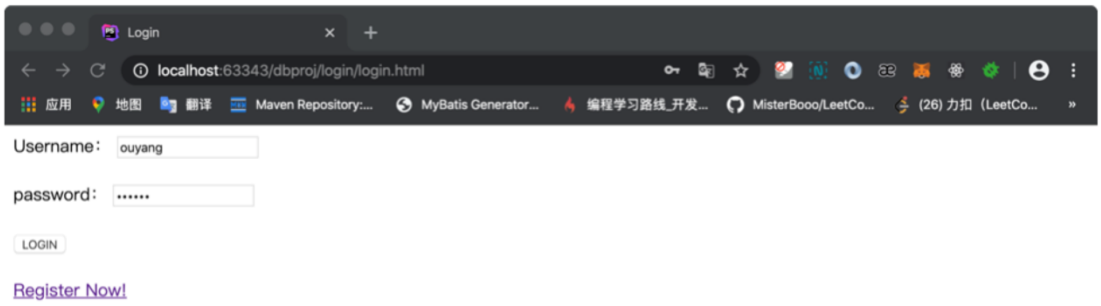
1. Login and create User (login.php, create_user.php)
   login.php
   screenshot ：



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, &...: $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
```

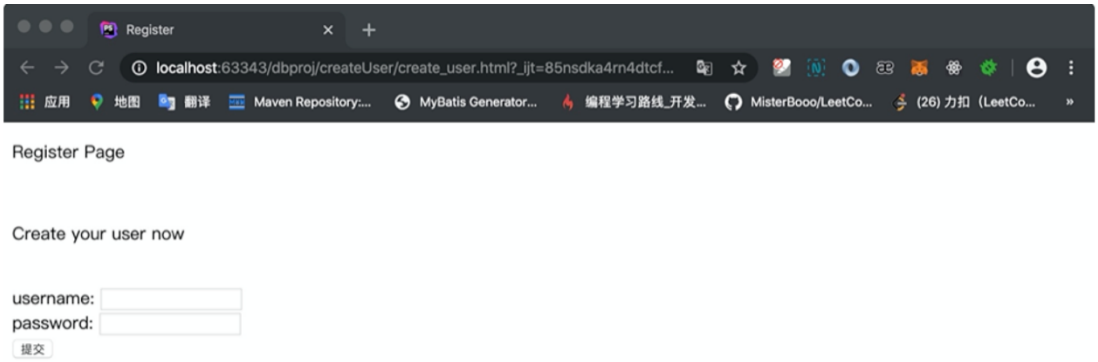
this way to SQL injection attacks:

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.

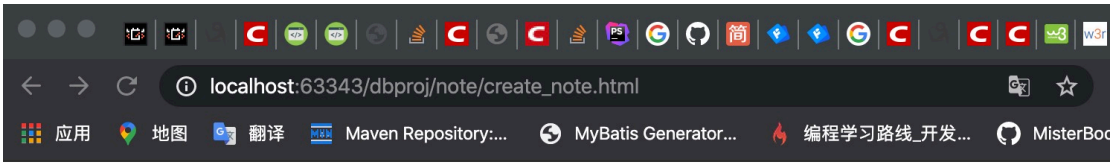
```
//username should be unique
$stmt = $conn->prepare( query: "select username from user where username = ?");
$stmt->bind_param( types: 's', &var1: $name);
$stmt->execute();
$result = $stmt->get_result();
```

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.

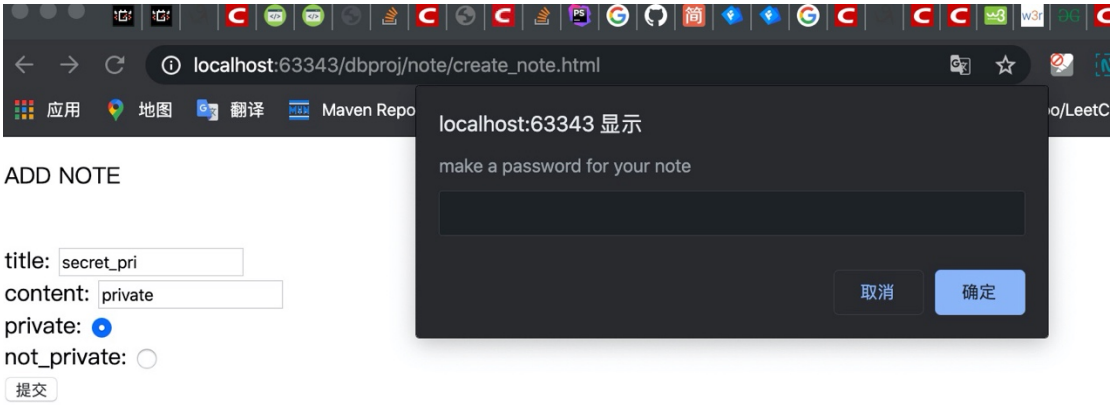
2. Notes creation (create_note.php)



ADD NOTE

title:
content:
private:
not_private:
提交

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==""){
```

