

Learning Report – Portfolio



GLOBAL
ENGINEERING
ACADEMY

Genesis



L&T Technology Services



Document History

Ver. Rel. No.	Release Date	Prepared. By	Reviewed By	Approved By	Remarks/Revision Details
1	31/01/2021	Prathviraj (99003123)			
2					
3					

Table of Contents

COURSE 1: SDLC.....	5
SDLC MINI PROJECT:	5
<i>CI/CD workflows for Project</i>	5
COURSE 2: CORE JAVA	6
ONLINE EXAMINATION MINI PROJECT:.....	6
<i>GitHub Link:</i>	7
COURSE 3: ADVANCED PYTHON	8
ROCK PAPER SCISSOR GAME:	8
<i>GitHub Link:</i>	8
COURSE 4: DATA ANALYTICS	9
ANALYSIS OF COMPLAINTS IN NEW YORK POLICE DEPARTMENT:	9
<i>GitHub link:</i>	15
COURSE 5: TDLC.....	16
BASIC CALCULATOR PROJECT WITH TEST CASES:	16
<i>GitHub Link:</i>	16
COURSE 6: ROBOT FRAMEWORK.....	17
1. SELENIUM:.....	17
<i>GitHub Link:</i>	18
2. APPIUM:	18
<i>GitHub Link:</i>	18
COURSE 7: NETWORKING	19
1. CISCO PACKET TRACER:	19
2. WIRE SHARK:	20

Table of Figures:

Figure 1: Medical store Management	5
Figure 2: Online Test	6
Figure 3: Test Result	7
Figure 4: Rock Paper Scissors	8
Figure 5: Cases in Particular time.....	9
Figure 6: Crime rate Map	10
Figure 7: Case status in each city	11
Figure 8: Cases in particular time.....	12
Figure 9: Case Status.....	12
Figure 10: Case status in each city	13
Figure 11: Types of cases.....	14
Figure 12: Types of cases in New York City.....	15
Figure 13; Test Cases For Calculator	16
Figure 14: TC 1	17
Figure 15: TC 2.....	17
Figure 16: Appium TC 1	18
Figure 17: 1 Router Configuration	19
Figure 18: 2 Router Configuration	20
Figure 19: Wire Shark.....	20

Course 1: SDLC

SDLC Mini Project:

In this course we made learning report of the mini project which we made in Step-in. The project consists of developing and implementing a Medical store management system that will automate all the process in a medical store. The developed system will reduce manual work that is required for keeping all the records of sellers and the customers. The system can able to provide any no. of repetitions and keep the record of the total customers.

This can also generate the bill for the customers.



The screenshot displays a terminal window titled "MEDICINE PURSHASE" (sic). It contains a form for entering medicine details. The fields and their values are as follows:

Field	Value
MEDICINE ID	1232
MEDICINE NAME	Sinex
ENTER RACK NO	12
CABNIT NO	104
COMPANY NAME	Herbal
SUPPLIER NAME	Om Pharmacy
UNIT COST Rs.:	10
SALE COST Rs.:	15
QUANTITY	50
MFG.DATE<dd-mm-yyyy>	2014-04-06
EXP.DATE<dd-mm-yyyy>	2014-12-30

Below the form, a summary line shows: TOTAL SALE COST = Rs. 750.00 TOTAL UNIT COST = Rs. 500.00. At the bottom, there are options to "Save" or "Cancel", and a prompt: "Press First charecter for the operation :".

Figure 1: Medical store Management

CI/CD workflows for Project

[Medical Store Management System](#)

Course 2: Core Java

Online Examination Mini Project:

This project is a mini exam conducting program which contains of 10 multiple choice question and options which concludes with the user shown the results as per the options selected.

