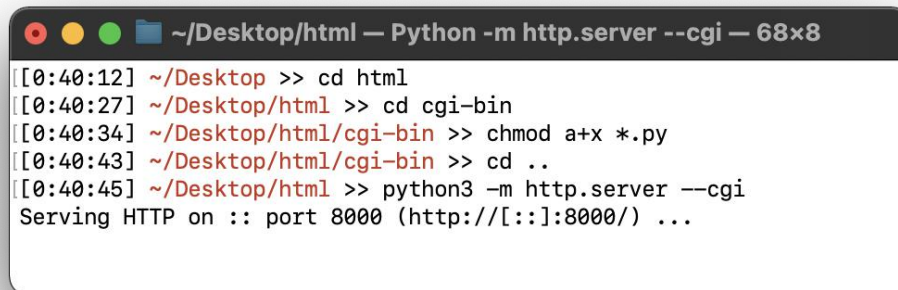


1. First of all, create a directory called "html".
2. Open terminal to create a local python server



```
~/Desktop/html — Python -m http.server --cgi — 68x8
[[0:40:12] ~/Desktop >> cd html
[[0:40:27] ~/Desktop/html >> cd cgi-bin
[[0:40:34] ~/Desktop/html/cgi-bin >> chmod a+x *.py
[[0:40:43] ~/Desktop/html/cgi-bin >> cd ..
[[0:40:45] ~/Desktop/html >> python3 -m http.server --cgi
Serving HTTP on :: port 8000 (http://[::]:8000/) ...
```

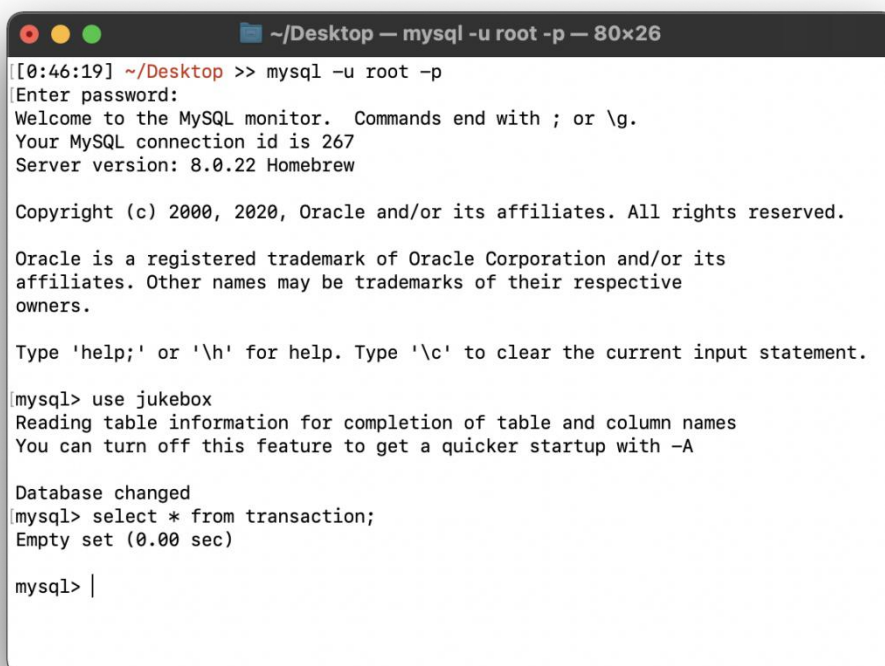
3. Open another terminal to access mysql database,
4. Type this command: `mysql -u root -p`
5. After you login as root, create a database called "JUKEBOX",

`CREATE DATABASE JUKEBOX;`

6. And create a table called "TRANSACTION",

```
CREATE TABLE `TRANSACTION` (
  `song` varchar(10) DEFAULT NULL,
  `cc_number` varchar(20) DEFAULT NULL);
```

