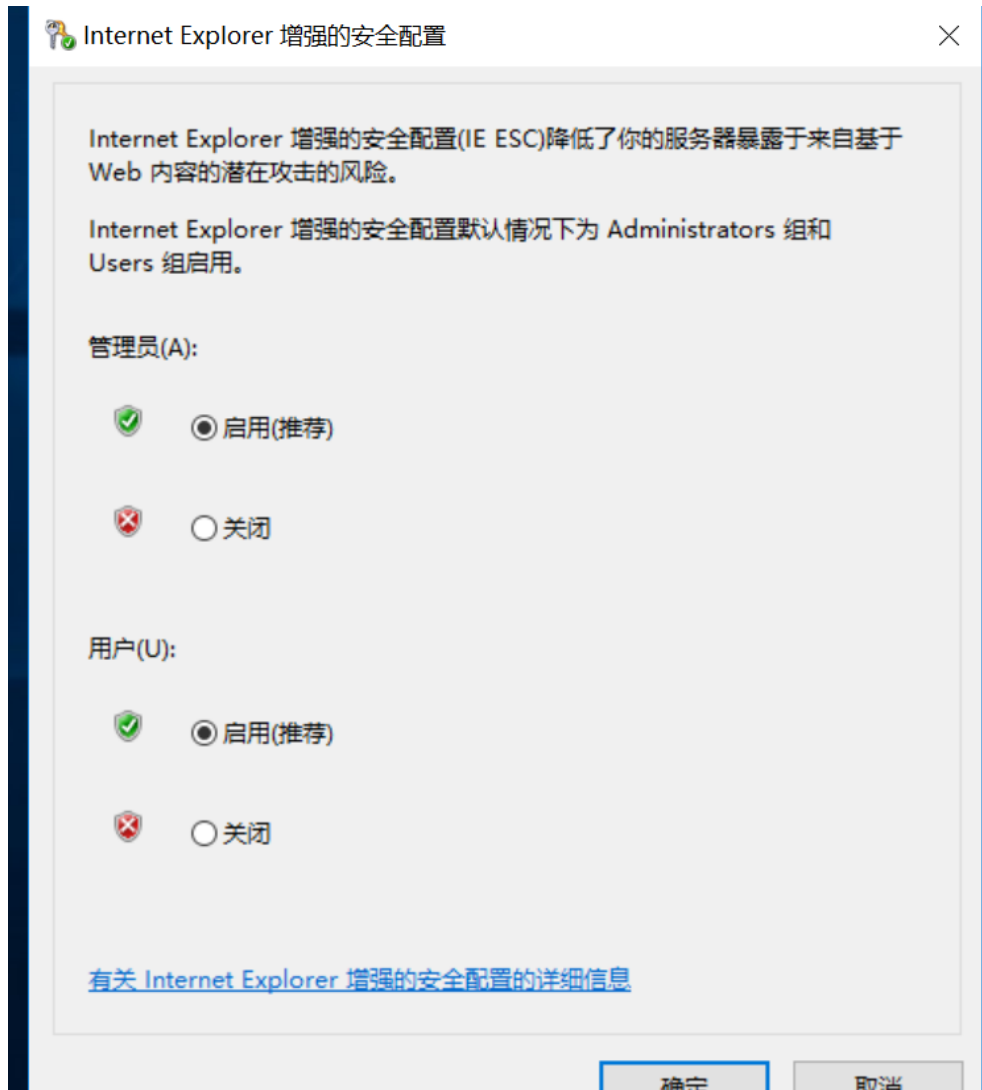


一、实验目的

1. 能够编写基本 SQL 语句管理 MySQL 数据库，掌握 HTML、JavaScript、CSS 和 PHP 编写的网页，进行静态和动态 PHP 网页开发。
2. 可以解释 Web 服务器平台的主流组合方式以及 Web 服务器平台的构成，能够实施 Web 服务器平台的安装与配置，具备基本的代码版本管理能力。

