

1. 引言

目的：本实现文档旨在详细描述“互联网数据库开发”课程设计项目的具体实现过程、结果以及遇到的问题与解决方案。

范围：本项目包括前后台功能的实现、数据库设计、MVC 架构的应用，并确保团队合作工具（如 Git）的有效使用。

2. 团队与任务分工

团队成员及角色

2210203(前/后端开发)

- 1.负责项目的整体架构设计，制定项目的开发计划和时间表。
- 2.使用 Yii2 框架构建服务器端逻辑，包括但不限于用户认证、权限管理、业务逻辑处理等。
- 3.设计并实现数据库模型，编写高效的查询语句。
- 4.设计网站的整体视觉风格，包括色彩搭配、图标设计等。

2211545(文档撰写，数据库管理)

- 1.编写详细的项目文档，如需求分析文档、设计文档、操作手册等。
- 2.准备演示材料和技术报告，为项目展示做准备。
- 3.负责数据库的搭建，确保数据的完整性和可用性。

任务分配

- 使用图表或列表形式展示每个成员的任务分配情况，可以包括 Tower 或 Trello 的截图来证明任务的分配和完成情况。

我们一般都是线下直接交流。

工作量评估

- 对各成员的工作量进行合理评估，并说明如何保证公平性。例如，通过定期的进度汇报会议确保每位成员的任务量均衡。

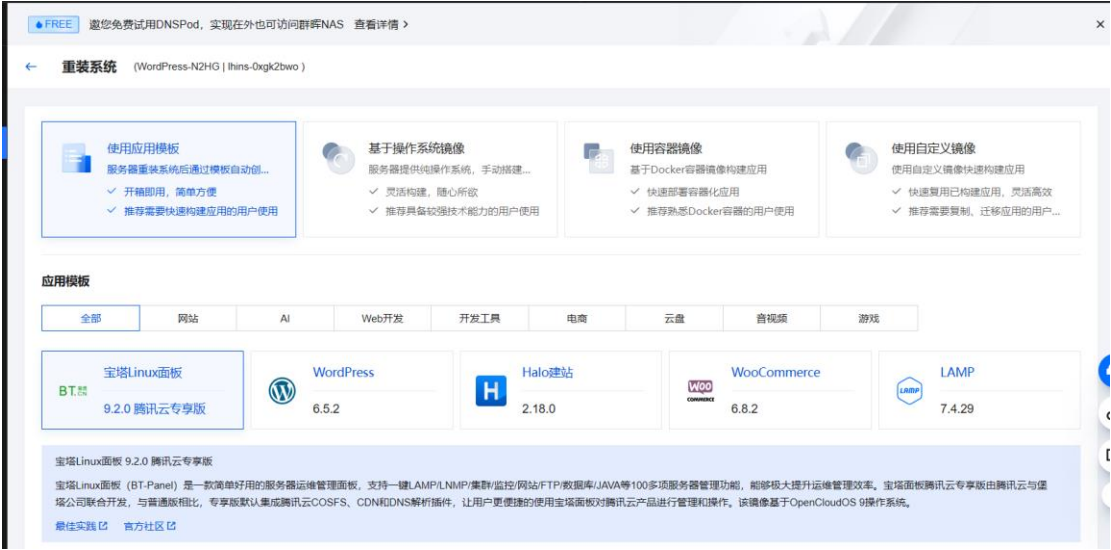
3. 开发环境搭建

-环境准备

- 操作系统：OpenCloudOS 9
- 编程语言：PHP 7.4.33, JavaScript, HTML5, CSS3
- 数据库管理系统：MySQL 5.7.44
- Web 服务器：Nginx 1.26.2

安装步骤

1.进入腾讯云服务器控制台，安装宝塔面板。



```
[lighthouse@VM-16-2-opencloudos ~]$ su
Password:
[root@VM-16-2-opencloudos lighthouse]# bt
=====宝塔面板命令行=====
(1) 重启面板服务          (8) 改面板端口          |
(2) 停止面板服务          (9) 清除面板缓存        |
(3) 启动面板服务          (10) 清除登录限制        |
(4) 重载面板服务          (11) 设置是否开启IP + User-Agent验证 |
(5) 修改面板密码          (12) 取消域名绑定限制    |
(6) 修改面板用户名        (13) 取消IP访问限制      |
(7) 强制修改MySQL密码     (14) 查看面板默认信息    |
(22) 显示面板错误日志     (15) 清理系统垃圾        |
(23) 关闭BasicAuth认证    (16) 修复面板(检查错误并更新面板文件到最新版) |
(24) 关闭动态口令认证     (17) 设置日志切割是否压缩 |
(25) 设置是否保存文件历史副本 (18) 设置是否自动备份面板 |
(26) 关闭面板ssl          (19) 关闭面板登录地区限制 |
(28) 修改面板安全入口     (29) 取消访问设备验证    |
(30) 取消访问UA验证       (32) 开启/关闭【80、443】端口访问面板 |
(0) 取消                  |
=====
请输入命令编号: 14
```

```
请输入命令编号: 14
=====
正在执行(14)...
=====
BT-Panel default info!
=====
外网面板地址: http://175.24.229.104:8888/tencentcloud
内网面板地址: http://10.0.16.2:8888/tencentcloud
username: e074ded3
password: d382fe33b0a5
Warning:
If you cannot access the panel,
release the following port (8888|888|80|443|20|21) in the security group
注意: 初始密码仅在首次登录面板前能正确获取, 其它时间请通过 bt 5 命令修改密码
=====
[root@VM-16-2-opencloudos lighthouse]# * Reconnecting *
[lighthouse@VM-16-2-opencloudos ~]$
```

