LOVELY PROFESSIONAL UNIVERSITY

PYTHON PROJECT ON

"TRAVELING SALESMAN PROBLEM IMPLEMENTATION"

Name : Amandeep Singh

Registration Number : 12013325

Github Link :https://github.com/amandeep12013325

Program : B. Tech (CSE)

Semester : Third

School : School of Computer Science & Engineering

University Name : Lovely Professional University

Date of Submission : 20th November 2021

TABLE OF CONTENT :-

ACKNOWLEDGEMENT

I would like to thank my teacher (Miss.Ankita Wadhawan) who gave me this opportunity to work on this project.

I got to learn a lot from this project about Shortest path finding system. I would also like to thankmy team member and friends who helped me a lot to go deeply under Python Project and explore more and more.

At last, I would like to extend my heartfelt thanks to my parents because without their help this project would not have been successful. Finally, I would like to thank my dear friends who have been with me all the time?

INTRODUCTION

1.1Context

This project has been done as part of my course for the CSE at Lovely Professional University. Supervised by Miss. Ankita Wadhawan, I havespend one month to fulfil the requirements in order to succeed the module.

1.2 Motivations

Being extremely interested in everything having a relation with the python project was a great occasion to give us the time to learn and confirm us interest for this field.

The fact that I can make estimations, predictions and give the ability for python to learn by themselves is both powerful and limitless in term of application possibilities.

1.3 Idea

As a first experience, I wanted to make my project as much didactic as possible by approaching every different step of the python process and trying to understand them deeply. Known as toy problem" the problems that are not immediate scientific interest but useful to illustrate and practice, I choose to take Shortest path finding system as approach. The goal was to predict to know about the shortest path for salesman who really needs it for his/her job..

Travelling Salesman Problem

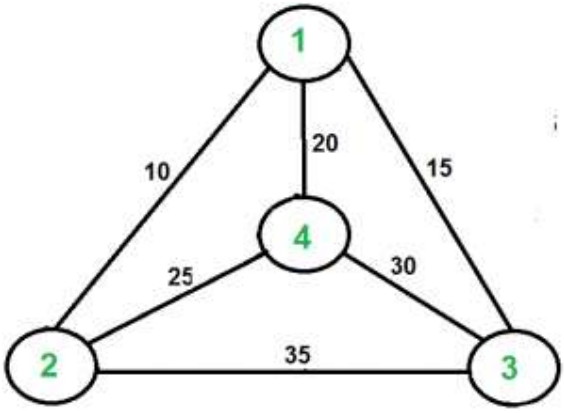
What is Travelling Salesman Problem ?

Travelling sales man problem is one of the challenging problems in the real life and also most well studied combinatorial optimization problem. Many Researches from different fields like operational research, algorithms design and including artificial intelligence attract by it.This problem has been studied by different researches and come up with different solutions and this problem has been solved by using different algorithms like Blind search, Branch and Bound Search , Heuristic algorithm and Genetic algorithms etc. the problem was formulated as a mathematical problem in 1930 and later it is used as bench mark for many optimal solution. Travelling Salesman Problem (TSP) : Given a set of cities and distances between every pair of cities, the problem is to find the shortest possible route that visits every city exactly once and returns to the starting point.

Note the difference between Hamiltonian Cycle and TSP. The Hamiltonian cycle problem is to find if there exists a tour that visits every city exactly once. Here we know that Hamiltonian Tour exists (because the graph is complete) and in fact, many such tours exist, the problem is to find a minimum weight Hamiltonian Cycle.

For example, consider the graph shown in the figure on the right side. A TSP tour in the graph is 1-2-4-3-1. The cost of the tour is 10+25+30+15 which is 80.

The problem is a famous NP-hard problem. There is no polynomial-time known solution for this problem.



Advantage of shortest path finder:

- It will reduce time of journey.
- Reduces money and fuel expenditure.
- It will also gives you some relief from tiredness.
- Motivates you to work more to get extra incentives.

CODE (SOLUTION OF THE ABOVE PROBLEM) :

```
import time
print("================================================================
================================================================")
time.sleep(0.5)
print("                                        W", end = '')
time.sleep(0.1)
print("E", end = '')
time.sleep(0.1)
print("L", end = '')
time.sleep(0.1)
print("C", end = '')
time.sleep(0.1)
print("O", end = '')
time.sleep(0.1)
print("M", end = '')
time.sleep(0.1)
print("E")
time.sleep(0.1)
print("                                          TO")
time.sleep(0.1)
print("                                  S", end = '')
time.sleep(0.1)
print("H", end = '')
time.sleep(0.1)
print("O", end = '')
time.sleep(0.1)
print("R", end = '')
time.sleep(0.1)
print("T", end = '')
```

```
time.sleep(0.1)
print("E", end = '')
time.sleep(0.1)
print("S", end = '')
time.sleep(0.1)
print("T", end = '')
time.sleep(0.1)
print(" ", end = '')
time.sleep(0.1)
print("D", end = '')
time.sleep(0.1)
print("I", end = '')
time.sleep(0.1)
print("S", end = '')
time.sleep(0.1)
print("T", end = '')
time.sleep(0.1)
print("A", end = '')
time.sleep(0.1)
print("N", end = '')
time.sleep(0.1)
print("C", end = '')
time.sleep(0.1)
print("E", end = '')
time.sleep(0.1)
print(" ", end = '')
time.sleep(0.1)
print("C", end = '')
time.sleep(0.1)
print("A", end = '')
```