7. Now we have an empty table,



```
~/Desktop — mysql -u root -p — 80x26
[[0:46:19] ~/Desktop >> mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 267
Server version: 8.0.22 Homebrew

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

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

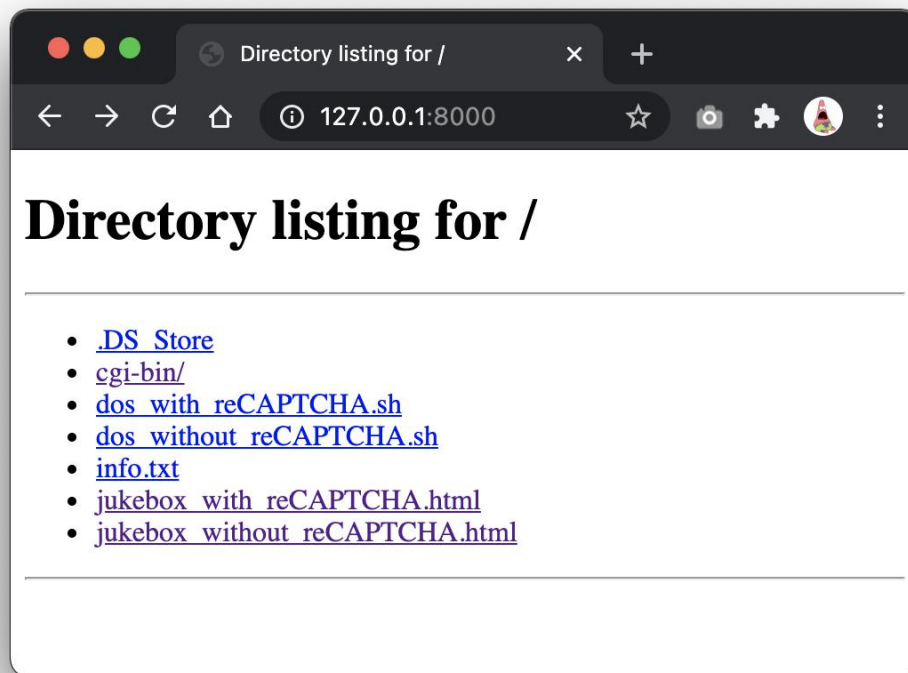
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use jukebox
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

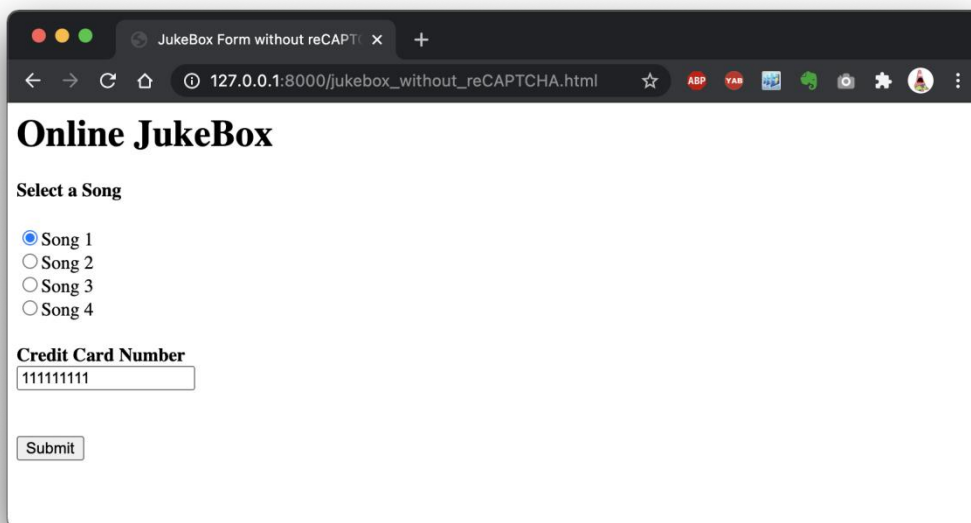
Database changed
mysql> select * from transaction;
Empty set (0.00 sec)

mysql> |
```

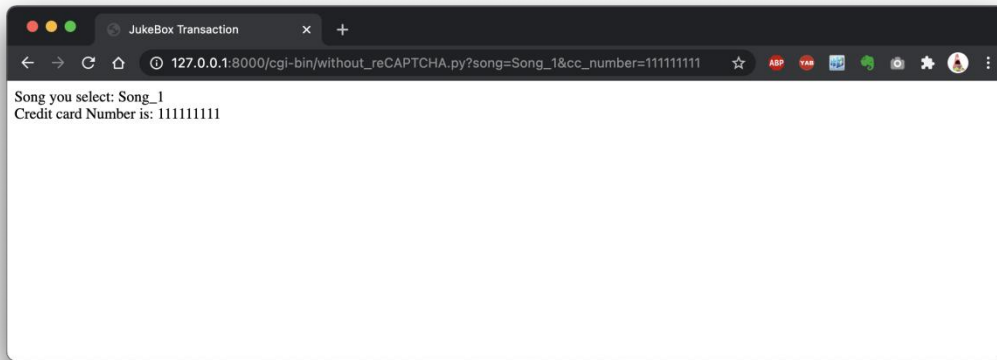
8. Then go to your browser to the URL "127.0.0.1:8000", you will see this page



9. Click `jukebox_without_reCAPTCHA.html` file, you will see the jukebox form.



10. After you select song number, type your credit card number and press submit, it will display your information like this.