记得提前开放相应端口

宝塔免费版试用DNSPod, 实现在外也可访问群晖NAS 查看详情 >

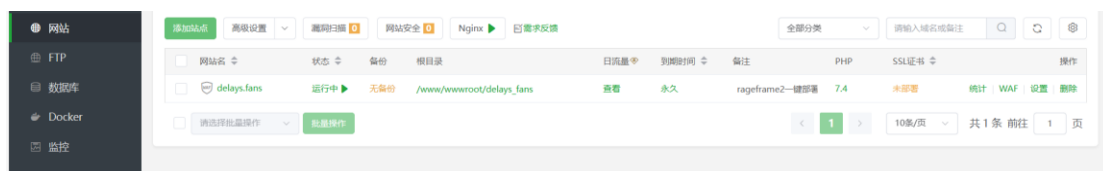
| | | | | | | | |
|--------------------------|--------------------|---------|------|-------|----|-----------------------------------|-------|
| <input type="checkbox"/> | FTP (21) | 0.0.0.0 | TCP | 21 | 允许 | FTP服务 (21) | 编辑 删除 |
| <input type="checkbox"/> | 自定义 | 0.0.0.0 | TCP | 888 | 允许 | | 编辑 删除 |
| <input type="checkbox"/> | 自定义 | 0.0.0.0 | TCP | 37185 | 允许 | | 编辑 删除 |
| <input type="checkbox"/> | 自定义 | 0.0.0.0 | TCP | 24444 | 允许 | | 编辑 删除 |
| <input type="checkbox"/> | 自定义 | 0.0.0.0 | TCP | 23333 | 允许 | | 编辑 删除 |
| <input type="checkbox"/> | 自定义 | 0.0.0.0 | TCP | 25565 | 允许 | | 编辑 删除 |
| <input type="checkbox"/> | 自定义 | 0.0.0.0 | TCP | 8888 | 允许 | | 编辑 删除 |
| <input type="checkbox"/> | 自定义 | 0.0.0.0 | UDP | 1194 | 允许 | | 编辑 删除 |
| <input type="checkbox"/> | 自定义 | 0.0.0.0 | TCP | 35164 | 允许 | | 编辑 删除 |
| <input type="checkbox"/> | 自定义 | 0.0.0.0 | TCP | 3000 | 允许 | | 编辑 删除 |
| <input type="checkbox"/> | HTTP (80) | 0.0.0.0 | TCP | 80 | 允许 | Web服务HTTP (80), 如 Apache, Nginx | 编辑 删除 |
| <input type="checkbox"/> | HTTPS (443) | 0.0.0.0 | TCP | 443 | 允许 | Web服务HTTPS (443), 如 Apache, Nginx | 编辑 删除 |
| <input type="checkbox"/> | Linux 登录 (22) | 0.0.0.0 | TCP | 22 | 允许 | Linux SSH登录 | 编辑 删除 |
| <input type="checkbox"/> | Windows登录 (3389) | 0.0.0.0 | TCP | 3389 | 允许 | Windows远程桌面登录 | 编辑 删除 |
| <input type="checkbox"/> | Windows登录优化 (3389) | 0.0.0.0 | UDP | 3389 | 允许 | Windows远程桌面登录优化 | 编辑 删除 |
| <input type="checkbox"/> | Ping | 0.0.0.0 | ICMP | ALL | 允许 | 通过Ping测试网络连通性 (即通ALL ICMP) | 编辑 删除 |

共 16 条20 条 / 页1 / 1 页

2.在宝塔面板的软件商店下载相关软件



3.添加站点，完成相关配置



设置伪静态

```
location / {  
    try_files $uri $uri/ /index.php$is_args$args;  
}  
  
location /backend {  
    try_files $uri $uri/ /backend/index.php$is_args$args;  
}  
  
location /api {  
    try_files $uri $uri/ /api/index.php$is_args$args;  
}
```

```
location /merchant {  
    try_files $uri $uri/ /merchant/index.php$is_args$args;  
}  
  
location /merapi {  
    try_files $uri $uri/ /merapi/index.php$is_args$args;  
}  
  
location /html5 {  
    try_files $uri $uri/ /html5/index.php$is_args$args;  
}  
  
location /oauth2 {  
    try_files $uri $uri/ /oauth2/index.php$is_args$args;  
}
```