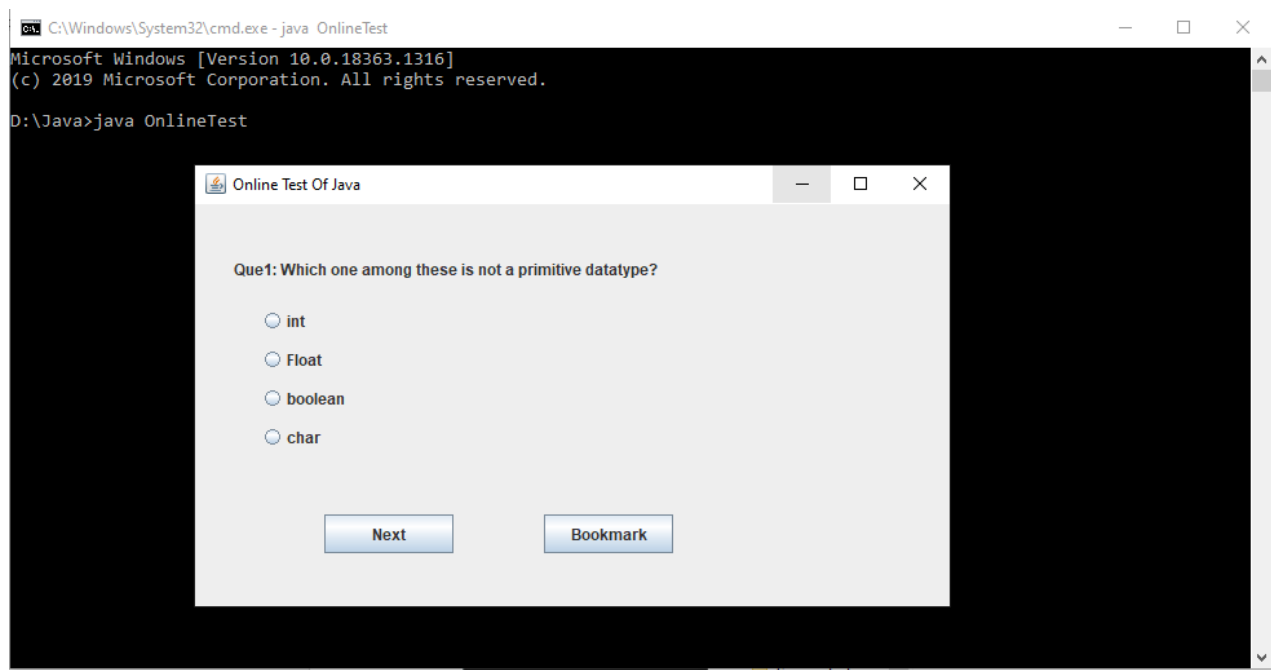


Figure 2: Online Test

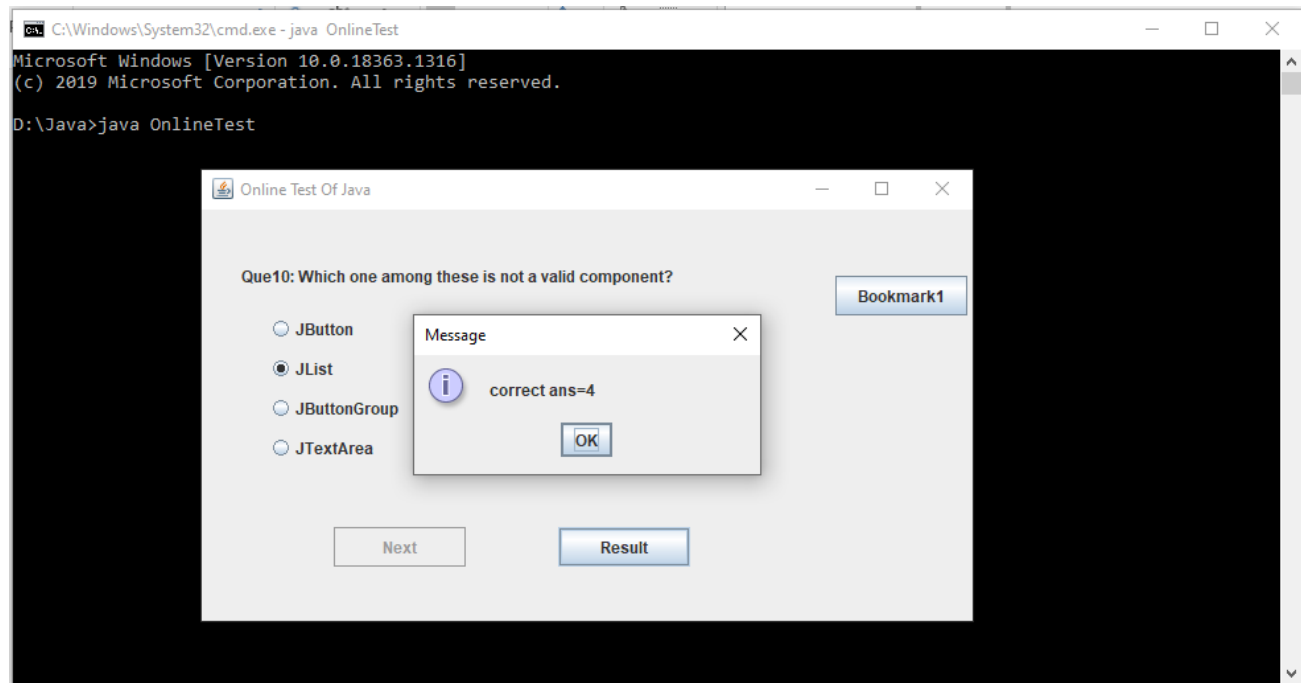


Figure 3: Test Result

GitHub Link:

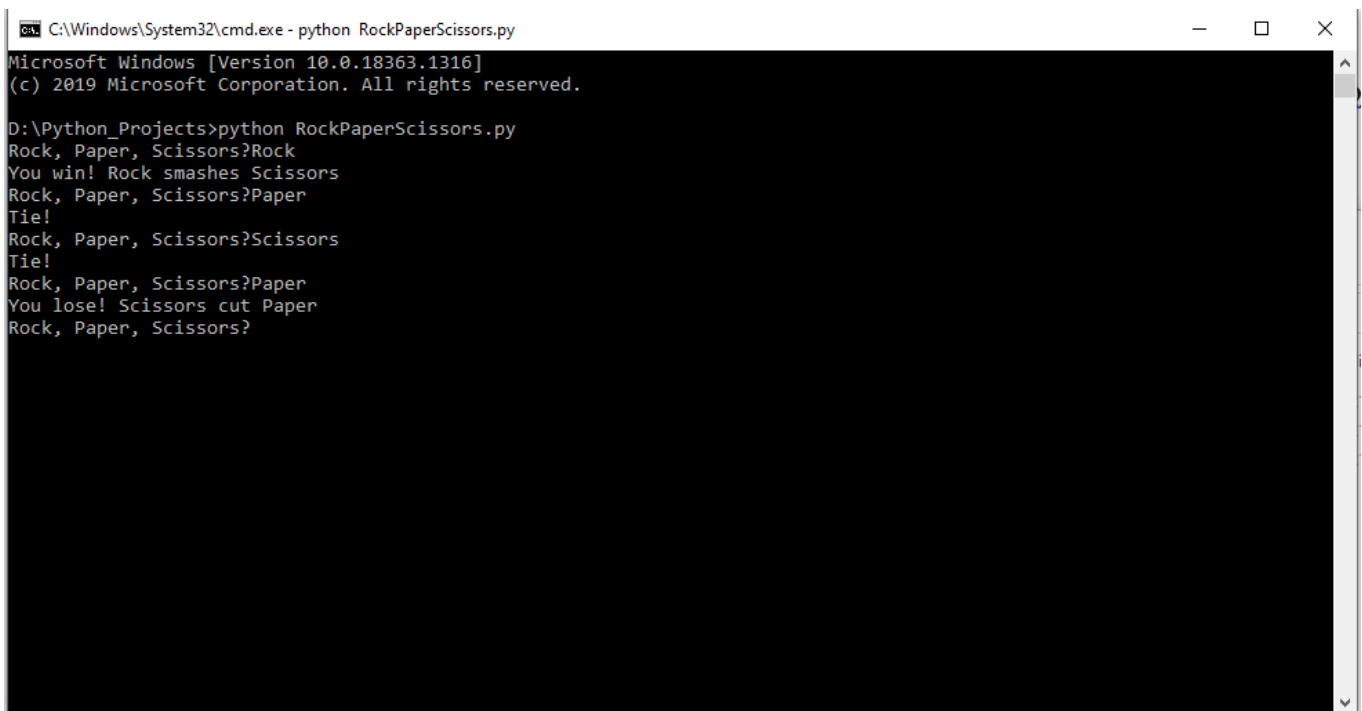
[Online Test](#)

Course 3: Advanced Python

Rock Paper Scissor Game:

Here I have used List to select any of rock paper or scissors.

