**《课程综合实践Ⅱ》实验报告三**

**LAMP网站搭建及交易系统API访问**

**姓名\_\_\_ 沈云昊 专业 电子信息工程**

**学号 3220104126 联系方式 2766636730@qq.com**

## 一、实验目的

1. 在Ubuntu系统上配置Apache、MySQL、php环境。

2. 在Linux系统基础上，搭建一个自己的网站。

3. 借助curl命令行工具，获取交易系统API数据。

## 二、实验内容及要求

**准备工作：为了方便后续的学习和实践，建议同学们自带电脑，在自己电脑上进行实验内容，并完成实验报告。**

**实验步骤（仔细阅读，按照步骤完成实验）**

**（本次实验全程在ubuntu上进行操作，无其余静态资源包）**

### Apache、MySQL、php环境配置

LAMP指的是Linux、Apache、MySQL、php构成的软件环境，是目前最流行的web应用平台之一，具有资源丰富、轻量、快速开发等特点，为在Ubuntu系统内用这四大工具进行网页构造，我们必须配置相应的环境，步骤如下（以20.04Ubuntu系统为例）。

### 1.1 Apache环境配置

**Apache** 是LAMP架构最核心的 Web Server，开源、稳定、模块丰富是 Apache 的优势，同时也是负载php的最佳选择，在Ubuntu系统内，我们配置环境如下：

1. 打开控制台，输入命令更新软件源，后安装Apache

sudo apt update

sudo apt install apache2

1. 安装完成后，Apache会自动启动，输入命令，查看运行状态

sudo systemctl status apache2

如果没有自行启动，请输入命令手动启动服务

sudo systemctl start apache2

1. 启动完成后，需要打开80端口和443端口用于http和https通讯，输入以下命令：

sudo ufw allow 80

sudo ufw allow 443

1. 配置完成后，可以打开虚拟机内的浏览器，输入网址

http://localhost

查看虚拟机ip，在宿主机的浏览器内，输入网址

http://虚拟机ip

应能跳出相同的Apache2默认页

### 1.2 MySQL环境配置

**MySQL**是现在最流行的轻量级关系型数据库，适合用于小型网站应用的数据储存，在本次实验中，我们需要用该数据库存储网页请求到的交易系统API数据，我们配置环境如下：

1. 进入控制台，安装MySQL

sudo apt install mysql-server

1. 安装完成后，MySQL会自动启动，输入命令，查看运行状态

sudo systemctl status mysql

如果没有自启，需要手动输入命令启动服务，参照Apache环境配置的启动

服务命令即可。

1. 安装完成后，MySQL生成了默认的登录用户和密码，输入命令查看

sudo cat /etc/mysql/debian.cnf

1. 用默认账户登录MySQL，输入命令

mysql -u 你的默认用户名 -p

输入密码后，进入MySQL命令行，对数据库进行一些基本操作，输入命令

create database hw3; ##创建名为hw3的数据库

show databases; ## 查看数据库是否创建成功

use hw3; ## 使用hw3数据库

在hw3数据库中创建数据表，输入命令

create table record(name VARCHAR(25),height VARCHAR(25),hash VARCHAR(64), time VARCHAR(32));

我们创建了一个有4个字段的表，分别为name、height、hash、

time，接下来我们输入命令查看表内数据和表结构。

select \* from record; ##输出应为empty set

desc record;

其余sql命令的学习，请同学们自行查阅菜鸟教程(<https://www.runoob.com/mysql/mysql-tutorial.html>)，并自己在数据库内多打多试

### 1.3 php环境配置

**PHP** 是一种创建动态交互性站点的强有力的服务器端脚本语言，并且对MySQL提供了完美的原生支持，我们配置环境如下：

1. 进入控制台，安装php及对apache2和MySQL的扩展支持包。

sudo apt install php

sudo apt install libapache2-mod-php

sudo apt install php-mysql

为后续实验做准备，安装其余支持包

sudo apt install php-curl php-gd php-xml php-mbstring php-xmlrpc php-zip php-soap php-intl

