WarmHearted

your ultimate place for public welfare

Gang Jiang, Yu Wang,
Zian Jiang, Hang Xie

Outline

- Introduction
- Front End
- User Interface
- Back End
- Database

Introduction

Social Network

?

Public Welfare

Introduction

We want a platform that can incorporate the public welfare and social network together

Social Network + Public Welfare = Peace & Love

Target User

WKU students who are passionate to public welfare

WKU students who will be passionate to public welfare

and

…

YOU

Your Voice Matters!

Framework - Front End

UI Component Library
Bootstrap v 4.4

User Interface - Register

Create Your Warmhearted Account

Email address

Please enter your email address

Password

Please enter your password

Comfirm password

Please enter your password again

Already has a account? Sign in!

User Interface - Log in

login
Email Address
Please enter your email address
Password
Please enter your password
Forget your password?
No account? Register!
Sign in

User Interface - Main Page #1

HOME
DISCOVER
Search
Notification
History
ME
Recommend
Follow
Hots
write articles
drafts

Recommend Title 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet. Proin gravida dolor sit amet lacus accumsan et viverra justo commodo. Proin sodales pulvinar sic tempor. Sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nam fermentum, nulla luctus pharetra vulputate, felis

User Interface - Main Page #2

Recommend
Follow
Hots
write articles
drafts
my collections
my likes

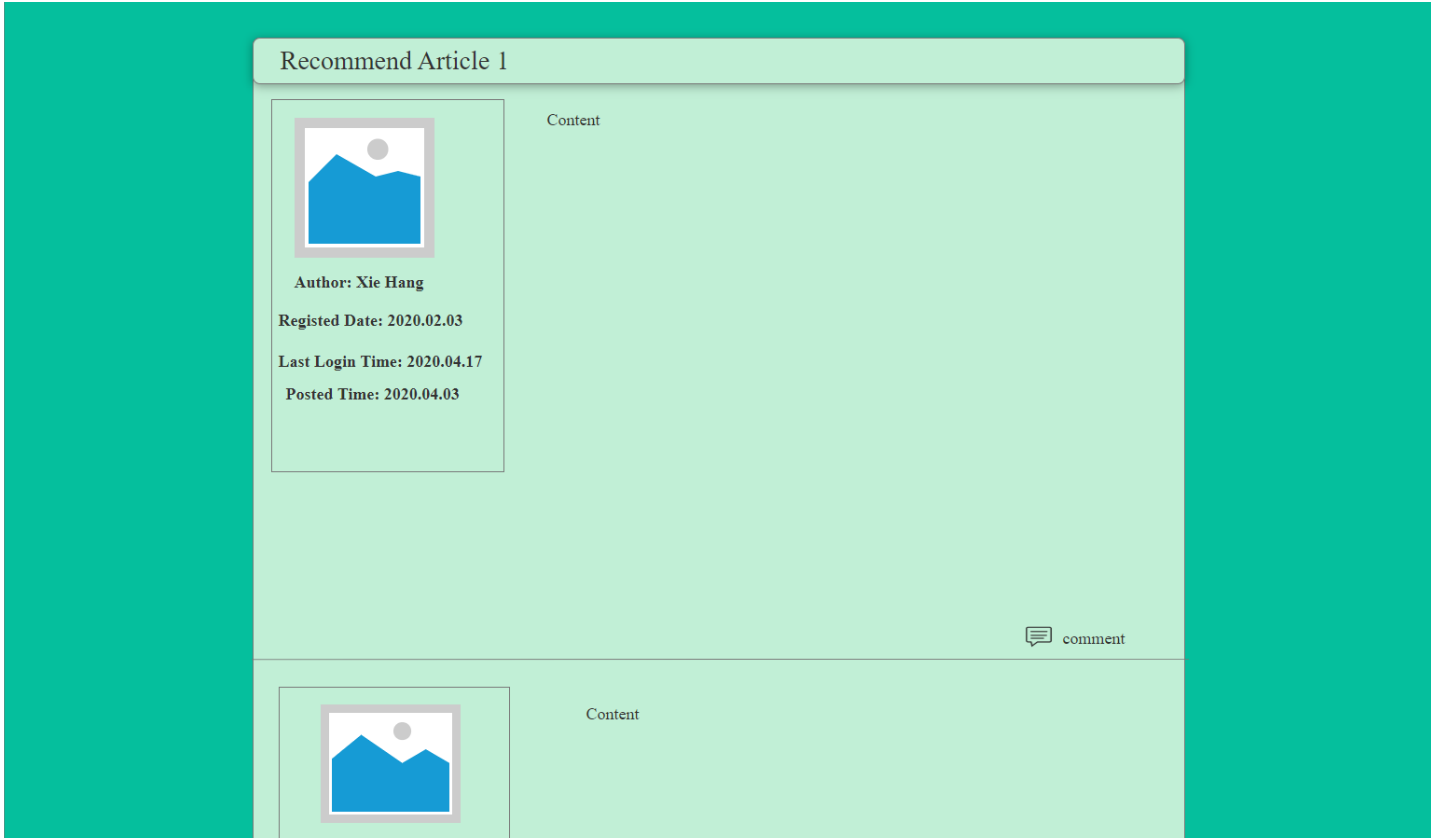
Follow Title 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet. Proin gravida dolor sit amet lacus accumsan et viverra justo commodo. Proin sodales pulvinar sic tempor. Sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nam fermentum, nulla luctus pharetra vulputate, felis tellus mollis orci, sed rhoncus pronin sapien nunc accuan eget.

