

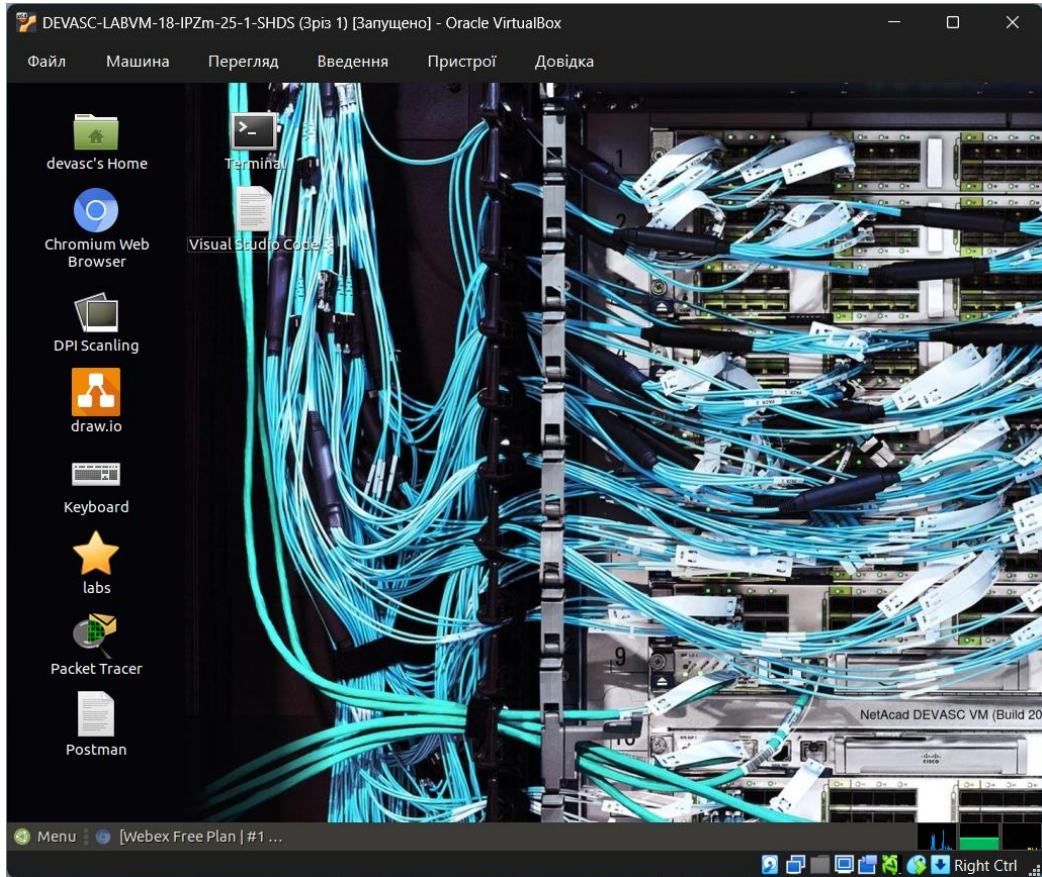
ЛАБОРАТОРНА РОБОТА № 2

Тема: Написання unit-тестів на мові Python.

Мета: отримання базових навичок по написанню unit-тестів на мові Python.

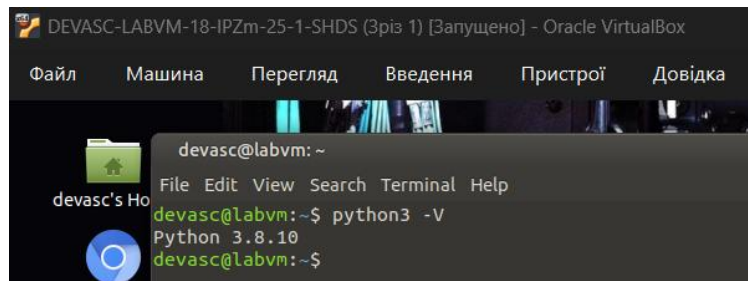
Хід роботи:

1. Запустити DEVASC VM.



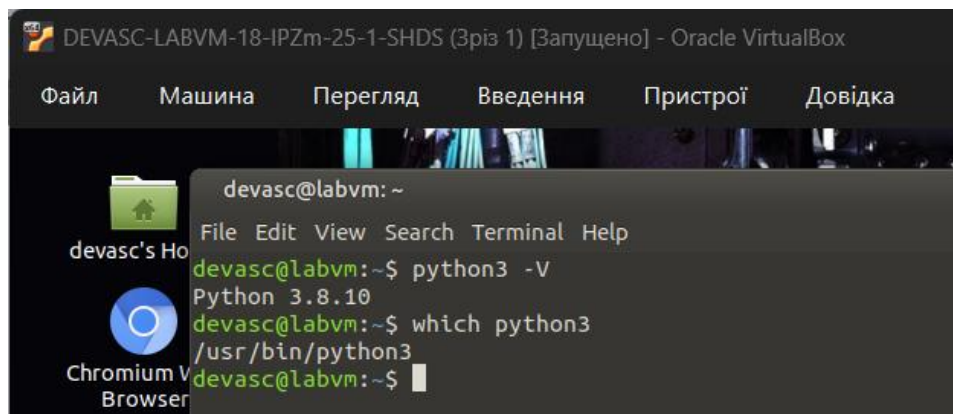
2. Перевірити встановлену версію Python.

Перевірте версію Python, яка вже встановлена у DEVASC VM, за допомогою команди `python3 -V`.



					Житомирська політехніка.26.121.19.000 – Лр2						
Змн.	Арк.	№ докум.	Підпис	Дата	Звіт з лабораторної роботи			Літ.	Арк.	Аркушів	
Розроб.		Шануріна Д.С.									
Перевір.		Морозов Д.С.								1	19
Керівник								ФІКТ Гр. ІПЗм-25-1			
Н. контр.											
Зав. каф.											

Щоб побачити каталог для локального середовища Python, скористайтеся командою `which python3`.

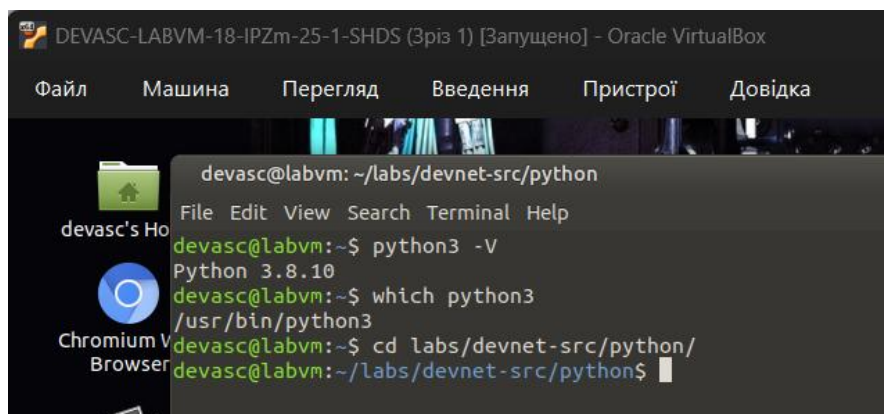


```
devasc@labvm: ~  
File Edit View Search Terminal Help  
devasc@labvm:~$ python3 -V  
Python 3.8.10  
devasc@labvm:~$ which python3  
/usr/bin/python3  
devasc@labvm:~$
```

3. Робота з PIP і Python Virtual Environments.