11. Now go to your database, your information stored successfully.

```
~/Desktop — mysql -u root -p — 80x26

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

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> use jukebox
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
[mysql> select * from transaction;
Empty set (0.00 sec)

[mysql> mysql> select * from transaction;
+-----+-----+
| song  | cc_number |
+-----+-----+
| Song_1 | 111111111 |
+-----+-----+
1 row in set (0.00 sec)

mysql> |
```

12. Run “bash dos_without_reCAPTCHA.sh” to conduct a DOS Attack.

```
~/Desktop/html -- zsh -- 134x49
[1:05:12] ~/Desktop >> cd html
[1:05:15] ~/Desktop/html >> bash dos_without_reCAPTCHA.sh
Song_1 111111111
--2020-11-17 01:05:36-- http://127.0.0.1:8000/cgi-bin/without_reCAPTCHA.py?song=Song_1&cc_number=111111111
Connecting to 127.0.0.1:8000... connected.
HTTP request sent, awaiting response... 200 Script output follows
Length: unspecified [text/html]
Saving to: 'without_reCAPTCHA.py?song=Song_1&cc_number=111111111'

without_reCAPTCHA.p  [ <=>          ] 143 --KB/s in 0s

2020-11-17 01:05:37 (19.5 MB/s) - 'without_reCAPTCHA.py?song=Song_1&cc_number=111111111' saved [143]

Song_2 222222222
--2020-11-17 01:05:45-- http://127.0.0.1:8000/cgi-bin/without_reCAPTCHA.py?song=Song_2&cc_number=222222222
Connecting to 127.0.0.1:8000... connected.
HTTP request sent, awaiting response... 200 Script output follows
Length: unspecified [text/html]
Saving to: 'without_reCAPTCHA.py?song=Song_2&cc_number=222222222'

without_reCAPTCHA.p  [ <=>          ] 143 --KB/s in 0s

2020-11-17 01:05:45 (19.5 MB/s) - 'without_reCAPTCHA.py?song=Song_2&cc_number=222222222' saved [143]

Song_3 333333333
--2020-11-17 01:05:50-- http://127.0.0.1:8000/cgi-bin/without_reCAPTCHA.py?song=Song_3&cc_number=333333333
Connecting to 127.0.0.1:8000... connected.
HTTP request sent, awaiting response... 200 Script output follows
Length: unspecified [text/html]
Saving to: 'without_reCAPTCHA.py?song=Song_3&cc_number=333333333'

without_reCAPTCHA.p  [ <=>          ] 143 --KB/s in 0s

2020-11-17 01:05:50 (19.5 MB/s) - 'without_reCAPTCHA.py?song=Song_3&cc_number=333333333' saved [143]

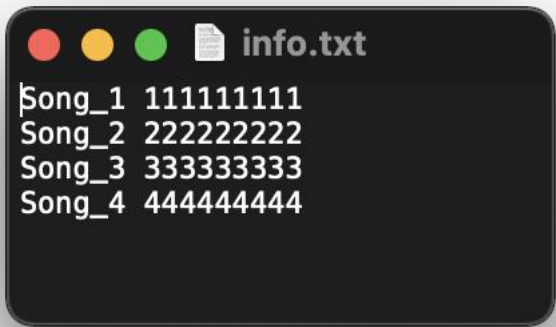
Song_4 444444444
--2020-11-17 01:05:53-- http://127.0.0.1:8000/cgi-bin/without_reCAPTCHA.py?song=Song_4&cc_number=444444444
Connecting to 127.0.0.1:8000... connected.
HTTP request sent, awaiting response... 200 Script output follows
Length: unspecified [text/html]
Saving to: 'without_reCAPTCHA.py?song=Song_4&cc_number=444444444'

without_reCAPTCHA.p  [ <=>          ] 143 --KB/s in 0s

2020-11-17 01:05:53 (19.5 MB/s) - 'without_reCAPTCHA.py?song=Song_4&cc_number=444444444' saved [143]

[1:07:36] ~/Desktop/html >> |
```