配置文件

```
server  
  
{  
    listen 80;  
  
    server_name delays.fans 175.24.229.104;  
  
    index index.html index.php index.htm default.php default.htm  
    default.html;  
  
    root /www/wwwroot/delays_fans/web;  
  
    #CERT-APPLY-CHECK--START
```

```
# 用于 SSL 证书申请时的文件验证相关配置 -- 请勿删除

include

/www/server/panel/vhost/nginx/well-known/delays.fans.conf;

#CERT-APPLY-CHECK--END


#SSL-START SSL 相关配置，请勿删除或修改下一行带注释的 404
规则

#error_page 404/404.html;

#SSL-END


#ERROR-PAGE-START 错误页配置，可以注释、删除或修改
error_page 404 /404.html;

#error_page 502 /502.html;

#ERROR-PAGE-END


#PHP-INFO-START PHP 引用配置，可以注释或修改

include enable-php-74.conf;

#PHP-INFO-END


#REWRITE-START URL 重写规则引用,修改后将导致面板设置的伪
静态规则失效

include /www/server/panel/vhost/rewrite/delays.fans.conf;
```

#REWRITE-END

#禁止访问的文件或目录

location

~

^/(\.user.ini|\.htaccess|\.git|\.env|\.svn|\.project|LICENSE|README.m

d)

{

return 404;

}

#一键申请 SSL 证书验证目录相关设置

location ~ /\.well-known{

allow all;

}

#禁止在证书验证目录放入敏感文件

if

(

\$uri

~

"^/\.well-known/.*\.(php|jsp|py|js|css|lua|ts|go|zip|tar\.gz|rar|7z|sq

|bak)\$") {

return 403;

}


```
location ~ .*\. (gif|jpg|jpeg|png|bmp|swf)$

{
    expires      30d;
    error_log /dev/null;
    access_log /dev/null;
}

location ~ .*\. (js|css)?$

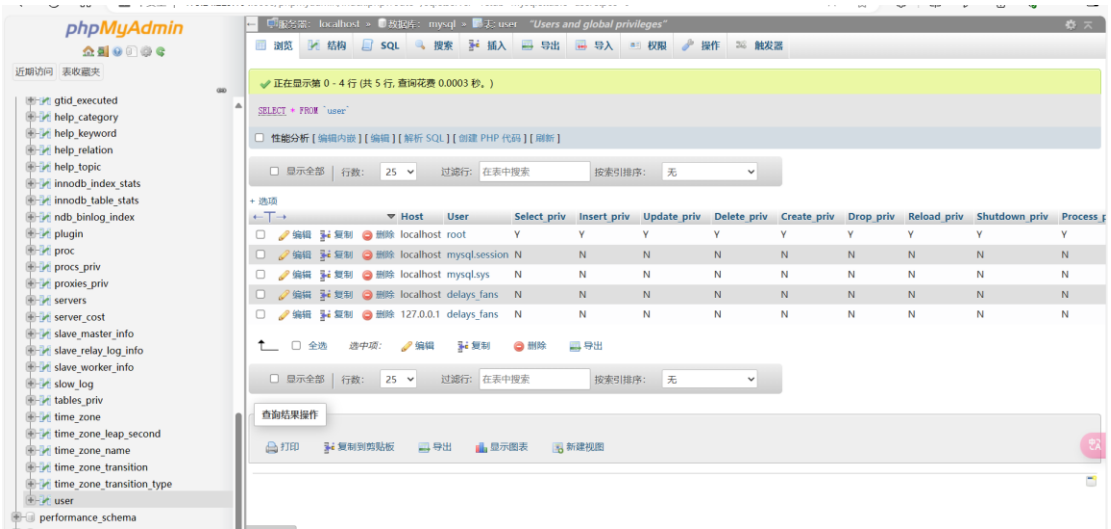
{
    expires      12h;
    error_log /dev/null;
    access_log /dev/null;
}

access_log  /www/wwwlogs/delays.fans.log;
error_log  /www/wwwlogs/delays.fans.error.log;
}
```

4. 数据库设计

数据表设计

展示数据库中所有表的设计，包括字段定义、主键、外键关系等。确保至少有 10 张表以满足课程要求。



phpMyAdmin interface showing the 'rf_rbac_auth_item' table. The table contains 14 rows of data, including columns like id, name, title, app_id, addons_name, pid, level, is_addon, sort, tree, status, created_at, and updated_at. The table is currently displaying rows 0-24 (共 172 行).