二、实验过程

首先安装 Windows Server 2016 Standard 标准版桌面操作系统，关闭 IE 浏览器增强安全模式，这是因为 IE 浏览器增强安全模式默认会严格限制网页访问、下载、脚本执行等行为。实验中需频繁访问本地网页、下载软件，开启 IE 浏览器增强安全模式会不断弹出安全警告、阻止操作，关闭后能让浏览器操作更顺畅，避免因安全拦截干扰实验步骤。



关闭防火墙，Windows 防火墙默认会拦截不明来源的网络连接、端口访问。实验中需配置 Apache 监听 80 等端口，让浏览器能访问本地 Web 服务；MySQL 也需监听 3306 端口供 PHP 连接。关闭防火墙可避免这些端口被拦截，确保 Apache、PHP、MySQL 之间的

网络通信正常。

控制面板 > 系统和安全 > Windows 防火墙 > 自定义设置

自定义各类网络的设置

你可以修改使用的每种类型的网络的防火墙设置。

专用网络设置



启用 Windows 防火墙

阻止所有传入连接，包括位于允许应用列表中的应用

Windows 防火墙阻止新应用时通知我



关闭 Windows 防火墙(不推荐)

公用网络设置



启用 Windows 防火墙

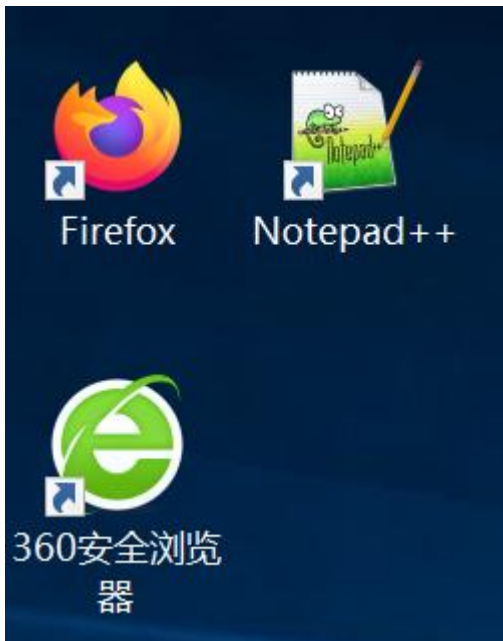
阻止所有传入连接，包括位于允许应用列表中的应用

Windows 防火墙阻止新应用时通知我



关闭 Windows 防火墙(不推荐)

安装 Firefox 浏览器和 360 安全浏览器（去官网安装即可），可以再安装一个 notepad++ 方便管理代码。



然后分别进行 PHP、MySQL、Apache 的安装与配置，主要是修改相关配置文件，使 PHP、MySQL、Apache 能够相互连接使用，以满足实验开发需求。

按照实验指导书内容配置 php 和 mysql。

实验问题：MySQL 安装后服务无法启动，发送系统错误 1067。

```
C:\Windows\system32>mysqld --install
Service successfully installed.

C:\Windows\system32>net start mysql
MySQL 服务正在启动 ...
MySQL 服务无法启动。

系统出错。

发生系统错误 1067。

进程意外终止。
```

打开“计算机管理“->”事件查看器”->”Windows 日志”->”应用程序”查看对应的报错，发现是路径问题，查看 mysql.ini 发现少添加了 datadir（数据存储目录）路径。

检查 my.ini，是否存在 datadir 且其是否指向实际存在的目录。发现 datadir 不存在。

```
basedir=C:\Users\Administrator\Desktop\Web\MYSQL\mysql-5.5.59-win32\mysql-5
datadir=C:\Users\Administrator\Desktop\Web\MYSQL\mysql-5.5.59-win32\mysql-5
character_set_server=utf8
```

添加 datadir 上去之后再次尝试开启 myssql 服务“net start mysql”，mysql 服务成功运行。

然后设置 mysql 密码为 root/123456，使用命令“mysqladmin -u root password 123456”。尝试登录，登陆成功。

```
C:\Users\Administrator>mysql -u root -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.5.59 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, 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>
```