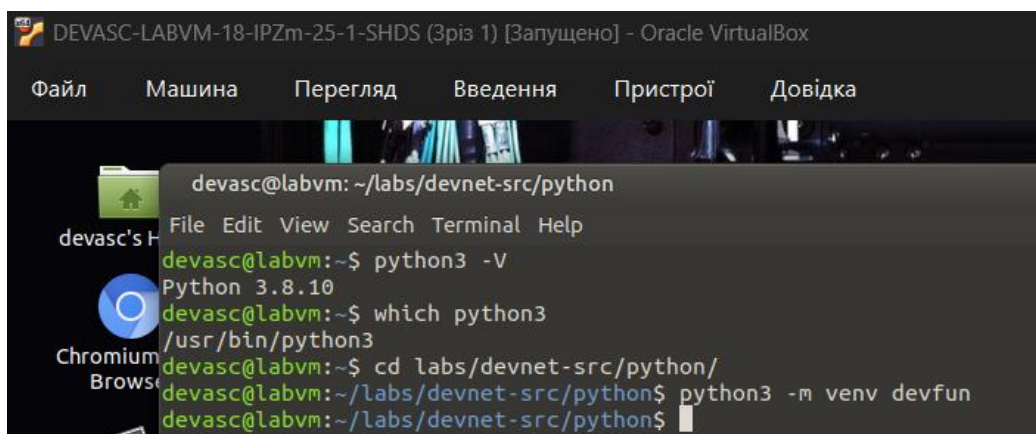
Створення Python virtual environment:

В DEVASC VM, перейдіть в каталог `labs/devnet-src/python`



```
devasc@labvm: ~/labs/devnet-src/python  
File Edit View Search Terminal Help  
devasc@labvm:~$ python3 -V  
Python 3.8.10  
devasc@labvm:~$ which python3  
/usr/bin/python3  
devasc@labvm:~$ cd labs/devnet-src/python/  
devasc@labvm:~/labs/devnet-src/python$
```

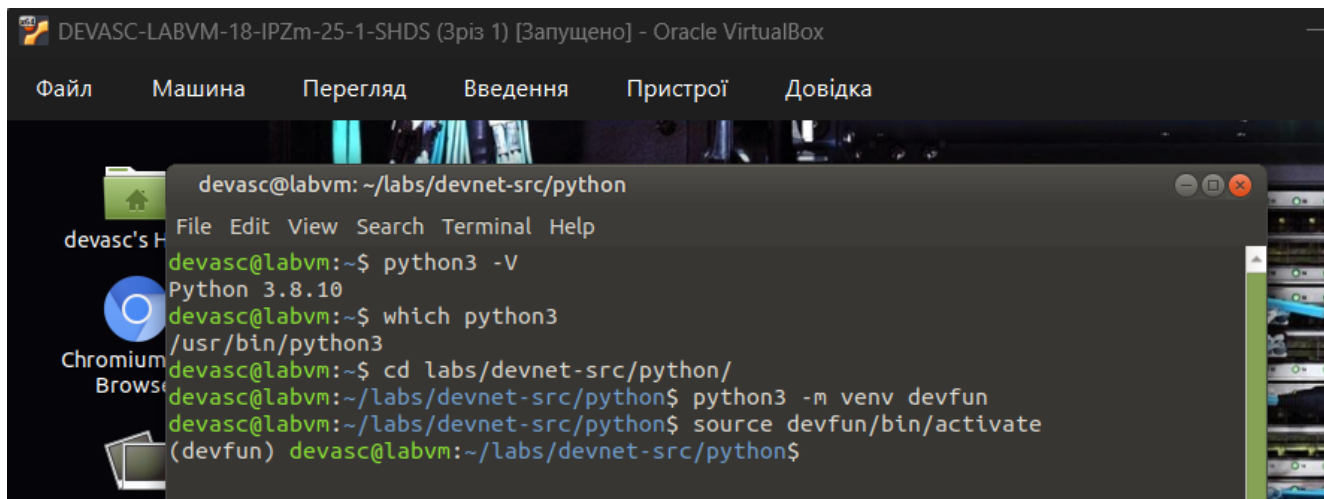
Введіть наступну команду, щоб використовувати інструмент `venv` для створення Python virtual environment з іменем `devfun`. Ключ `-m` наказує Python запустити модуль `venv`.



```
devasc@labvm: ~/labs/devnet-src/python  
File Edit View Search Terminal Help  
devasc@labvm:~$ python3 -V  
Python 3.8.10  
devasc@labvm:~$ which python3  
/usr/bin/python3  
devasc@labvm:~$ cd labs/devnet-src/python/  
devasc@labvm:~/labs/devnet-src/python$ python3 -m venv devfun  
devasc@labvm:~/labs/devnet-src/python$
```

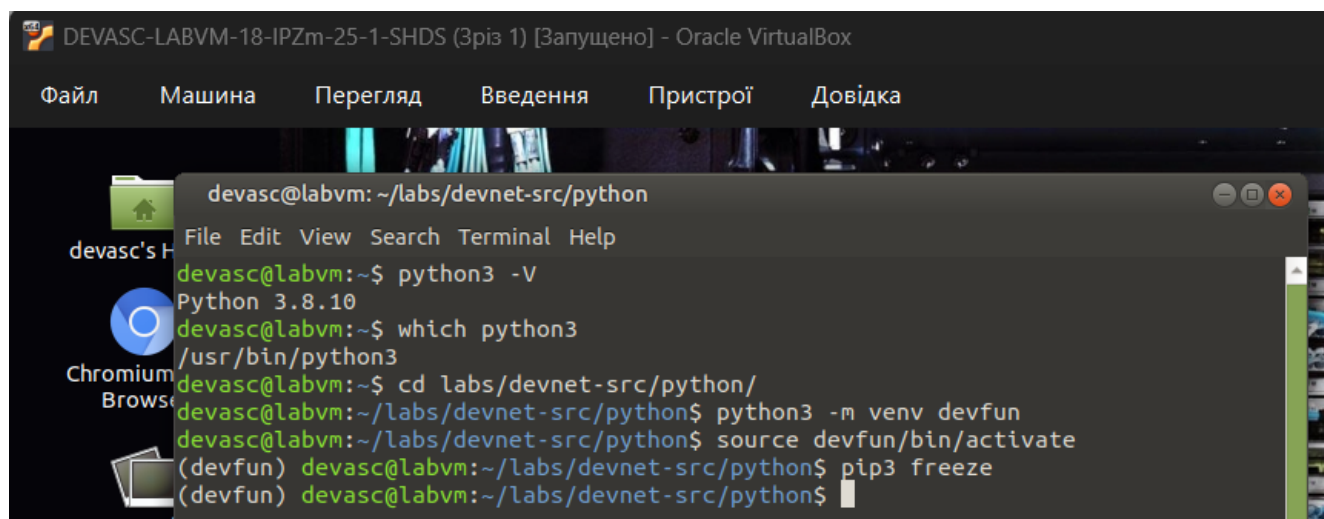
Активация та перевірка Python virtual environment.