| id | name | title | app_id | addons_name | pid | level | is_addon | sort | tree | status | created_at | updated_at |
|----|--------------------------|-------|---------|-------------|-----|-------|----------|------|------------------------|--------|------------|------------|
| 1 | base | 系统基础 | backend | | 0 | 1 | 0 | 0 | tr_0 | 1 | 1582376897 | 1582376897 |
| 2 | /base/member/personal | 个人中心 | backend | | 1 | 2 | 0 | 0 | tr_0 tr_1 | 1 | 1582376897 | 1582376897 |
| 3 | /base/member/up-password | 修改密码 | backend | | 1 | 2 | 0 | 1 | tr_0 tr_1 | 1 | 1582376897 | 1582376897 |
| 4 | /main/clear-cache | 清理缓存 | backend | | 1 | 2 | 0 | 2 | tr_0 tr_1 | 1 | 1582376897 | 1582376897 |
| 5 | baseAnnounce | 查看消息 | backend | | 1 | 2 | 0 | 3 | tr_0 tr_1 | 1 | 1582376897 | 1582376897 |
| 6 | /notify/announce | 公告列表 | backend | | 5 | 3 | 0 | 0 | tr_0 tr_1 tr_5 | 1 | 1582376897 | 1582376897 |
| 7 | /notify/announce-view | 公告详情 | backend | | 5 | 3 | 0 | 1 | tr_0 tr_1 tr_5 | 1 | 1582376898 | 1582376898 |
| 8 | /notify/message | 私信列表 | backend | | 5 | 3 | 0 | 2 | tr_0 tr_1 tr_5 | 1 | 1582376898 | 1582376898 |
| 9 | /notify/remind | 提醒列表 | backend | | 5 | 3 | 0 | 3 | tr_0 tr_1 tr_5 | 1 | 1582376898 | 1582376898 |
| 10 | cate:1 | 平台首页 | backend | | 0 | 1 | 0 | 1 | tr_0 | 1 | 1582376898 | 1582376898 |
| 11 | indexMember | 会员管理 | backend | | 10 | 2 | 0 | 0 | tr_0 tr_10 | 1 | 1582376898 | 1582376898 |
| 12 | indexMemberInfo | 会员信息 | backend | | 11 | 3 | 0 | 0 | tr_0 tr_10 tr_11 | 1 | 1582376898 | 1582376898 |
| 13 | /member/member/recharge | 充值 | backend | | 12 | 4 | 0 | 0 | tr_0 tr_10 tr_11 tr_12 | 1 | 1582376898 | 1582376898 |
| 14 | /member/member/index | 首页 | backend | | 12 | 4 | 0 | 1 | tr_0 tr_10 tr_11 tr_12 | 1 | 1582376898 | 1582376898 |

phpMyAdmin interface showing the 'rf_rbac_auth_item_child' table. The table contains 14 rows of data, including columns like role_id, item_id, name, app_id, is_addon, and addons_name. The table is currently displaying rows 0-24 (共 123 行).

| role_id | item_id | name | app_id | is_addon | addons_name |
|---------|---------|-------------------------------|---------|----------|-------------|
| 1 | 1 | base | backend | 0 | |
| 1 | 2 | /base/member/personal | backend | 0 | |
| 1 | 6 | /notify/announce | backend | 0 | |
| 1 | 11 | indexMember | backend | 0 | |
| 1 | 12 | indexMemberInfo | backend | 0 | |
| 1 | 13 | /member/member/recharge | backend | 0 | |
| 1 | 20 | /member/address/index | backend | 0 | |
| 1 | 25 | /member/auth/index | backend | 0 | |
| 1 | 30 | /member/credits-log/index | backend | 0 | |
| 1 | 34 | /member/recharge-config/index | backend | 0 | |
| 1 | 42 | commonAddons | backend | 0 | |
| 1 | 43 | /common/addons/index | backend | 0 | |
| 1 | 52 | /common/config/index | backend | 0 | |
| 1 | 57 | /common/config-cate/index | backend | 0 | |
| 1 | 62 | /common/menu/index | backend | 0 | |
| 1 | 67 | /common/menu-cate/index | backend | 0 | |
| 1 | 72 | /base/notify-announce/index | backend | 0 | |

服务器: localhost x 数据库: delays_fans

结构 SQL 搜索 查询 导出 导入 操作 权限 程序 事件 触发器 设计器

过滤器

包含文字:

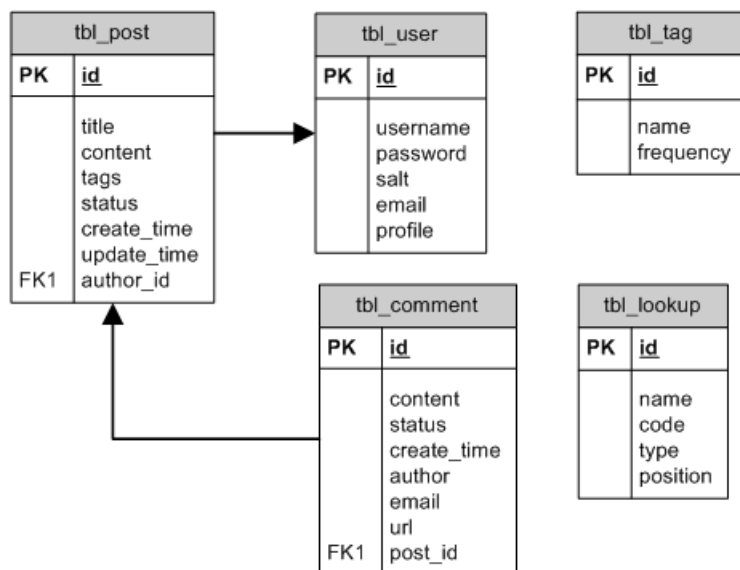
| 表 | 操作 | 行数 | 类型 | 排序规则 | 大小 | 冗余 |
|--|---------------------|-------|--------|--------------------|----------|----|
| <input type="checkbox"/> rf_api_access_token | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 48.0 KB | - |
| <input checked="" type="checkbox"/> rf_backend_auth | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 48.0 KB | - |
| <input type="checkbox"/> rf_backend_member | ★ 浏览 结构 搜索 插入 清空 删除 | 2 | InnoDB | utf8mb4_general_ci | 16.0 KB | - |
| <input type="checkbox"/> rf_backend_notify | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 32.0 KB | - |
| <input type="checkbox"/> rf_backend_notify_member | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 32.0 KB | - |
| <input type="checkbox"/> rf_backend_notify_pull_time | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 32.0 KB | - |
| <input type="checkbox"/> rf_backend_notify_subscription_config | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 16.0 KB | - |
| <input type="checkbox"/> rf_common_action_behavior | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 16.0 KB | - |
| <input type="checkbox"/> rf_common_action_log | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 16.0 KB | - |
| <input type="checkbox"/> rf_common_addons | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 48.0 KB | - |
| <input type="checkbox"/> rf_common_addons_binding | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 32.0 KB | - |
| <input type="checkbox"/> rf_common_addons_config | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 32.0 KB | - |
| <input type="checkbox"/> rf_common_attachment | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 32.0 KB | - |
| <input type="checkbox"/> rf_common_config | ★ 浏览 结构 搜索 插入 清空 删除 | 117 | InnoDB | utf8mb4_general_ci | 96.0 KB | - |
| <input type="checkbox"/> rf_common_config_cate | ★ 浏览 结构 搜索 插入 清空 删除 | 52 | InnoDB | utf8mb4_general_ci | 16.0 KB | - |
| <input type="checkbox"/> rf_common_config_value | ★ 浏览 结构 搜索 插入 清空 删除 | 14 | InnoDB | utf8mb4_general_ci | 32.0 KB | - |
| <input type="checkbox"/> rf_common_ip_blacklist | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 16.0 KB | - |
| <input type="checkbox"/> rf_common_log | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 64.0 KB | - |
| <input type="checkbox"/> rf_common_menu | ★ 浏览 结构 搜索 插入 清空 删除 | 41 | InnoDB | utf8mb4_general_ci | 32.0 KB | - |
| <input type="checkbox"/> rf_common_menu_cate | ★ 浏览 结构 搜索 插入 清空 删除 | 5 | InnoDB | utf8mb4_general_ci | 16.0 KB | - |
| <input type="checkbox"/> rf_common_pay_log | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 16.0 KB | - |
| <input type="checkbox"/> rf_common_provinces | ★ 浏览 结构 搜索 插入 清空 删除 | 3,676 | MyISAM | utf8mb4_general_ci | 397.9 KB | - |
| <input type="checkbox"/> rf_common_report_log | ★ 浏览 结构 搜索 插入 清空 删除 | 3 | InnoDB | utf8mb4_general_ci | 32.0 KB | - |
| <input type="checkbox"/> rf_common_sms_log | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 32.0 KB | - |
| <input type="checkbox"/> rf_member | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 80.0 KB | - |
| <input type="checkbox"/> rf_member_account | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 32.0 KB | - |
| <input type="checkbox"/> rf_member_address | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 32.0 KB | - |

| | | | | | | |
|---|---------------------|--------------|---------------|---------------------------|---------------|-------------|
| <input type="checkbox"/> rf_member_auth | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 48.0 KB | - |
| <input type="checkbox"/> rf_member_balance_withdraw | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8_general_ci | 16.0 KB | - |
| <input type="checkbox"/> rf_member_bank_account | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8_general_ci | 32.0 KB | - |
| <input type="checkbox"/> rf_member_credits_log | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 16.0 KB | - |
| <input type="checkbox"/> rf_member_invoice | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8_general_ci | 16.0 KB | - |
| <input type="checkbox"/> rf_member_level | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_bin | 16.0 KB | - |
| <input type="checkbox"/> rf_member_recharge_config | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8_general_ci | 16.0 KB | - |
| <input type="checkbox"/> rf_merapi_access_token | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 48.0 KB | - |
| <input type="checkbox"/> rf_merchant | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 16.0 KB | - |
| <input type="checkbox"/> rf_merchant_account | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 16.0 KB | - |
| <input type="checkbox"/> rf_merchant_auth | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 48.0 KB | - |
| <input type="checkbox"/> rf_merchant_base_config | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 16.0 KB | - |
| <input type="checkbox"/> rf_merchant_cate | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 16.0 KB | - |
| <input type="checkbox"/> rf_merchant_credits_log | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 16.0 KB | - |
| <input type="checkbox"/> rf_merchant_member | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 16.0 KB | - |
| <input type="checkbox"/> rf_migration | ★ 浏览 结构 搜索 插入 清空 删除 | 51 | InnoDB | utf8_general_ci | 16.0 KB | - |
| <input type="checkbox"/> rf_oauth2_access_token | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 48.0 KB | - |
| <input type="checkbox"/> rf_oauth2_authorization_code | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 32.0 KB | - |
| <input type="checkbox"/> rf_oauth2_client | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 32.0 KB | - |
| <input type="checkbox"/> rf_oauth2_refresh_token | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 32.0 KB | - |
| <input type="checkbox"/> rf_rbac_auth_assignment | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 16.0 KB | - |
| <input type="checkbox"/> rf_rbac_auth_item | ★ 浏览 结构 搜索 插入 清空 删除 | 172 | InnoDB | utf8mb4_general_ci | 48.0 KB | - |
| <input type="checkbox"/> rf_rbac_auth_item_child | ★ 浏览 结构 搜索 插入 清空 删除 | 123 | InnoDB | utf8mb4_general_ci | 48.0 KB | - |
| <input type="checkbox"/> rf_rbac_auth_role | ★ 浏览 结构 搜索 插入 清空 删除 | 0 | InnoDB | utf8mb4_general_ci | 16.0 KB | - |
| 51 张表 | 总计 | 4,256 | InnoDB | utf8mb4_general_ci | 1.9 MB | 0 字节 |