然后配置 apache2.4，加载 php7.1.16，然后进入 apache2.4 的安装目录 bin，打开 cmd，数日命令安装并启动服务：httpd -k install httpd -k start。

```
C:\Users\Administrator\Desktop\Web\apache2.4\httpd-2.4.33-o102o-x86-vc14-r2\Apache24\bin>httpd -k sta
t
(OS 10048) 通常每个套接字地址(协议/网络地址/端口)只允许使用一次。 : AH00072: make_sock: could not bind
to address [::]:80
(OS 10048) 通常每个套接字地址(协议/网络地址/端口)只允许使用一次。 : AH00072: make_sock: could not bind
to address 0.0.0.0:80
AH00451: no listening sockets available, shutting down
AH00015: Unable to open logs
```

最后完成项目代码的版本管理与测试：

先去 GIT 官网下载 git 软件包 [Git- 下载 - Git 版本控制系统](#)，下载完成后安装即可。
然后用 git 打开 apache2.4 安装目录下的 htdocs 目录

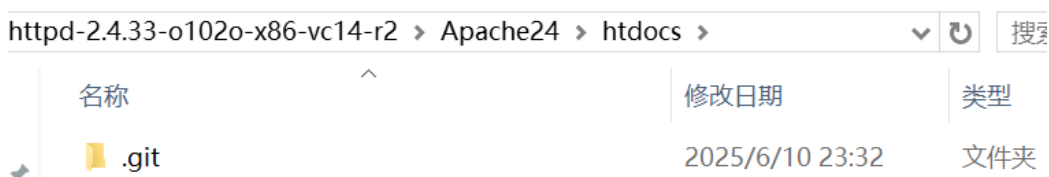


初始化仓库。

git init

```
Administrator@WIN-DEM9HKQN6U1 MINGW64 ~/Desktop/Web/apache2.4/httpd-2.4.33-o102o-x86-vc14-r2/Apache24/htdocs (master)
$ git init
Reinitialized existing Git repository in C:/Users/Administrator/Desktop/Web/apache2.4/httpd-2.4.33-o102o-x86-vc14-r2/Apache24/htdocs/.git/
```

htdocs 目录下出现.git 文件夹则初始化成功。

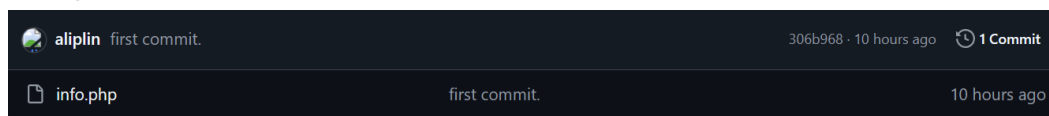


然后配置 github 相关账号配置之后就可以进行提交，提交命令如下：

git add .

git commit -m "first commit"

查看 github 页面发现提交成功。



任务一：学习基本 SQL 语言和 MySQL 数据库的使用，包括数据库和数据表的创建及数据操作。

进行 users 表的创建操作。通过在 MySQL 命令行输入 create table users 语句，定义了包含 id、username、passcode 字段的表结构。

```
mysql> create table users
-> (
-> id int not null auto_increment,
-> username char(32) not null,
-> passcode char(32) not null,
-> primary key(id)
-> );
Query OK, 0 rows affected (0.17 sec)
```

执行后提示 Query OK，表明表创建成功。

用 desc users 查看表结构，清晰展示出各字段的类型、是否允许为空、主键及额外属性等信息。

```
mysql> desc users;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
username	char(32)	NO		NULL	
passcode	char(32)	NO		NULL	

```
3 rows in set (0.11 sec)
```

表结构符合预期。

对 users 表进行扩展与数据操作。先通过 alter table 语句为表添加 status 布尔型字段并设默认值 true，执行成功后用 select * 查询，能看到新增字段。

```
mysql> alter table users add status boolean default true;
Query OK, 1 row affected (0.09 sec)
Records: 1 Duplicates: 0 Warnings: 0

mysql> select * from users;
```

id	username	passcode	email	status
1	admin	admin123	admin@sys.com	1

```
1 row in set (0.00 sec)
```

向 user 表插入 admin:admin123 的用户信息，初始 email 为 NULL。

```
mysql> insert into users(username,passcode) values('admin','admin123');
Query OK, 1 row affected (0.00 sec)

mysql> select * from users;
```

id	username	passcode	email
1	admin	admin123	NULL

