Lab 2 Web Application Security

Name: Mohamad Nour Shahin

Group number: B22-CBS-01

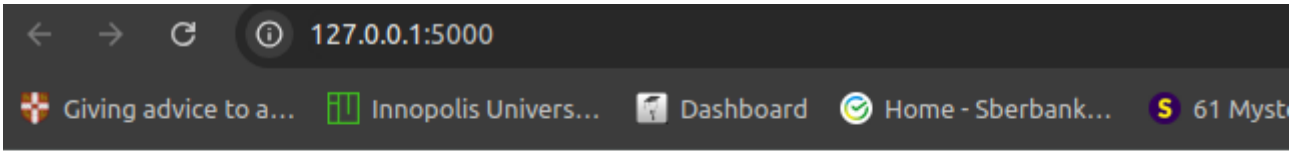
Questions to Answer

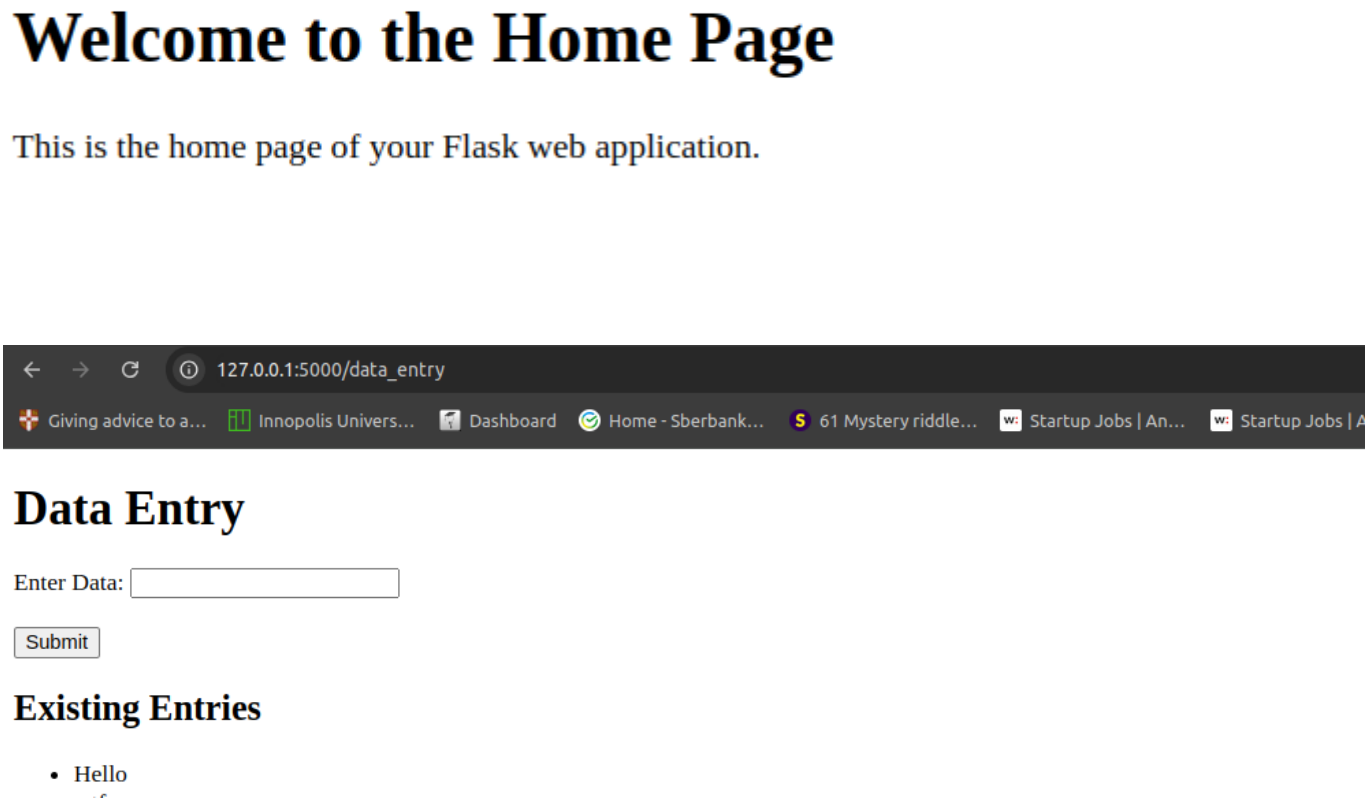
Task 1

Setup a simple web application with a database connectivity (any application works).

Solution:

For solving this task I used the classwork app and i will attached to my solution. you can check the code inside the zip file.