☐ 全选
 选中项:

打印 数据字典

ER 图



5. 模块实现

5.1 用户管理模块

模型（Model）：

- `User.php`：负责用户数据的 CRUD 操作，包括验证用户输入、与数据库交互以存储或检索用户信息。

```
```php
```

```
public function register($username, $password, $email) {

 // 验证输入并保存新用户到数据库

}
```

```
public function login($username, $password) {
 // 验证用户名和密码，并返回用户对象或错误信息
}
...
```

视图（View）：

- 使用 Bootstrap 框架构建响应式布局，确保不同设备上的良好用户体验。
- React.js 用于动态组件渲染，如表单验证反馈、实时显示用户状态等。
- `register.html`：提供用户注册界面。
- `login.html`：提供用户登录界面。

控制器（Controller）：

- `UserController.php`：处理所有与用户相关的 HTTP 请求，调用相应的模型方法进行业务逻辑处理，并根据结果选择适当的视图展示给用户。

```
```php  
public function handleRegister() {  
    if ($this->request->isPost()) {  
        $data = $this->request->getPost();  
        $user = new User();
```

```

        $result      =      $user->register($data['username'],
$data['password'], $data['email']);

        if ($result) {

            return $this->view->render('success.html');

        } else {

            return $this->view->render('error.html');

        }

    }

    return $this->view->render('register.html');

}

```

```

public function handleLogin() {

    if ($this->request->isPost()) {

        $data = $this->request->getPost();

        $user = new User();

        $authResult      =      $user->login($data['username'],
$data['password']);

        if ($authResult) {

            $_SESSION['user_id'] = $authResult->getId();

            return $this->redirect('/dashboard');

        } else {

            return $this->view->render('login.html', ['error' =>

```

```
'Invalid credentials']);  
  
        }  
  
    }  
  
    return $this->view->render('login.html');  
  
}  
...
```

5.2 内容发布模块

模块划分

功能描述：内容发布模块允许用户创建、编辑和删除文章或帖子。它还支持文章分类管理和标签分配。

模型（Model）：

- `Post.php`：处理文章数据的 CRUD 操作，同时管理分类和标签关联。

```
```php
```

```
public function createPost($title, $content, $category_id, $tags) {

 // 创建新文章并关联分类和标签

}
```



```

public function getPostsByCategory($category_id) {

 // 根据分类 ID 获取文章列表

}

...

```

视图（View）：

- 使用富文本编辑器（如 TinyMCE）增强文章编辑体验。
- React.js 用于实现实时预览和互动元素，如投票、点赞按钮等。
- `create-post.html`：提供文章创建界面。
- `edit-post.html`：提供文章编辑界面。

控制器（Controller）：

- `PostController.php`：接收来自前端的请求，调用模型完成相应的业务逻辑，然后呈现适当的结果给用户。

```

```php

public function handleCreatePost() {

    if ($this->request->isPost()) {

        $data = $this->request->getPost();

        $post = new Post();

        $result = $post->createPost($data['title'], $data['content'],
$data['category_id'], $data['tags']);

        if ($result) {

```

```

        return $this->redirect('/posts');

    } else {

        return $this->view->render('create-post.html',
['error' => 'Failed to create post']);

    }

}

return $this->view->render('create-post.html');

}

public function handleEditPost($id) {

    // 编辑文章逻辑...

}

...