Follow Title 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet. Proin gravida dolor sit amet lacus accumsan et viverra justo commodo. Proin sodales pulvinar sic tempor. Sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nam fermentum, nulla luctus pharetra vulputate, felis tellus mollis orci, sed rhoncus pronin sapien nunc accuan eget.

User Interface - Article Page



Recommend Article 1

Author: Xie Hang

Registed Date: 2020.02.03

Last Login Time: 2020.04.17

Posted Time: 2020.04.03

Content

comment

Content

User Interface - Personal Profile Page #1

HOME
DISCOVER
Search
Notification
History
ME
NickName
Personal Information
Account Management
Notification
Settings
Profile Photo

Email Address: 123@163.com

NickName: Beautiful Girl

User Interface - Personal Profile Page #2

HOME
DISCOVER
Search
Notification
History
ME
NickName
Personal Information
Account Management
Notification
Settings
Account Management

WarmHearted原型 - Axure RP 9 Enterprise Edition : ABC 已授权

Framework - Back End

PHP is the best programming language in the world!

Python, byte me :)

Back End - Main Functions

1. Login function
2. Registration function
3. Delete/Modify/Upload post
4. Delete/Modify/Upload reply
5. Modify personal Information

Login Function

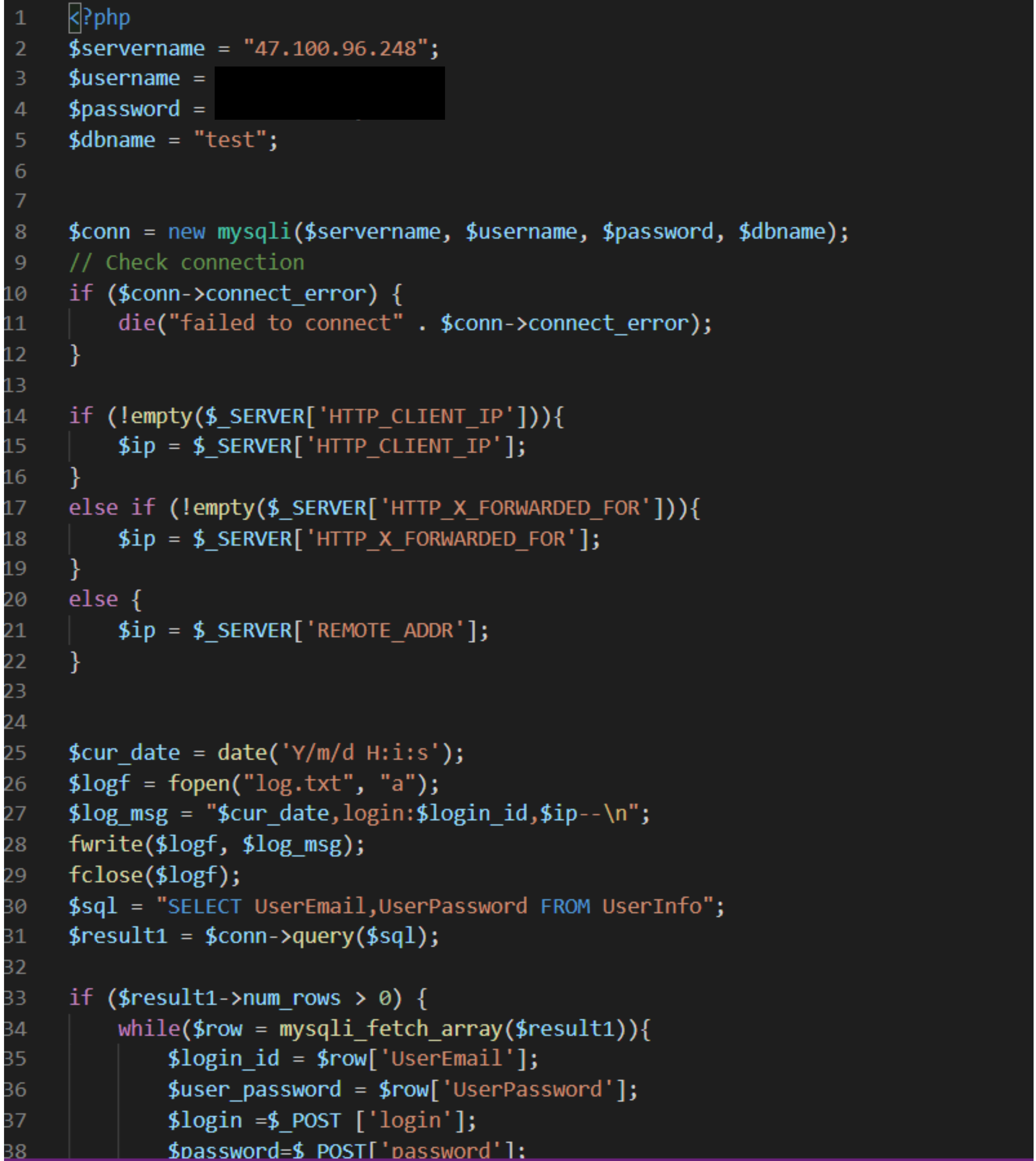