Активуйте віртуальне середовище. Підказка змінюється, щоб вказати назву середовища, в якому ви зараз працюєте (в даному випадку *devfun*). Використавши команду *pip3 install*, система встановлюватиме лише пакети для активного віртуального середовища.



```
devasc@labvm: ~/labs/devnet-src/python
File Edit View Search Terminal Help
devasc@labvm:~$ python3 -V
Python 3.8.10
devasc@labvm:~$ which python3
/usr/bin/python3
devasc@labvm:~$ cd labs/devnet-src/python/
devasc@labvm:~/labs/devnet-src/python$ python3 -m venv devfun
devasc@labvm:~/labs/devnet-src/python$ source devfun/bin/activate
(devfun) devasc@labvm:~/labs/devnet-src/python$
```

Виконайте команду *pip3 freeze*, щоб переконатися, що в середовищі *devfun* наразі не містить додаткових пакетів Python.



```
devasc@labvm: ~/labs/devnet-src/python
File Edit View Search Terminal Help
devasc@labvm:~$ python3 -V
Python 3.8.10
devasc@labvm:~$ which python3
/usr/bin/python3
devasc@labvm:~$ cd labs/devnet-src/python/
devasc@labvm:~/labs/devnet-src/python$ python3 -m venv devfun
devasc@labvm:~/labs/devnet-src/python$ source devfun/bin/activate
(devfun) devasc@labvm:~/labs/devnet-src/python$ pip3 freeze
(devfun) devasc@labvm:~/labs/devnet-src/python$
```

Тепер встановіть пакет *requests* у середовищі *devfun*.

```

devasc@labvm: ~/labs/devnet-src/python
File Edit View Search Terminal Help
devasc@labvm:~/labs/devnet-src/python$ source devfun/bin/activate
(devfun) devasc@labvm:~/labs/devnet-src/python$ pip3 freeze
(devfun) devasc@labvm:~/labs/devnet-src/python$ pip3 install requests
Collecting requests
  Downloading requests-2.32.4-py3-none-any.whl (64 kB)
    |-----| 64 kB 1.4 MB/s
Collecting urllib3<3,>=1.21.1
  Downloading urllib3-2.2.3-py3-none-any.whl (126 kB)
    |-----| 126 kB 4.0 MB/s
Collecting charset_normalizer<4,>=2
  Downloading charset_normalizer-3.4.7-cp38-cp38-manylinux2014_x86_64_manylinux_
2_17_x86_64_manylinux_2_28_x86_64.whl (199 kB)
    |-----| 199 kB 14.4 MB/s
Collecting idna<4,>=2.5
  Downloading idna-3.13-py3-none-any.whl (68 kB)
    |-----| 68 kB 5.6 MB/s
Collecting certifi>=2017.4.17
  Downloading certifi-2026.4.22-py3-none-any.whl (135 kB)
    |-----| 135 kB 88.7 MB/s
Installing collected packages: urllib3, charset-normalizer, idna, certifi, requ
sts
Successfully installed certifi-2026.4.22 charset-normalizer-3.4.7 idna-3.13 requ
ests-2.32.4 urllib3-2.2.3
(devfun) devasc@labvm:~/labs/devnet-src/python$

```

Повторно введіть команду `pip3 freeze`, щоб побачити пакети, які зараз встановлені в середовищі `devfun`. Примітка. Ваш список пакетів і номери версій можуть децю відрізнятись.

```

devasc@labvm: ~/labs/devnet-src/python
File Edit View Search Terminal Help
devasc@labvm:~/labs/devnet-src/python$ pip3 freeze
certifi==2026.4.22
charset-normalizer==3.4.7
idna==3.13
requests==2.32.4
urllib3==2.2.3
(devfun) devasc@labvm:~/labs/devnet-src/python$

```

Щоб деактивувати віртуальне середовище та повернутися до системи, введіть команду *deactivate*.

```

devasc@labvm: ~/labs/devnet-src/python
File Edit View Search Terminal Help
Downloading urllib3-2.2.3-py3-none-any.whl (126 kB)
Collecting charset_normalizer<4,>=2
Downloading charset_normalizer-3.4.7-cp38-cp38-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl (199 kB)
Collecting idna<4,>=2.5
Downloading idna-3.13-py3-none-any.whl (68 kB)
Collecting certifi>=2017.4.17
Downloading certifi-2026.4.22-py3-none-any.whl (135 kB)
Installing collected packages: urllib3, charset-normalizer, idna, certifi, requests
Successfully installed certifi-2026.4.22 charset-normalizer-3.4.7 idna-3.13 requests-2.32.4 urllib3-2.2.3
(devfun) devasc@labvm:~/labs/devnet-src/python$ pip3 freeze
certifi==2026.4.22
charset-normalizer==3.4.7
idna==3.13
requests==2.32.4
urllib3==2.2.3
(devfun) devasc@labvm:~/labs/devnet-src/python$ deactivate
devasc@labvm:~/labs/devnet-src/python$
  
```

Перевірка пакетів встановлених в системному оточенні.

Введіть загальносистемну команду *python3 -m pip freeze*, щоб побачити, які пакети встановлені в системному середовищі.

```

devasc@labvm: ~/labs/devnet-src/python
File Edit View Search Terminal Help
devasc@labvm:~/labs/devnet-src/python$ python3 -m pip freeze
aiohttp==3.6.2
ansible==2.9.9
apache-libcloud==2.8.0
appdirs==1.4.3
apturl==0.5.2
argcomplete==1.8.1
arrow==0.15.6
astroid==2.3.3
async-timeout==3.0.1
attrs==19.3.0
backports.ssl==0.0.9
bcrypt==3.1.7
binaryornot==0.4.4
blinker==1.4
Brlapi==0.7.0
cajaarena==19.7.15
certifi==2019.11.28
cffi==1.14.0
chardet==3.0.4
click==7.1.2
command-not-found==0.3
configobj==5.0.6
cookiecutter==1.7.2
cryptography==2.8
cupshelpers==1.0
dbus-python==1.2.16
defer==1.0.6
dill==0.3.2
  
```

		Шануріна Д.С.			Житомирська політехніка.26.121.19.000 – Лр2	Арк.
		Морозов Д.С.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

Якщо ви хочете швидко знайти версію встановленого пакета, передайте вихід команді `grep`. Введіть наступну команду, щоб побачити поточну версію пакета запитів.

```

devasc@labvm: ~/labs/devnet-src/python
File Edit View Search Terminal Help
devasc's H
yamllint==1.23.0
yarl==1.4.2
zipp==1.0.0
devasc@labvm:~/labs/devnet-src/python$ python3 -m pip freeze | grep requests
requests==2.22.0
requests-kerberos==0.12.0
requests-ntlm==1.1.0
requests-toolbelt==0.9.1
requests-unixsocket==0.2.0
devasc@labvm:~/labs/devnet-src/python$
  
```

Перенесення віртуального оточення.