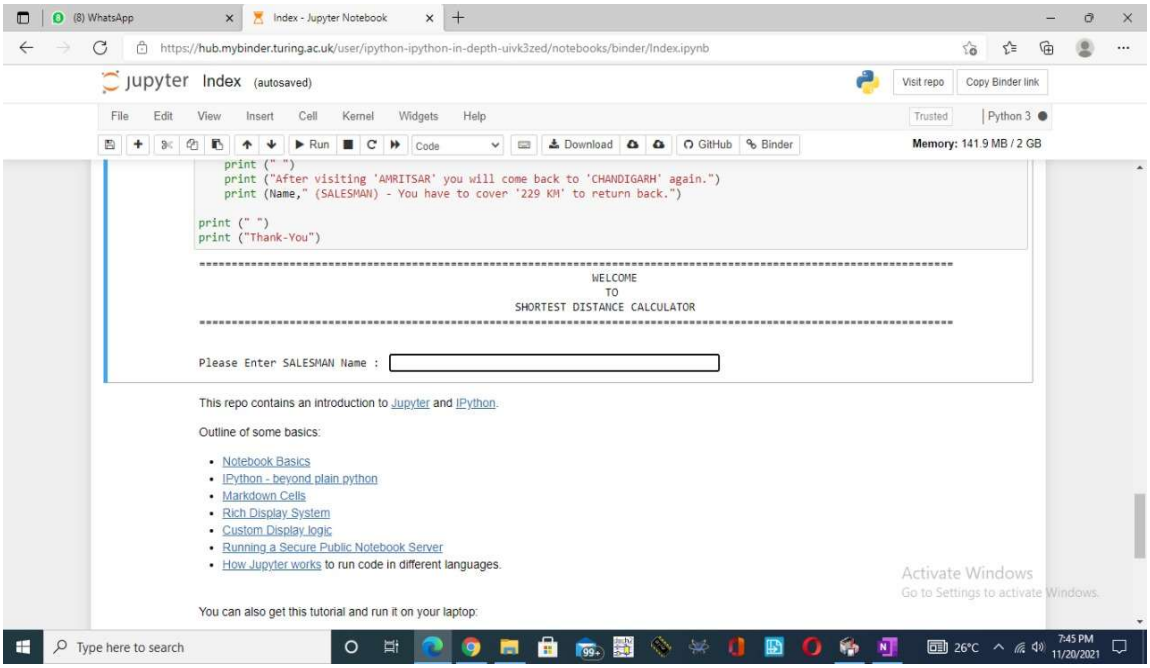
```
time.sleep(0.1)
print("L", end = '')
time.sleep(0.1)
print("C", end = '')
time.sleep(0.1)
print("U", end = '')
time.sleep(0.1)
print("L", end = '')
time.sleep(0.1)
print("A", end = '')
time.sleep(0.1)
print("T", end = '')
time.sleep(0.1)
print("O", end = '')
time.sleep(0.1)
print("R")
print("================================================================
============================================================")
print (" ")
Name=input("Please Enter SALESMAN Name : ");
print ("Hello, ",Name)
print("\n")


SerialNo1=1


InitialCityLudhiana = {"LUDHIANA" : 0, "JALANDHAR" : 61, "PATIALA" : 93,
"CHANDIGARH" : 106, "AMRITSAR" : 140}
InitialCityJalandhar = {"JALANDHAR" : 0, "LUDHIANA" : 61, "AMRITSAR" : 80,
"CHANDIGARH" : 149, "PATIALA" : 154}
InitialCityAmritsar = {"AMRITSAR" : 0, "JALANDHAR" : 80, "LUDHIANA" : 140,
"CHANDIGARH" : 229, "PATIALA" : 235}
```

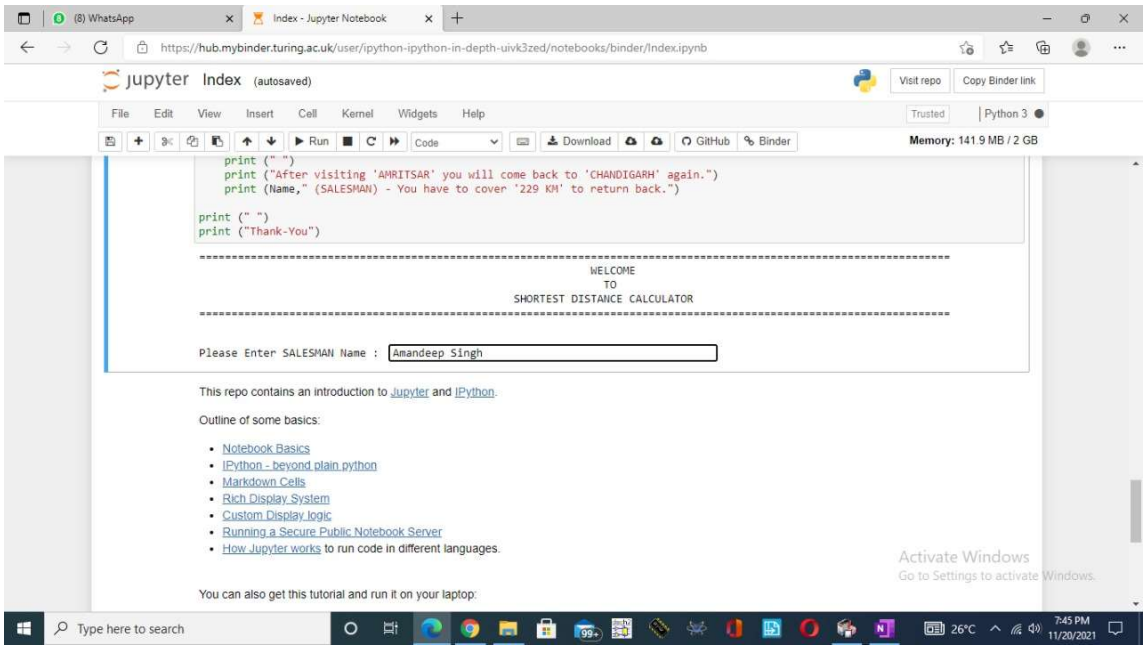
```
InitialCityPatiala = {"PATIALA" : 0, "CHANDIGARH" : 75, "LUDHIANA" : 93,
"JALANDHAR" : 154, "AMRITSAR" : 235}

InitialCityChandigarh = {"CHANDIGARH" : 0, "PATIALA" : 75, "LUDHIANA" :
106, "JALANDHAR" : 149, "AMRITSAR" : 229}


print ("Please enter from which city You are traveling")

print (" ")

UserResponce=input("City Name : ")



if UserResponce=="LUDHIANA" or UserResponce=="ludhiana" or
UserResponce=="Ludhiana":

   print (" ")

   print ("SHORTEST DISTANCE - From 'LUDHIANA' ")

   print ("----------------------------------------")

   print (" ")

   for City in InitialCityLudhiana:

      print (SerialNo1,". ",City," : ",InitialCityLudhiana[City],"Kilo Meters")

      SerialNo1=SerialNo1+1


   print (" ")

   print ("After visiting 'AMRITSAR' you will come back to 'LUDHIANA' again.")

   print (Name," (SALESMAN) - You have to cover '140 KM' to return back.")


if UserResponce=="JALANDHAR" or UserResponce=="jalandhar" or
UserResponce=="Jalandhar":

   print (" ")

   print ("SHORTEST DISTANCE - From 'JALANDHAR' ")

   print ("----------------------------------------")

   print (" ")

   for City in InitialCityJalandhar:

      print (SerialNo1,". ",City," : ",InitialCityJalandhar[City],"Kilo Meters")
```