The computer uses the random value for list and selects winner if the Paper covers Rock or Rock smashes scissors or Scissors tears the paper. It will give the output as Tie if the both selects the same otherwise It displays that I lost the game.



```
C:\Windows\System32\cmd.exe - python RockPaperScissors.py
Microsoft Windows [Version 10.0.18363.1316]
(c) 2019 Microsoft Corporation. All rights reserved.

D:\Python_Projects>python RockPaperScissors.py
Rock, Paper, Scissors?Rock
You win! Rock smashes Scissors
Rock, Paper, Scissors?Paper
Tie!
Rock, Paper, Scissors?Scissors
Tie!
Rock, Paper, Scissors?Paper
You lose! Scissors cut Paper
Rock, Paper, Scissors?
```

Figure 4: Rock Paper Scissors

GitHub Link:

[Rock Paper Scissor](#)

Course 4: Data Analytics

Analysis of Complaints in New York Police Department:

Here in this I tried to perform different operations with roughly 3 lac records in csv format.

How much better we can understand the data using pandas, numpy & matplotlib. Below are the screenshots of the plot.

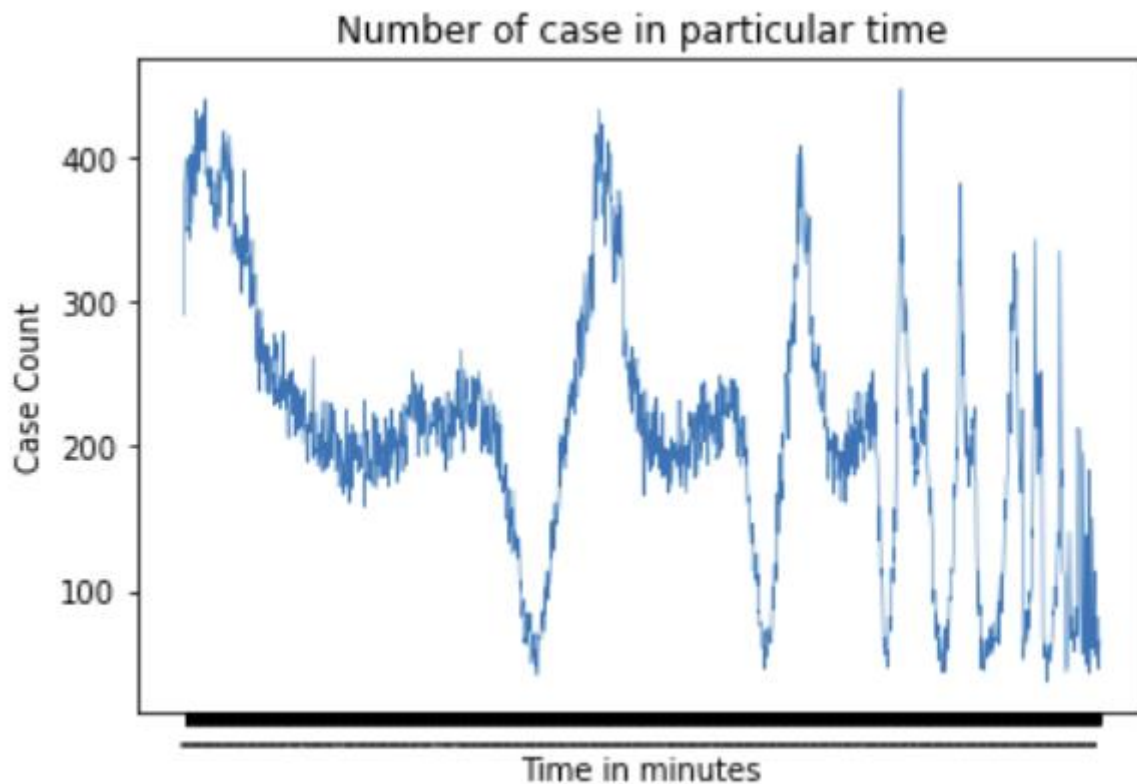


Figure 5: Cases in Particular time

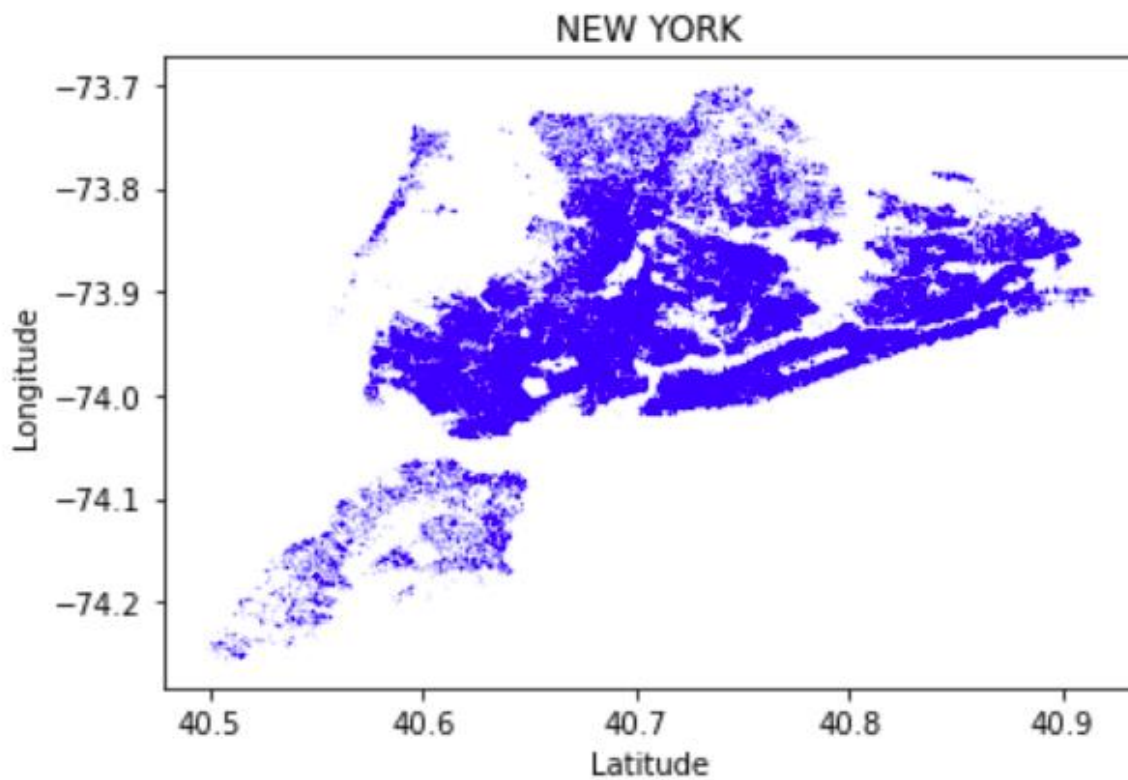


Figure 6: Crime rate Map

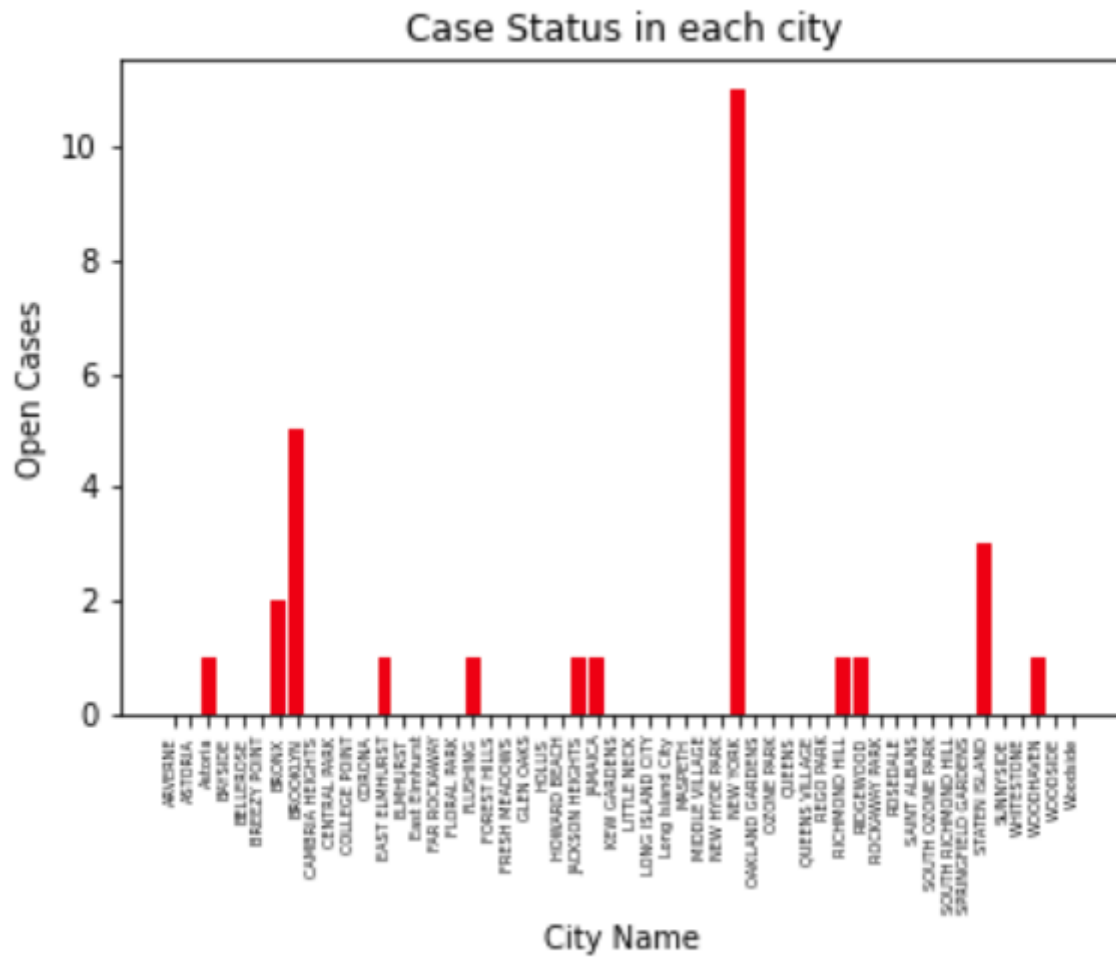


Figure 7: Case status in each city

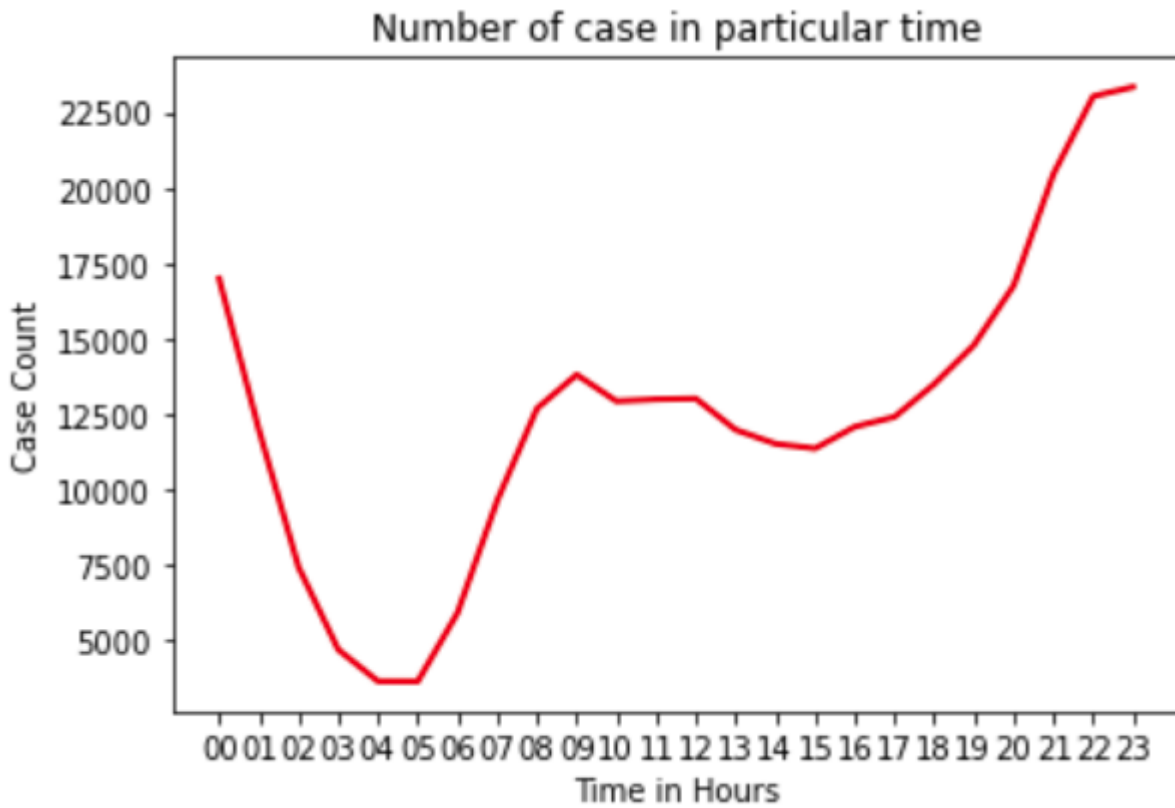