Знову активуйте `devfun` virtual environment.

```

devasc@labvm: ~/labs/devnet-src/python
File Edit View Search Terminal Help
devasc's H
devasc@labvm:~/labs/devnet-src/python$ python3 -m pip freeze | grep requests
requests==2.22.0
requests-kerberos==0.12.0
requests-ntlm==1.1.0
requests-toolbelt==0.9.1
requests-unixsocket==0.2.0
devasc@labvm:~/labs/devnet-src/python$ source devfun/bin/activate
(devfun) devasc@labvm:~/labs/devnet-src/python$
  
```

Запишіть результат виведення команди `pip3 freeze` в файл `requirements.txt`.

```

devasc@labvm: ~/labs/devnet-src/python
File Edit View Search Terminal Help
devasc's H
requests-ntlm==1.1.0
requests-toolbelt==0.9.1
requests-unixsocket==0.2.0
devasc@labvm:~/labs/devnet-src/python$ source devfun/bin/activate
(devfun) devasc@labvm:~/labs/devnet-src/python$ pip3 freeze > requirements.txt
(devfun) devasc@labvm:~/labs/devnet-src/python$
  
```

Вимкніть віртуальне середовище *devfun*. Ви можете скористатися командою *ls*, щоб побачити, що файл *requirements.txt* знаходиться в каталозі */python*.

```

devasc@labvm: ~/labs/devnet-src/python
File Edit View Search Terminal Help
requests-unixsocket==0.2.0
devasc@labvm:~/labs/devnet-src/python$ source devfun/bin/activate
(devfun) devasc@labvm:~/labs/devnet-src/python$ pip3 freeze > requirements.txt
(devfun) devasc@labvm:~/labs/devnet-src/python$ deactivate
devasc@labvm:~/labs/devnet-src/python$ ls
devfun  devices.txt  requirements.txt
devasc@labvm:~/labs/devnet-src/python$
  
```

Створіть і активуйте нове *virtual environment* з назвою *devnew*.

```

devasc@labvm: ~/labs/devnet-src/python
File Edit View Search Terminal Help
(devfun) devasc@labvm:~/labs/devnet-src/python$ pip3 freeze > requirements.txt
(devfun) devasc@labvm:~/labs/devnet-src/python$ deactivate
devasc@labvm:~/labs/devnet-src/python$ ls
devfun  devices.txt  requirements.txt
devasc@labvm:~/labs/devnet-src/python$ python3 -m venv devnew
devasc@labvm:~/labs/devnet-src/python$ source devnew/bin/activate
(devnew) devasc@labvm:~/labs/devnet-src/python$
  
```

Використовуйте команду *pip3 install -r requirements.txt*, щоб встановити ті самі пакети, які встановлені у віртуальному середовищі *devfun*.

```

devasc@labvm: ~/labs/devnet-src/python
File Edit View Search Terminal Help
devasc@labvm:~/labs/devnet-src/python$ source devnew/bin/activate
(devnew) devasc@labvm:~/labs/devnet-src/python$ pip3 install -r requirements.txt
Collecting certifi==2026.4.22
  Using cached certifi-2026.4.22-py3-none-any.whl (135 kB)
Collecting charset-normalizer==3.4.7
  Using cached charset-normalizer-3.4.7-cp38-cp38-manylinux2014_x86_64_manylinux
  _2_17_x86_64_manylinux_2_28_x86_64.whl (199 kB)
Collecting idna==3.13
  Using cached idna-3.13-py3-none-any.whl (68 kB)
Collecting requests==2.32.4
  Using cached requests-2.32.4-py3-none-any.whl (64 kB)
Collecting urllib3==2.2.3
  Using cached urllib3-2.2.3-py3-none-any.whl (126 kB)
Installing collected packages: certifi, charset-normalizer, idna, urllib3, requ
sts
Successfully installed certifi-2026.4.22 charset-normalizer-3.4.7 idna-3.13 requ
ests-2.32.4 urllib3-2.2.3
(devnew) devasc@labvm:~/labs/devnet-src/python$
(devnew) devasc@labvm:~/labs/devnet-src/python$
  
```

Під час введення `pip3 freeze` у середовищі `devnew` ви повинні побачити наступний результат.

```

devasc@labvm: ~/labs/devnet-src/python
(devnew) devasc@labvm:~/labs/devnet-src/python$ pip3 freeze
certifi==2026.4.22
charset-normalizer==3.4.7
idna==3.13
requests==2.32.4
urllib3==2.2.3
(devnew) devasc@labvm:~/labs/devnet-src/python$

```

Деактивуйте `devnew` virtual environment.

```

devasc@labvm: ~/labs/devnet-src/python
certifi==2026.4.22
charset-normalizer==3.4.7
idna==3.13
requests==2.32.4
urllib3==2.2.3
(devnew) devasc@labvm:~/labs/devnet-src/python$ deactivate
devasc@labvm:~/labs/devnet-src/python$

```

4. Проведення тестування за допомогою фреймворку unittest.

Перегляньте файл `test_data.py` file

```

devasc@labvm: ~/labs/devnet-src
devasc@labvm:~/labs/devnet-src$ more unittest/test_data.py
key1 = "issueSummary"
key2 = "XY&^$#@!1234%^&"

data = {
    "id": "AWcvsjx864kVeDHDi2gB",
    "instanceId": "E-NETWORK-EVENT-AWcvsjx864kVeDHDi2gB-1542693469197",
    "category": "Warn",
    "status": "NEW",
    "timestamp": 1542693469197,
    "severity": "P1",
    "domain": "Availability",
    --More-- (5%)

```

Напишіть функцію `json_search()`, що буде тестуватися. Відкрийте файл `~/labs/devnet-src/unittest/recursive_json_search.py`

```
devasc@labvm: ~/labs/devnet-src
File Edit View Search Terminal Help
devasc's H
key1 = "issueSummary"
key2 = "XV&^$#@!1234%^&"

data = {
  "id": "AWcvsjx864kVeDHDi2gB",
  "instanceId": "E-NETWORK-EVENT-AWcvsjx864kVeDHDi2gB-1542693469197",
  "category": "Warn",
  "status": "NEW",
  "timestamp": 1542693469197,
  "severity": "P1",
  "domain": "Availability",
  "source": "DNAC",
  "priority": "P1",
  "type": "Network",
}
devasc@labvm:~/labs/devnet-src$ vi unittest/recursive_json_search.py
```

Скопіюйте код вище у файл і збережіть його

```
devasc@labvm: ~/labs/devnet-src
File Edit View Search Terminal Help
# Fill the Python code in this file
from test_data import *
def json_search(key,input_object):
    ret_val=[]
    if isinstance(input_object, dict): # Iterate dictionary
        for k, v in input_object.items(): # searching key in the dict
            if k == key:
                temp={k:v}
                ret_val.append(temp)
            if isinstance(v, dict): # the value is another dict so repeat
                json_search(key,v)
            elif isinstance(v, list): # it's a list
                for item in v:
                    if not isinstance(item, (str,int)): # if dict or list repeat
                        json_search(key,item)
    else: # Iterate a list because some APIs return JSON object in a list
        for val in input_object:
            if not isinstance(val, (str,int)):
                json_search(key,val)
    return ret_val
print(json_search("issueSummary",data))
```

