**Name: Manthan Dabadgaonkar**

**Contact:**

**Ph: 9422595634**

**Email:** [**manthand1511@gmail.com**](mailto:manthand1511@gmail.com)

**ISAC Task**

**Task:**

Write a python application that takes domain name as an input and returns back the following:  
1. The IP of the domain name

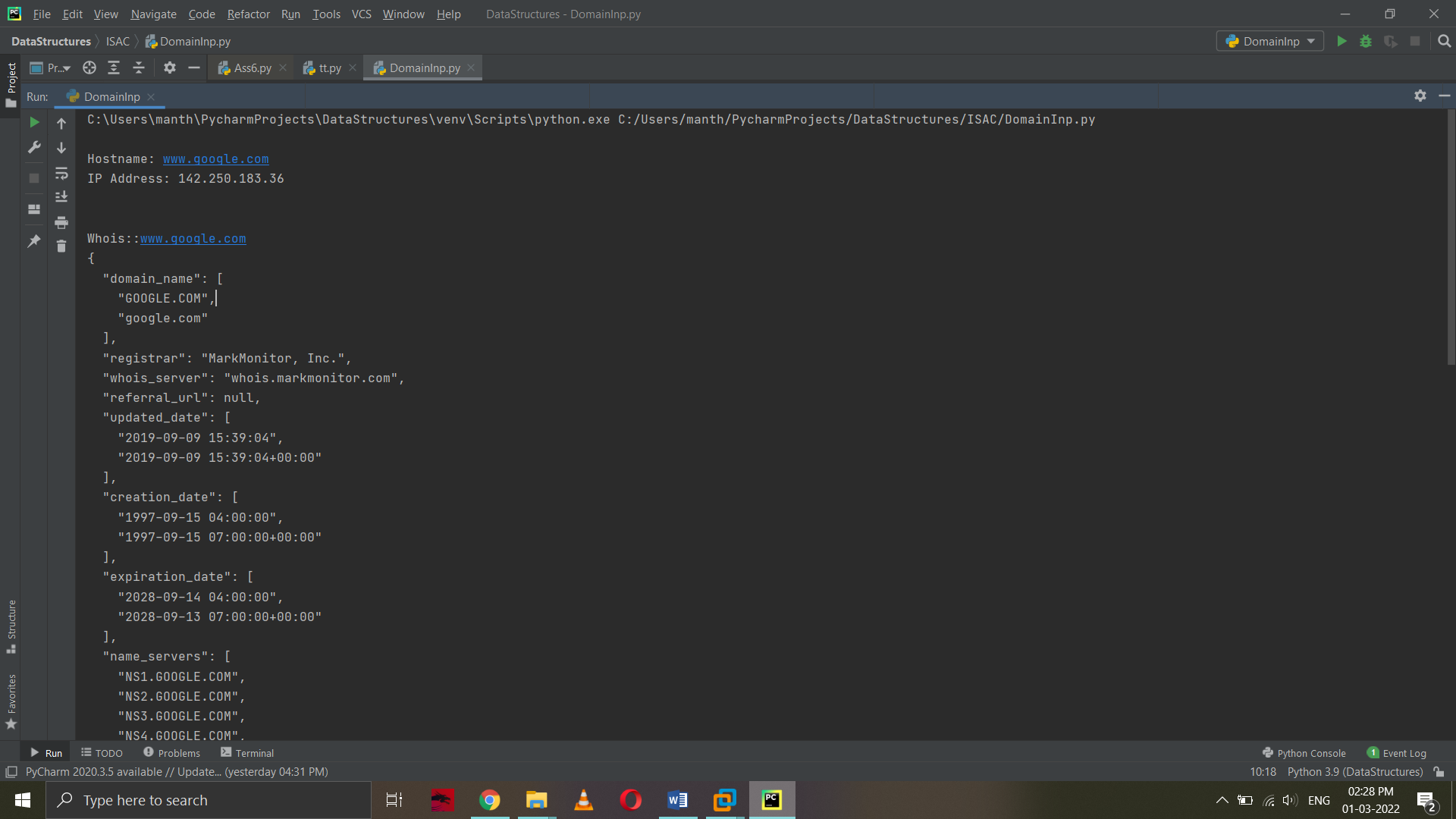
2. Whois information of the domain

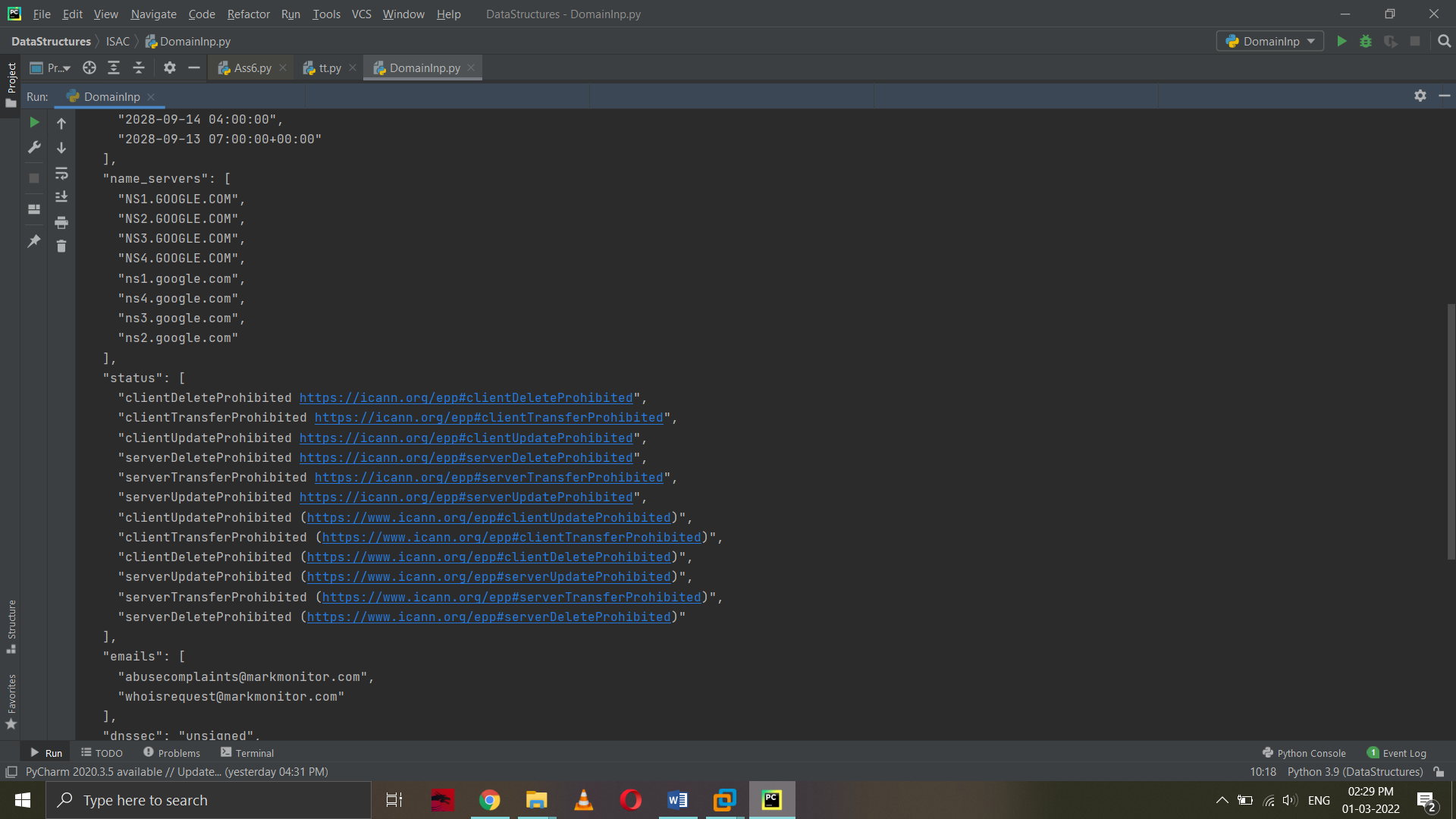
3. Ping details of the domain