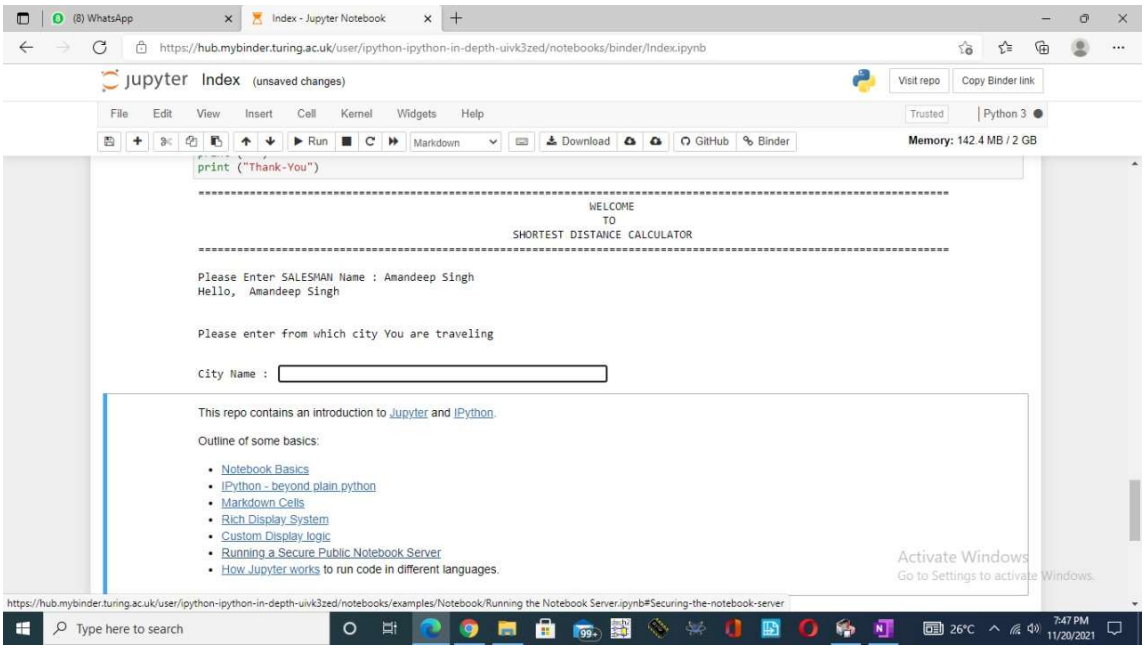
```
        SerialNo1=SerialNo1+1


    print (" ")
    print ("After visiting 'PATIALA' you will come back to 'JALANDHAR' again.")
    print (Name," (SALESMAN) - You have to cover '154 KM' to return back.")


if UserResponce=="AMRITSAR" or UserResponce=="amritsar" or
UserResponce=="Amritsar":
    print (" ")
    print ("SHORTEST DISTANCE - From 'AMRITSAR' ")
    print ("----------------------------------------")
    print (" ")
    for City in InitialCityAmritsar:
        print (SerialNo1,". ",City," : ",InitialCityAmritsar[City],"Kilo Meters")
        SerialNo1=SerialNo1+1


    print (" ")
    print ("After visiting 'PATIALA' you will come back to 'AMRITSAR' again.")
    print (Name," (SALESMAN) - You have to cover '235 KM' to return back.")


if UserResponce=="PATIALA" or UserResponce=="patiala" or
UserResponce=="Patiala":
    print (" ")
    print ("SHORTEST DISTANCE - From 'PATIALA' ")
    print ("----------------------------------------")
    print (" ")
    for City in InitialCityPatiala:
        print (SerialNo1,". ",City," : ",InitialCityPatiala[City],"Kilo Meters")
        SerialNo1=SerialNo1+1


    print (" ")
```

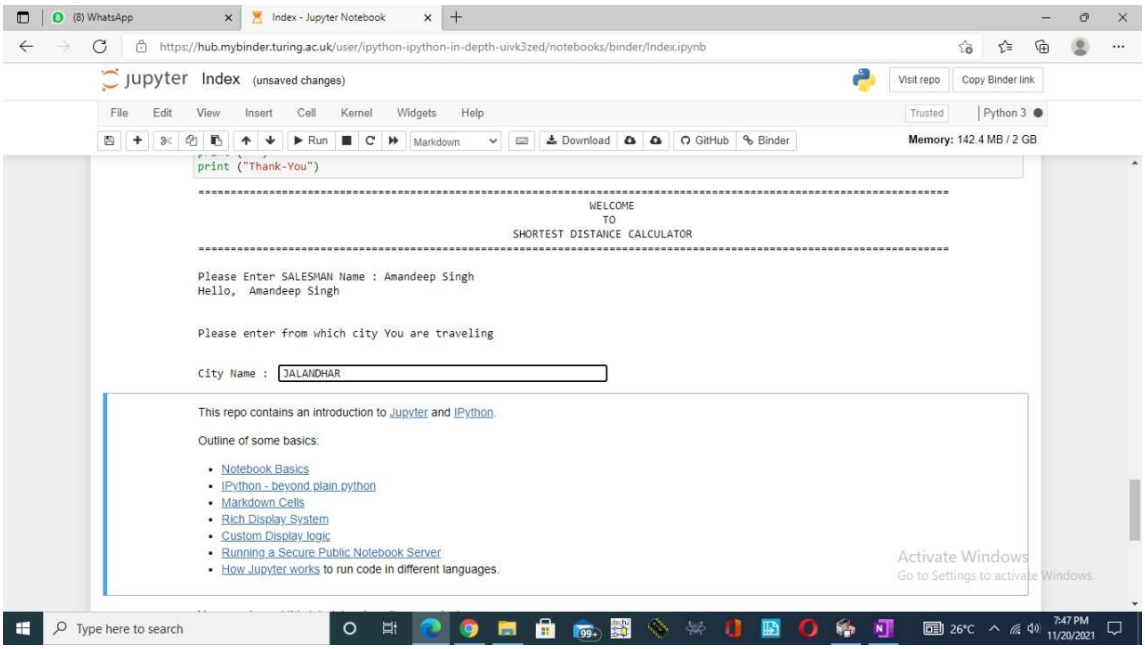
```
    print ("After visiting 'AMRITSAR' you will come back to 'PATIALA' again.")
    print (Name," (SALESMAN) - You have to cover '235 KM' to return back.")


if UserResponce=="CHANDIGARH" or UserResponce=="chandigarh" or
UserResponce=="Chandigarh":
    print (" ")
    print ("SHORTEST DISTANCE - From 'CHANDIGARH' ")
    print ("----------------------------------------")
    print (" ")
    for City in InitialCityChandigarh:
        print (SerialNo1,". ",City," : ",InitialCityChandigarh[City],"Kilo Meters")
        SerialNo1=SerialNo1+1


    print (" ")
    print ("After visiting 'AMRITSAR' you will come back to 'CHANDIGARH' again.")
    print (Name," (SALESMAN) - You have to cover '229 KM' to return back.")


print (" ")
print ("Thank-You")
```