安装完成后，重启apache2服务，使扩展支持包生效。

sudo service apache2 restart

### 简单的php网站搭建（请截图保存在实验记录中）

请同学们参照之前学过的Linux命令相关知识，完成以下几步：

1. 进入Apache2的网站根目录/var/www/html下，并查看该目录下包含的文件。
2. 在根目录下，新建一个文件，命名为phpinfo.php作为新的网页。
3. 对该文件进行内容编辑，输入以下内容：

<?php

phpinfo();

?>

1. 打开虚拟机内的浏览器，输入网址：

http://localhost/phpinfo.php

打开宿主机的浏览器，输入网址：

http://你的虚拟机IP/phpinfo.php

应该可以看到php在你虚拟机里的配置页面，

随后请同学们查阅菜鸟教程（https://www.runoob.com/php/php-tutorial.html），学习php的相关知识，并在此基础上，写一个简单的网页，**保存页面效果截图至实验记录中，保存php源文件随实验报告一起上交。**

### 交易系统API访问及数据库操作（请截图保存在实验记录中）

### 3.1 curl工具安装

curl是常用的命令行工具，用于请求Web服务器，支持的协议包括DICT、FILE、FTP、HTTP、HTTPS等十余种，实验中，我们将使用curl访问某交易系统提供的API接口并获取其返回结果。curl安装配置过程如下：

1. 打开终端，输入命令，安装curl

sudo apt install curl

1. 安装完成后，输入比特币链API进行测试，输入命令

curl https://api.blockcypher.com/v1/btc/main

结果应该能返回区块高度、时间、哈希值等相关信息。(考虑到该API为国外网站提供，有概率被墙，在不同的时间段多试几次即可)

### 3.2 API数据网页回显

在3.1中，我们成功的在控制台中输出了交易系统API返回的信息，现在我们要把这些信息以表格的形式显示在网页上，新建一个网页可参考步骤如下：

1. 进入/var/www/html，新建一个名为curlbtc.php的文件。
2. 在文件中输入以下内容：

<?php

$loginUrl = 'https://api.blockcypher.com/v1/btc/main';

$ch = curl\_init(); //初始化一个cURL会话

curl\_setopt($ch, CURLOPT\_SSL\_VERIFYPEER, false);

curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, true);

curl\_setopt($ch, CURLOPT\_URL,$loginUrl); //为cURL会话设置选项

$result=curl\_exec($ch); //获得cURL会话返回结果

curl\_close($ch);

echo "After running the API url: ".$loginUrl. "<br>";

echo "The result of JSON data:<hr>";

var\_dump($result); //将变量result输出到页面上

echo "<br><br>";

echo "<hr>";

echo "To JSON object data:<hr>";

$str=json\_decode($result,true); //对JSON格式的字符串进行编码

var\_dump($str);

echo "<br><br>";

echo "<hr>";

echo "<table border=1>"; //开始构建表格

echo "<tr><td>Field</td><td></td></tr>";

$tm=$str["name"];

echo "<tr><td>Name:</td><td>".$tm."</td></tr>";

echo "<tr><td>height:</td><td>".$str["height"]."</td></tr>";