13. Shell script automatically read song number and credit card number from info.txt and fill this jukebox form.



It's clearly to see it's easy to be DOS attacked for some simple web forms.

```
~/Desktop — mysql -u root -p — 80x26

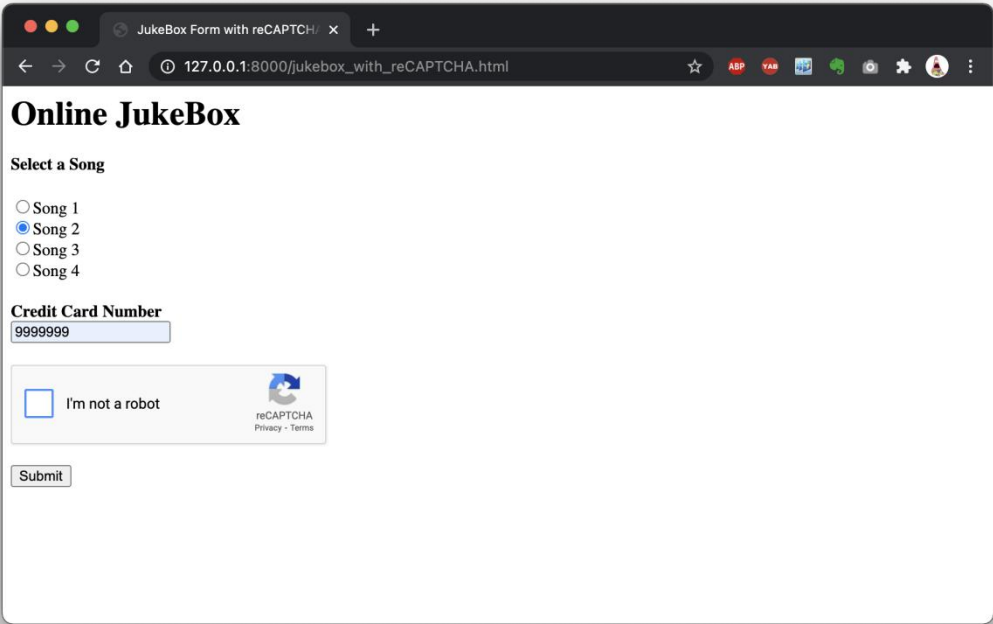
Database changed
[mysql> select * from transaction;
Empty set (0.00 sec)

[mysql> mysql> select * from transaction;
+-----+-----+
| song  | cc_number |
+-----+-----+
| Song_1 | 111111111 |
+-----+-----+
1 row in set (0.00 sec)

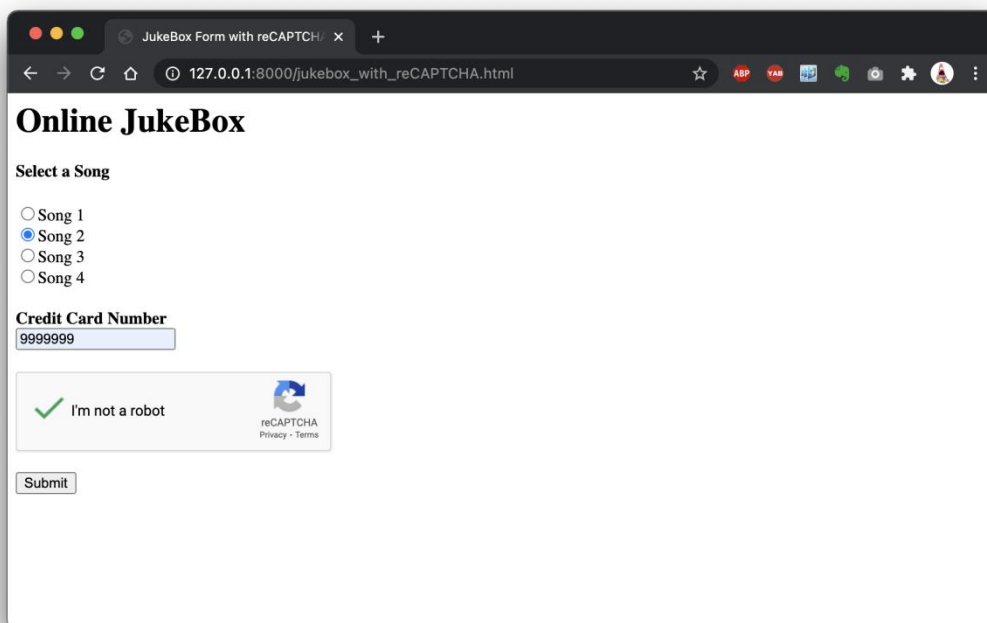