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'];
```

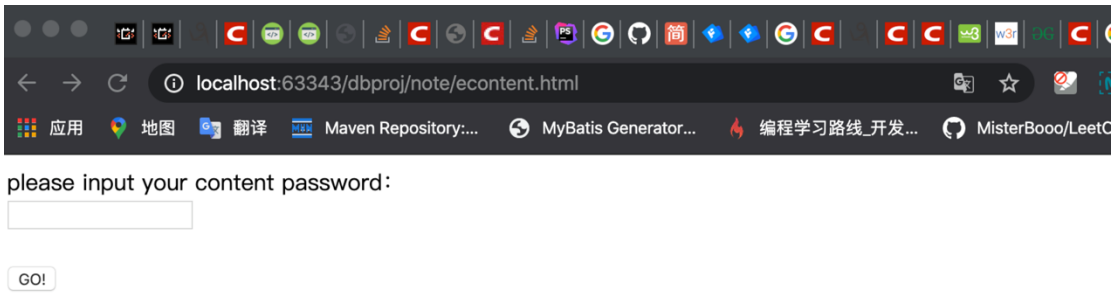
```
($private_note == 1){
  $sql_create_note_private="insert into note (username,title,content,private_note) VALUES
            ('$username_this','$title',AES_ENCRYPT('$content','$note_pass'),'$private_note'
  if ($conn->query($sql_create_note_private) === TRUE) {
      //写入成功
      echo"<script type="."\""."text/javascript"."\"".">"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

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

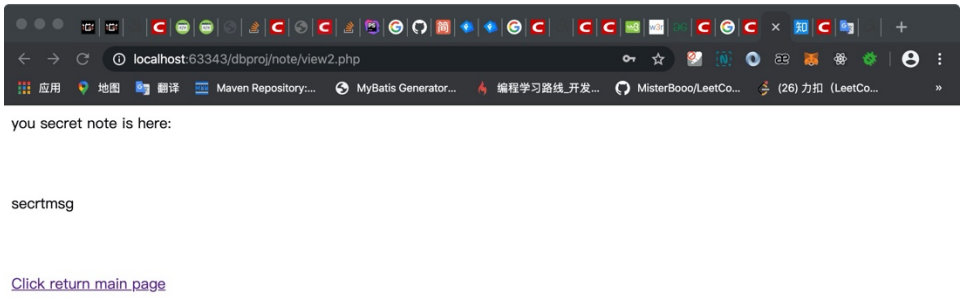
3. View note(view.php)

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:

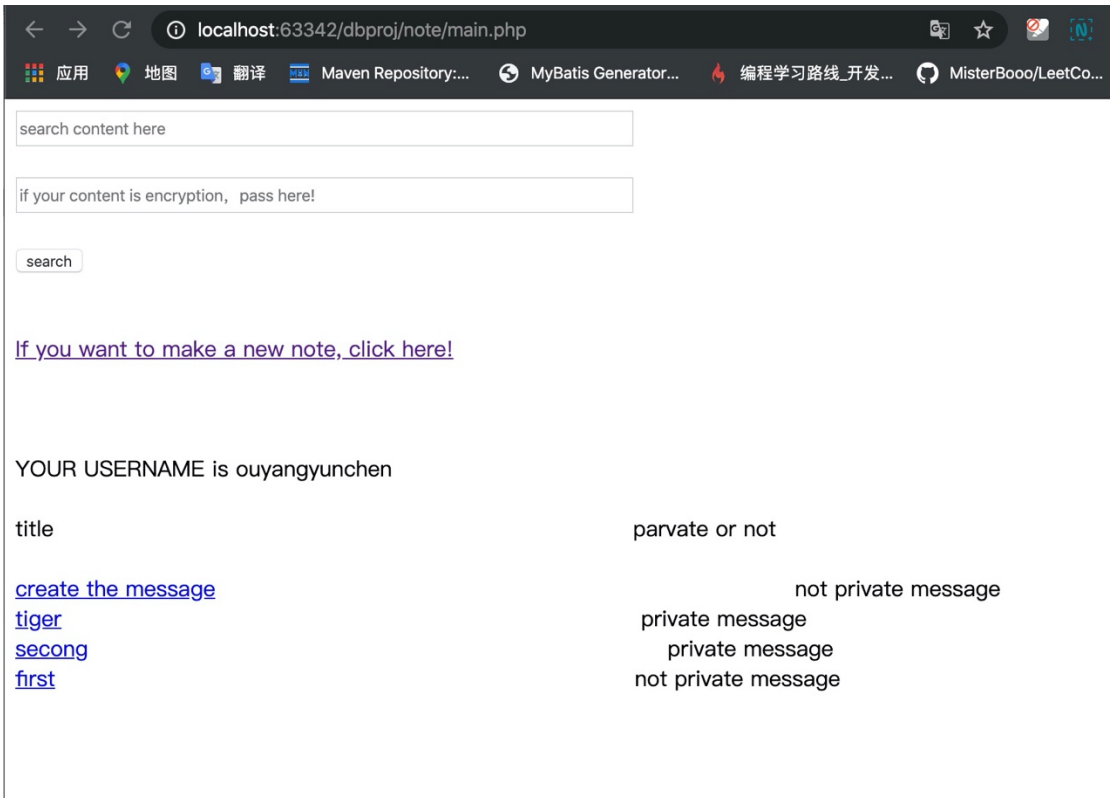


After enter password. I can see the result. Like this



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

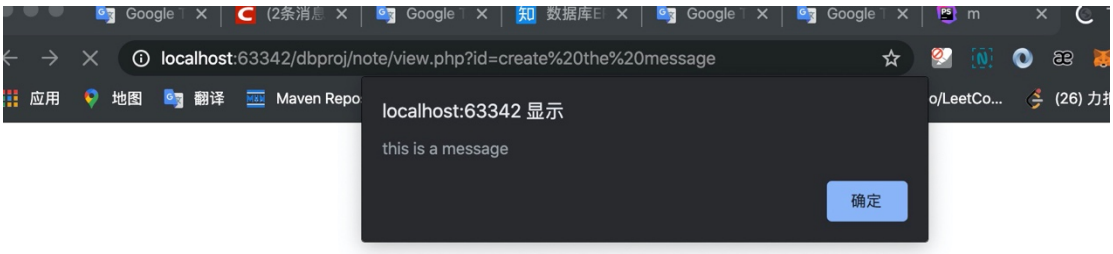
If you want to make a new note, click here!

YOUR USERNAME is ouyangyunchen

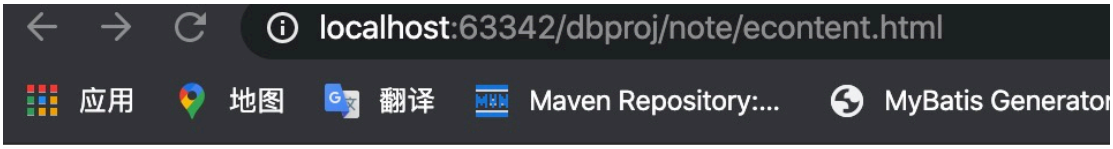
title	parvate or not
create the message	not private r
tiger	private message
secong	private message
first	not private message

