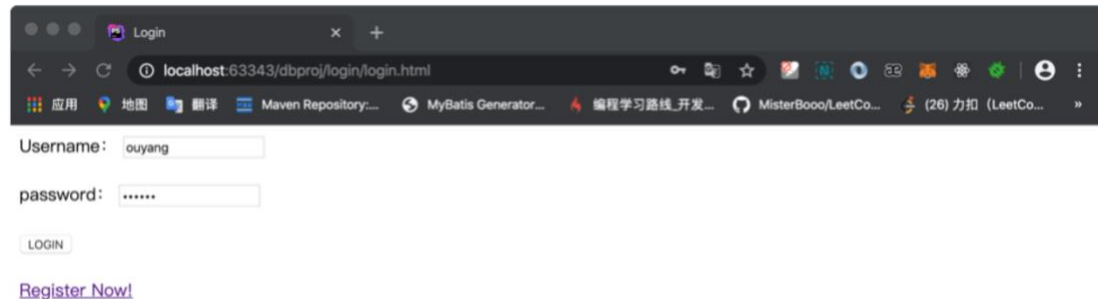


All .html and .php are in scripts folder.

1. Login and create User (login.php, create_user.php)

login.php

screenshot :



We can input the username and password to login in to the system. And if you do not have the account. You can "Register Now!" to create a new one.

In login.php I use

```
$stmt = $conn->prepare( query: "select * from user where username=? and password=?");  
$stmt->bind_param( types: 'ss', &var1: $name, &var2: $password1);  
$stmt->execute();  
$result = $stmt->get_result();  
if ($result->num_rows > 0)//判断密码与注册时密码是否一致
```

this way to SQL injection attacks:

By this way, I limit the degree of freedom of the query statement: the parameter is only the parameter, nobody can add a semicolon into a command. So I can defense the SQL injection attacks,

However, if I do in this way:

```
/* $str = "select * from user where username='$name' and password='$password1'";  
$result = $conn->query($str);*/
```

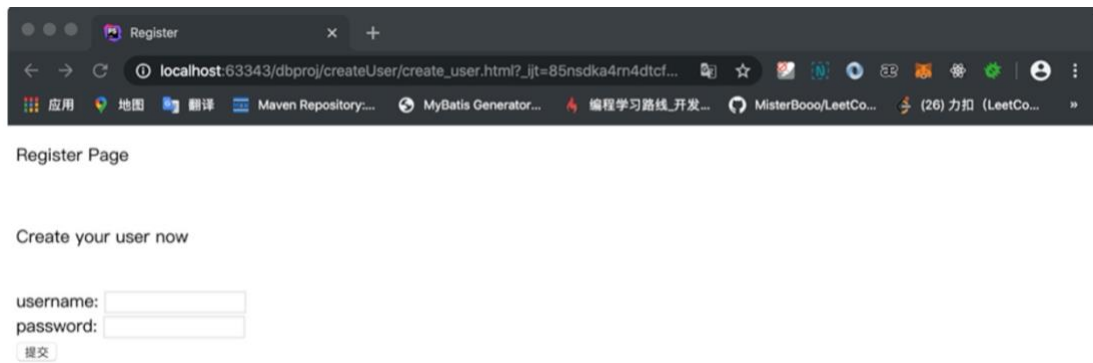
if the attacker send this attack"; Drop table table_name; "" password;

The SQL will just like "select * from user where username= attack"; Drop table table_name;

" It will drop your table. This is unsafe.

create_user.php:

screenshot :



If I want to make sure every username is unique. I can do I by this way:

```
//username should be unique
$stmt = $conn->prepare( query: "select username from user where username = ?");
$stmt->bind_param( types: 's', &var1: $name);
$stmt->execute();
$result = $stmt->get_result();
```

Here, I also use the same way to defend the SQL injection. The principle is the same as above.