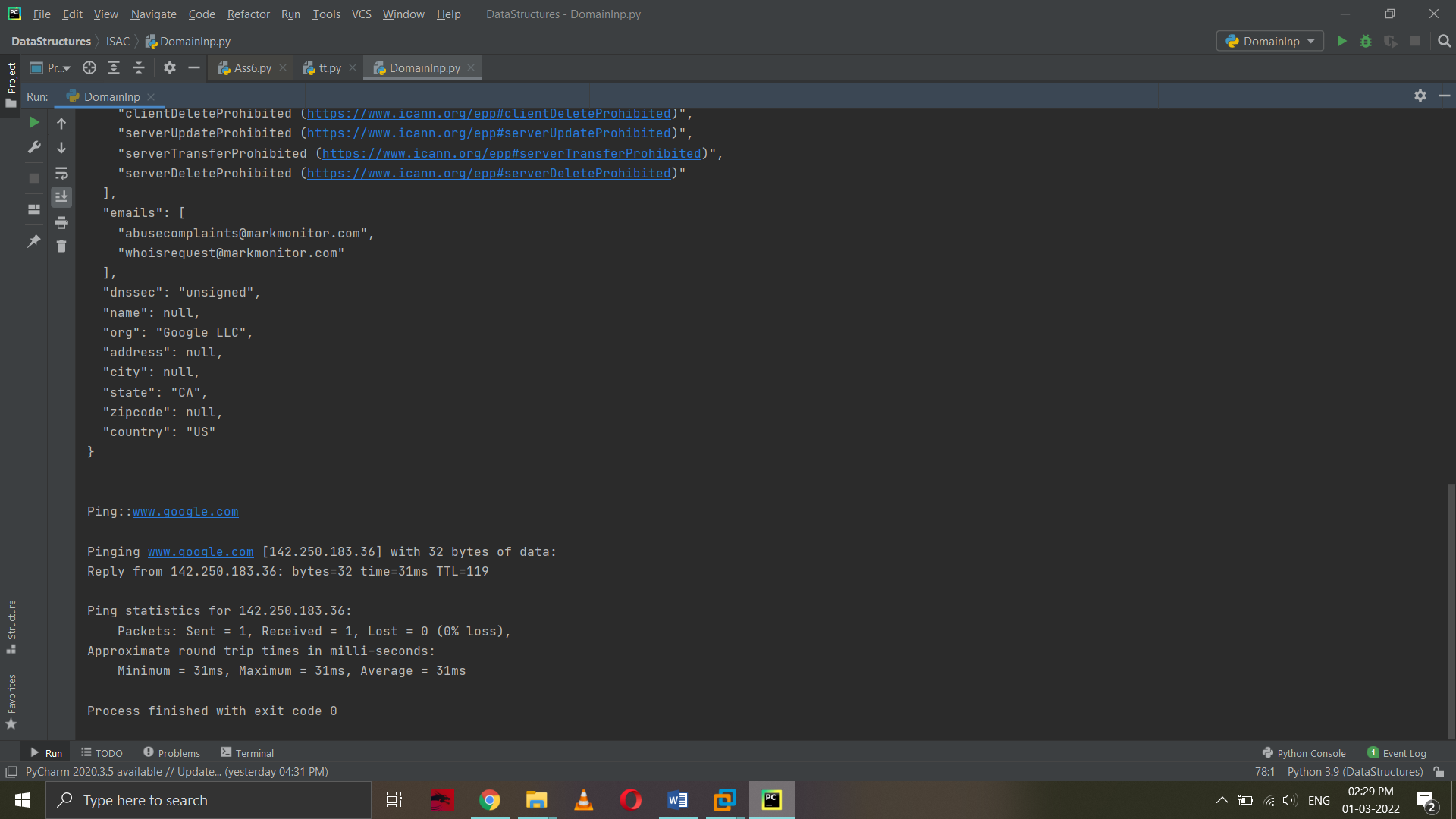
Program:

import socket #for ip of host machine  
import whois #whois  
import platform #ping  
import subprocess #ping  
  
def ping(host):  
 #Returns True if host (str) responds to a ping request.  
  