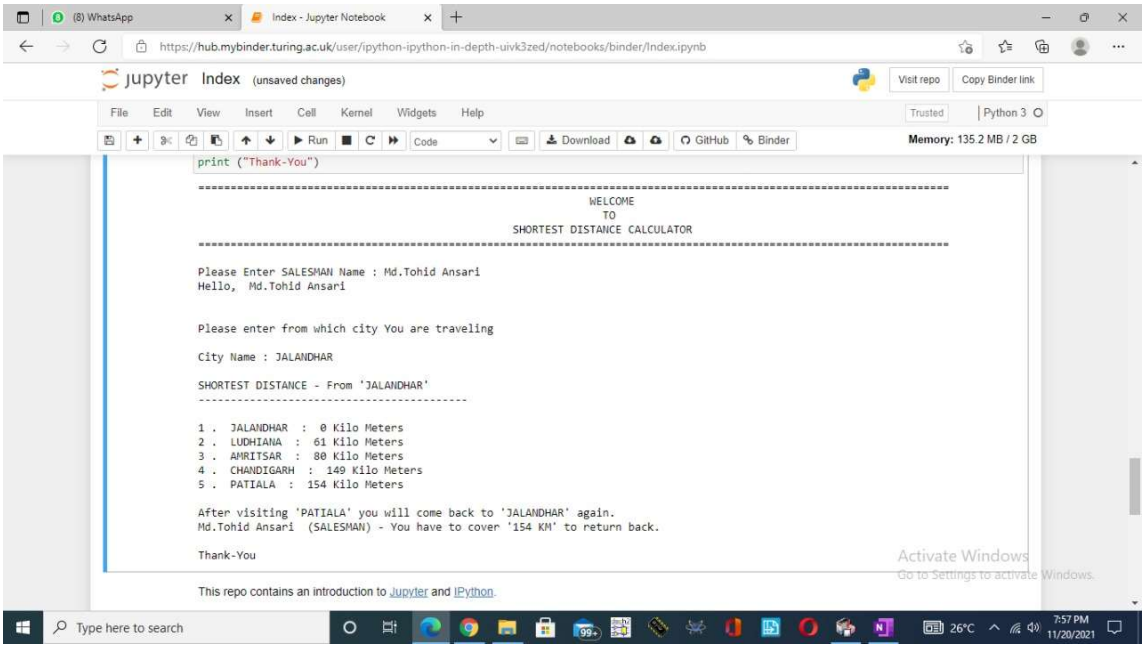
OUTPUT SCREENSHOTS:

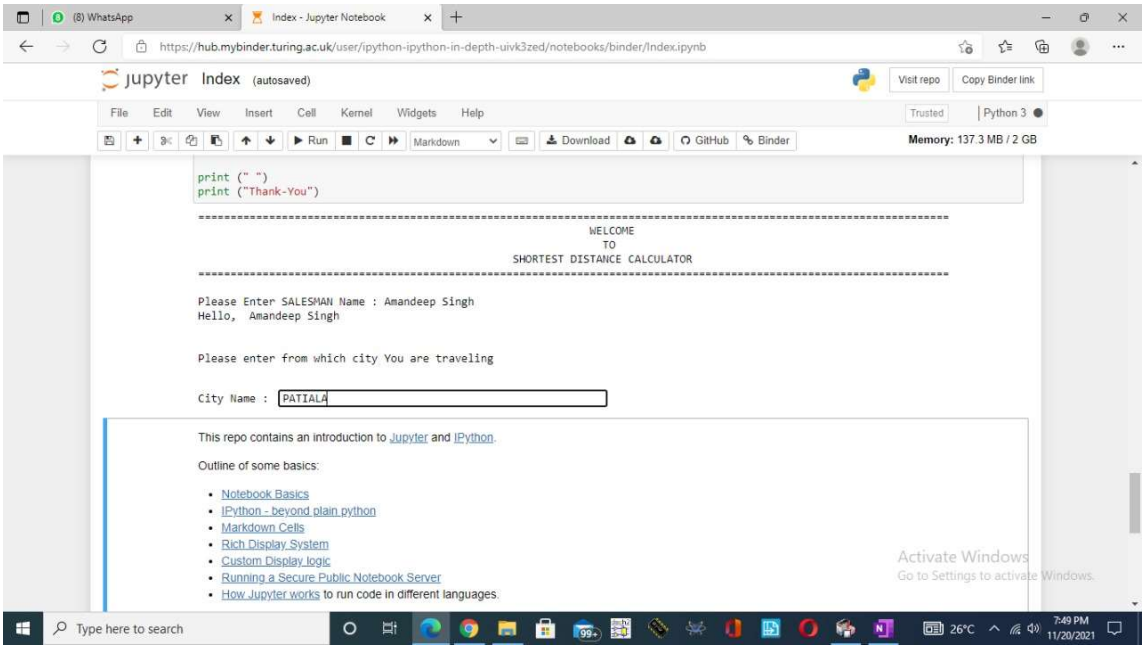


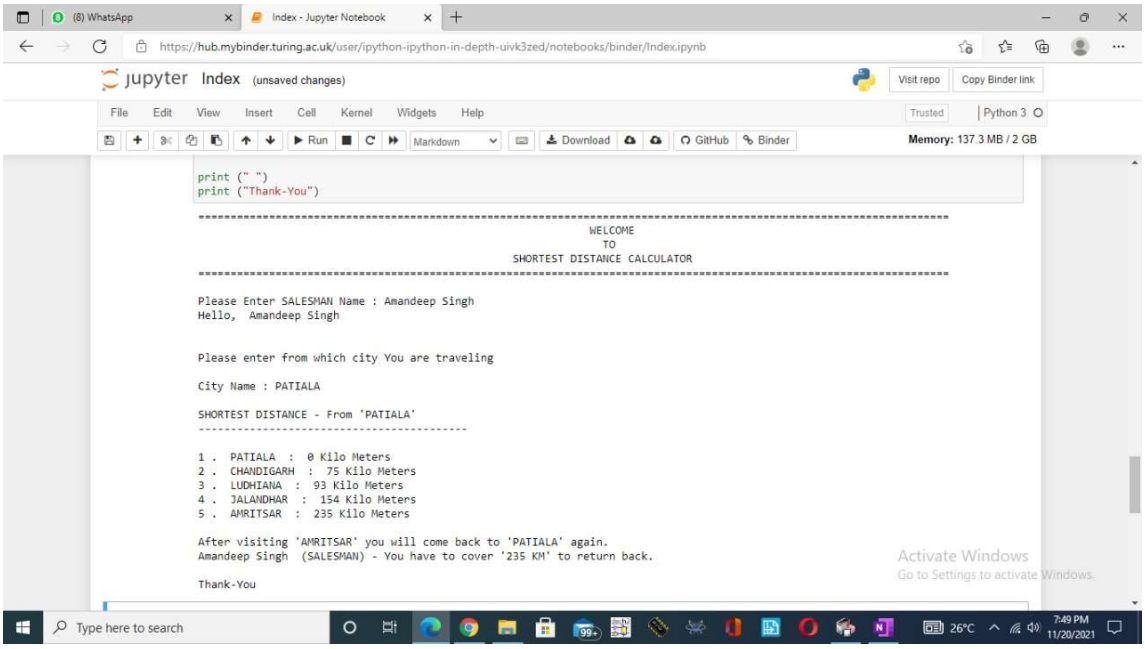


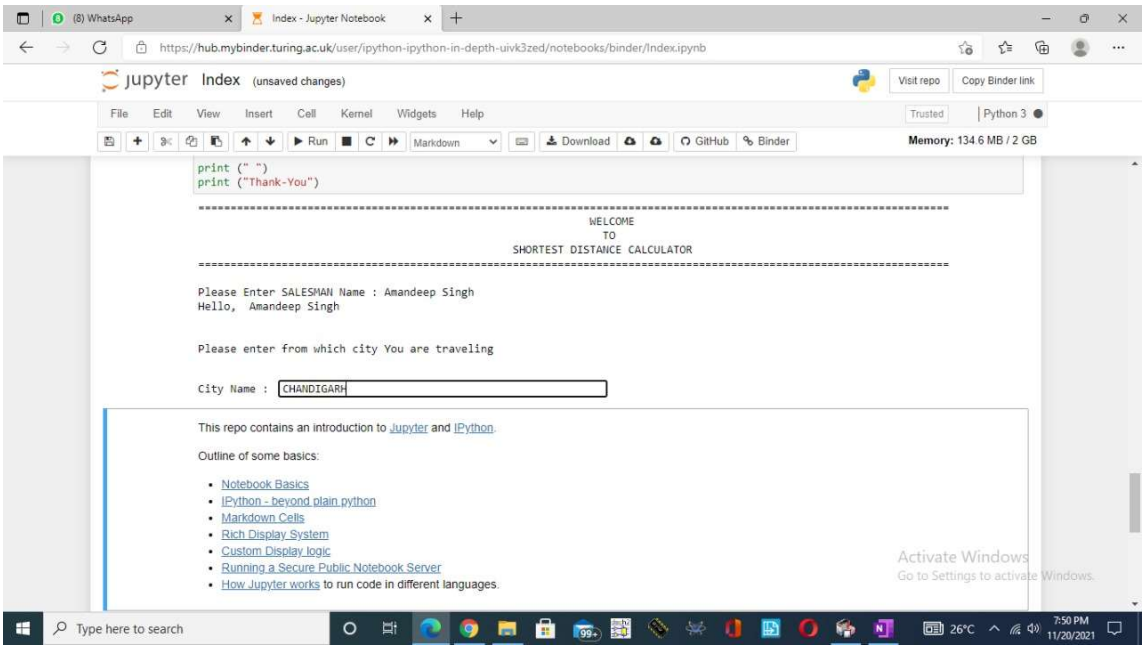

(8) WhatsApp
Index - Jupyter Notebook
https://hub.mybinder.turing.ac.uk/user/ipython-ipython-in-depth-uivk3zed/notebooks/binder/Index.ipynb
jupyter Index (unsaved changes)
Visit repo
Copy Binder link
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Markdown
Download
GitHub
Binder
Memory: 142.4 MB / 2 GB
print ("Thank-You")
WELCOME
TO
SHORTEST DISTANCE CALCULATOR
Please Enter SALESMAN Name : Amandeep Singh
Hello, Amandeep Singh
Please enter from which city You are traveling
City Name :
This repo contains an introduction to Jupyter and IPython.
Outline of some basics:
Notebook Basics
IPython - beyond plain python
Markdown Cells
Rich Display System
Custom Display logic
Running a Secure Public Notebook Server
How Jupyter works to run code in different languages.
Activate Windows
Go to Settings to activate Windows.
https://hub.mybinder.turing.ac.uk/user/ipython-ipython-in-depth-uivk3zed/notebooks/examples/Notebook/Running the Notebook Server.ipynb#Securing-the-notebook-server
Type here to search
26°C
7:47 PM
11/20/2021

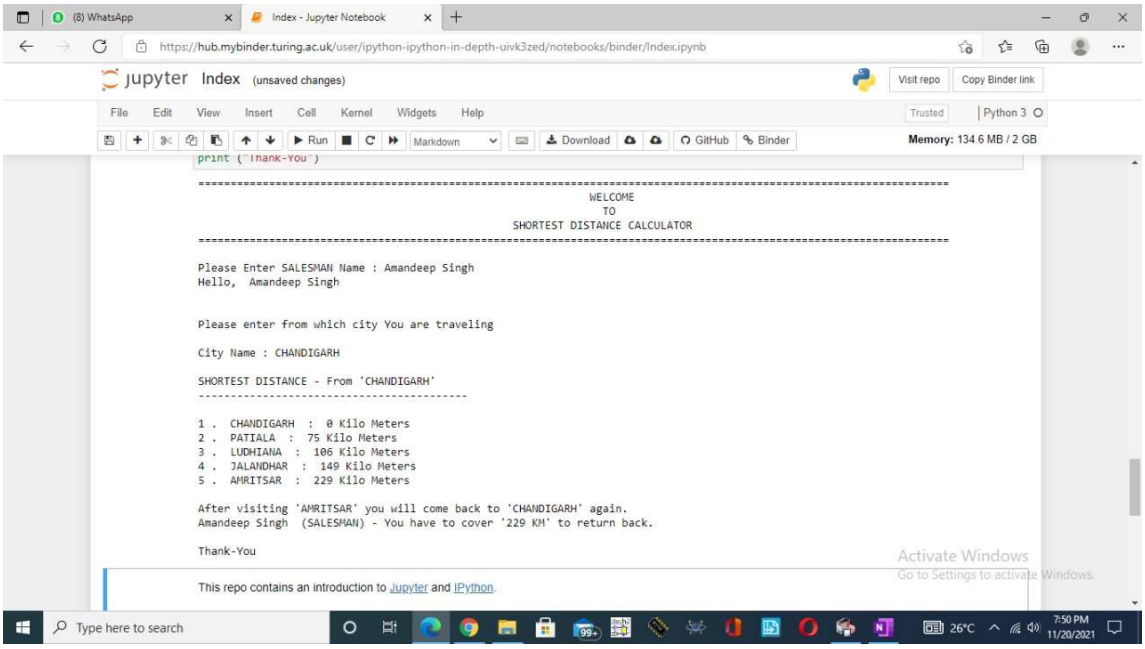

(8) WhatsApp
Index - Jupyter Notebook
https://hub.mybinder.turing.ac.uk/user/ipython-ipython-in-depth-uivk3zed/notebooks/binder/Index.ipynb
jupyter Index (unsaved changes)
Visit repo
Copy Binder link
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Markdown
Download
GitHub
Binder
Memory: 142.4 MB / 2 GB
print ("Thank-You")
WELCOME
TO
SHORTEST DISTANCE CALCULATOR
Please Enter SALESMAN Name : Amandeep Singh
Hello, Amandeep Singh
Please enter from which city You are traveling
City Name : JALANDHAR
This repo contains an introduction to Jupyter and IPython.
Outline of some basics:
Notebook Basics
IPython - beyond plain python
Markdown Cells
Rich Display System
Custom Display logic
Running a Secure Public Notebook Server
How Jupyter works to run code in different languages.
Activate Windows
Go to Settings to activate Windows.
Type here to search
26°C
7:47 PM
11/20/2021