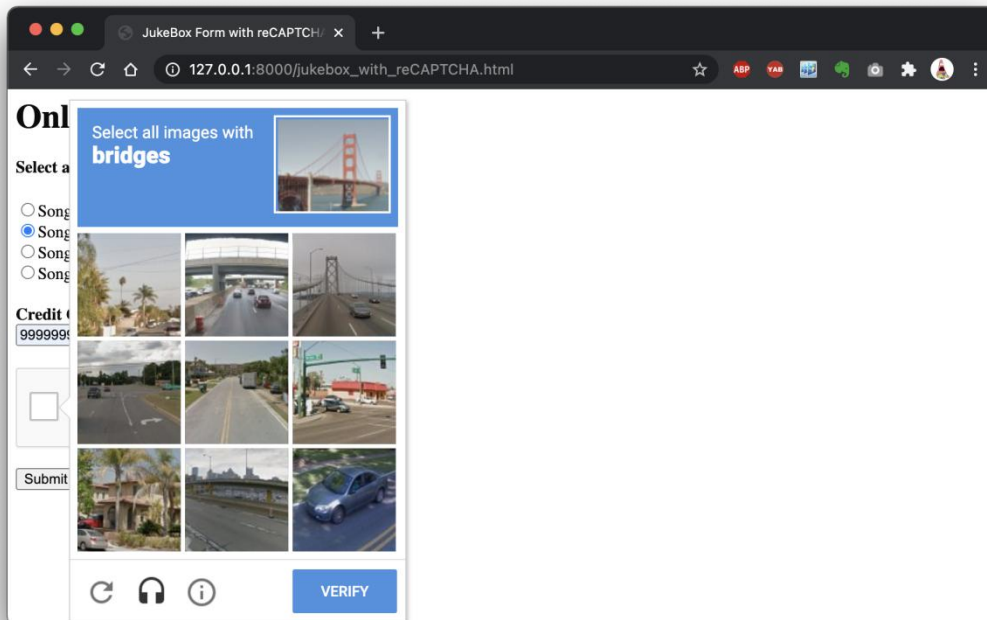
[mysql> select * from transaction;
+-----+-----+
| song  | cc_number |
+-----+-----+
| Song_1 | 111111111 |
| Song_1 | 111111111 |
| Song_2 | 222222222 |
| Song_3 | 333333333 |
| Song_4 | 444444444 |
+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

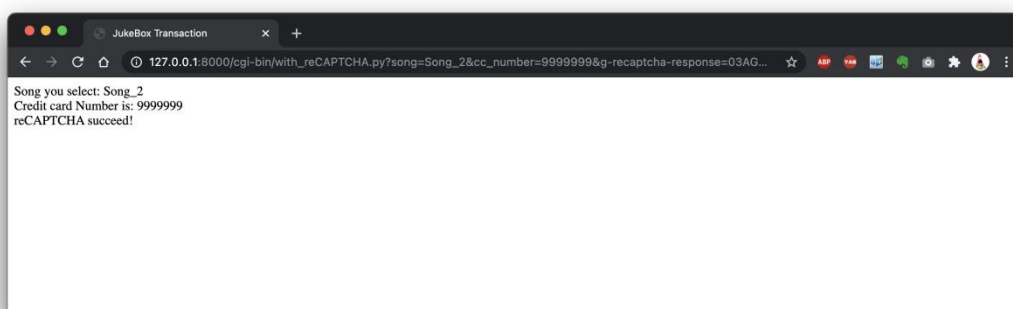
14. Let's add reCAPTCHA to prevent from DOS Attack:



15. You have to pass the reCAPTCHA verification to make sure you are not a robot.



16. It tells you reCAPTCHA succeed. Now your form has been stored to database.