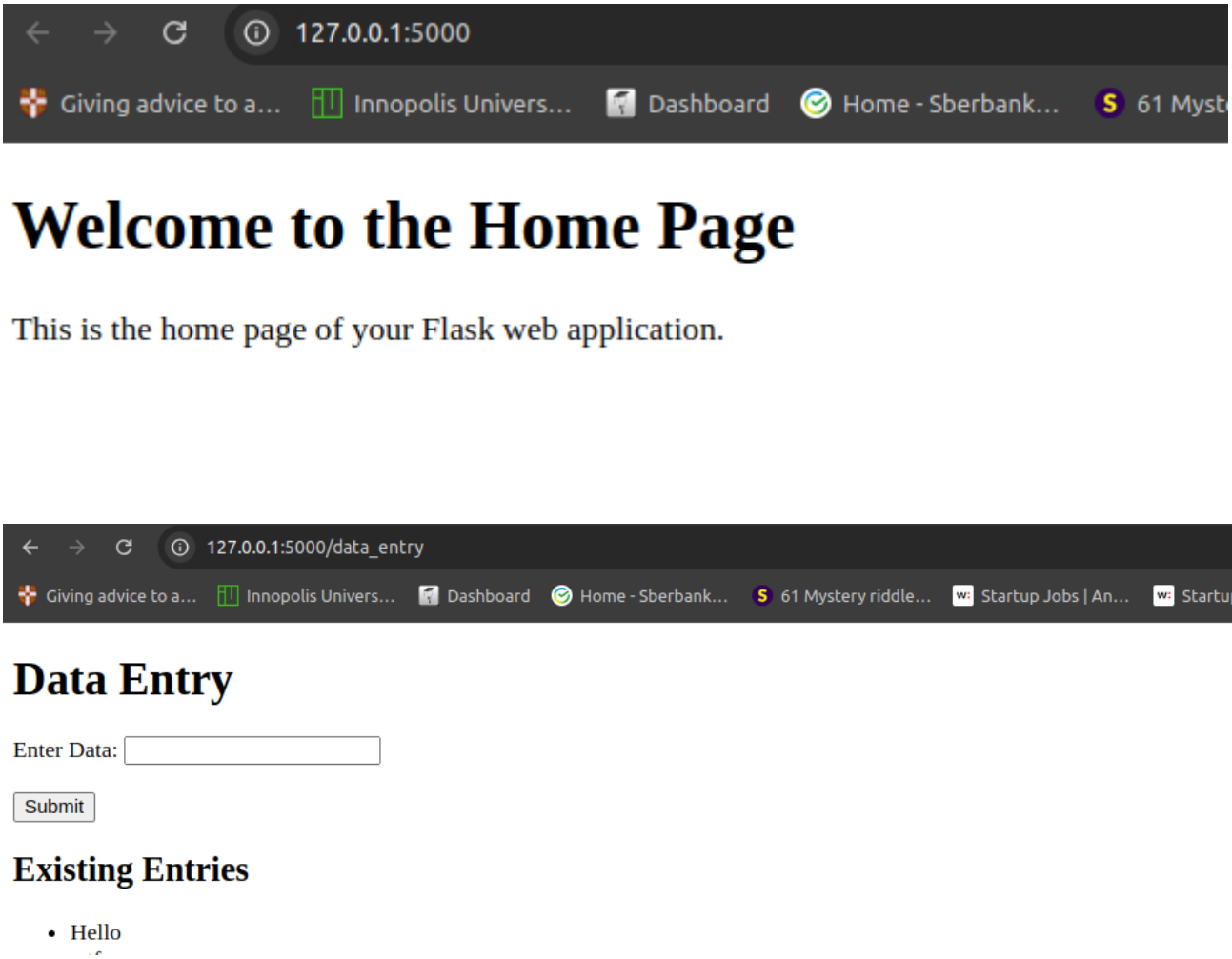
Task 2

Setup authentication to DB using one of the following methods:

- Basic
- Digest
- Certificate

Solution:

For solving this task I used the classwork app and i will attached to my solution. you can check the code inside the zip file.



Task 3

Secure your database:

- Disable remote Root Login of the SQL server
- Disable SSH Password Authentication on the host machine
- Assign a password to the SQL Root User
- Perform the MySQL Secure Installation
- Bind the Database to Localhost

Solution:

1. Disable remote Root Login of the SQL server: Firstly, I will Log in to the MySQL server as the root user from locale machine and entering the password using command:

```
sudo mysql -u root -p
```

```
127.0.0.1 - - [04/Sep/2024 11:27:55] "GET /data_entry HTTP/1.1" 200 -
^C(venv) mohamad@mohamad-Lenovo-ideapad-520-15IKB:~/my_flask_appsudo mysql -u root -p-p
[sudo] password for mohamad:
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 35
Server version: 8.0.39-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Secondly, Run the following SQL script against the MySQL server, to remove all access from remote hosts for the 'root' user account:

```
DELETE FROM mysql.user WHERE User='root' AND Host NOT IN ('localhost',
'127.0.0.1', '::1');
```

```
mysql> DELETE FROM mysql.user WHERE User='root' AND Host NOT IN ('localhost', '127.0.0.1', '::1');
Query OK, 0 rows affected (0.01 sec)
```

Finally, After making changes to permissions/user accounts, make sure you flush the provilege tables using the following command:

```
FLUSH PRIVILEGES;
```

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.02 sec)
```

Testing to connect to from my friend laptop Ali using command in his laptop connecting to same network:

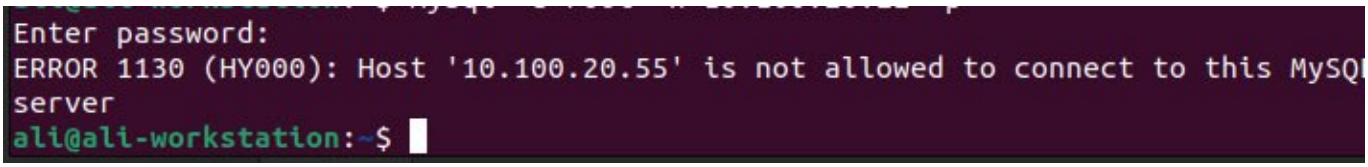
- to know my ip.

```
ifconfig
```

```
wlp3s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.100.20.22  netmask 255.255.254.0  broadcast 10.100.21.255
        inet6 fe80::65c0:b6f4:987a:e014  prefixlen 64  scopeid 0x20<link>
        ether 40:a3:cc:10:09:7d  txqueuelen 1000  (Ethernet)
        RX packets 122685  bytes 136395081 (136.3 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 77102  bytes 18610576 (18.6 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

mohamad@mohamad-Lenovo-ideapad-520-15IKB:~/my flask app$
```

- running command to connect to my local database using command:

```
mysql -u root -h 10.100.20.22 -p
```