We mainly implement our login function by four steps:

1. Build the connection between back end and database

2. Receive the arguments which passed from front end and websites.

3. Search the database for these parameters

4. If these arguments are found, log in successfully. On the contrary, if there is no account in the database or your password is wrong, the corresponding error data will be returned



```php
<?php
$servername = "47.100.96.248";
$username = [redacted]
$password =
$dbname = "test";


$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("failed to connect" . $conn->connect_error);
}

if (!empty($_SERVER['HTTP_CLIENT_IP'])){
    $ip = $_SERVER['HTTP_CLIENT_IP'];
}
else if (!empty($_SERVER['HTTP_X_FORWARDED_FOR'])){
    $ip = $_SERVER['HTTP_X_FORWARDED_FOR'];
}
else {
    $ip = $_SERVER['REMOTE_ADDR'];
}


$cur_date = date('Y/m/d H:i:s');
$logf = fopen("log.txt", "a");
$log_msg = "$cur_date,login:$login_id,$ip--\n";
fwrite($logf, $log_msg);
fclose($logf);
$sql = "SELECT UserEmail,UserPassword FROM UserInfo";
$result1 = $conn->query($sql);

if ($result1->num_rows > 0) {
    while($row = mysqli_fetch_array($result1)){
        $login_id = $row['UserEmail'];
        $user_password = $row['UserPassword'];
        $login =$_POST ['login'];
        $password=$_POST['password'];
```