```

5.3 评论系统模块

模块划分

功能描述：评论系统模块允许用户对文章或其他内容发表评论，支持多级回复和点赞功能。

模型（Model）：

- `Comment.php`：处理评论数据的 CRUD 操作，以及与文章或其它实体的关联关系。

```
```php

 public function addComment($post_id, $user_id, $content,
$parent_id = null) {

 // 添加新评论，指定父评论 ID（如果有的话）

 }

 public function getCommentsByPostId($post_id) {

 // 获取指定文章下的所有评论，包括子评论

 }

 ...
```
```

视图（View）：

- 使用 AJAX 技术实现实时评论加载，无需刷新页面即可查看最新评论。

- React.js 用于构建交互性强的评论组件，如展开/折叠回复、点赞按钮等。

- `comments.html`：嵌入在文章详情页中，显示评论列表并提供评论框。

控制器（Controller）：

- `CommentController.php`: 处理评论相关的请求，如添加新评论、获取评论列表等。

```
```php

public function handleAddComment() {

 if ($this->request->isPost()) {

 $data = $this->request->getPost();

 $comment = new Comment();

 $result = $comment->addComment($data['post_id'],
$_SESSION['user_id'], $data['content'], $data['parent_id']);

 if ($result) {

 return json_encode(['status' => 'success', 'message' =>
'Comment added successfully']);

 } else {

 return json_encode(['status' => 'error', 'message' =>
'Failed to add comment']);

 }

 }

 return json_encode(['status' => 'error', 'message' => 'Invalid
request']);

}

public function handleGetComments($post_id) {
```

```

$comment = new Comment();

$comments = $comment->getCommentsByPostId($post_id);

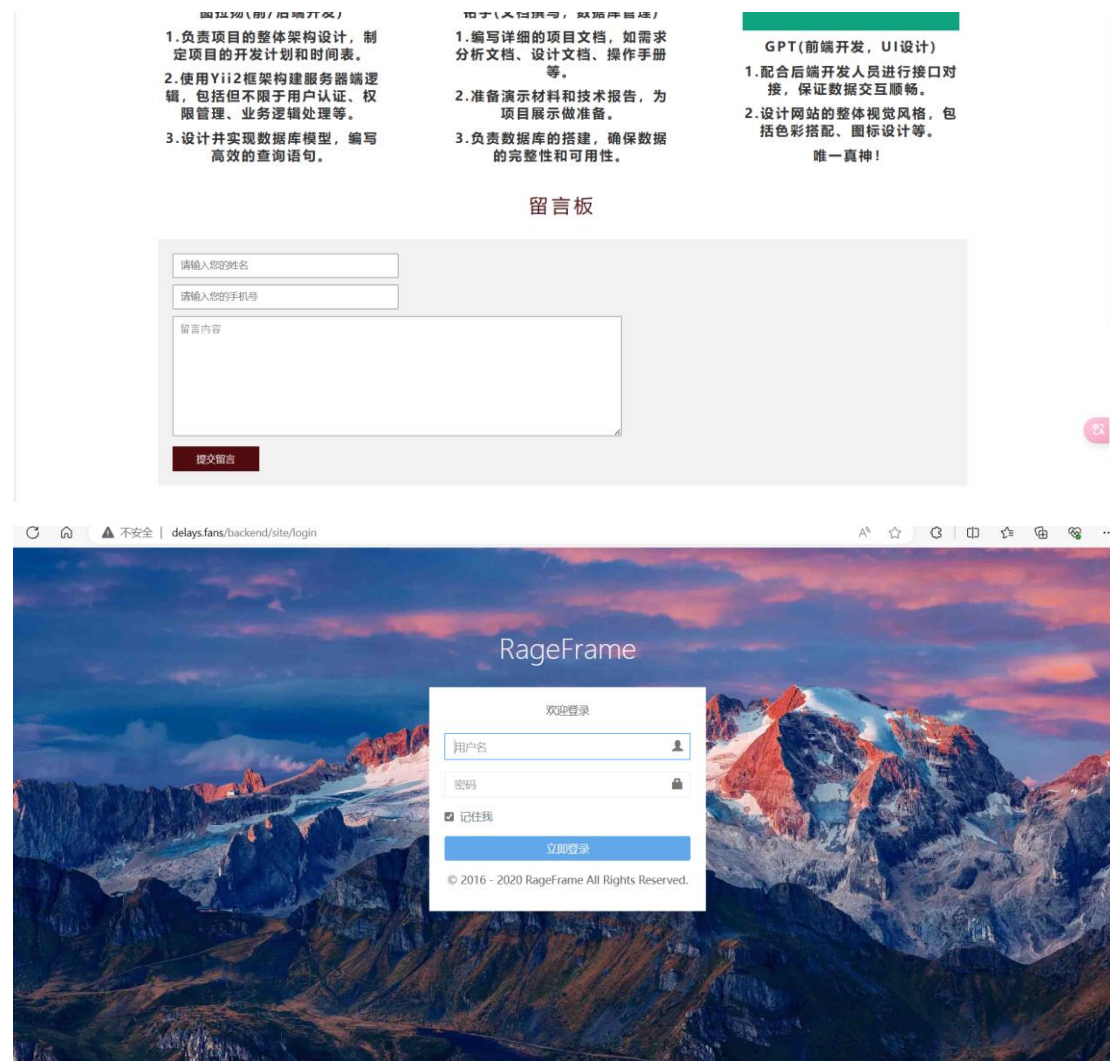
return json_encode($comments);