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：

•••••

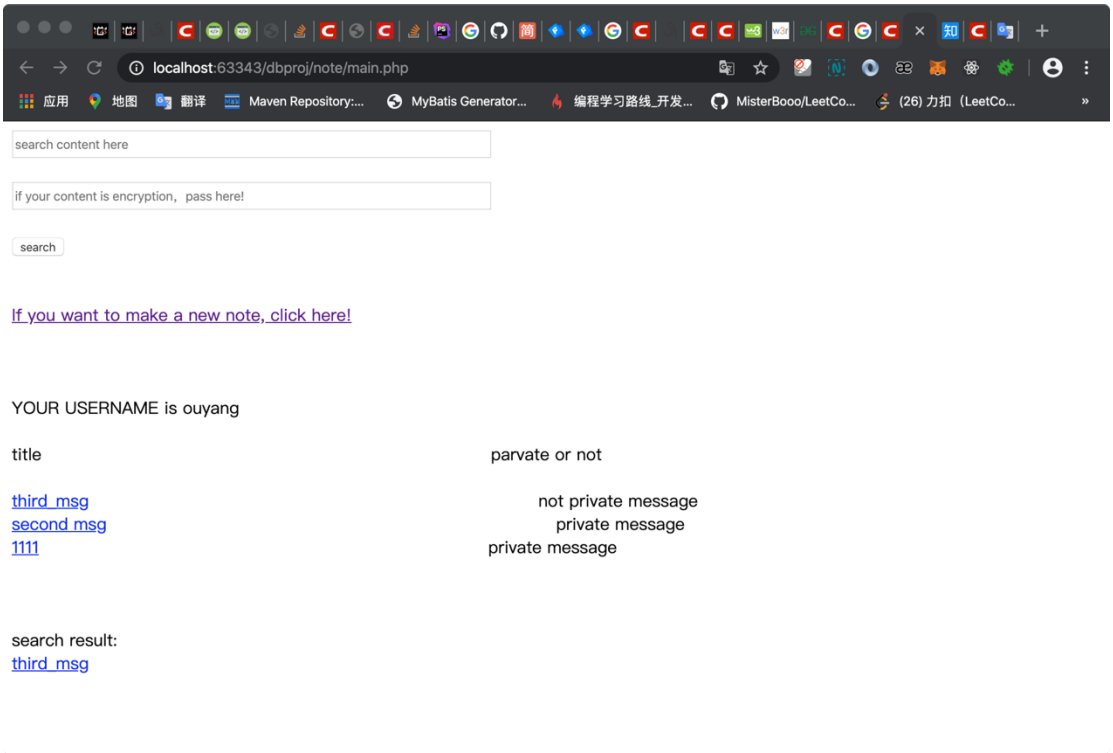
GO!

You can see the result by the password:

you secret note is here:

tiger and pig

Click return main page

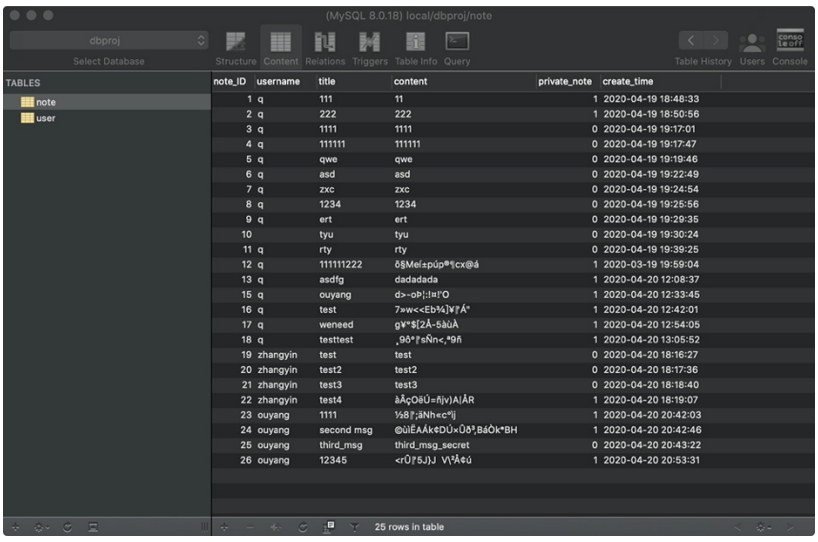


And I also can go to create a new note by click the link.

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;
```



```sql
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;
```