Запустіть код. Ви не повинні отримати жодних помилок і вивести [], що вказує на порожній список. Якби функція `json_search()` була написана правильно (а це не так), це означало б, що немає даних із ключем «`issueSummary`», повідомленим даними JSON, що повертаються API Cisco DNA Center. Іншими словами, немає проблем, про які слід повідомляти.

```

devasc@labvm: ~/labs/devnet-src/unittest
File Edit View Search Terminal Help
devasc@labvm:~/labs/devnet-src$ vi unittest/recursive_json_search.py
devasc@labvm:~/labs/devnet-src$ python3 recursive_json_search.py
python3: can't open file 'recursive_json_search.py': [Errno 2] No such file or directory
devasc@labvm:~/labs/devnet-src$ cd unittest
devasc@labvm:~/labs/devnet-src/unittest$ python3 recursive_json_search.py
[]
devasc@labvm:~/labs/devnet-src/unittest$

```

Але як дізнатися, що функція `json_search()` працює належним чином? Ви можете відкрити файл `test_data.py` і знайти ключ «`issueSummary`», як показано нижче. Якби ви це зробили, ви справді виявили б, що є проблема. Це невеликий набір даних і відносно простий рекурсивний пошук. Проте виробничі дані та код рідко бувають такими простими. Тому тестування коду є життєво важливим для швидкого пошуку та виправлення помилки у вашому коді.

```

devasc@labvm: ~/labs/devnet-src/unittest
File Edit View Search Terminal Help
devasc@labvm:~/labs/devnet-src$ python3 recursive_json_search.py
python3: can't open file 'recursive_json_search.py': [Errno 2] No such file or directory
devasc@labvm:~/labs/devnet-src$ cd unittest
devasc@labvm:~/labs/devnet-src/unittest$ python3 recursive_json_search.py
[]
devasc@labvm:~/labs/devnet-src/unittest$ vi test_data.py

```

```

devasc@labvm: ~/labs/devnet-src/unittest
File Edit View Search Terminal Help
"enrichmentInfo": {
  "issueDetails": {
    "issue": [
      {
        "issueId": "AWcvsjx864kVeDHDi2gB",
        "issueSource": "Cisco DNA",
        "issueCategory": "Availability",
        "issueName": "snmp_device_down",
        "issueDescription": "This network device leaf2.abc.inc is unreachable
from controller. The device role is ACCESS.",
        "issueEntity": "network_device",
        "issueEntityValue": "10.10.20.82",
        "issueSeverity": "HIGH",
        "issuePriority": "",
        "issueSummary": "Network Device 10.10.20.82 Is Unreachable From Contro
ller",
        "issueTimestamp": 1542693469197,
        "suggestedActions": [
          {
            "message": "From the controller, verify whether the last hop is re
achable.",
            "steps": []
          }
        ]
      }
    ]
  }
}
32,10 18%

```

5. Написання модульних тестів для перевірки коректності роботи функції.

Відкрийте файл `~labs/devnet-src/unittest/test_json_search.py`

```

devasc@labvm: ~/labs/devnet-src/unittest
File Edit View Search Terminal Help
devasc@labvm:~/labs/devnet-src$ cd unittest
devasc@labvm:~/labs/devnet-src/unittest$ python3 recursive_json_search.py
[]
devasc@labvm:~/labs/devnet-src/unittest$ vi test_data.py
devasc@labvm:~/labs/devnet-src/unittest$ vi test_json_search.py

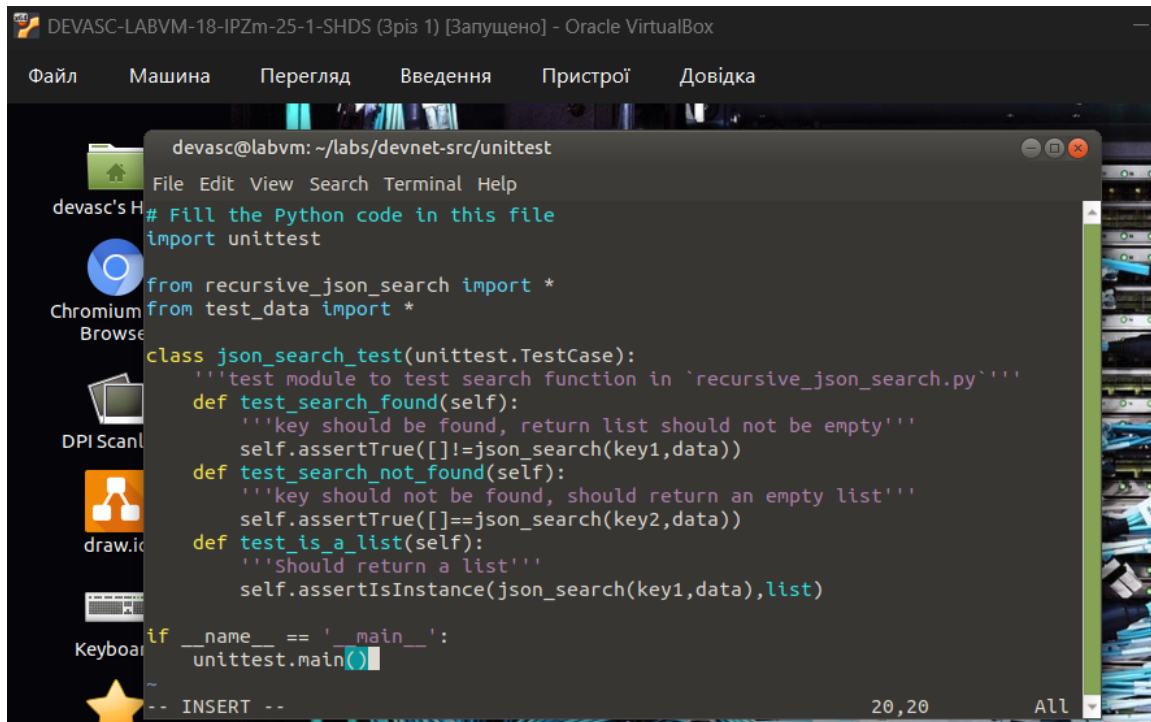
```

```

devasc@labvm: ~/labs/devnet-src/unittest
File Edit View Search Terminal Help
# Fill the Python code in this file