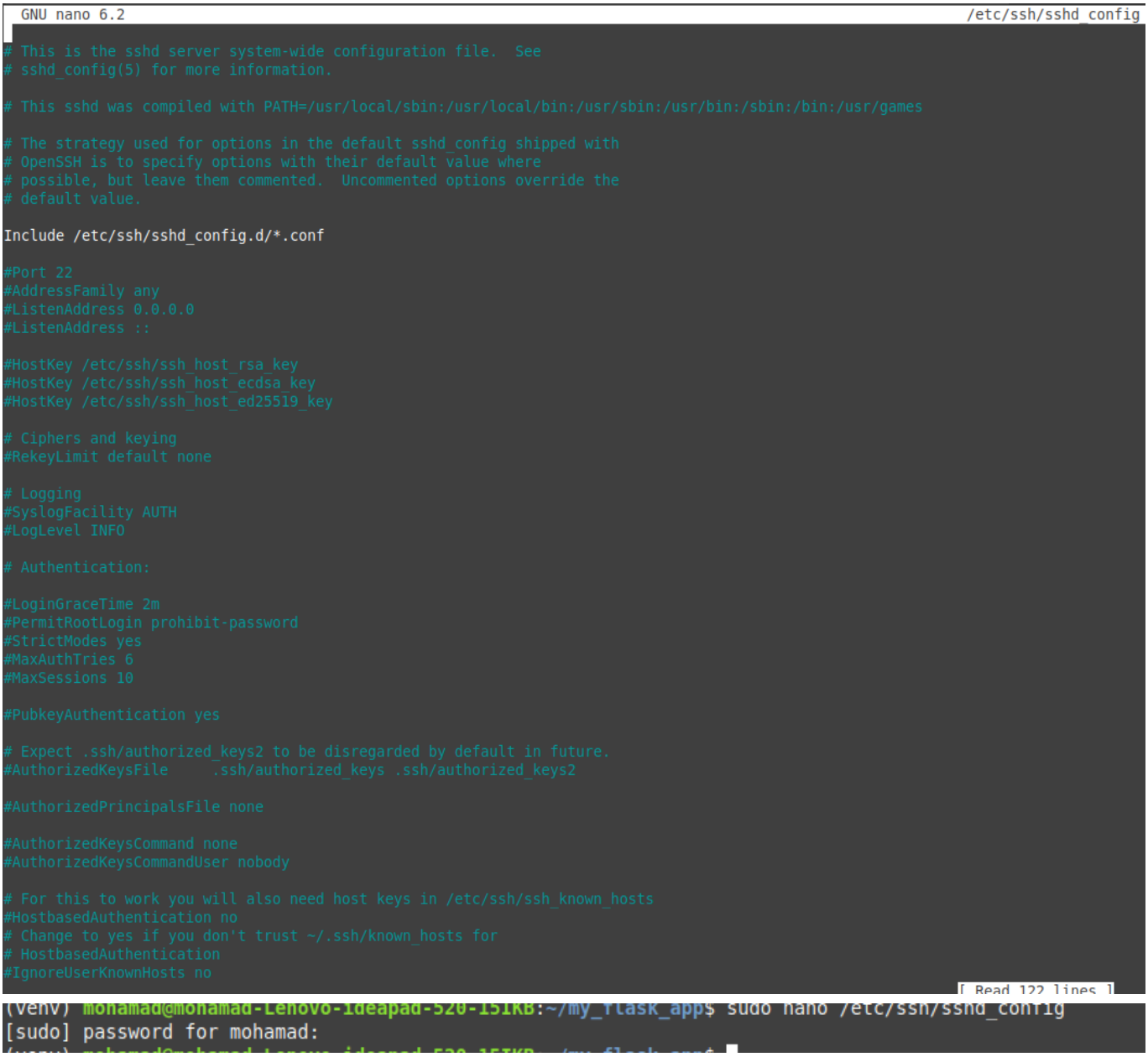
- So we can say that Ali couldn't connect to my database, because I disabled it

---

2. Disable SSH Password Authentication on the host machine:

- Firstly, we need to edit the SSH configiration file where we will open its file and edit it using command:

```
sudo nano /etc/ssh/sshd_config
```



- Secondly, we need to find line #PasswordAuthentication yes and change it to PasswordAuthentication no:

```
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
KbdInteractiveAuthentication no

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no

# GSSAPI options
#GSSAPIAuthentication no
#GSSAPICleanupCredentials yes
#GSSAPIStrictAcceptorCheck yes
#GSSAPIKeyExchange no

# Set this to 'yes' to enable PAM authentication, account processing,
```

- Thirdly, we need to restart the SSH service using command:

```
service ssh restart
```

```
(venv) mohamad@mohamad-Lenovo-ideapad-520-15IKB:~/my_flask_app$ sudo nano /etc/ssh/ssh
(venv) mohamad@mohamad-Lenovo-ideapad-520-15IKB:~/my_flask_app$ service ssh restart
```

Finally, we will test it with Ali device and I gave him my ip before, and connect to same network:

```
ali@ali-workstation:~$ ssh mohamad@10.100.20.22
```

- So we can say that Ali couldn't connect to device using SSH.

---

3. Assign a password to the SQL Root User:

- we did it before while installing the MySql in the device, but we can change it using commands:
    - Login into Mysql

```
sudo mysql -u root -p
ALTER USER 'root'@'localhost' IDENTIFIED WITH caching_sha2_password BY
'123123123';
FLUSH PRIVILEGES;
```

```
(venv) mohamad@mohamad-Lenovo-ideapad-520-15IKB:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 23
Server version: 8.0.39-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>     ALTER USER 'root'@'localhost' IDENTIFIED WITH caching_sha2_password
Y '123123123';
Query OK, 0 rows affected (0.03 sec)

mysql>     FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)
```