```
~/Desktop — mysql -u root -p — 80x26

[mysql> select * from transaction;

+-----+-----+
| song | cc_number |
+-----+-----+
| Song_1 | 111111111 |
| Song_1 | 111111111 |
| Song_2 | 222222222 |
| Song_3 | 333333333 |
| Song_4 | 444444444 |
+-----+-----+
5 rows in set (0.00 sec)

[mysql> select * from transaction;

+-----+-----+
| song | cc_number |
+-----+-----+
| Song_1 | 111111111 |
| Song_1 | 111111111 |
| Song_2 | 222222222 |
| Song_3 | 333333333 |
| Song_4 | 444444444 |
| Song_2 | 9999999 |
+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

17. Now, let's try DOS attack one more time.

```
~/Desktop/html — -zsh — 130x50

[1:11:39] ~/Desktop >> cd html
[1:11:45] ~/Desktop/html >> bash dos_with_reCAPTCHA.sh
Song_1 111111111
--2020-11-17 01:12:08-- http://127.0.0.1:8000/cgi-bin/with_reCAPTCHA.py?song=Song_1&cc_number=111111111
Connecting to 127.0.0.1:8000... connected.
HTTP request sent, awaiting response... 200 Script output follows
Length: unspecified [text/html]
Saving to: 'with_reCAPTCHA.py?song=Song_1&cc_number=111111111'

with_reCAPTCHA.py?song=Song_1& [ <=> ] 161 --KB/s in 0s

2020-11-17 01:12:09 (21.9 MB/s) - 'with_reCAPTCHA.py?song=Song_1&cc_number=111111111' saved [161]

Song_2 222222222
--2020-11-17 01:12:28-- http://127.0.0.1:8000/cgi-bin/with_reCAPTCHA.py?song=Song_2&cc_number=222222222
Connecting to 127.0.0.1:8000... connected.
HTTP request sent, awaiting response... 200 Script output follows
Length: unspecified [text/html]
Saving to: 'with_reCAPTCHA.py?song=Song_2&cc_number=222222222'