```

Скопіюйте в нього наступний код



```
devasc@labvm: ~/labs/devnet-src/unittest
File Edit View Search Terminal Help
# Fill the Python code in this file
import unittest

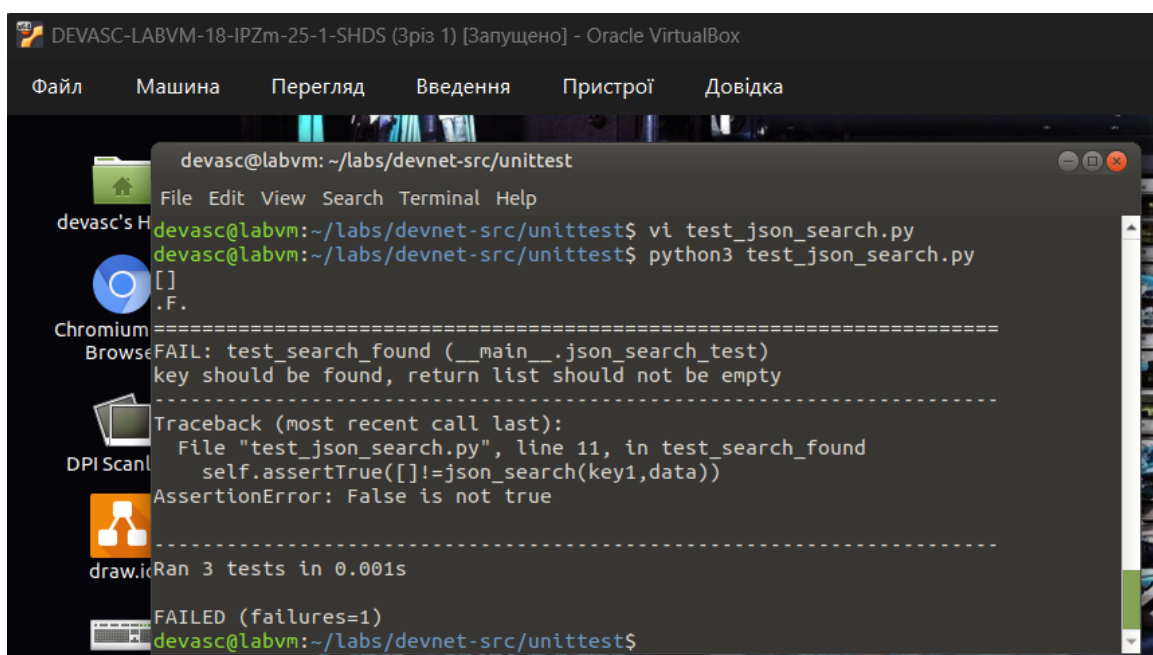
from recursive_json_search import *
from test_data import *

class json_search_test(unittest.TestCase):
    '''test module to test search function in 'recursive_json_search.py'''
    def test_search_found(self):
        '''key should be found, return list should not be empty'''
        self.assertTrue([]!=json_search(key1,data))
    def test_search_not_found(self):
        '''key should not be found, should return an empty list'''
        self.assertTrue([]==json_search(key2,data))
    def test_is_a_list(self):
        '''should return a list'''
        self.assertIsInstance(json_search(key1,data),list)

if __name__ == '__main__':
    unittest.main()
```

6. Запуск тесту для перевірки результатів.

Запустіть тестовий сценарій у його поточному стані, щоб побачити, які результати він зараз повертає. По-перше, ви бачите порожній список. По-друге, ви бачите *.F.* підсвічено у виводі. Крапка (.) означає, що тест пройдено, а *F* означає, що тест не пройшов. Отже, перший тест пройшов, другий — провалено, а третій — пройшов.



```
devasc@labvm: ~/labs/devnet-src/unittest
File Edit View Search Terminal Help
devasc@labvm:~/labs/devnet-src/unittest$ vi test_json_search.py
devasc@labvm:~/labs/devnet-src/unittest$ python3 test_json_search.py
[]
.F.
=====
FAIL: test_search_found (__main__.json_search_test)
key should be found, return list should not be empty
-----
Traceback (most recent call last):
  File "test_json_search.py", line 11, in test_search_found
    self.assertTrue([]!=json_search(key1,data))
AssertionError: False is not true
-----
Ran 3 tests in 0.001s

FAILED (failures=1)
devasc@labvm:~/labs/devnet-src/unittest$
```

Щоб перерахувати кожен тест і його результати, запустіть сценарій ще раз у `unittest` з опцією `-v`. Зверніть увагу, що вам не потрібне розширення `.py` для сценарію `test_json_search.py`. Ви можете побачити, що ваш метод тестування `test_search_found` не працює. Примітка: Python не обов'язково запускає ваші тести по порядку. Тести виконуються в алфавітному порядку на основі назв методів тестування.

```

devasc@labvm: ~/labs/devnet-src/unittest
File Edit View Search Terminal Help
devasc@labvm:~/labs/devnet-src/unittest$ python3 -m unittest -v test_json_search
[]
test_is_a_list (test_json_search.json_search_test)
Should return a list ... ok
test_search_found (test_json_search.json_search_test)
key should be found, return list should not be empty ... FAIL
test_search_not_found (test_json_search.json_search_test)
key should not be found, should return an empty list ... ok
=====
FAIL: test_search_found (test_json_search.json_search_test)
key should be found, return list should not be empty
=====
Traceback (most recent call last):
  File "/home/devasc/labs/devnet-src/unittest/test_json_search.py", line 11, in
test_search_found
    self.assertTrue([]!=json_search(key1,data))
AssertionError: False is not true
=====
Ran 3 tests in 0.001s
FAILED (failures=1)
devasc@labvm:~/labs/devnet-src/unittest$

```

7. Дослідження і виправлення помилки в файлі `recursive_json_search.py`.