---

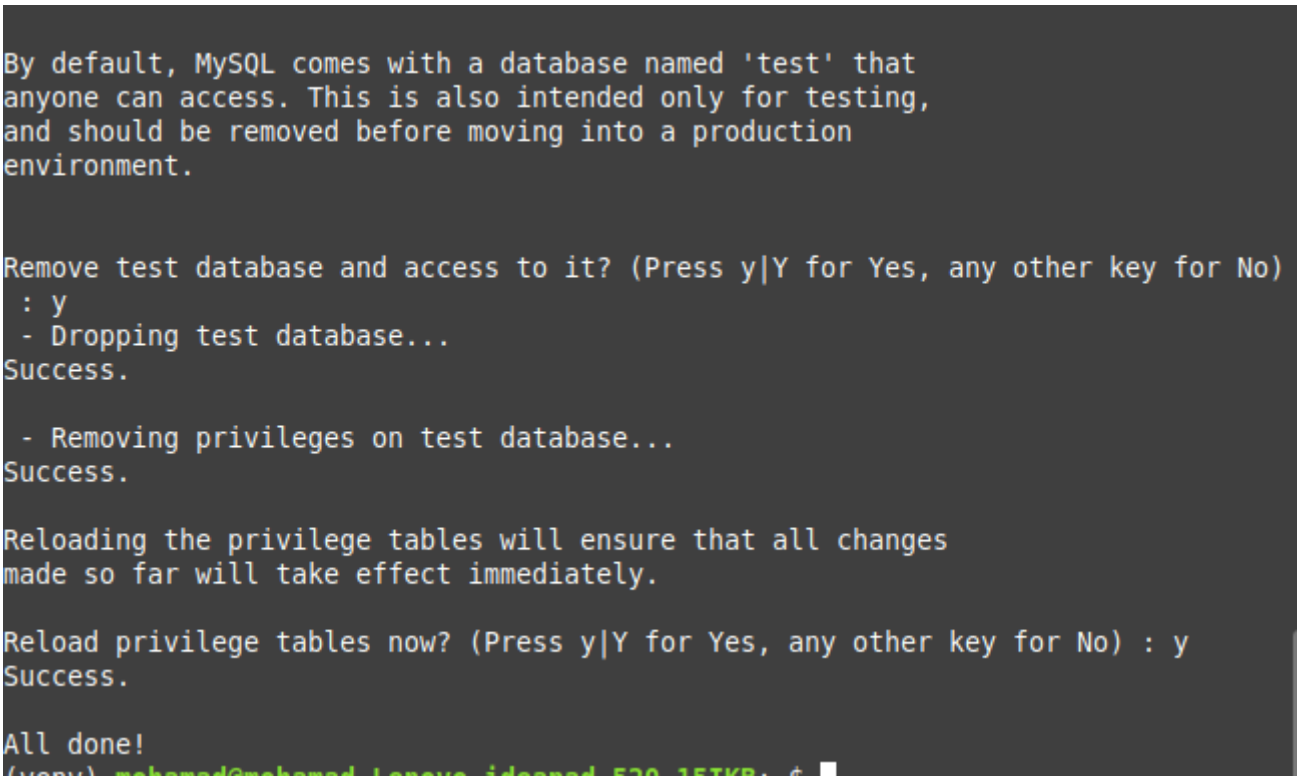
4. Perform the MySQL Secure Installation:

- I will provide the screenshots:

```
(venv) mohamad@mohamad-Lenovo-ideapad-520-15IKB:~$ mysql_secure_installation

Securing the MySQL server deployment.

Enter password for user root:

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: y

There are three levels of password validation policy:

LOW    Length >= 8
MEDIUM Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary
            file

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 0
Using existing password for root.

Estimated strength of the password: 50
```

```
Estimated strength of the password: 50
Change the password for root ? ((Press y|Y for Yes, any other key for No) : y

New password:

Re-enter new password:
Sorry, passwords do not match.

New password:

Re-enter new password:
Sorry, passwords do not match.

New password:

Re-enter new password:

Estimated strength of the password: 100
Do you wish to continue with the password provided?(Press y|Y for Yes, any other
 key for No) : y
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
```

```
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.


Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.


Remove test database and access to it? (Press y|Y for Yes, any other key for No)
```



---

5. Bind the Database to Localhost using the command:

Firstly, we will check the configuration file and update the bind-address value to 127.0.0.1 (localhost) on it using command:

```
sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
cat /etc/mysql/mysql.conf.d/mysqld.cnf
sudo systemctl restart mysql
```

```
(venv) mohamad@mohamad-Lenovo-ideapad-520-15IKB:~$ sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
[sudo] password for mohamad:
```

```
(venv) mohamad@mohamad-Lenovo-ideapad-520-15IKB:~$ cat /etc/mysql/mysql.conf.d/mysqld.cnf
#
# The MySQL database server configuration file.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

# Here is entries for some specific programs
# The following values assume you have at least 32M ram

[mysqld]
#
# * Basic Settings
#
user            = mysql
# pid-file      = /var/run/mysqld/mysqld.pid
# socket        = /var/run/mysqld/mysqld.sock
# port          = 3306
# datadir       = /var/lib/mysql


# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_tmpdir
# tmpdir                = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 127.0.0.1
mysqlx-bind-address     = 127.0.0.1
#
```

```
(venv) mohamad@mohamad-Lenovo-ideapad-520-15IKB:~$ sudo systemctl restart mysql
(venv) mohamad@mohamad-Lenovo-ideapad-520-15IKB:~$ sudo netstat -tulnp | grep mysql
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      413028/mysqld
tcp        0      0 127.0.0.1:33060         0.0.0.0:*               LISTEN      413028/mysqld
(venv) mohamad@mohamad-Lenovo-ideapad-520-15IKB:~$
```

Task 4

Configure rate limiting on your web application and run a benchmark utility against the application to test and see when the application starts dropping requests.

> You can use tools like Apache benchmark.

Solution:

firstly, I will use Flask-Limiter to add rate limiting to the web application using commands:

```
pip install Flask-Limiter
nano my_flask_app/app.py
flask limiter config
flask limiter limits
```

```
(venv) mohamad@mohamad-Lenovo-ideapad-520-15IKB:~$ pip install Flask-Limiter
Collecting Flask-Limiter
  Downloading Flask_Limiter-3.8.0-py3-none-any.whl.metadata (6.1 kB)
Collecting limits>=3.13 (from Flask-Limiter)
  Downloading limits-3.13.0-py3-none-any.whl.metadata (7.2 kB)
Requirement already satisfied: Flask>=2 in ./venv/lib/python3.10/site-packages (
from Flask-Limiter) (3.0.3)
Collecting ordered-set<5,>4 (from Flask-Limiter)
  Downloading ordered_set-4.1.0-py3-none-any.whl.metadata (5.3 kB)
Collecting rich<14,>=12 (from Flask-Limiter)
  Downloading rich-13.8.0-py3-none-any.whl.metadata (18 kB)
Collecting typing-extensions>=4 (from Flask-Limiter)
  Downloading typing_extensions-4.12.2-py3-none-any.whl.metadata (3.0 kB)
Requirement already satisfied: Werkzeug>=3.0.0 in ./venv/lib/python3.10/site-pac
kages (from Flask>=2->Flask-Limiter) (3.0.4)
Requirement already satisfied: Jinja2>=3.1.2 in ./venv/lib/python3.10/site-packa
ges (from Flask>=2->Flask-Limiter) (3.1.4)
Requirement already satisfied: itsdangerous>=2.1.2 in ./venv/lib/python3.10/site
```

```
[notice] To update, run: pip install --upgrade pip
(venv) mohamad@mohamad-Lenovo-ideapad-520-15IKB:~$ nano my_flask_app/app.py
(venv) mohamad@mohamad-Lenovo-ideapad-520-15IKB:~$
```