Figure 8: Cases in particular time

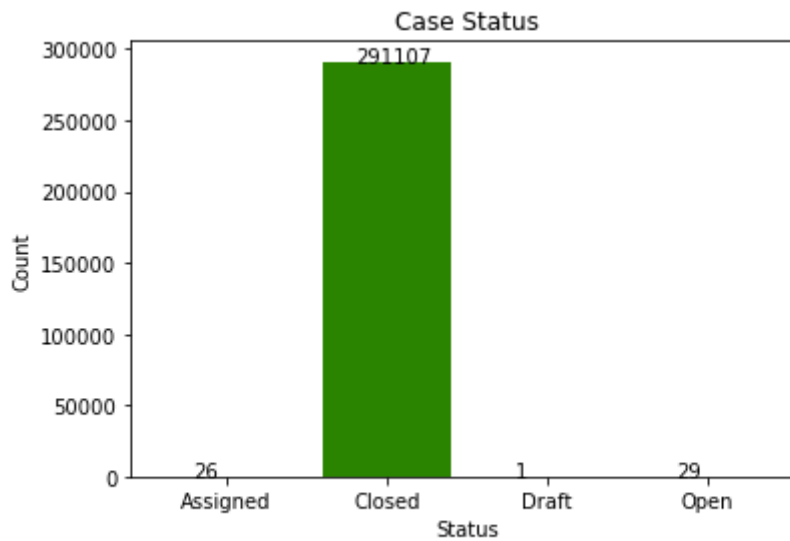


Figure 9: Case Status

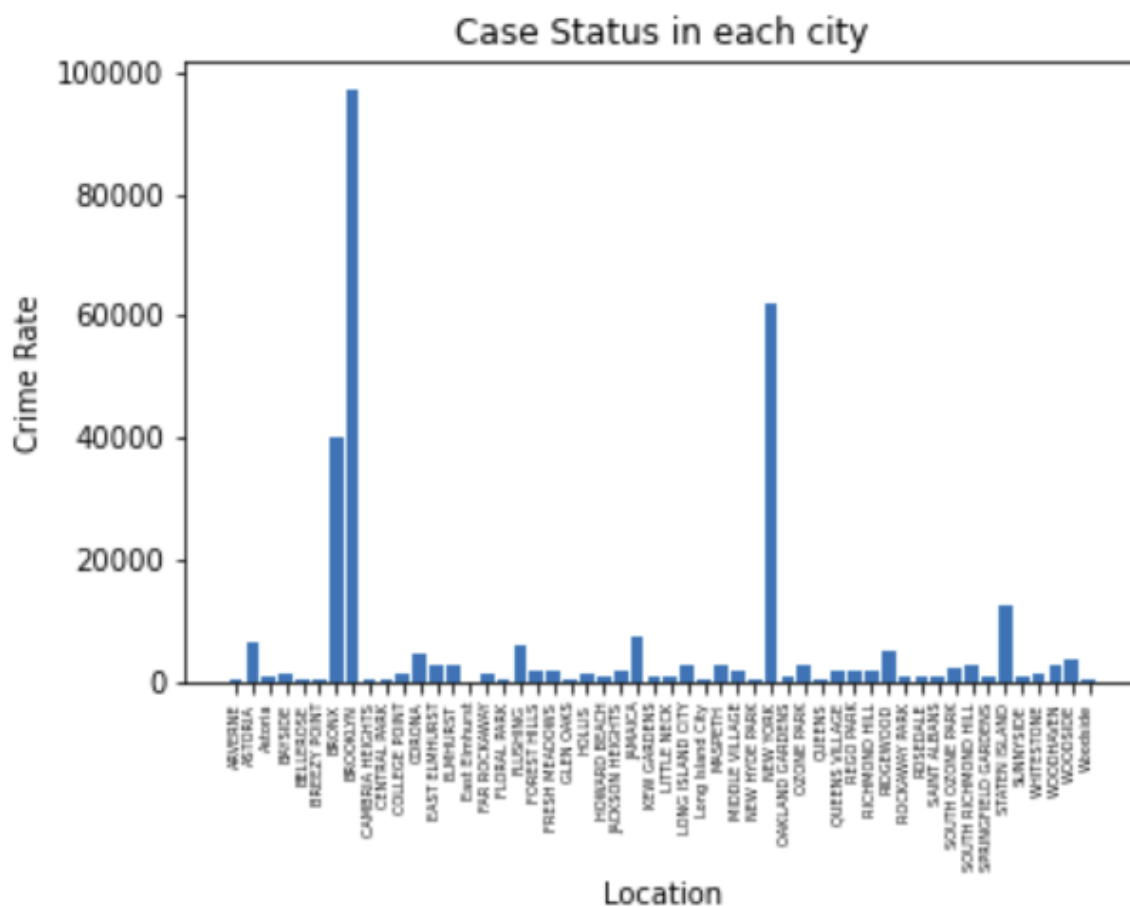


Figure 10: Case status in each city

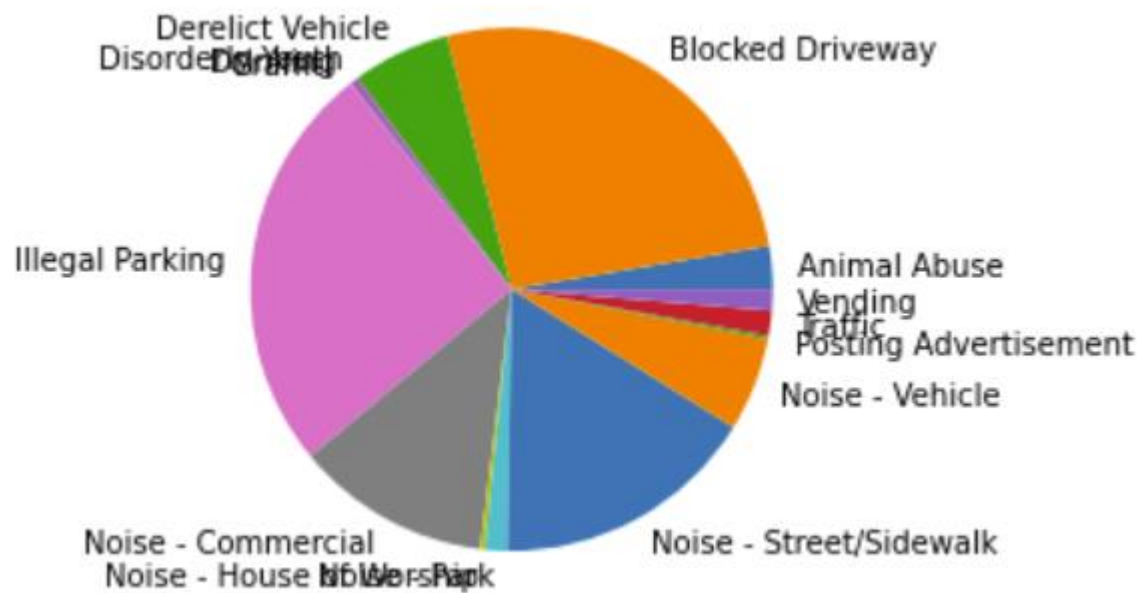


Figure 11: Types of cases

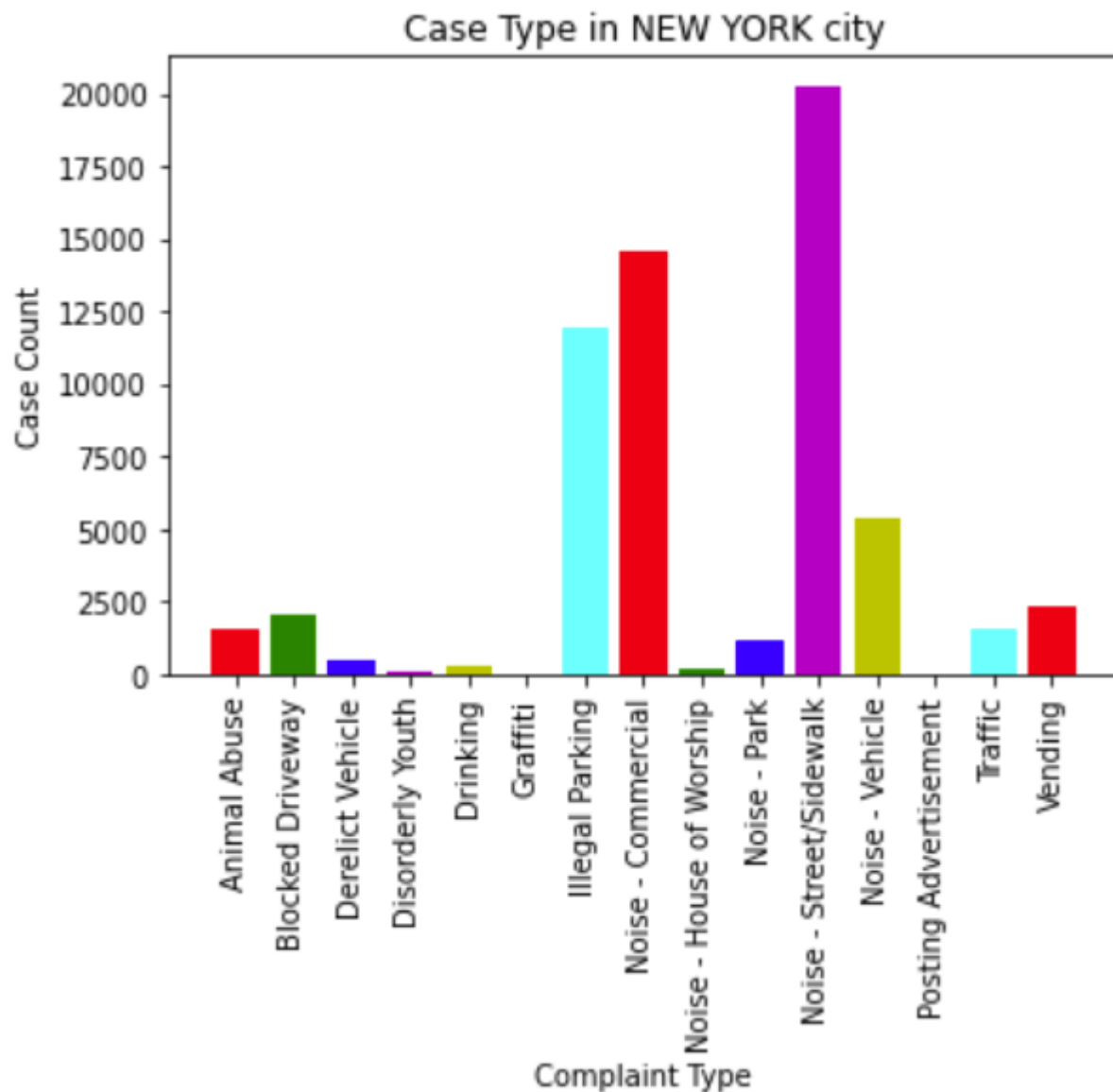


Figure 12: Types of cases in New York City

GitHub link:

[DataAnalytics_With_Python](#)

Course 5: TDLC

Basic Calculator Project with Test Cases:

In this first we made the test cases and then we had written the code. Below is the screenshot of the output.

```
./all.out
make: execvp: ./all.out: Permission denied
make: *** [Makefile:14: run] Error 127
sanket@sanket:~/Downloads/calculator-main/3_Implementation$ make
gcc *.c Unity/unity.c calc.h -IUnity -o all.out
sanket@sanket:~/Downloads/calculator-main/3_Implementation$ ./all.out
test.c:73:sum:PASS
test.c:74:sum1:PASS
test.c:75:sum2:PASS
test.c:76:subtract:PASS
test.c:77:subtract1:PASS
test.c:78:multi:PASS
test.c:79:div:PASS
test.c:80:sq:PASS
test.c:81:sq1:PASS
test.c:82:cu:PASS
test.c:83:cu1:PASS
test.c:84:cube_rt:PASS
test.c:85:sq_rt:PASS

-----
13 Tests 0 Failures 0 Ignored
OK
sanket@sanket:~/Downloads/calculator-main/3_Implementation$
```

Figure 13; Test Cases For Calculator

GitHub Link:

[Calculator](#)