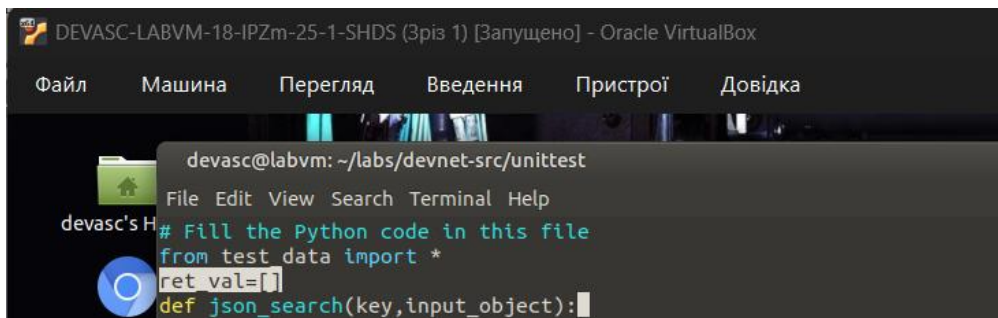
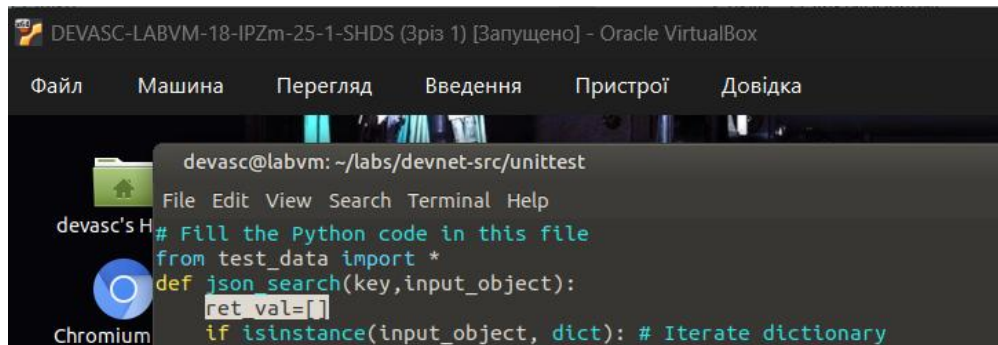
В файлі `recursive_json_search.py` перемістіть `ret_val=[]` з функції, щоб ітерація щоразу не перезаписувала накопичений список.

```

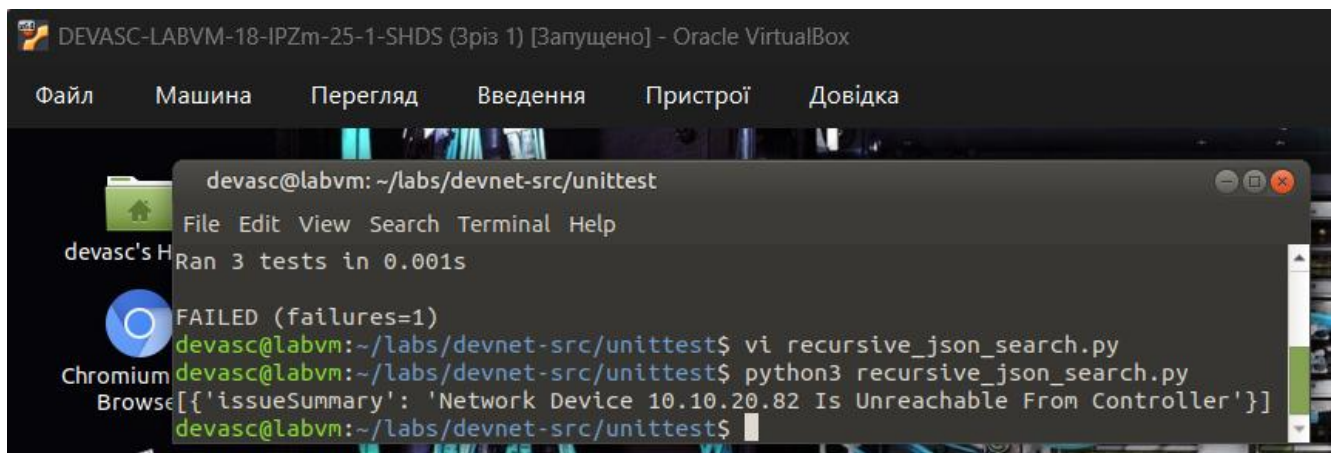
devasc@labvm: ~/labs/devnet-src/unittest
File Edit View Search Terminal Help
devasc@labvm:~/labs/devnet-src/unittest$ python3 -m unittest -v recursive_json_search
Ran 3 tests in 0.001s
FAILED (failures=1)
devasc@labvm:~/labs/devnet-src/unittest$ vi recursive_json_search.py

```

		Шануріна Д.С.			Житомирська політехніка.26.121.19.000 – Лр2	Арк.
		Морозов Д.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		13



Збережіть і запустіть скрипт. Ви повинні отримати наступний вивід, який підтверджує, що ви вирішили проблему. Тобто список більше не порожній після запуску сценарію.



8. Запуск файлу і перевірка виправлення помилок.

Запустіть unittest знову без параметра -v, щоб перевірити, чи повертає test_json_search будь-які помилки. Як правило, ви не використовуватимете параметр -v, щоб мінімізувати вивід в консолі та зробити тести швидшими. На початку звіту ви можете побачити ..F, що означає, що третій тест не пройшов. Також зверніть увагу, що список все ще друкується. Ви можете припинити цю поведінку, видаливши функцію print() у файлі resursive_json_search.py.

		Шануріна Д.С.			Житомирська політехніка.26.121.19.000 – Лр2	Арк.
		Морозов Д.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		14

```

devasc@labvm: ~/labs/devnet-src/unittest
File Edit View Search Terminal Help
devasc@labvm:~/labs/devnet-src/unittest$ python3 -m unittest test_json_search
[{'issueSummary': 'Network Device 10.10.20.82 Is Unreachable From Controller'}]
..F
=====
FAIL: test_search_not_found (test_json_search.json_search_test)
key should not be found, should return an empty list
-----
Traceback (most recent call last):
  File "/home/devasc/labs/devnet-src/unittest/test_json_search.py", line 14, in
test_search_not_found
    self.assertTrue([]==json_search(key2,data))
AssertionError: False is not true
-----
Ran 3 tests in 0.002s
FAILED (failures=1)
devasc@labvm:~/labs/devnet-src/unittest$

```

Відкрийте файл `test_data.py` і знайдіть `issueSummary`, яке є значенням для `key1`. Ви повинні знайти його двічі, але лише один раз в об'єкті даних JSON. Але якщо ви шукаєте значення для `key2`, тобто `XY&^$#@!1234%^&`, ви знайдете його лише вгорі, де воно визначено, оскільки його немає в об'єкті JSON. Третій тест - це перевірка, щоб переконатися, що його там немає. У третьому тестовому коментарі вказано, що ключ повинен не знайдено, має повернути порожній список. Однак функція повернула непорожній список.

```

devasc@labvm: ~/labs/devnet-src/unittest
File Edit View Search Terminal Help
devasc@labvm:~/labs/devnet-src/unittest$ Ran 3 tests in 0.002s
=====
FAILED (failures=1)
devasc@labvm:~/labs/devnet-src/unittest$ vi test_data.py

```

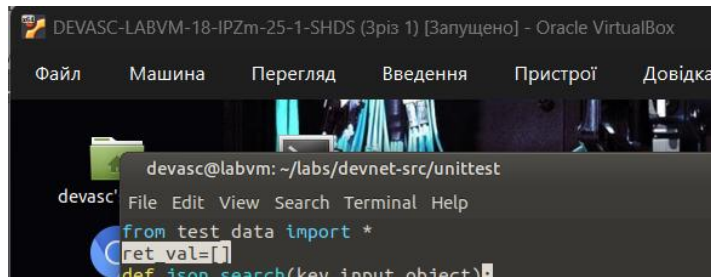
```
devasc@labvm: ~/labs/devnet-src/unittest
key1 = "issueSummary"
key2 = "XY&^$#@!1234%^&"
Chromid = {
  "id": "AWcvsjx864kVeDHDi2qB",
```

```
"issueDescription": "This network device leaf2.abc.inc is unreachable
from controller. The device role is ACCESS.",
"issueEntity": "network_device",
"issueEntityValue": "10.10.20.82",
"issueSeverity": "HIGH",
"issuePriority": "",
"issueSummary": "Network Device 10.10.20.82 Is Unreachable From Contro
```