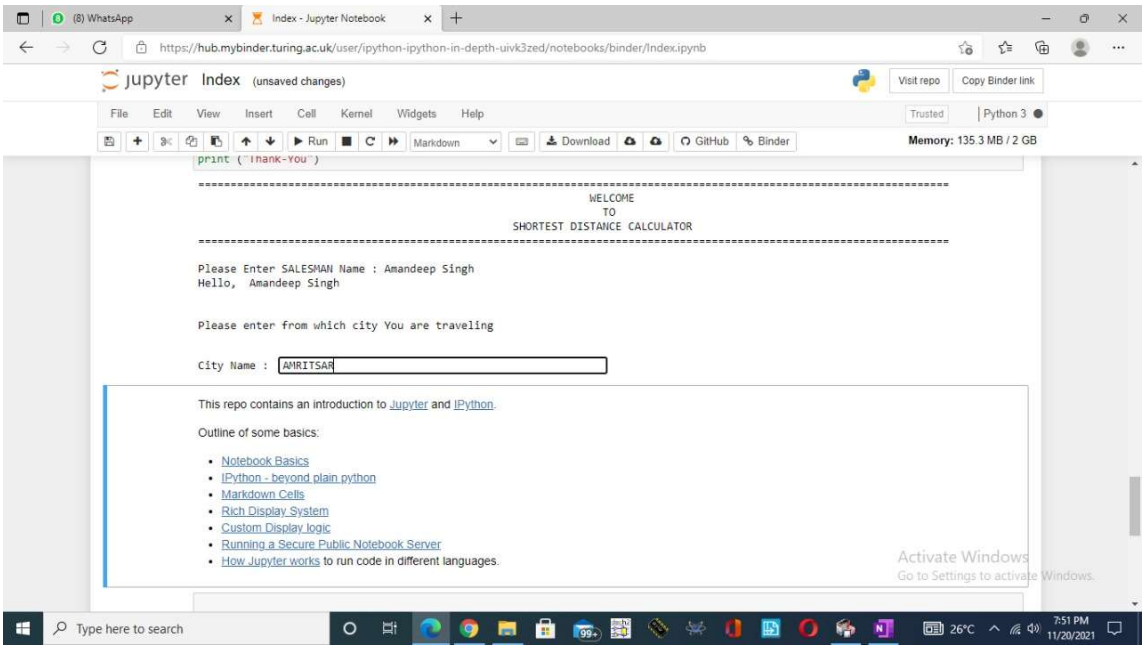

(8) WhatsApp
Index - Jupyter Notebook
https://hub.mybinder.turing.ac.uk/user/ipython-ipython-in-depth-uivk3zed/notebooks/binder/Index.ipynb
jupyter Index (unsaved changes)
Visit repo
Copy Binder link
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Download
GitHub
Binder
Memory: 135.2 MB / 2 GB
print ("Thank-You")
WELCOME
TO
SHORTEST DISTANCE CALCULATOR
Please Enter SALESMAN Name : Md.Tohid Ansari
Hello, Md.Tohid Ansari
Please enter from which city You are traveling
City Name : JALANDHAR
SHORTEST DISTANCE - From 'JALANDHAR'
1 . JALANDHAR : 0 Kilo Meters
2 . LUDHIANA : 61 Kilo Meters
3 . AMRITSAR : 80 Kilo Meters
4 . CHANDIGARH : 149 Kilo Meters
5 . PATIALA : 154 Kilo Meters
After visiting 'PATIALA' you will come back to 'JALANDHAR' again.
Md.Tohid Ansari (SALESMAN) - You have to cover '154 KM' to return back.
Thank-You
This repo contains an introduction to Jupyter and IPython.
Activate Windows
Go to Settings to activate Windows.
Type here to search
26°C
7:57 PM
11/20/2021

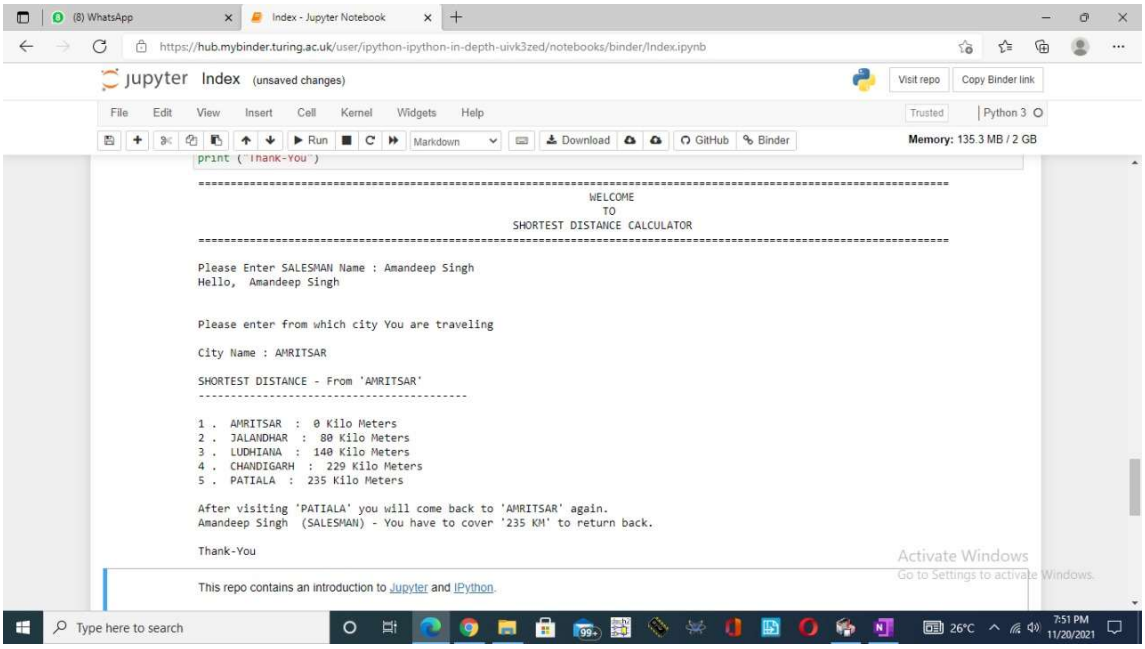

(8) WhatsApp
Index - Jupyter Notebook
https://hub.mybinder.turing.ac.uk/user/ipython-ipython-in-depth-uivk3zed/notebooks/binder/Index.ipynb
jupyter Index (autosaved)
Visit repo
Copy Binder link
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Markdown
Download
GitHub
Binder
Memory: 137.3 MB / 2 GB
print (" ")
print ("Thank-You")
WELCOME
TO
SHORTEST DISTANCE CALCULATOR
Please Enter SALESMAN Name : Amandeep Singh
Hello, Amandeep Singh
Please enter from which city You are traveling
City Name : PATIALA
This repo contains an introduction to Jupyter and IPython.
Outline of some basics:
Notebook Basics
IPython - beyond plain python
Markdown Cells
Rich Display System
Custom Display logic
Running a Secure Public Notebook Server
How Jupyter works to run code in different languages.
Activate Windows
Go to Settings to activate Windows.
Type here to search
26°C
7:49 PM
11/20/2021