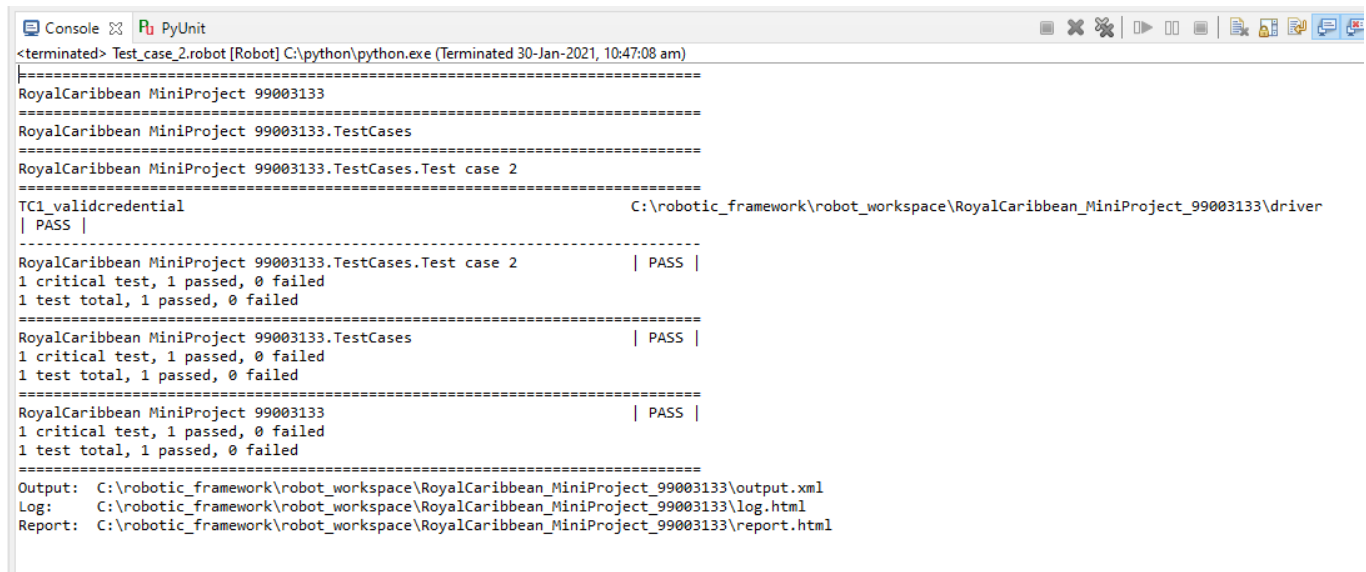
Course 6: Robot Framework

1. Selenium:

In this I had made testcases for KOHLS website in which one fails and other pass. Below is the screenshot of the test cases.

```
Command: C:\python\python.exe -m robot.run --listener C:\Users\CTEA\AppData\Local\Temp\RobotTempDir17478739480335971078\TestRunnerAgent.py:59414 --argumentf
Suite Executor: Robot Framework 3.2.2 (Python 3.7.9 on win32)
=====
RoyalCaribbean MiniProject 99003133
=====
RoyalCaribbean MiniProject 99003133.TestCases
=====
RoyalCaribbean MiniProject 99003133.TestCases.Login suite
=====
TC1_validcredential                                     C:\robotic_framework\robot_workspace\RoyalCaribbean_MiniProject_99003133\driver
February 2nd, 2021 - April 3rd, 2021
April 10th, 2021 - April 16th, 2022
| FAIL |
Does not meet my requirements
-----
RoyalCaribbean MiniProject 99003133.TestCases.Login suite | FAIL |
1 critical test, 0 passed, 1 failed
1 test total, 0 passed, 1 failed
-----
RoyalCaribbean MiniProject 99003133.TestCases | FAIL |
1 critical test, 0 passed, 1 failed
1 test total, 0 passed, 1 failed
-----
RoyalCaribbean MiniProject 99003133 | FAIL |
1 critical test, 0 passed, 1 failed
1 test total, 0 passed, 1 failed
```

Figure 14: TC 1



```
<terminated> Test_case_2.robot [Robot] C:\python\python.exe (Terminated 30-Jan-2021, 10:47:08 am)
=====
RoyalCaribbean MiniProject 99003133
=====
RoyalCaribbean MiniProject 99003133.TestCases
=====
RoyalCaribbean MiniProject 99003133.TestCases.Test case 2
=====
TC1_validcredential                                     C:\robotic_framework\robot_workspace\RoyalCaribbean_MiniProject_99003133\driver
| PASS |
-----
RoyalCaribbean MiniProject 99003133.TestCases.Test case 2 | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
-----
RoyalCaribbean MiniProject 99003133.TestCases | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
-----
RoyalCaribbean MiniProject 99003133 | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
-----
Output: C:\robotic_framework\robot_workspace\RoyalCaribbean_MiniProject_99003133\output.xml
Log: C:\robotic_framework\robot_workspace\RoyalCaribbean_MiniProject_99003133\log.html
Report: C:\robotic_framework\robot_workspace\RoyalCaribbean_MiniProject_99003133\report.html
```

Figure 15: TC 2

Github Link:

[Selenium](#)

2. Appium:

In Appium I had tested Carinfo app by adding some details of the car to the console. Below is the screenshot of the test case.

```
<terminated> Flipkart.Robot (Selected Test Cases) [Robot] C:\python\python.exe (terminated 30-Jan-2021, 11:03:00 am)
Command: C:\python\python.exe -m robot.run --listener C:\Users\CTEA\AppData\Local\Temp\RobotTempDir174787394803
Suite Executor: Robot Framework 3.2.2 (Python 3.7.9 on win32)
=====
Flipkart App
=====
Flipkart App.NativeAppSuite
=====
Flipkart App.NativeAppSuite.Flipkart
=====
TC1_Verify_Invalid_login | PASS |
-----
Flipkart App.NativeAppSuite.Flipkart | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
=====
Flipkart App.NativeAppSuite | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
=====
Flipkart App | PASS |
1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed
=====
Output: C:\robotic framework\robot workspace\Flipkart App\output.xml
Log: C:\robotic framework\robot workspace\Flipkart App\log.html
Report: C:\robotic framework\robot workspace\Flipkart App\report.html
```

Figure 16: Appium TC 1

GitHub Link:

[CarInfo](#)

Course 7: Networking

1. Cisco Packet Tracer:

In this software we had configure 1 router with Pc & 2 routers with Pc and tried to ping all of them. Below is the screenshot of the same.