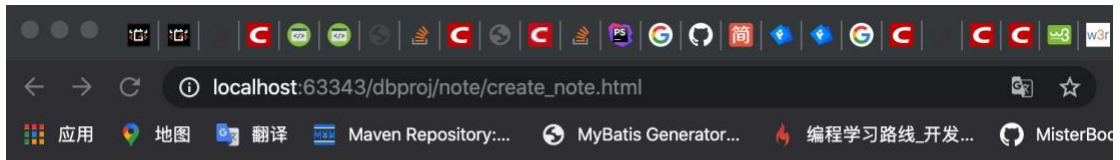
```
//username should be unique
$stmt = $conn->prepare( query: "select username from user where username = ?");
$stmt->bind_param( types: 's', &var1: $name);
$stmt->execute();
$result = $stmt->get_result();
```

However, if I do the register SQL by this way, I will not able to defend the SQL injection:

```
$sql="insert into user (username,password) VALUES ('$name','$password')";
if ($conn->query($sql) === TRUE) {
    //echo "New record created successfully";
    echo"<script type=\"\".\"text/javascript\".\"\".\">\".\"window.location=\".\"\".\"reg
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}
```

If I want to make sure there is no SQL injection. I can do this by the way as above. So, as fellow I will not talk about the SQL injection.

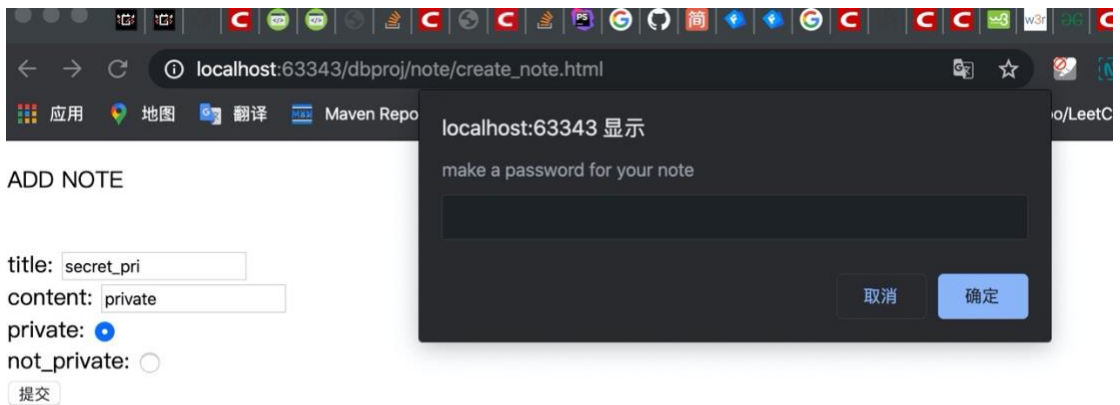
2. Notes creation (create_note.php)



ADD NOTE

title:
content:
private: ☐
not_private: ☐

Here I can create a new note. And if I want to make it private. I can do like this



So, I can add a password to encrypt this content. After I make a note. I will go back to main.php.

By the way, I use the password for note is blank or not to make sure the is private not or not.

```
if($title=="||$content==""){  
    echo"<script type=\"\".\"text/javascript\".\"\".\">\"  
        .\"window.alert\".\"(\".\"\".\"NOT BLANK IN TITLE OR CONTENT! \".\"\".\").\";\".\"</scrip  
    echo"<script type=\"\".\"text/javascript\".\"\".\">\".\"window.location=\".\"\".\"create  
    exit;  
}  
/* if($private note=="")f
```

By this way I make sure the title and content is not blank.

```

        exit;
    }

}*/
//username should be unique
$sql_unique_title = "select title from note where title =
                    '$title' and username = '$username_this'";
$result_title = $conn->query($sql_unique_title);

if ($result_title->num_rows>0){

    echo"<script type=\"\".\"text/javascript\".\"\".\">\"
        .\"window.alert\".\"(\".\"\".\"you have the same title!\".\"\".\").\";\".\"</script>\";

```

Here, I can make sure you can only see your note.