with_reCAPTCHA.py?song=Song_2& [ <=> ] 161 --KB/s in 0s

2020-11-17 01:12:29 (21.9 MB/s) - 'with_reCAPTCHA.py?song=Song_2&cc_number=222222222' saved [161]

Song_3 333333333
--2020-11-17 01:12:31-- http://127.0.0.1:8000/cgi-bin/with_reCAPTCHA.py?song=Song_3&cc_number=333333333
Connecting to 127.0.0.1:8000... connected.
HTTP request sent, awaiting response... 200 Script output follows
Length: unspecified [text/html]
Saving to: 'with_reCAPTCHA.py?song=Song_3&cc_number=333333333'

with_reCAPTCHA.py?song=Song_3& [ <=> ] 161 --KB/s in 0s

2020-11-17 01:12:32 (19.2 MB/s) - 'with_reCAPTCHA.py?song=Song_3&cc_number=333333333' saved [161]

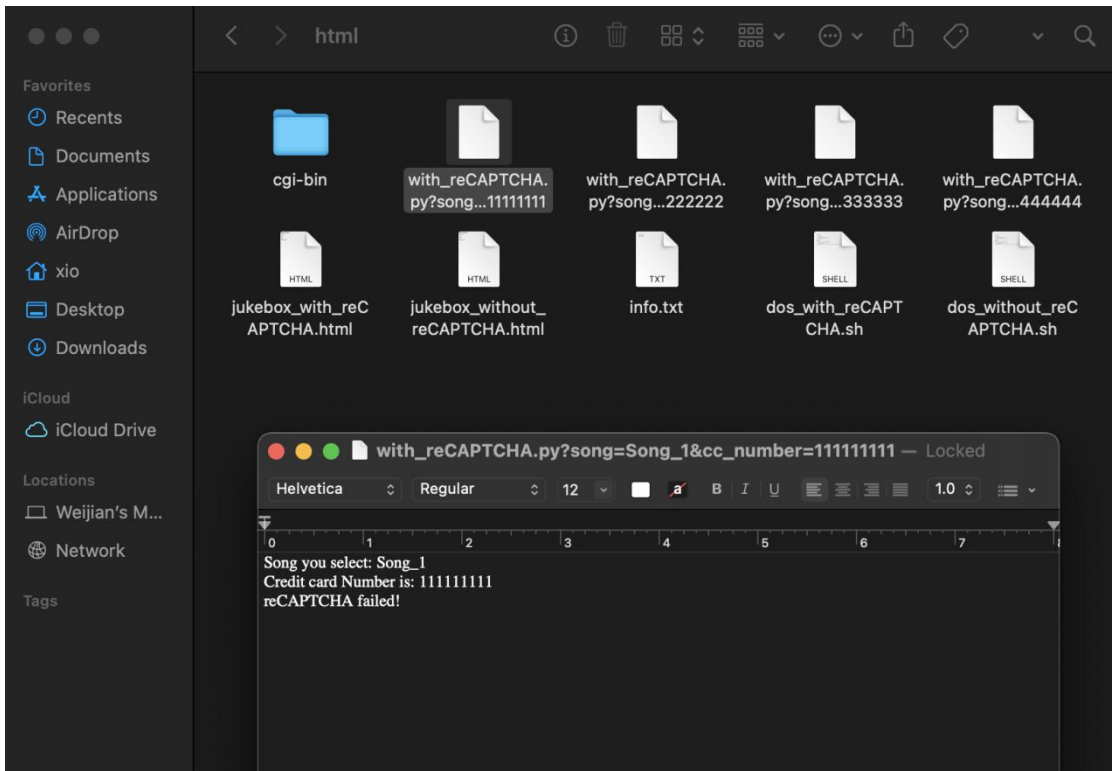
Song_4 444444444
--2020-11-17 01:12:48-- http://127.0.0.1:8000/cgi-bin/with_reCAPTCHA.py?song=Song_4&cc_number=444444444
Connecting to 127.0.0.1:8000... connected.
HTTP request sent, awaiting response... 200 Script output follows
Length: unspecified [text/html]
Saving to: 'with_reCAPTCHA.py?song=Song_4&cc_number=444444444'

with_reCAPTCHA.py?song=Song_4&cc [ <=> ] 161 --KB/s in 0s

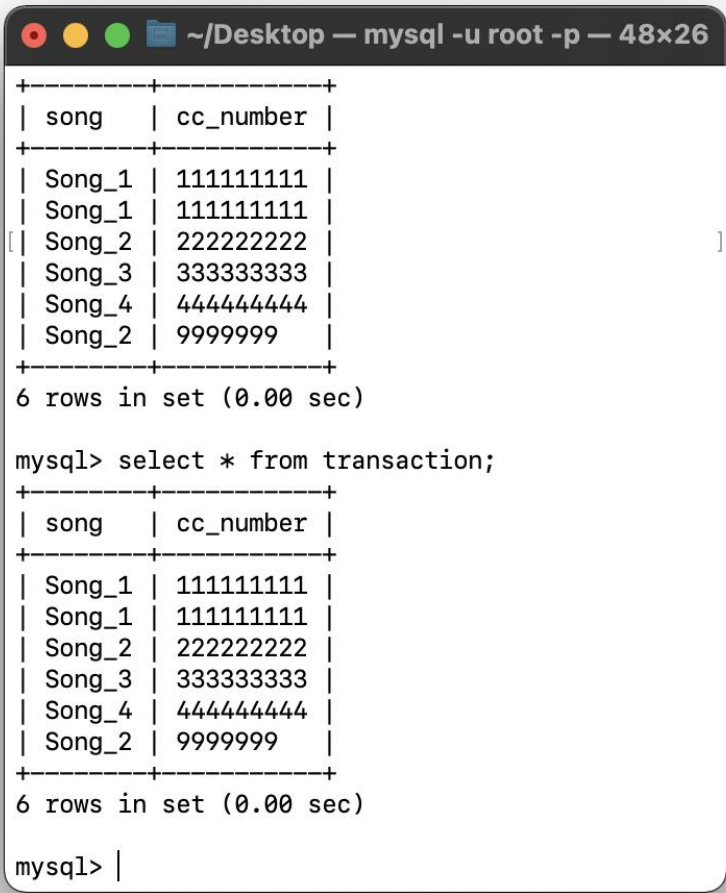
2020-11-17 01:12:48 (19.2 MB/s) - 'with_reCAPTCHA.py?song=Song_4&cc_number=444444444' saved [161]

[1:13:00] ~/Desktop/html >>
```

18. It tells you reCAPTCHA failed.



19. And database didn't allow any new data modification from DOS attack



20. reCAPTCHA prevents you from DOS attack !!!

Register reCAPTCHA:

<https://www.google.com/recaptcha/admin/create>

If you use local server, add “127.0.0.1” and “localhost” to your domains.