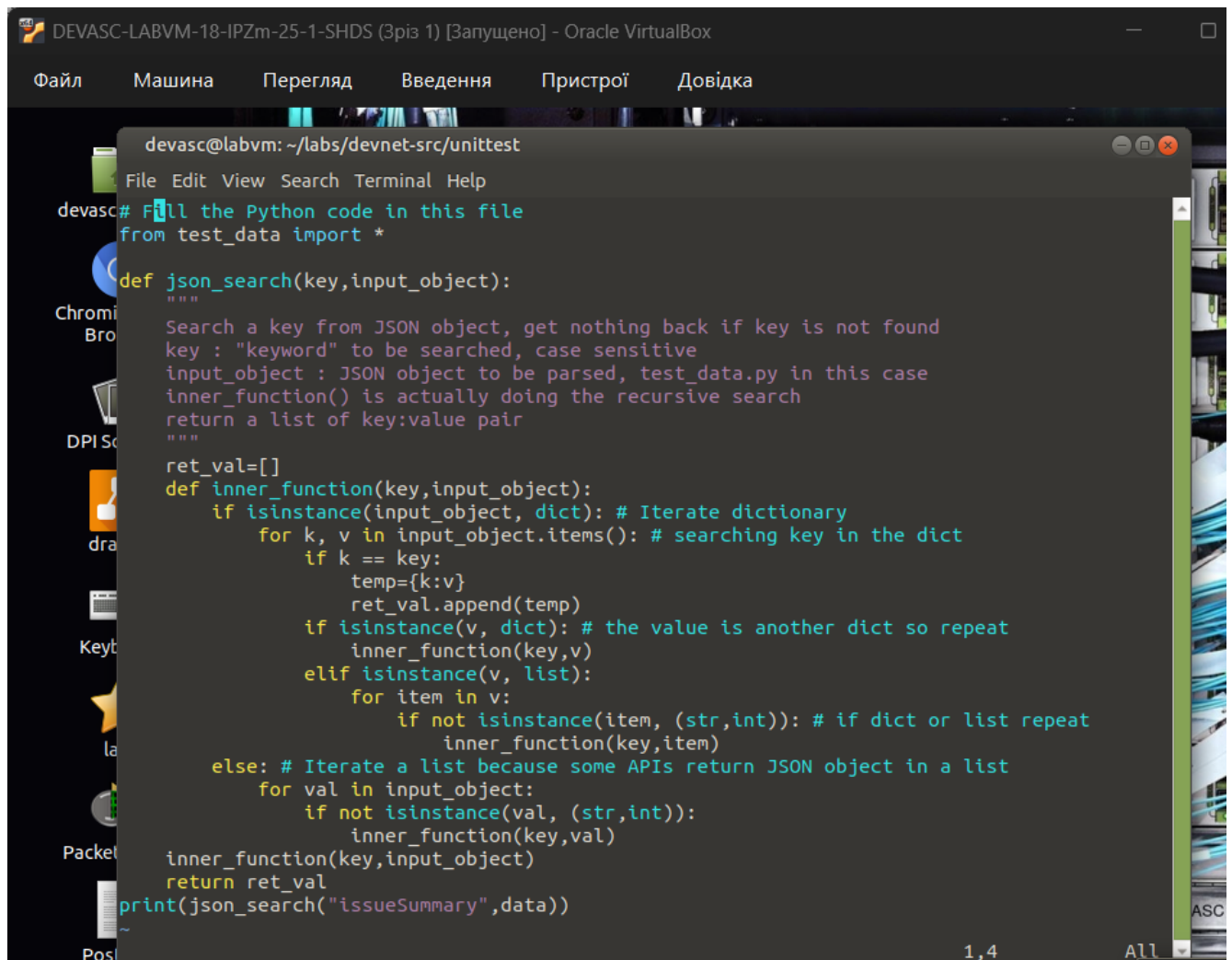
9. Пошук і виправлення другої помилки в recursive_json_search.py.

Перегляньте код `recursive_json_search.py` ще раз. Зверніть увагу, що `ret_val` тепер є глобальною змінною після того, як ви виправили її на попередньому кроці. Це означає, що його значення зберігається при кількох викликах функції `json_search()`. Це хороший приклад того, чому використання глобальних змінних у функціях є поганою практикою.

```
devasc@labvm: ~/labs/devnet-src/unittest
Ran 3 tests in 0.002s
FAILED (failures=1)
devasc@labvm:~/labs/devnet-src/unittest$ vi test_data.py
devasc@labvm:~/labs/devnet-src/unittest$ vi recursive_json_search.py
```



Щоб вирішити цю проблему, оберніть функцію `json_search()` зовнішньою функцією. Видаліть наявну функцію `json_search()` і замініть її на модифіковану функцію нижче:



Збережіть файл і запустіть `unittest`. Ім'я якогось конкретного файлу не потрібно тому, що функція `unittest Test Discovery` запускає будь-який знайдений локальний файл, ім'я якого починається з `test`. Ви повинні отримати наступний результат. Зверніть увагу, що тепер усі тести пройшли, і список для ключа

		Шануріна Д.С.			Житомирська політехніка.26.121.19.000 – Лр2	Арк.
		Морозов Д.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		17

«issueSummary» заповнено. Ви можете безпечно видалити функцію `print()`, оскільки вона зазвичай не використовується, коли цей тест об'єднується з іншими тестами для більшого тесту.

```

devasc@labvm: ~/labs/devnet-src/unittest
File Edit View Search Terminal Help
devasc@labvm:~/labs/devnet-src/unittest$ vi recursive_json_search.py
devasc@labvm:~/labs/devnet-src/unittest$ python3 -m unittest
[{'issueSummary': 'Network Device 10.10.20.82 Is Unreachable From Controller'}]
...
-----
Ran 3 tests in 0.000s
OK
devasc@labvm:~/labs/devnet-src/unittest$

```

```

devasc@labvm: ~/labs/devnet-src/unittest
File Edit View Search Terminal Help
devasc@labvm:~/labs/devnet-src/unittest$ vi recursive_json_search.py
    for val in input_object:
        if not isinstance(val, (str,int)):
            inner_function(key,val)
        inner_function(key,input_object)
    return ret_val
print(json_search("issueSummary",data))
-- INSERT --
31,4 Bot

```

```

devasc@labvm: ~/labs/devnet-src/unittest
File Edit View Search Terminal Help
devasc@labvm:~/labs/devnet-src/unittest$ vi recursive_json_search.py
    for val in input_object:
        if not isinstance(val, (str,int)):
            inner_function(key,val)
        inner_function(key,input_object)
    return ret_val
"recursive_json_search.py" 31L, 1300C written
31,0-1 Bot

```

Висновки: Під час виконання лабораторної роботи було набуто навички роботи з віртуальними середовищами Python, менеджером пакетів PIP, а також написання та використання unit-тестів для виявлення і виправлення помилок у програмному коді. В результаті було налаштовано віртуальні середовища Python за допомогою модуля `venv`, встановлено додаткові бібліотеки, а також виконано

		Шануріна Д.С.			Житомирська політехніка.26.121.19.000 – Лр2	Арк.
		Морозов Д.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		18

експорт та імпорт залежностей через файл requirements.txt. Також було вивчено принципи роботи фреймворку unittest та створено модульні тести для перевірки функції рекурсивного пошуку в JSON-об'єкті.

У процесі тестування було виявлено логічні помилки у функції json_search(), пов'язані з неправильним використанням локальної змінної та глобального стану. Було виправлено помилки шляхом винесення списку результатів за межі рекурсивних викликів та реалізації вкладеної функції для уникнення використання глобальних змінних. Після внесення змін усі unit-тести завершилися успішно.

		Шануріна Д.С.			Житомирська політехніка.26.121.19.000 – Лр2	Арк.
		Морозов Д.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		19