The username, I use session like this:

```

}

session_start(); //开启session
$username_this = $_SESSION['username'];
$title_id = $_GET['id'];

```

```

if($private_note == 1){
    $sql_create_note_private="insert into note (username,title,content,private_note) VALUES
                            ('$username_this', '$title', AES_ENCRYPT('$content', '$note_pass'), '$private_note'
    if ($conn->query($sql_create_note_private) === TRUE) {
        //写入成功
        echo"<script type=\"\".\"text/javascript\".\"\".\">\"
            .\"window.alert\".\"(\".\"\".\"write note success!\".\"\".\").\";\".\"</script>\";

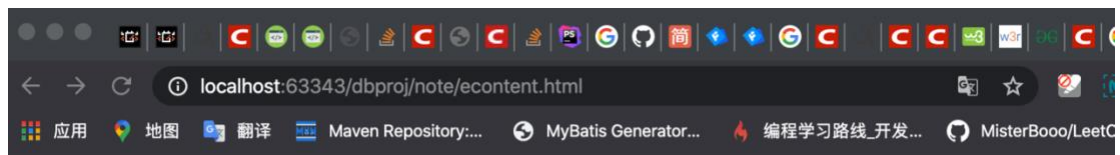
```

Here, I use `AES_ENCRYPT('$content', '$note_pass')` to encrypt the content.

3. View note(view.php)

After I click the title in main.php. I can view the note:

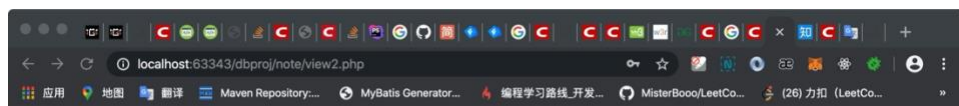
If the note is encrypted, I need a password for the content(the password is not login password). Like this:



please input your content password:

GO!

After enter password. I can see the result. Like this



you secret note is here:

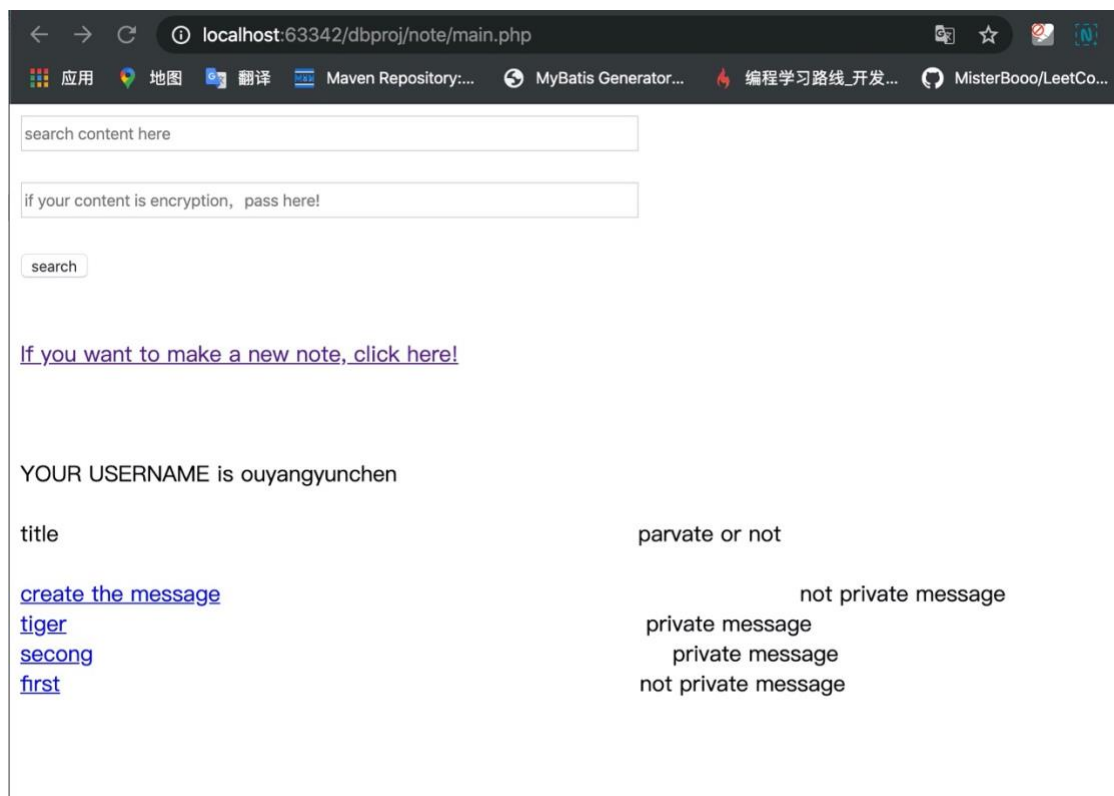
secretnsg

[Click return main page](#)