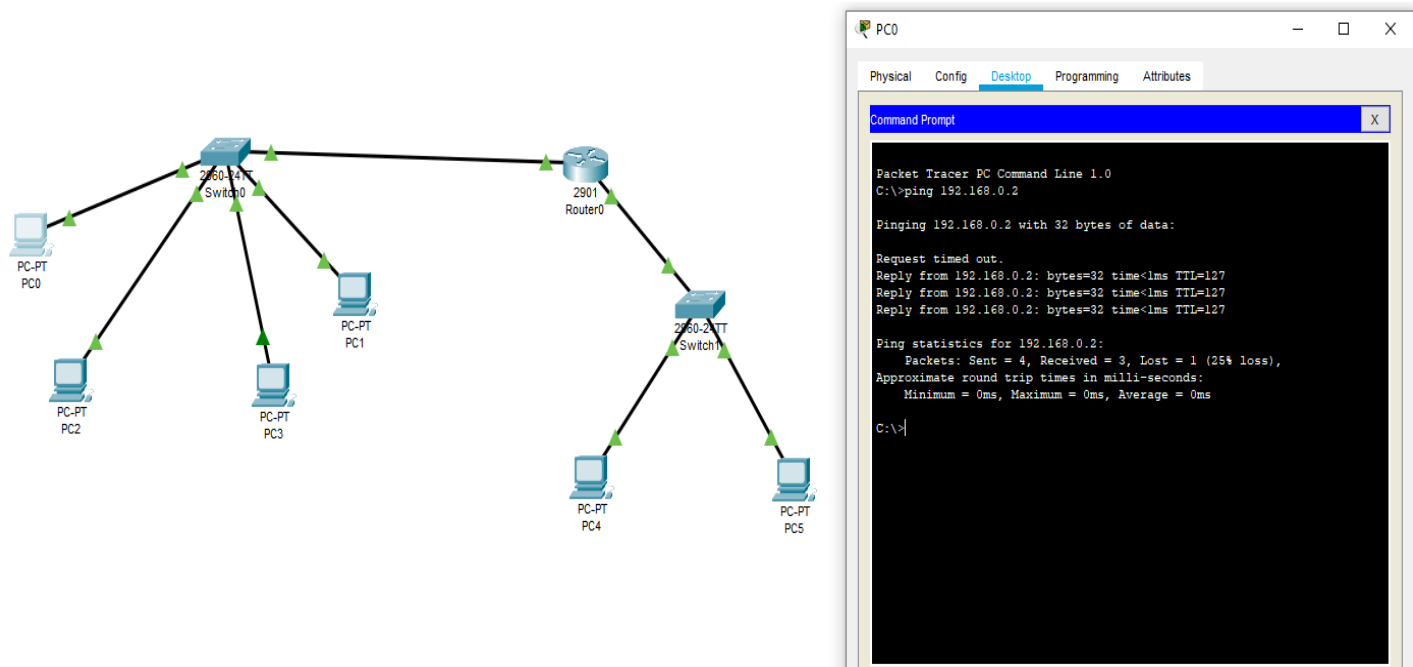


Figure 17: 1 Router Configuration

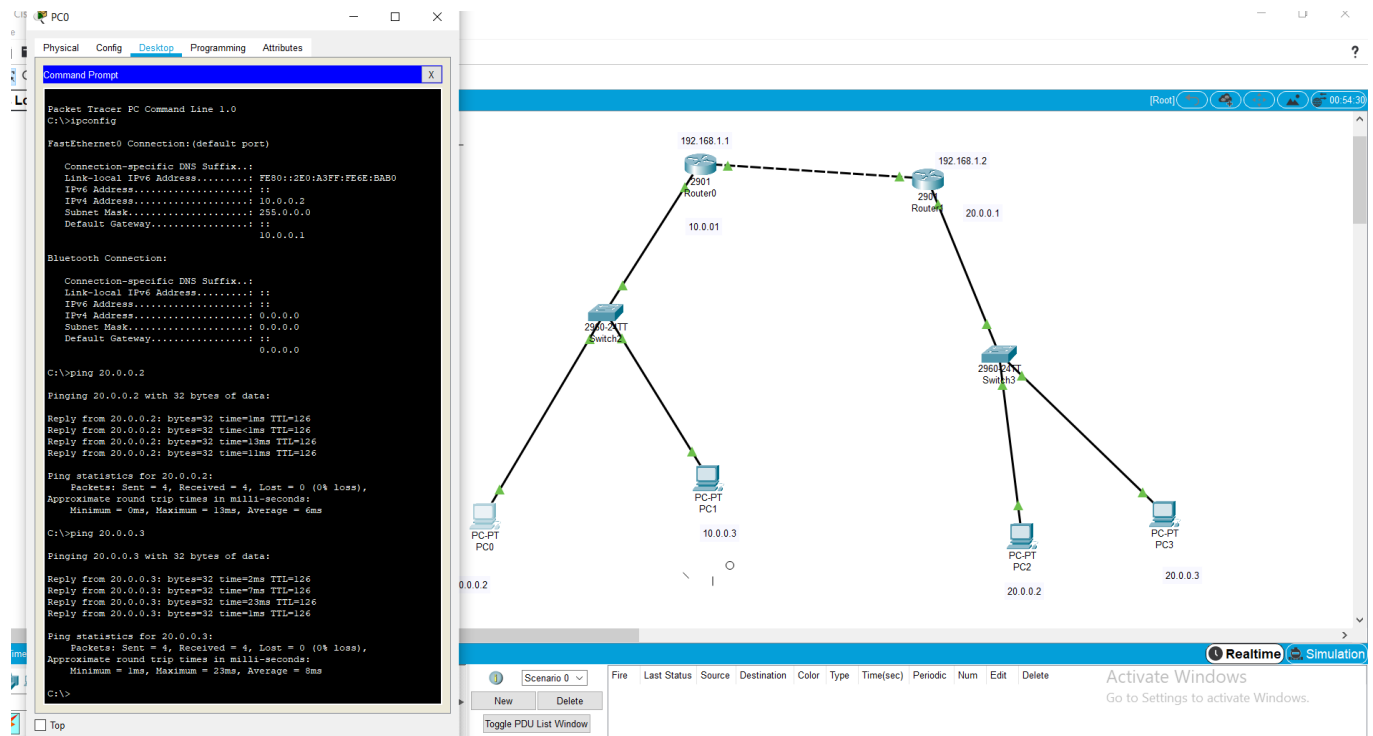


Figure 18: 2 Router Configuration

2. Wire Shark:

Here we had ping www.google.com and seen how it's happening at back end.

402	18.925/61	8.8.8.8	192.168.90.155	DNS	90 Standard query response 0x4749 A www.google.com A 142.250.67.68
403	18.935195	8.8.8.8	192.168.90.155	DNS	90 Standard query response 0x4f49 A www.google.com A 142.250.67.68
404	18.935309	192.168.90.155	8.8.8.8	ICMP	118 Destination unreachable (Port unreachable)
405	18.955896	192.168.90.155	142.250.67.68	ICMP	74 Echo (ping) request id=0x0001, seq=1/256, ttl=128 (reply in 406)
406	18.979250	142.250.67.68	192.168.90.155	ICMP	74 Echo (ping) reply id=0x0001, seq=1/256, ttl=118 (request in 405)
407	19.961134	192.168.90.155	142.250.67.68	ICMP	74 Echo (ping) request id=0x0001, seq=2/512, ttl=128 (reply in 408)
408	19.984505	142.250.67.68	192.168.90.155	ICMP	74 Echo (ping) reply id=0x0001, seq=2/512, ttl=118 (request in 407)
409	20.472812	192.168.90.155	52.109.60.16	TLSv1.2	83 Application Data
410	20.485981	52.109.60.16	192.168.90.155	TLSv1.2	79 Application Data

Figure 19: Wire Shark