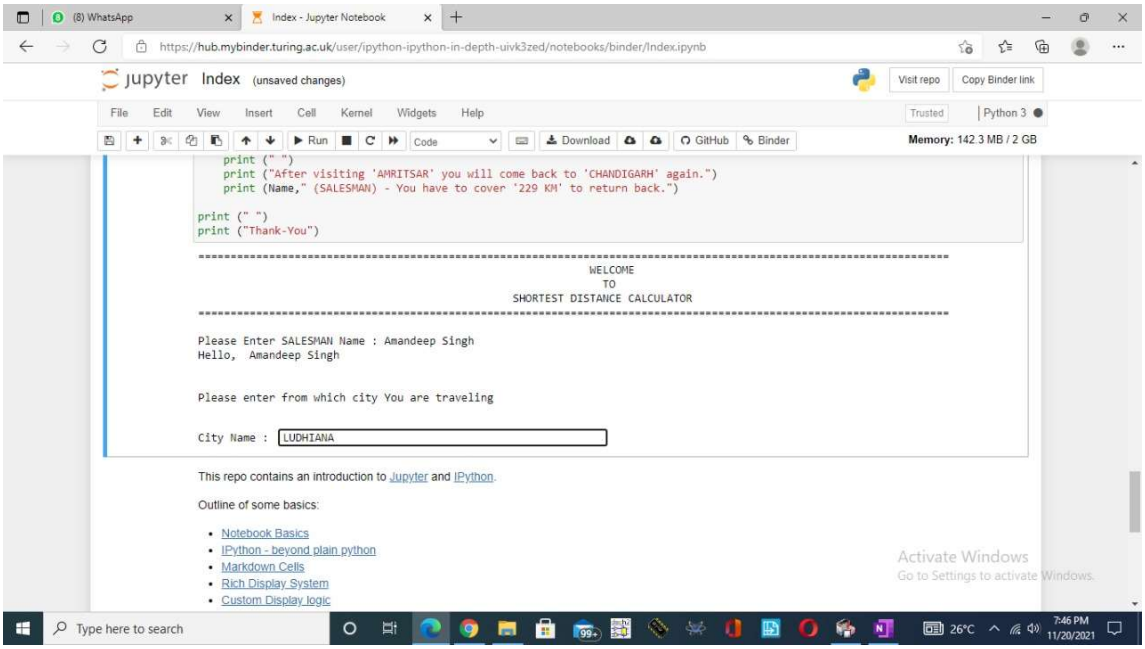

(8) WhatsApp
Index - Jupyter Notebook
https://hub.mybinder.turing.ac.uk/user/ipython-ipython-in-depth-uivk3zed/notebooks/binder/Index.ipynb
jupyter Index (unsaved changes)
Visit repo
Copy Binder link
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Markdown
Download
GitHub
Binder
Memory: 137.3 MB / 2 GB
print (" ")
print ("Thank-You")
WELCOME
TO
SHORTEST DISTANCE CALCULATOR
Please Enter SALESMAN Name : Amandeep Singh
Hello, Amandeep Singh
Please enter from which city You are traveling
City Name : PATIALA
SHORTEST DISTANCE - From 'PATIALA'
1 . PATIALA : 0 Kilo Meters
2 . CHANDIGARH : 75 Kilo Meters
3 . LUDHIANA : 93 Kilo Meters
4 . JALANDHAR : 154 Kilo Meters
5 . AMRITSAR : 235 Kilo Meters
After visiting 'AMRITSAR' you will come back to 'PATIALA' again.
Amandeep Singh (SALESMAN) - You have to cover '235 KM' to return back.
Thank-You
Activate Windows
Go to Settings to activate Windows.
Type here to search
26°C
7:49 PM
11/20/2021

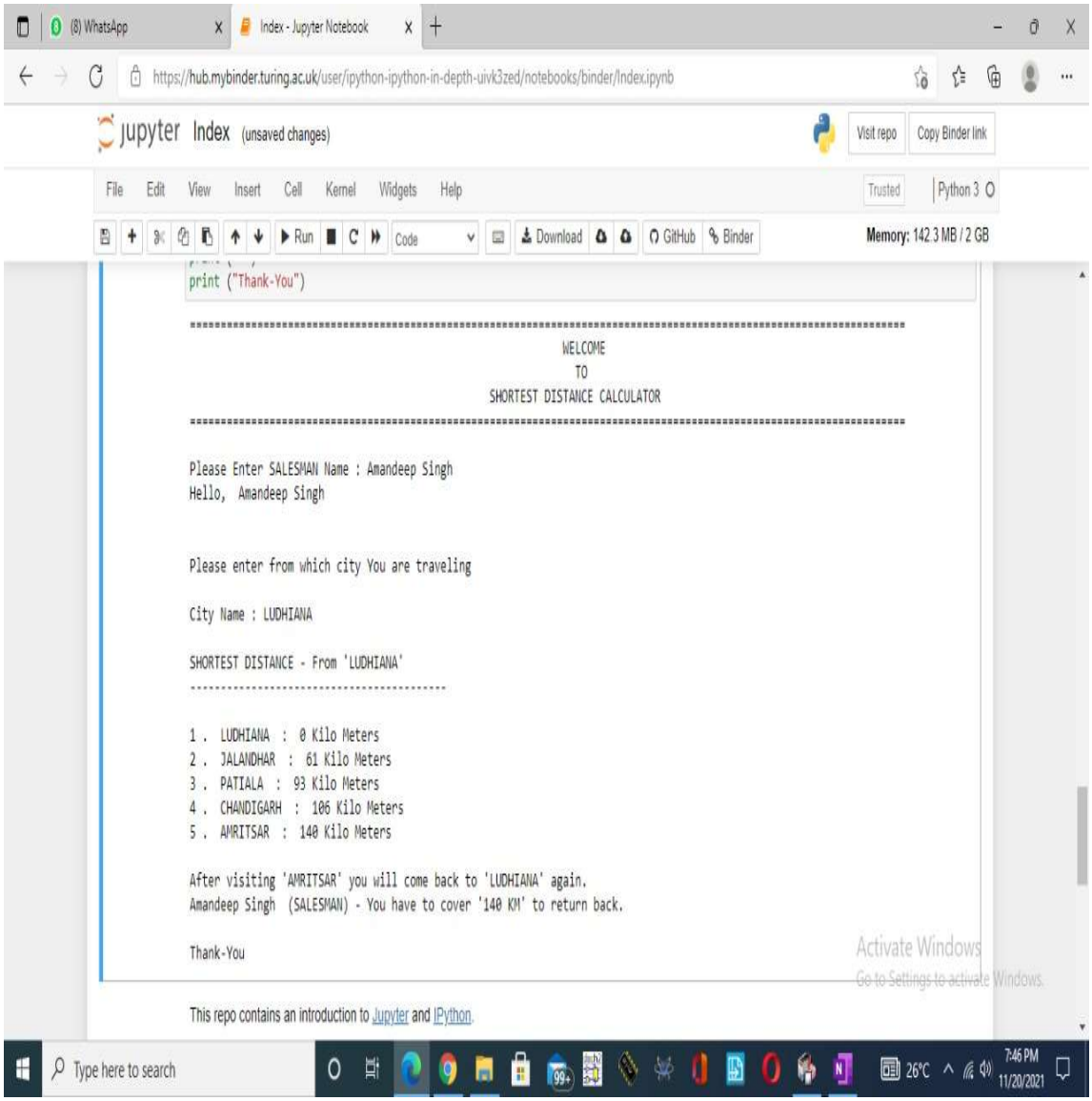

(8) WhatsApp
Index - Jupyter Notebook
https://hub.mybinder.turing.ac.uk/user/ipython-ipython-in-depth-uivk3zed/notebooks/binder/Index.ipynb
jupyter Index (unsaved changes)
Visit repo
Copy Binder link
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Markdown
Download
GitHub
Binder
Memory: 134.6 MB / 2 GB
print (" ")
print ("Thank-You")
WELCOME
TO
SHORTEST DISTANCE CALCULATOR
Please Enter SALESMAN Name : Amandeep Singh
Hello, Amandeep Singh
Please enter from which city You are traveling
City Name : CHANDIGARH
This repo contains an introduction to Jupyter and IPython.
Outline of some basics:
Notebook Basics
IPython - beyond plain python
Markdown Cells
Rich Display System
Custom Display logic
Running a Secure Public Notebook Server
How Jupyter works to run code in different languages.
Activate Windows
Go to Settings to activate Windows.
Type here to search
26°C
7:50 PM
11/20/2021