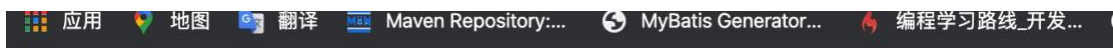
If the note is not encryption, just click , I can see the result:



4. Notes browsing and searching (main.php)



On the top, I can search. Just like, if I want to search the content like "this is a message". This message is not private. You do not have to input all message. Just like "this is a" will work.



this is a

if your content is encryption, pass here!

search

[If you want to make a new note, click here!](#)

YOUR USERNAME is ouyangyunchen

title

parvate or not

[create the message](#)

[tiger](#)

[secong](#)

[first](#)

not private r

private message

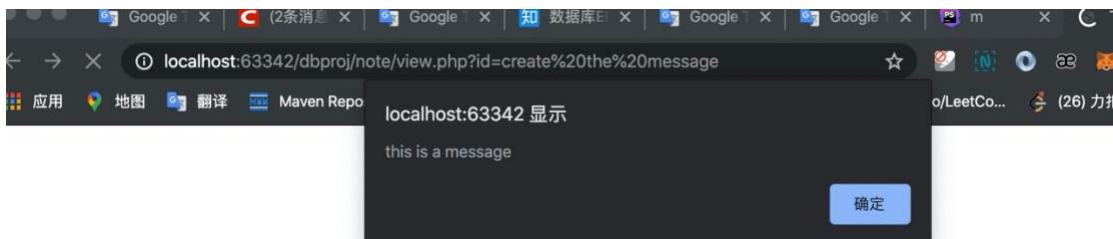
private message

not private message

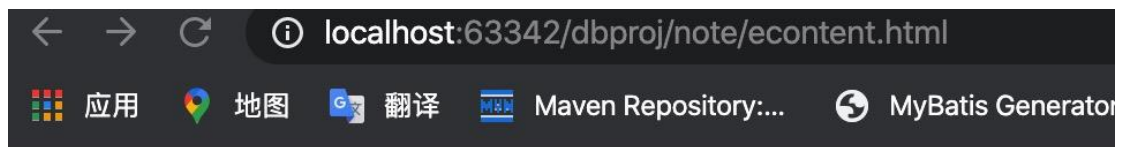
search result:

[create the message](#)

The result is at the bottom. Click it, you will get the message:



If it is a private message. You need enter a password for the content:



please input your content password:

GO!

You can see the result by the password:



you secret note is here:

tiger and pig

[Click return main page](#)

localhost:63343/dbproj/note/main.php

search content here

if your content is encryption, pass here!

search

[If you want to make a new note, click here!](#)

YOUR USERNAME is ouyang

title

parvate or not

[third_msg](#) not private message
[second_msg](#) private message
[1111](#) private message

search result:
[third_msg](#)

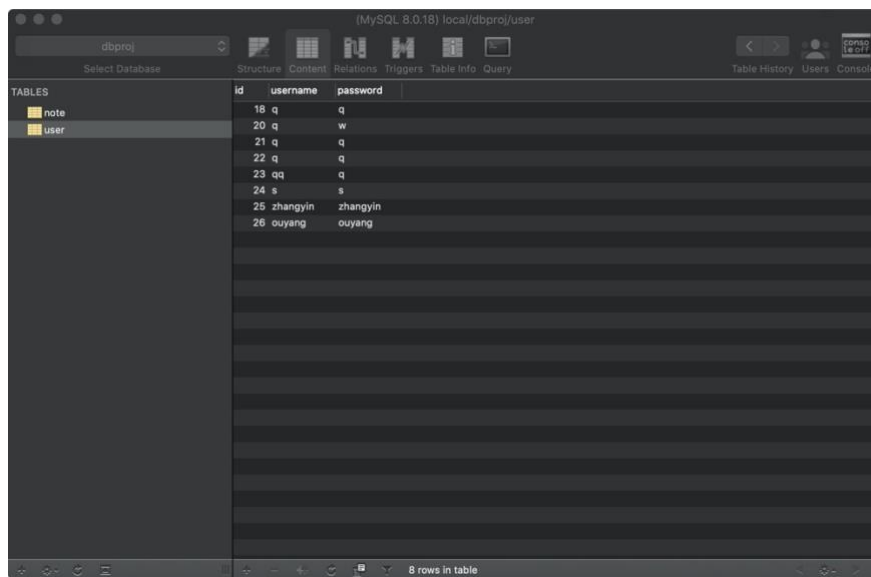
And I also can go to create a new note by click the link.

