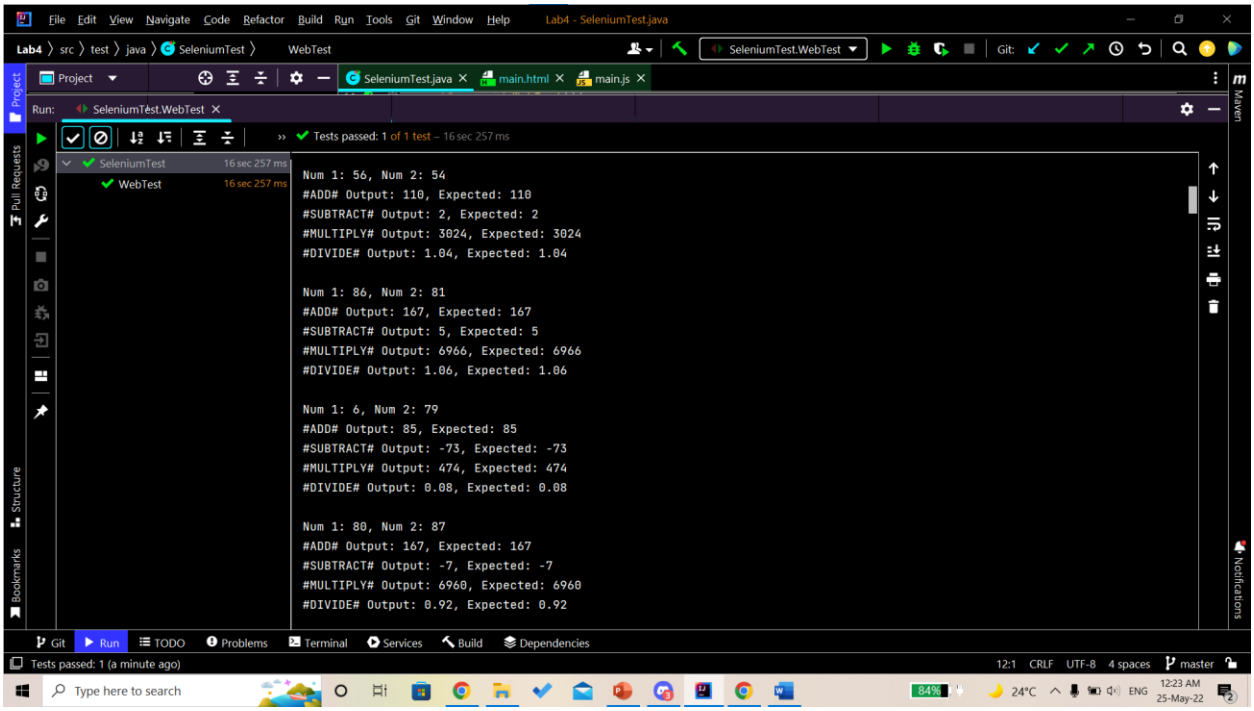
Description

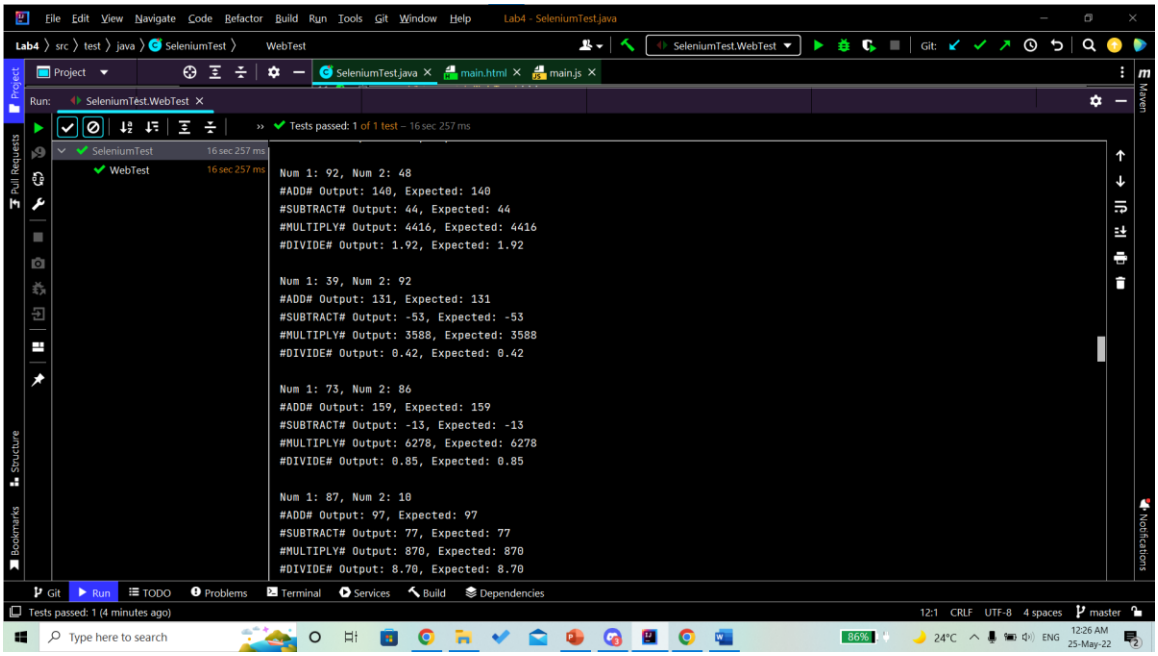
This project is about a web-based calculator. It uses selenium combined with Junit to automate the testing of this calculator. We randomly generated 2 numbers and send them to the calculator using selenium then we take the result of adding, subtracting, and dividing these numbers and compare it with the expected output using Junit.



Testing Output

We are generating 100 random numbers for each operand and testing them using assertEquals() in Junit.





Code

```
24          for (int k=0;k<100;k++){
25              int i = (int)(Math.random()*100);
26              int j = (int)(Math.random()*100);
27              driver.findElement(By.id("firstnum")).clear();
28              driver.findElement(By.id("secondnum")).clear();
29              driver.findElement(By.id("firstnum")).sendKeys(String.valueOf(i));
30              driver.findElement(By.id("secondnum")).sendKeys(String.valueOf(j));
31              driver.findElement(By.id("calcbtn")).click();
32              String add = driver.findElement(By.id("add")).getText();
33              String sub = driver.findElement(By.id("sub")).getText();
34              String mult = driver.findElement(By.id("mult")).getText();
35              String divide = driver.findElement(By.id("divide")).getText();
36              System.out.printf("Num 1: %d, Num 2: %d\n", i, j);
37              System.out.printf("#ADD# Output: %d, Expected: %d\n", Integer.parseInt(add),i+j);
38              assertEquals(Integer.parseInt(add), actual: i+j);
39              System.out.printf("#SUBTRACT# Output: %d, Expected: %d\n", Integer.parseInt(sub),i-j);
40              assertEquals(Integer.parseInt(sub), actual: i-j);
41              System.out.printf("#MULTIPLY# Output: %d, Expected: %d\n", Integer.parseInt(mult),i*j);
42              assertEquals(Integer.parseInt(mult), actual: i*j);
43              System.out.printf("#DIVIDE# Output: %.2f, Expected: %.2f\n\n", Double.parseDouble(divide),(double)i/j);
44              if(j==0){
45                  if(i==0){
46                      assertEquals(divide, actual: "NaN");
47                      continue;
48                  }
49                  assertEquals(divide, actual: "Infinity");
50                  continue;
51              }
52              assertEquals(Double.parseDouble(divide), actual: (double) i/j, delta: 0.01);
53          }
54      }
55  }
```