```
1 row in set (0.00 sec)
```

更新 admin 用户信息中的 email 为 admin@sys.com

```
mysql> update users set email='admin@sys.com' where id=1;
Query OK, 1 row affected (0.36 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from users;
```

id	username	passcode	email
1	admin	admin123	admin@sys.com

```
1 row in set (0.00 sec)
```

用用 select 查询语句验证，发现数据准确变更。

创建本地用户 firstuser，该用户只能本地连接数据库。通过 create user 语句指定用户名、主机及密码，并用 use mysql 和 select user,host from user 查看，确认新用户创建及权限相关记录。

```
mysql> create user 'firstuser'@'localhost' identified by 'fist*2018~USER';
Query OK, 0 rows affected (0.32 sec)

mysql> use mysql;
Database changed
mysql> select user,host from user;
+-----+-----+
| user      | host      |
+-----+-----+
| root      | 127.0.0.1 |
| root      | :::1      |
| firstuser | localhost |
| root      | localhost |
+-----+-----+
5 rows in set (0.00 sec)
```

任务二：综合使用 HTML、CSS 和 JavaScript 设计一个静态网页，展示用户信息。

运用 HTML 构建页面结构。

通过标签定义“欢迎您访问！”标题、用户信息表格内容，明确数据展示的布局框架。CSS 负责样式渲染，设置蓝色标题栏、表格样式等，美化页面视觉效果，提升可读性。JavaScript 实现数据校验、表格交互等效果。

将固定的用户信息静态嵌入页面，完成信息展示。

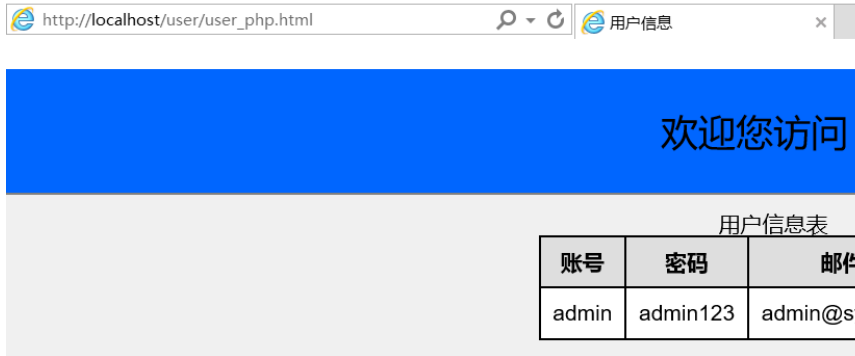


任务三：使用 PHP 语言将静态网页修改为动态网页，实现从数据库读取数据并动态展示。

在静态基础上，引入 PHP 实现动态化。

PHP 作为服务端脚本语言，可与 MySQL 数据库建立连接，从数据库中读取用户数据，再将读取的数据 动态插入 HTML 结构，生成包含实时数据库内容的网页。

由于数据库中目前只存在 admin 的用户数据，因此页面只能生成 admin 的用户数据。访问页面，与猜想一致。



相比静态网页，关键在于服务端与数据库交互，数据不再硬编码，而是按需从数据库获取并渲染，实现内容**动态**更新展示。

三、实验结论

本次实验围绕 Web 服务器平台与 Web 开发基础展开。成功完成 Windows Server 2016 环境搭建，解决 MySQL 服务启动报错等问题，实现 PHP、MySQL、Apache 协同配置。通过 SQL 熟练操作 MySQL 数据库，完成表创建、字段修改、数据增删改及用户权限管理；用 HTML、CSS、JavaScript 构建静态网页，再经 PHP 改造为动态网页，实现数据库数据动态渲染。全面掌握 Web 服务器主流组合及原理，熟悉 Git 版本管理流程。实验验证了数据库与 Web 开发技术的协同应用，清晰梳理静态到动态网页的转换逻辑，为后续 Web 安全开发与数据库攻防实践夯实基础，显著提升数据库操作与 Web 开发融合的实践能力。

四、思考题

1. 你还了解哪些 Web 服务器平台组合？
 - Node.js + MongoDB+ Express: Node.js (实现服务端逻辑)+MongoDB+Express (Web 开发框架)。Node.js 借助事件驱动、非阻塞 I/O 处理高并发请求，Express 框架简化路由、中间件等 Web 开发流程；MongoDB 以文档存储数据，与 Node.js 生态的 JSON 数据格式天然契合，便于快速开发实时交互、数据结构灵活的应用。适用于实时应用。
 - Python+Flash/Django+数据库: Python 搭配 Django 或 Flask，结合关系型或非关系型数据库。客户端请求抵达 Web 服务器，经框架路由分发到对应视图函数，函数通过 ORM 操作数据库获取/处理数据，再返回响应；Django 还内置模板引擎可渲染动态页面，Flask 常结合 Jinja2 模板或纯 API 开发。Django 适合快速搭建完整功能的企业级应用；Flask 适用于轻量级 API 开发、小型项目或需灵活扩展的场景，像数据分析平台的后端服务。
 - J2EE: Java+Tomcat/WebSphere/WebLogic+数据库。Java 代码部署到应用服务器，服务器负责加载、运行代码，处理 HTTP 请求；通过 JDBC 连接数据库操作数据，再将结果以 HTML 或 JSON 等形式响应；企业级服务器还可实现事务管理、负载均衡等。适用于大型企业级应用。
 - LAMP/LNMP: Linux+Apache/Nginx+MySQL/MariaDB+PHP，是经典开源 Web 开发组合。LAMP 中用 Apache 处理请求，LNMP 则以轻量级、高并发的 Nginx

替代 Apache。Apache/Nginx 接收客户端请求，静态资源直接返回；动态请求交给 PHP 处理，PHP 连接数据库读写数据，最终将处理结果生成 HTML 响应给客户端。适用于中小企业官网。

2. PHP 代码可以混写 CSS 吗？尝试在自定义的 outputusers() 函数中将状态为 0 的记录用红色显示。

可以混写 CSS 代码。

借助 outputusers() 函数，遍历从数据库获取的用户记录，判断 status 字段值，若为 0，则该行 <td> 添加红色背景样式 (#FF0000)。