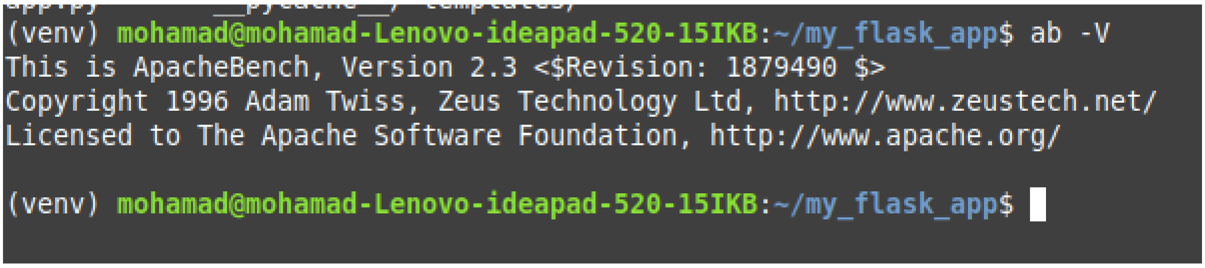
```
(venv) mohamad@mohamad-Lenovo-ideapad-520-15IKB:~/my_flask_app$ flask limiter config
/home/mohamad/venv/lib/python3.10/site-packages/flask_limiter/extension.py:333: UserWarning: Using the in-m
ps://flask-limiter.readthedocs.io#configuring-a-storage-backend for documentation about configuring the sto
  warnings.warn(
                                    Flask-Limiter Config
```

Notes	Configuration	Value
Enabled	RATELIMIT_ENABLED	True
Key Function	RATELIMIT_KEY_FUNC	flask_limiter.util.get_remote_address()
Key Prefix	RATELIMIT_KEY_PREFIX	''
Rate Limiting Config	RATELIMIT_STRATEGY	FixedWindowRateLimiter
	├─ RATELIMIT_STORAGE_URI	└─ N/A
	│ ├─ Instance	│ ├─ MemoryStorage
	│ └─ Backend	│ ├─ Counter()
	├─ RATELIMIT_STORAGE_OPTIONS	│ ├─ {}
	└─ Status	│ └─ OK
ApplicationLimits Limits	RATELIMIT_APPLICATION	[]
Default Limits	RATELIMIT_DEFAULT	['200 per 1 day', '50 per 1 hour']
	RATELIMIT_DEFAULTS_PER_METHOD	False
	RATELIMIT_DEFAULTS_EXEMPT_WHEN	None
	RATELIMIT_DEFAULTS_DEDUCT_WHEN	None
	RATELIMIT_DEFAULTS_COST	1
Header configuration	RATELIMIT_HEADERS_ENABLED	False
Fail on first breach	RATELIMIT_FAIL_ON_FIRST_BREACH	True
On breach callback	RATELIMIT_ON_BREACH_CALLBACK	None

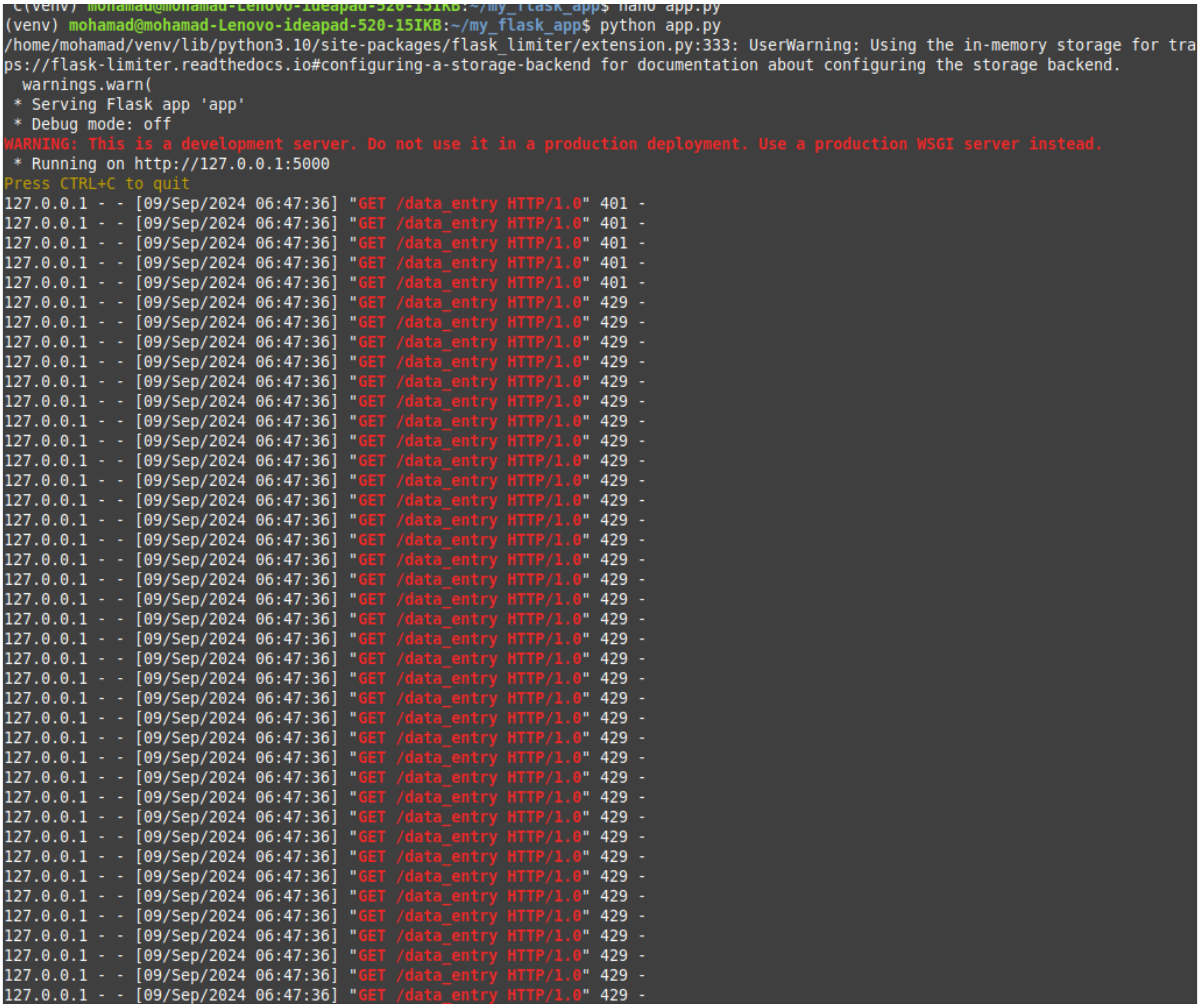
```
(venv) mohamad@mohamad-Lenovo-ideapad-520-15IKB:~/my_flask_app$ flask limiter limits
/home/mohamad/venv/lib/python3.10/site-packages/flask_limiter/extension.py:333: UserWarn
ps://flask-limiter.readthedocs.io#configuring-a-storage-backend for documentation about
  warnings.warn(
app
├── home: /
│   └── Exempt
└── data_entry: /data_entry
    └── 5 per 1 minute
```

