To get started with ShodanTerm, first be sure to have already installed the following python packages:

-Shodan
$ ***pip install shodan***

This package is used to make the api requests from https://shodan.io

-Termcolor
$ ***pip install termcolor***

This package is used to make the python terminal colorful.

The text editor I used is ***Vim***.

These are the steps I followed to make the ShodanTerm program,

1.
The first step I took was to read the official documentation at https://shodan.readthedocs.io and think of a way to implement the program myself, my first thought was creating a gui but if I made a gui I could not record an asciinema,and so I implemented a terminal version of the program,

The first few lines of code are the following:

```
from termcolor import colored
import shodan
from subprocess import *

print(colored("___       ____  __   __   _    ___ __ __ _  _ ","green"))
print(colored("|__ |__| |  | |  \ /  \ | \ |  | |__|__||\/|   ","green"))
print(colored("___||  | |__| |__/ |--| |  \|  | |__|  \|  | ","green"))
print("")
sh_key=input("[*]please enter your shodan key---->")
api=shodan.Shodan(sh_key)
print(colored("1.What's my ip","green"))
print(colored("2.Ip scan","red"))
print(colored("3.Custom Search","yellow"))
opt=input(colored("[*]Enter the option you want to choose+-->","cyan"))
```

I just import the needed modules then declare some needed variables like the api key and the option the user wants to choose.
As you can see, for each input or output I use the colored() method from the termcolor module.

Then I first started by implementing the ”what’s my ip” section of the program;
here is the code of this section

```
while opt!="0":
 if opt=="1":
   print("your ip is: ")
   print(colored(str(check_output(["shodan","myip"])),"green"))
   opt=input(colored("[@]reenter the same or an other option--->","red"))
```

On the first line of this section I made a loop that kills the program if the user gives ‘0’ as the option.
On line 4 I used the subprocess method ‘check_output’ to execute terminal commands from a python file and checks the output.
The user then reenters an option or kills the program with ‘0’.

2.

Then I implemented the ‘IP scan’ section which was certainly the most difficult part I had yet to implement, I needed to do more than simply reading the docs, I started to test the api myself by doing ip scans and carefully reading the output to see which section of the outputted dictionary I needed, I then started writing the code and doing some tests myself, here it is:

```
elif opt=="2":
 ip=input("ip to scan->")
 try:
  host=api.host(ip)
  print(colored("""
  IP: {}
  Organization: {}
  OS: {}
  City: {}
  country: {}
  Hostnames: {}
  Open ports: {}
  Ports: {}
  """.format(host['ip_str'],host.get('org','n/a'),host.get('os','n/a'),host['city'],host['country_name'],','.join(host['hostnames']),len(host['
ports']),str(host['ports'])),'green'))
 except shodan.APIError, e:
       print('Error: {}'.format(e))
```

I put the code in a try→except statement because sometimes for one or another case the program might raise a shodan.APIError exception so I had to catch it before it kills the process.
As you can see I printed the results in a formatted format for readability purposes.

3.

Lastly I implemented the “custom search” section. This one was pretty tricky because it didn’t work like the “ip scan” one so I had to go back to the official documentation to see what I could find, then I found out that the query returns a dictionnary in this state:

```
{
        'total': 8669969,
        'matches': [
                {
                        'data': 'HTTP/1.0 200 OK\r\nDate: Mon, 08 Nov 2010 05:09:59 GMT\r\nSer...
                        'hostnames': ['pl4t1n.de'],
                        'ip': 3579573318,
                        'ip_str': '89.110.147.239',
                        'os': 'FreeBSD 4.4',
                        'port': 80,
                        'timestamp': '2014-01-15T05:49:56.283713'
                },
                ...
        ]
}
```

So it returned a dictionary containing the total number of results and a list containing dictionaries for each match, the tricky thing was how to find the organization name when it was not in the given dictionary so I had to figure out a way to overcome it.
Here is the code:

```
elif opt=="3":
 search=input(colored("[#]enter the search to perform--->","cyan"))
 try:
  res=api.search(search)
  print(colored("Total results Found: "+str(res['total']),"green"))
  print("Please presss ctrl+c to stop the search")
  for result in res['matches']:
   host=api.host(result['ip_str'])
   print(colored(str(result['ip_str']),"green")+" "+colored(str(result['port']),"yellow")+" "+" "+colored(host.get('org','n/a'),"cyan")+" "+col
ored(str(','.join(result['hostnames'])),"red"))
 except shodan.APIError, e:
       print('Error: {}'.format(e))
 opt=input(colored("[@]reenter the same or an other option--->","red"))
-- INSERT --                                                                    49,1          Bot
```

For each result I recuperated the IP address and then ran “***host.get('org','n/a'***” to get the organization name,

These are the steps that I followed to make this program,

Thanks for your time.