Database:

```
Create syntax:
CREATE TABLE `note` (
  `note_ID` int(10) NOT NULL AUTO_INCREMENT,
  `username` varchar(10) NOT NULL DEFAULT '',
  `title` varchar(300) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL DEFAULT '',
  `content` varbinary(400) NOT NULL DEFAULT '',
  `private_note` int(2) NOT NULL,
  `create_time` datetime DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`note_ID`)
) ENGINE=InnoDB AUTO_INCREMENT=28 DEFAULT CHARSET=utf8;
```

note_ID	username	title	content	private_note	create_time
1	q	111	11	1	2020-04-19 18:48:33
2	q	222	222	1	2020-04-19 18:50:56
3	q	1111	1111	0	2020-04-19 19:17:01
4	q	111111	111111	0	2020-04-19 19:17:47
5	q	qwe	qwe	0	2020-04-19 19:19:46
6	q	asd	asd	0	2020-04-19 19:22:49
7	q	zxc	zxc	0	2020-04-19 19:24:54
8	q	1234	1234	0	2020-04-19 19:25:56
9	q	ert	ert	0	2020-04-19 19:29:35
10	q	tyu	tyu	0	2020-04-19 19:30:24
11	q	rty	rty	0	2020-04-19 19:39:25
12	q	111111222	88Meisapup#icx@a	1	2020-03-19 19:59:04
13	q	asdlg	dadadada	1	2020-04-20 12:08:37
15	q	ouyang	ds-qbJieIO	1	2020-04-20 12:33:46
16	q	test	7wcc<5bNv7A*	1	2020-04-20 12:42:01
17	q	waneed	gk*8(2A-S8uA	1	2020-04-20 12:54:05
18	q	testtest	,80*7aRnc,*98	1	2020-04-20 13:05:52
19	zhangyin	test	test	0	2020-04-20 18:16:27
20	zhangyin	test2	test2	0	2020-04-20 18:17:36
21	zhangyin	test3	test3	0	2020-04-20 18:18:40
22	zhangyin	test4	AAcO8U=flvAlAR	1	2020-04-20 18:19:07
23	ouyang	1111	1/87-jNhec*lj	1	2020-04-20 20:42:03
24	ouyang	second_msg	@UEAAktDUx08*B80k*BH	1	2020-04-20 20:42:46
25	ouyang	third_msg	third_msg_secret	0	2020-04-20 20:43:22
26	ouyang	12345	<r0f5JjJ V194e8	1	2020-04-20 20:53:31

```
CREATE TABLE `user` (  
  `id` int(10) NOT NULL AUTO_INCREMENT,  
  `username` varchar(30) DEFAULT NULL,  
  `password` varchar(40) DEFAULT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=27 DEFAULT CHARSET=utf8;
```



The screenshot shows a MySQL database management tool interface. The title bar indicates the connection is to 'localhost/dbproj/user'. The left sidebar shows a tree view with 'dbproj' selected, containing 'note' and 'user' tables. The main area displays the 'user' table structure and data. The table has three columns: 'id', 'username', and 'password'. The data is as follows:

id	username	password
18	q	q
20	q	w
21	q	q
22	q	q
23	qq	q
24	s	s
25	zhangyin	zhangyin
26	ouyang	ouyang

The status bar at the bottom indicates '8 rows in table'.