secondly, from the images you can see that I added the limits to my app, now we will move to test it using commands:

```
ab -V
ab -n 100 -c 10  http://localhost:5000/data_entry
ab -n 100 -c 10  http://localhost:5000/
```



```
(venv) mohamad@mohamad-Lenovo-ideapad-520-15IKB:~/my_flask_app$ ab -V
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

(venv) mohamad@mohamad-Lenovo-ideapad-520-15IKB:~/my_flask_app$
```



```
(venv) mohamad@mohamad-Lenovo-ideapad-520-15IKB:~/my_flask_app$ python app.py
/home/mohamad/venv/lib/python3.10/site-packages/flask_limiter/extension.py:333: UserWarning: Using the in-memory storage for tra
ps://flask-limiter.readthedocs.io#configuring-a-storage-backend for documentation about configuring the storage backend.
  warnings.warn(
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 401 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 401 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 401 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 401 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 401 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
127.0.0.1 - - [09/Sep/2024 06:47:36] "GET /data_entry HTTP/1.0" 429 -
```


```
mohamad@mohamad-Lenovo-ideapad-520-15IKB:~/my_flask_app$ ab -n 100 -c 10  http://localhost:5000/data_entry
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient).....done


Server Software:        Werkzeug/3.0.4
Server Hostname:        localhost
Server Port:            5000

Document Path:          /data_entry
Document Length:        0 bytes

Concurrency Level:      10
Time taken for tests:   0.159 seconds
Complete requests:      100
Failed requests:        95
   (Connect: 0, Receive: 0, Length: 95, Exceptions: 0)
Non-2xx responses:      100
Total transferred:      30250 bytes
HTML transferred:       11115 bytes
Requests per second:    628.88 [#/sec] (mean)
Time per request:       15.901 [ms] (mean)
Time per request:       1.590 [ms] (mean, across all concurrent requests)
Transfer rate:          185.78 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   0.1      0       0
Processing:    11   15   2.3     15      21
Waiting:        1    3   1.9      3       9
Total:         11   15   2.3     15      21

Percentage of the requests served within a certain time (ms)
  50%     15
  66%     16
  75%     17
  80%     17
  90%     18
  95%     19
  98%     21
  99%     21
 100%     21 (longest request)
mohamad@mohamad-Lenovo-ideapad-520-15IKB:~/my_flask_app$
```

```
mohamad@mohamad-Lenovo-ideapad-520-15IKB:~/my_flask_app$ ab -n 100 -c 10  http://localhost:5000/
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient).....done


Server Software:        Werkzeug/3.0.4
Server Hostname:        localhost
Server Port:            5000

Document Path:          /
Document Length:        191 bytes

Concurrency Level:      10
Time taken for tests:   0.139 seconds
Complete requests:      100
Failed requests:        0
Total transferred:      36600 bytes
HTML transferred:       19100 bytes
Requests per second:    718.48 [#/sec] (mean)
Time per request:       13.918 [ms] (mean)
Time per request:       1.392 [ms] (mean, across all concurrent requests)
Transfer rate:          256.80 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   0.1      0       0
Processing:    11   13   1.8     13      20
Waiting:        1    2   1.6      2       9
Total:         11   13   1.9     13      20

Percentage of the requests served within a certain time (ms)
  50%     13
  66%     13
  75%     14
  80%     14
  90%     15
  95%     17
  98%     20
  99%     20
 100%     20 (longest request)
```

```
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
127.0.0.1 - - [09/Sep/2024 10:32:22] "GET / HTTP/1.0" 200 -
```

Finally, in the screenshots you can notice that limter was working.

Task 5

Monitor/protect the webserver using any of the following tools:

- Network Firewall.
- Intrusion Detection System such as Snort or Suricata.
- Web Application Firewalls such as IronBee, WebKnight, Shadow Daemon.
- Monitoring tools such as Prometheus, Grafana.
- Honeypots; Glastopf, WebTrap, honeyhttpd

> https://github.com/paralax/awesome-honeypots

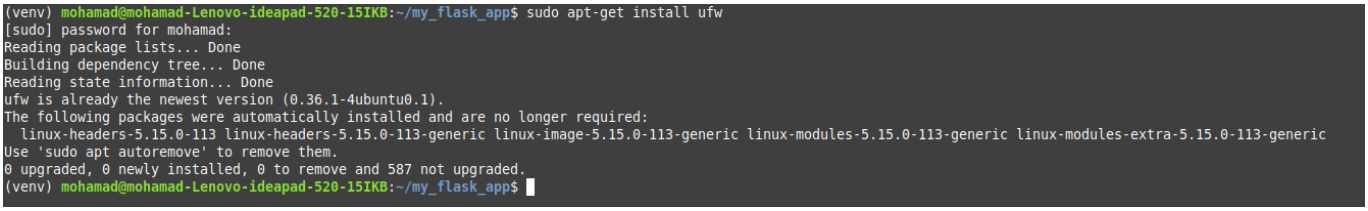
> https://owasp.org/www-community/Web_Application_Firewall