(8) WhatsApp
Index - Jupyter Notebook
https://hub.mybinder.turing.ac.uk/user/ipython-ipython-in-depth-uivk3zed/notebooks/binder/Index.ipynb
jupyter Index (unsaved changes)
Visit repo
Copy Binder link
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Markdown
Download
GitHub
Binder
Memory: 134.6 MB / 2 GB
print ("Thank-You")
WELCOME
TO
SHORTEST DISTANCE CALCULATOR
Please Enter SALESMAN Name : Amandeep Singh
Hello, Amandeep Singh
Please enter from which city You are traveling
City Name : CHANDIGARH
SHORTEST DISTANCE - From 'CHANDIGARH'
1 . CHANDIGARH : 0 Kilo Meters
2 . PATIALA : 75 Kilo Meters
3 . LUDHIANA : 106 Kilo Meters
4 . JALANDHAR : 149 Kilo Meters
5 . AMRITSAR : 229 Kilo Meters
After visiting 'AMRITSAR' you will come back to 'CHANDIGARH' again.
Amandeep Singh (SALESMAN) - You have to cover '229 KM' to return back.
Thank-You
This repo contains an introduction to Jupyter and IPython.
Activate Windows
Go to Settings to activate Windows.
Type here to search
26°C
7:50 PM
11/20/2021


(8) WhatsApp
Index - Jupyter Notebook
https://hub.mybinder.turing.ac.uk/user/ipython-ipython-in-depth-uivk3zed/notebooks/binder/Index.ipynb
jupyter Index (unsaved changes)
Visit repo
Copy Binder link
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Markdown
Download
GitHub
Binder
Memory: 135.3 MB / 2 GB
print ("Thank-You")
WELCOME
TO
SHORTEST DISTANCE CALCULATOR
Please Enter SALESMAN Name : Amandeep Singh
Hello, Amandeep Singh
Please enter from which city You are traveling
City Name : AMRITSAR
This repo contains an introduction to Jupyter and IPython.
Outline of some basics:
Notebook Basics
IPython - beyond plain python
Markdown Cells
Rich Display System
Custom Display logic
Running a Secure Public Notebook Server
How Jupyter works to run code in different languages.
Activate Windows
Go to Settings to activate Windows.
Type here to search
26°C
7:51 PM
11/20/2021


(8) WhatsApp
Index - Jupyter Notebook
https://hub.mybinder.turing.ac.uk/user/ipython-ipython-in-depth-uivk3zed/notebooks/binder/Index.ipynb
jupyter Index (unsaved changes)
Visit repo
Copy Binder link
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Markdown
Download
GitHub
Binder
Memory: 135.3 MB / 2 GB
print ("Thank-You")
WELCOME
TO
SHORTEST DISTANCE CALCULATOR
Please Enter SALESMAN Name : Amandeep Singh
Hello, Amandeep Singh
Please enter from which city You are traveling
City Name : AMRITSAR
SHORTEST DISTANCE - From 'AMRITSAR'
1 . AMRITSAR : 0 Kilo Meters
2 . JALANDHAR : 80 Kilo Meters
3 . LUDHIANA : 140 Kilo Meters
4 . CHANDIGARH : 229 Kilo Meters
5 . PATIALA : 235 Kilo Meters
After visiting 'PATIALA' you will come back to 'AMRITSAR' again.
Amandeep Singh (SALESMAN) - You have to cover '235 KM' to return back.
Thank-You
This repo contains an introduction to Jupyter and IPython.
Activate Windows
Go to Settings to activate Windows.
Type here to search
26°C
7:51 PM
11/20/2021


(8) WhatsApp
Index - Jupyter Notebook
https://hub.mybinder.turing.ac.uk/user/ipython-ipython-in-depth-uivk3zed/notebooks/binder/Index.ipynb
jupyter Index (unsaved changes)
Visit repo
Copy Binder link
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Download
GitHub
Binder
Memory: 142.3 MB / 2 GB
print (" ")
print ("After visiting 'AMRITSAR' you will come back to 'CHANDIGARH' again.")
print (Name," (SALESMAN) - You have to cover '229 KM' to return back.")
print (" ")
print ("Thank-You")
WELCOME
TO
SHORTEST DISTANCE CALCULATOR
Please Enter SALESMAN Name : Amandeep Singh
Hello, Amandeep Singh
Please enter from which city You are traveling
City Name : LUDHIANA
This repo contains an introduction to Jupyter and IPython.
Outline of some basics:
Notebook Basics
IPython - beyond plain python
Markdown Cells
Rich Display System
Custom Display logic
Activate Windows
Go to Settings to activate Windows.
Type here to search
26°C
7:46 PM
11/20/2021


(8) WhatsApp
Index - Jupyter Notebook
https://hub.mybinder.turing.ac.uk/user/ipython-ipython-in-depth-uivk3zed/notebooks/binder/Index.ipynb
jupyter Index (unsaved changes)
Visit repo
Copy Binder link
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Download
GitHub
Binder
Memory: 142.3 MB / 2 GB
print ("Thank-You")
WELCOME
TO
SHORTEST DISTANCE CALCULATOR
Please Enter SALESMAN Name : Amandeep Singh
Hello, Amandeep Singh
Please enter from which city You are traveling
City Name : LUDHIANA
SHORTEST DISTANCE - From 'LUDHIANA'
1 . LUDHIANA : 0 Kilo Meters
2 . JALANDHAR : 61 Kilo Meters
3 . PATIALA : 93 Kilo Meters
4 . CHANDIGARH : 106 Kilo Meters
5 . AMRITSAR : 140 Kilo Meters
After visiting 'AMRITSAR' you will come back to 'LUDHIANA' again.
Amandeep Singh (SALESMAN) - You have to cover '140 KM' to return back.
Thank-You
Activate Windows
Go to Settings to activate Windows.
This repo contains an introduction to Jupyter and IPython.
Type here to search
26°C
7:46 PM
11/20/2021

CONCLUSION:

In this project I have used so many concepts of “PYTHON PROGRAMMING” Likewise-

- Dictionary – (Key and Value pair)
- For loop
- While loop
- If else concept
- time. Sleep function

Along with these concepts I have also learnt so many new things regarding python and its working.

My topic was “Travelling Salesman Problem Implementation”. I have done a little bit research about it and used all those concepts which are used in real life while finding the shortest route between a set of points and locations that must be visited.

Thank-You Mam.