Registration Function

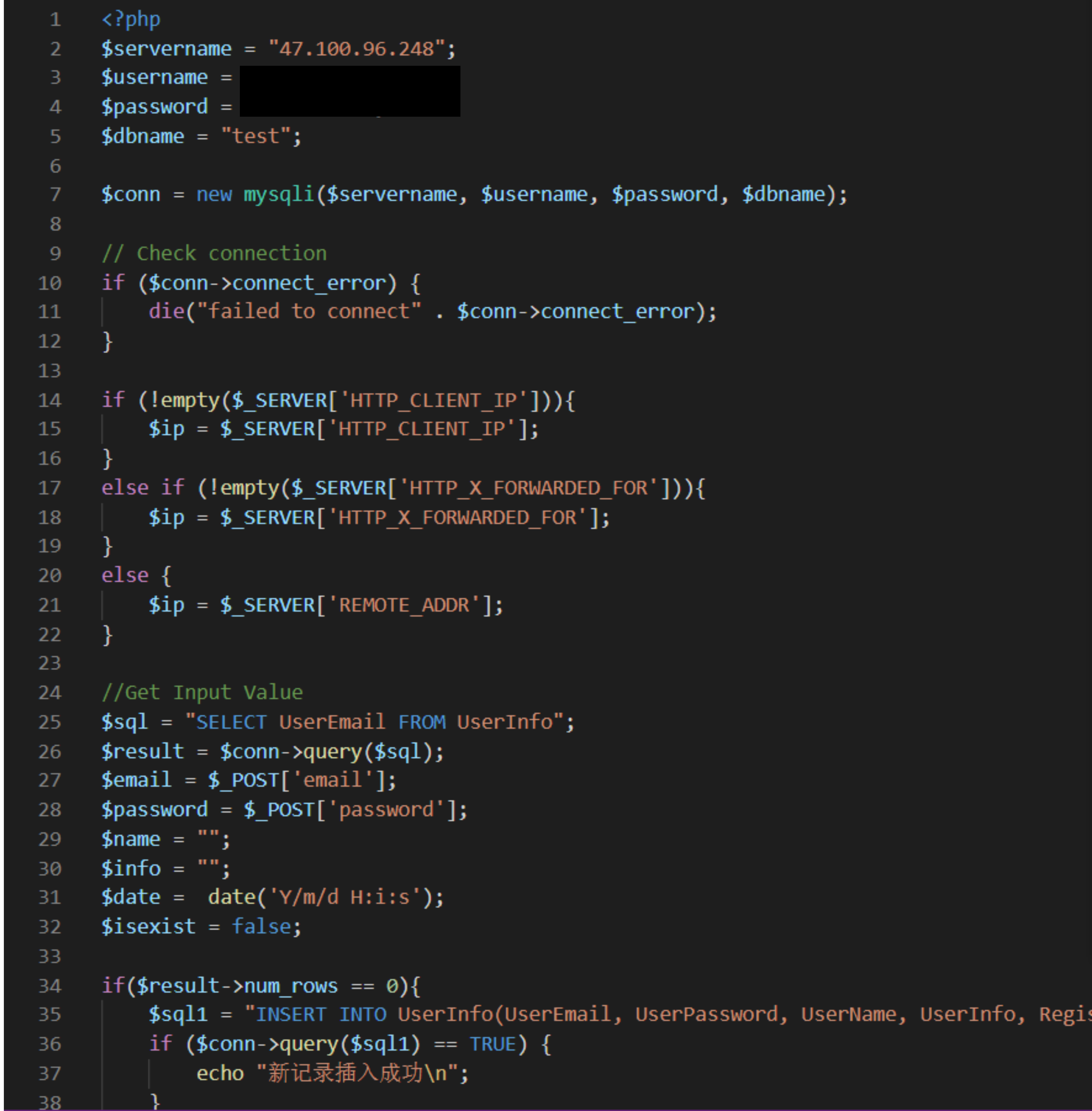
In registration function, it has the similar steps to the login functions:

1. Build the connection between back end and database

2. Receive the arguments which passed from front end and websites

3. If database is empty, then add these values to database and register successfully.

4. If database is no empty, then check whether database has this new account, if so, return wrong message, and if not, register successfully



```php
<?php
$servername = "47.100.96.248";
$username = [redacted]
$password =
$dbname = "test";

$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("failed to connect" . $conn->connect_error);
}

if (!empty($_SERVER['HTTP_CLIENT_IP'])){
    $ip = $_SERVER['HTTP_CLIENT_IP'];
}
else if (!empty($_SERVER['HTTP_X_FORWARDED_FOR'])){
    $ip = $_SERVER['HTTP_X_FORWARDED_FOR'];
}
else {
    $ip = $_SERVER['REMOTE_ADDR'];
}

//Get Input Value
$sql = "SELECT UserEmail FROM UserInfo";
$result = $conn->query($sql);
$email = $_POST['email'];
$password = $_POST['password'];
$name = "";
$info = "";
$date =  date('Y/m/d H:i:s');
$isexist = false;

if($result->num_rows == 0){
    $sql1 = "INSERT INTO UserInfo(UserEmail, UserPassword, UserName, UserInfo, Regis
    if ($conn->query($sql1) == TRUE) {
        echo "新记录插入成功\n";
    }
```

Delete/Modify/Upload Post and Reply

The steps to achieve the function of deleting / modifying / uploading posts and replies are similar, and both of them has following steps:

1. Build connect between back end and database

2. Receive arguments which passed from front end and websites

3. Add these values to database

Because both of post and reply have their own id, so each time author upload a post or reply, the id of this post and reply will automatically add one. For uploading reply, it will also use the id of post which this reply refer to.

We will also use ajax to dynamically load web pages, so that each time users enter this webpage, the content and length of the web page will be automatically modified according to the number of posts and replies in the database

Modify Personal Information

We mainly implement this function by three steps:

1. Build the connection between back end and database

2. Receive the arguments which passed from front end and websites.

3. Update database based on new personal message

Database

1. User Information
2. Post Information
3. Reply Information

User Information

	#	名字	类型	排序规则	属性	空	默认	额外	操作
□	1	**UserEmail**	varchar(50)	utf8mb4_general_ci		否	*无*		修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)
□	2	**UserPassword**	varchar(50)	utf8mb4_general_ci		否	*无*		修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)
□	3	**UserName**	varchar(50)	utf8mb4_general_ci		否			修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)
□	4	**UserInfo**	varchar(50)	utf8mb4_general_ci		否			修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)
□	5	**RegisterDate**	date			是	*NULL*		修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)

1. UserEmail: Users use their email to sign up for accounts. Is also the representative of the account
2. UserPassword: The password of account
3. UserName: The name of user
4. UserInfo: User profile
5. RegisterDate: The date of sign up.

Post Information

	#	名字	类型	排序规则	属性	空	默认	额外	操作
□	1	**PostID**	int(11)			否	*无*		修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)
□	2	**PostContent**	varchar(1000)	utf8mb4_general_ci		否	*无*		修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)
□	3	**PostAuthor**	varchar(100)	utf8mb4_general_ci		否	*无*		修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)
□	4	**PostTitle**	varchar(100)	utf8mb4_general_ci		否	*无*		修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)
□	5	**PostDate**	date			否	*无*		修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)

1. PostID: The id of post, and each post has its own unique ID for identification and searching.
2. PostContent: The main article of this post
3. PostAuthor: The author of post
4. PostTitle: The title of post
5. PostDate: Time when the author uploaded the post

Reply Information

	#	名字	类型	排序规则	属性	空	默认	额外	操作
□	1	**ReplyID**	int(11)			否	无		修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)
□	2	**PostID**	int(11)			否	无		修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)
□	3	**ReplyContent**	varchar(1000)	utf8mb4_general_ci		否	无		修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)
□	4	**ReplyAuthor**	varchar(100)	utf8mb4_general_ci		否	无		修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)
□	5	**ReplyTitle**	varchar(100)	utf8mb4_general_ci		否	无		修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)
□	6	**ReplyDate**	date			否	无		修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)

1. ReplyID: The id of reply, and each reply has its own unique ID for identification and searching.
2. PostID: It decides which post this reply belongs to.
3. ReplyContent: The main article of this reply
4. ReplyAuthor: The author of reply
5. ReplyTitle: The title of reply
6. ReplyDate: Time when the author uploaded the reply

Questions!