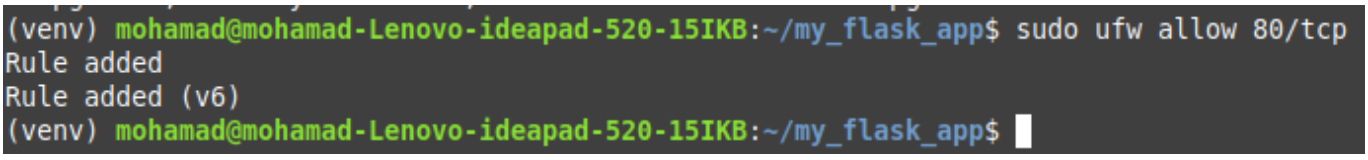
Solution:

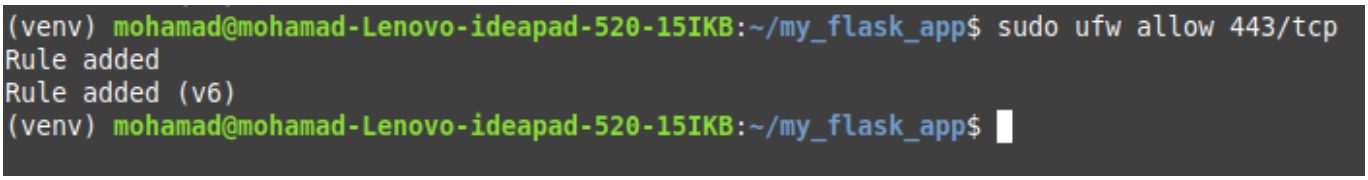
I will use network firewall especially ufw using commands:

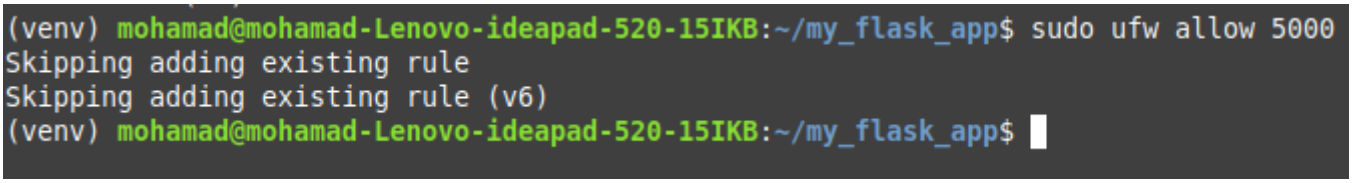
```
sudo apt-get install ufw
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp
sudo ufw allow 5000
sudo ufw deny 3306
sudo ufw enable
sudo ufw status
```

- Firstly, i will install the package, and allowing connecting to 80, 443, 5000 (our localhost), and denying connecting to our database using outside connection using ssh
- here is the screenshots:









```
(venv) mohamad@mohamad-Lenovo-ideapad-520-15IKB:~/my_flask_app$ sudo ufw deny 3306
Rule added
Rule added (v6)
(venv) mohamad@mohamad-Lenovo-ideapad-520-15IKB:~/my_flask_app$
```

```
Rule added (v6)
(venv) mohamad@mohamad-Lenovo-ideapad-520-15IKB:~/my_flask_app$ sudo ufw enable
Firewall is active and enabled on system startup
(venv) mohamad@mohamad-Lenovo-ideapad-520-15IKB:~/my_flask_app$
```

```
(venv) mohamad@mohamad-Lenovo-ideapad-520-15IKB:~/my_flask_app$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
3306/tcp                   ALLOW       Anywhere
5000                       ALLOW       Anywhere
5432                       DENY        Anywhere
80/tcp                     ALLOW       Anywhere
443/tcp                    ALLOW       Anywhere
3306                       DENY        Anywhere
3306/tcp (v6)              ALLOW       Anywhere (v6)
5000 (v6)                  ALLOW       Anywhere (v6)
5432 (v6)                  DENY        Anywhere (v6)
80/tcp (v6)                ALLOW       Anywhere (v6)
443/tcp (v6)               ALLOW       Anywhere (v6)
3306 (v6)                  DENY        Anywhere (v6)

(venv) mohamad@mohamad-Lenovo-ideapad-520-15IKB:~/my_flask_app$
```

- Finally, you can see that the firewall is active and the ports tracking.

Task 6

Salting:

- Use the sha256sum utility to hash your name at least twice. Are the output the same?
- Create two random strings A and B of 6 characters each. These strings are your salts.
- Append string A to your name and hash it.
- Append string B to your name and hash it.
- How do you think such salting practices will help if usernames and passwords get exposed in a database leak.
- Find any database leak with user credentials on the internet. Use only openly available information from sources such as pastbein. Don't perform any active reconnaissance.

Solution:

1. Use the sha256sum utility to hash your name at least twice. Are the output the same using commands:

```
echo -n "Mohamad Nour Shahin" | sha256sum
echo -n "Mohamad Nour Shahin" | sha256sum
```

```
mohamad@mohamad-Lenovo-ideapad-520-15IKB:~$ echo -n "Mohamad Nour Shahin" | sha256sum
d9c2e2f9f61decc0abfe4d3846d60e0cfc355864303b02a7c432a4e9feed1e2a  -
mohamad@mohamad-Lenovo-ideapad-520-15IKB:~$ echo -n "Mohamad Nour Shahin" | sha256sum
d9c2e2f9f61decc0abfe4d3846d60e0cfc355864303b02a7c432a4e9feed1e2a  -
mohamad@mohamad-Lenovo-ideapad-520-15IKB:~$
```

it was the smae hashing

2. Create two random strings A and B of 6 characters each. These strings are your salts using command:.

```
openssl rand -hex 3
```

```
mohamad@mohamad-Lenovo-ideapad-520-15IKB:~$ openssl rand -hex 3
cf8eff
mohamad@mohamad-Lenovo-ideapad-520-15IKB:~$ openssl rand -hex 3
58f79e
mohamad@mohamad-Lenovo-ideapad-520-15IKB:~$
```

3. Append string A to your name and hash it and Append string B to your name and hash it using commands:

```
echo -n "Mohamad Nour Shahincf8eff" | sha256sum
echo -n "Mohamad Nour Shahin58f79e" | sha256sum
```

```
mohamad@mohamad-Lenovo-ideapad-520-15IKB:~$ echo -n "Mohamad Nour Shahincf8eff" | sha256sum
d17c23a838ea50e0a253f6635f8e91ede5990a09d5715b7416547513c5445123  -
mohamad@mohamad-Lenovo-ideapad-520-15IKB:~$ echo -n "Mohamad Nour Shahin58f79e" | sha256sum
164c862ab2bbb1a00cf50185a82934a25ec88495b8cb5141aa8a13e69bec510c  -
mohamad@mohamad-Lenovo-ideapad-520-15IKB:~$
```

now it is different hashing, after adding the slating.