 para = '-n' if platform.system().lower()== 'windows' else '-c' #setting the paramater in accordance with the OS  
  
 # Building the command. Ex: "ping -c 1 google.com"  
 comm = ['ping', para, '1', host]  
 print ("\n\nPing::" + host)  
 return subprocess.call(comm) == 0  
  
def whoiss(host):  
 w = whois.whois(host) # getting whois information  
 print("\nWhois::"+host)  
 print(w)  
  
hostname = socket.gethostname() #getting the hostname by socket.gethostname() method that is host machine  
hostname2 = "www.google.com" #domain name hardcoded  
ip\_address = socket.gethostbyname(hostname2) #getting the IP address using socket.gethostbyname() method  
  
  
## printing IP and hostname  
print(f"\nHostname: {hostname2}")  
print(f"IP Address: {ip\_address}\n")  
  
whoiss(hostname2)  
ping(hostname2)

Output Screen Shots:







**Output Text:**

Hostname: www.google.com

IP Address: 142.250.183.36

Whois::www.google.com

{

"domain\_name": [

"GOOGLE.COM",

"google.com"

],

"registrar": "MarkMonitor, Inc.",

"whois\_server": "whois.markmonitor.com",

"referral\_url": null,

"updated\_date": [

"2019-09-09 15:39:04",

"2019-09-09 15:39:04+00:00"

],

"creation\_date": [

"1997-09-15 04:00:00",

"1997-09-15 07:00:00+00:00"

],

"expiration\_date": [

"2028-09-14 04:00:00",

"2028-09-13 07:00:00+00:00"

],

"name\_servers": [

"NS1.GOOGLE.COM",

"NS2.GOOGLE.COM",

"NS3.GOOGLE.COM",

"NS4.GOOGLE.COM",

"ns1.google.com",

"ns4.google.com",

"ns3.google.com",

"ns2.google.com"

],

"status": [

"clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited",

"clientTransferProhibited https://icann.org/epp#clientTransferProhibited",

"clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited",

"serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited",

"serverTransferProhibited https://icann.org/epp#serverTransferProhibited",

"serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited",

"clientUpdateProhibited (https://www.icann.org/epp#clientUpdateProhibited)",

"clientTransferProhibited (https://www.icann.org/epp#clientTransferProhibited)",

"clientDeleteProhibited (https://www.icann.org/epp#clientDeleteProhibited)",

"serverUpdateProhibited (https://www.icann.org/epp#serverUpdateProhibited)",

"serverTransferProhibited (https://www.icann.org/epp#serverTransferProhibited)",

"serverDeleteProhibited (https://www.icann.org/epp#serverDeleteProhibited)"

],

"emails": [

"abusecomplaints@markmonitor.com",

"whoisrequest@markmonitor.com"

],

"dnssec": "unsigned",

"name": null,

"org": "Google LLC",

"address": null,

"city": null,

"state": "CA",

"zipcode": null,

"country": "US"

}

Ping::www.google.com

Pinging www.google.com [142.250.183.36] with 32 bytes of data:

Reply from 142.250.183.36: bytes=32 time=31ms TTL=119

Ping statistics for 142.250.183.36:

Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 31ms, Maximum = 31ms, Average = 31ms

Process finished with exit code 0

**Explanation:**

Used packages:

* Socket. for host machines ip if needed
* Whois. To collect the information through whois command
* Platform. To check the Os of host machine.
* Subprocess. To ping the domain

Step1:

* Take domain name as input or hardcode it in the host variable.
* If you need to take domain as the host machine use socket.gethostname()

Step 2:

* To read the ip address of the domain use socket.gethostbyname(hostname2)
* Print the ip address to complete task 1

Step 3:

* To retrieve the whois information associated with the domains ip address use whois.whois(hostname)
* Print the whois information to complete task 2

Step 4:

* Identify the OS of the host machine and accordingly assign the parameter as ‘-n’ for windows or ‘-c’ for other than windows
* Design the command as ‘ping’ + parameter + ‘1’ + host
* Pass the command through subprocess.call()
* This will print the ping command thus completing all tasks