}

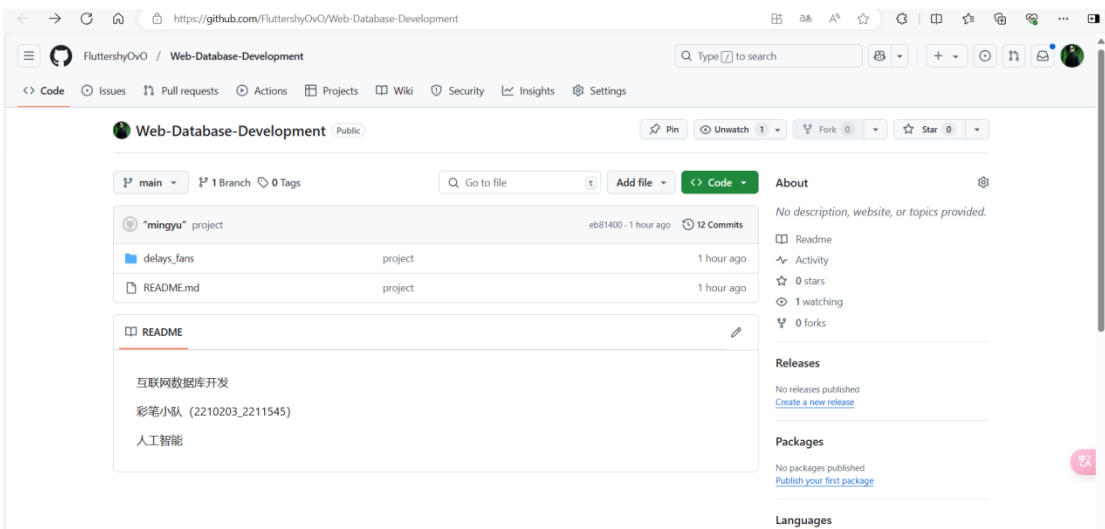
...

```

## 6. 实现结果



## 7. Git 提交记录



## 8.总结与反思

### 成果总结

在本次“互联网数据库开发”课程设计项目中，我们团队成功实现了预期的多个关键目标，并达到了课程设计的要求。以下是具体成果的回顾：

功能完整：我们构建了一个具备前后台功能的团队主页，前台包括团队和个人展示、留言功能，后台则实现了团队和个人信息编辑、信息发布等功能。所有功能均按计划完成，并且用户体验良好。

数据库设计：根据课程要求，我们设计了超过 10 张数据表，涵盖了用户信息、文章发布、评论管理等多个方面。通过精心设计的数据模

型和实体关系图（ER 图），确保了系统的扩展性和高效性。

**MVC 架构应用：**严格遵循 MVC 架构进行开发，每个成员至少编写了一个 MVC 层的文件，保证了代码结构清晰、易于维护。例如，UserController.php 处理用户登录注册，PostController.php 负责内容发布管理，而 CommentController.php 则管理评论系统。

**团队合作工具使用：**我们选择了 Git 作为版本控制系统，Tower 作为项目管理工具，有效地跟踪任务进度并管理代码变更。团队成员之间的沟通顺畅，分工明确，任务分配合理，确保了项目的顺利推进。

**文档齐全：**按照课程要求，提交了需求文档、设计文档、实现文档、用户手册和部署文档等全部必要文档，每份文档都详细记录了相关工作内容，为后续维护提供了重要参考。

**性能优化与安全措施：**对应用程序进行了性能测试和安全审查，采取了一系列优化措施以提高响应速度，并加强了数据保护机制，如输入验证、SQL 注入防护等。

**动态图形展示：**为了增加项目的亮点，我们在后台加入了动态图形展示功能，能够直观地呈现团队活动统计数据，得到了额外加分。

一键部署功能：实现了简单的自动化部署脚本，简化了环境搭建过程，提高了部署效率，也为项目加分不少。

## 挑战与应对

尽管项目总体进展顺利，但在实施过程中我们也遇到了一些挑战：

技术难题：初期对于某些新技术的理解不够深入，导致开发进度稍有延迟。通过查阅资料、在线学习以及向导师请教，最终克服了这些障碍。

团队沟通：由于团队成员时间安排不同，有时难以及时协调解决问题。为此，我们建立了每日简短的线上会议制度，确保信息同步，问题快速解决。

代码整合：在集成各模块时发现部分接口定义不一致，经过讨论决定统一 API 规范，重新调整了部分代码逻辑，保证了系统的稳定性。

## 未来改进方向

展望未来，我们可以考虑以下几个方面的改进：



功能增强：根据用户反馈继续丰富网站功能，如添加社交分享、个人空间定制等功能，提升用户粘性。

性能优化：进一步优化数据库查询语句，减少不必要的网络请求，提高整体响应速度。

安全性强化：持续关注最新的安全威胁趋势，定期更新防护策略，确保用户数据的安全。

用户体验提升：收集更多用户意见，不断优化界面设计和交互流程，打造更加友好便捷的用户体验。