4. How do you think such salting practices will help if usernames and passwords get exposed in a database leak.

Password salting increases password complexity, making them unique and secure without affecting user experience. It also helps prevent hash table attacks and slows down brute-force and dictionary attacks.

5. Find any database leak with user credentials on the internet. Use only openly available information from sources such as pastbein. Don't perform any active reconnaissance.

here is the link to the leak data and a screenshot.

https://pastebin.com/6DENG8xS

```
This leak includes the emails and passwords for 66 MILLION tumblr Users accounts.
Just open up the database in your favorite text editor and Ctrl + F for the email you want to hack.

Proof of content 100 lines of accounts:
Format is email:password (password is SH1 hashed)




jesselperry@gmail.com:5a529448b621ce306fc29503abde0da437d7be05
picchiaccio@hotmail.com:5529f8a2630dbfb5071601ff78c22930fcdddd96
tomharpel@hotmail.com:1d5e6e28a3426fab2bad8a95d22752b405039200
benzinger@gmail.com:919db65a0f3fb2115d8b5ac7c3fb0e856f0ae218
ryan.cates@gmail.com:a20f5d605573101b051f3a68707cf762ed61bdfd
mkis78@gmail.com:2348c738ba4aa2ee93c65d09c8dbcbfecf8b6582
anil@dashes.com:f90d5735980869026a8f02ce6cfd0ad7117c3182
laumerritt@yahoo.com.mx:180d747ab5833054f60725836c5e9eff6a5891f7
adrianoamalfi@gmail.com:11d8061e005d546e8412c298d11d72443cc24b59
skott@email.it:7880b1c01c0724d9ff0ed937810dfc17abb5f6a0
soulunwind@me.com:25a8bc188cef346e3b18d505bd909662583ad30c
c4tacomb@gmail.com:319b7029ad2593ed41bce1d454869b9f61ff5ae3
bibano@virgilio.it:f7ec12947f4523ea2f919f7202da04ec48592369
nathans@gmail.com:8a45437b4dd5aedde79839cb58724e138b8cc380
massimo.andretta@gmail.com:e9050e68cb2caf9a3c295f5dddb1615974dd1d32
gerosa.roberto@gmail.com:852fc84336eab1d20a1ca992a1b90ca9956435f8
andreasiron@virgilio.it:964fd863d750d0e8ad894d718ea80df6b4822095
cmarina@gmail.com:800e8da844950a13ba5f32c18d8ea35729d78d2f
info@smashmedia.it:0ae69e649d5ecfe8d7be3ee98497d9c97c37d17e
me@filippocorti.net:b5e0c1c5974475147e255a6cac7ee1aa8a0ff07c
mth_dir@yahoo.it:bc754725ea81d97738e36402b982ea8a39f6c2e8
cstomaci@alice.it:9dfe1aab2a4ca756c7bbb17f459dd918469837ef
ms_michel@hotmail.com:f998dc05ae9073973ddce48c40e6cb55920acefe
aborruso@gmail.com:0dc16199d11ce789d18946e0c0d36591c786f2c6
l_decrescienzo@virgilio.it:ecfaa9d278f3db2118d5942fd756613374f97b67
```

Task 7

Describe OWASP Top 10 in your own words.

Open Web Application Security Project https://owasp.org/Top10/

Solution:

1. **Broken Access Control**
   When people can access files or areas they shouldn't. Example: Someone gets into private files that should be hidden.

2. **Cryptographic Failures**
   Weak or missing encryption that makes it easy to steal data. Example: Not encrypting passwords properly allows hackers to guess them.

3. **Injection**
   Unfiltered user inputs lead to attacks. Example: A hacker can trick the system using a SQL query.

4. **Insecure Design**
   Bad system design leads to security problems. Example: Two users buying the same item at the same time and causing issues.

5. **Security Misconfiguration**
   Mistakes in setup leave security holes. Example: Leaving unnecessary ports open on a server.

6. **Vulnerable and Outdated Components**
   Using old, unsafe software parts. Example: An old version of a plugin has known issues hackers can exploit.

7. **Identification and Authentication Failures**
   Issues with verifying users. Example: Allowing weak passwords or letting hackers guess passwords without limits.

8. **Software and Data Integrity Failures**
   Failure to keep data and software safe from tampering. Example: Relying on untrusted sources for updates.

9. **Security Logging and Monitoring Failures**
   Not tracking system activity well enough. Example: Failing to log failed login attempts, missing signs of attacks.

10. **Server-Side Request Forgery (SSRF)**
    The server makes unsafe requests on behalf of a user. Example: Hackers trick the server into connecting to malicious sites.

Task 8

> Is it okay to hardcode secrets such as database passwords in your application source code?

No, hardcoding secrets like database passwords in your application source code is not a good practice. This can expose your sensitive information to attackers, especially if the source code is shared or compromised. It increases the risk of accidental leaks (through version control systems like Git) and can lead to security breaches.

> How can you manage such secrets?

To securely manage secrets like database credentials, you can use a **Secret Management Solution**. Some common methods are:

- **Environment Variables**: Store secrets outside of the source code, typically in environment variables.
- **Encrypted Configuration Files**: Use encrypted files to store secrets and decrypt them at runtime.

- **Secrets Management Tools**: Use dedicated tools such as **HashiCorp Vault**, **AWS Secrets Manager**, or **Azure Key Vault**.

Referneces:

Refernece 1 Refernece 2 Refernece 3 Refernece 4 -->