```
10 function outputusers() {
11     global $result;
12     while(($row = mysqli_fetch_array($result))) {
13         $style = "";
14         if($row[4]==0) {
15             $style = "style='background-color: #FF0000'";
16         }
17         echo "<tr>";
18         echo "<td $style>".$row[1]."</td>";
19         echo "<td $style>".$row[2]."</td>";
20         echo "<td $style>".$row[3]."</td>";
21         echo "<td $style>".$row[4]."</td>";
22         echo "</tr>";
23     }
24 }
```

接着通过 MySQL 语句 update users set status='flase' where id=1 修改 admin 用户的 status 为 0 进行测试。使用 select 查询语句查询 users 表中 admin 信息，status 已经变更为 0。

```
mysql> update users set status='flase' where id=1;
Query OK, 1 row affected, 1 warning (0.13 sec)
Rows matched: 1 Changed: 1 Warnings: 1

mysql> select * from users;
+----+-----+-----+-----+-----+
| id | username | passcode | email | status |
+----+-----+-----+-----+-----+
| 1 | admin | admin123 | admin@sys.com | 0 |
+----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

查看页面中 admin 用户对应的行呈现红色背景，测试成功！

欢迎您访问！

用户信息表

账号	密码	邮件	状态
admin	admin123	admin@sys.com	0

3. 如何收回 MySQL 数据库 firstuser 用户 create 创建表格的权限。

通过 REVOKE 语句收回权限。

输入 `revoke create on firstlab.* from 'firstuser'@'localhost';`

明确收回 firstuser 用户在 firstlab 数据库下创建表格（CREATE 权限）的权限。

-执行后提示 Query OK 表示语句生效。

然后用 `flush privileges` 命令刷新权限，让系统立即应用权限变更。

```
mysql> revoke create on firstlab.* from 'firstuser'@'localhost';
Query OK, 0 rows affected (0.02 sec)

mysql> flush privileges
-> ;
Query OK, 0 rows affected (0.14 sec)
```