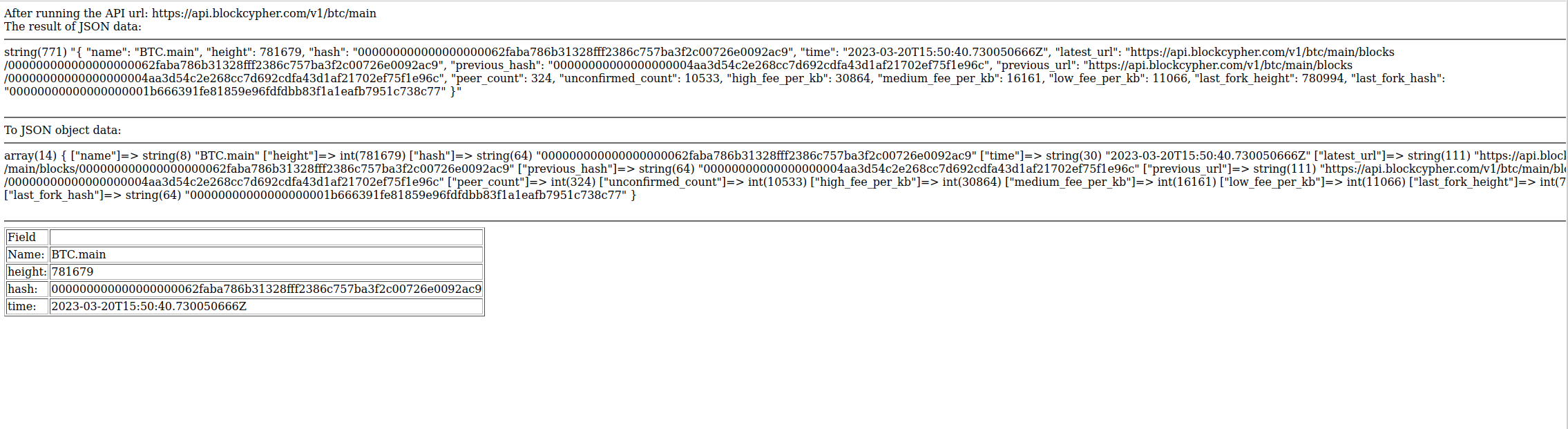
echo "<tr><td>hash:</td><td>".$str["hash"]."</td></tr>";

echo "<tr><td>time:</td><td>".$str["time"]."</td></tr>";

echo "</table>"

?>

保存后，在虚拟机浏览器和宿主机浏览器中输入对应的url，应该能看到页面如下：



参考页面效果和同学们自学的php知识，应该能大致看懂这段代码的含义。

### 3.3 数据库存储

我们已经从交易系统的API获取到了返回信息，现在我们需要把这部分在页面上显示的信息保存在本地的MySQL数据库中，具体操作如下，

在我们之前编辑的curlbtc.php文件内追加部分代码，追加位置放在语句

$str=json\_decode($result,true);

之后。

追加代码如下：

$servername = "localhost";

$username = " "; //填入你的数据库用户名

$password = " "; //填入你的数据库密码

$dbname = "hw3"; //填入要使用的数据库名称

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

if ($conn->connect\_error) {

die("连接失败: " . $conn->connect\_error);

}

$sql = "INSERT INTO record (name, height, hash, time) VALUES ('$str[name]','$str[height]', '$str[hash]', '$str[time]')";

echo "<br><br>";

if ($conn->query($sql) === TRUE) {

echo "新记录插入成功";

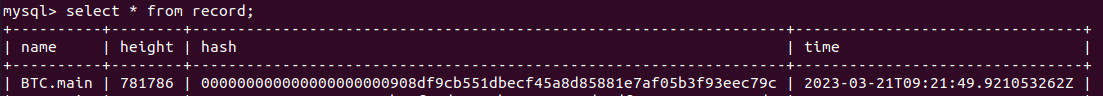
} else {

echo "Error: " . $sql . "<br>" . $conn->error;

}

 重新在浏览器内打开curlbtc.php页面，如果能看到

即为插入数据库数据成功，我们可以在控制台中打开数据库查看数据进行验证，



下面请同学们完成以下任务，

由于区块链的出块速度并不是很快，导致我们每次打开curlbtc.php网页时可能会有重复的块写入数据库，导致数据库的数据冗余，为避免这一情况的发生，请同学们在原来的php文件基础上增加数据库内是否已经有对应记录的判断，并保证只有在没有记录的情况下，才将信息写入，**保存php源文件随实验报告一起上交。**

**作业上交内容与事项：**

1．实验报告文档：按照要求完成实验并将关键步骤实验结果进行截图记录。注意文档工整；

2．程序代码：如果有程序代码或其他相关材料，可汇总压缩所有文件并上传压缩包。注意文件命名；

**本期实验作业上交期限：**

请在实验设置的截止日期内提交实验报告，若逾期提交，成绩会被适当打折。

**本次作业上交内容：**

* 实验报告文档(.docx)
* 两个源代码文件(.php)

## 三、实验感受及记录

* 1. **实验感受（本次实验遇到的问题、主要收获等内容）**

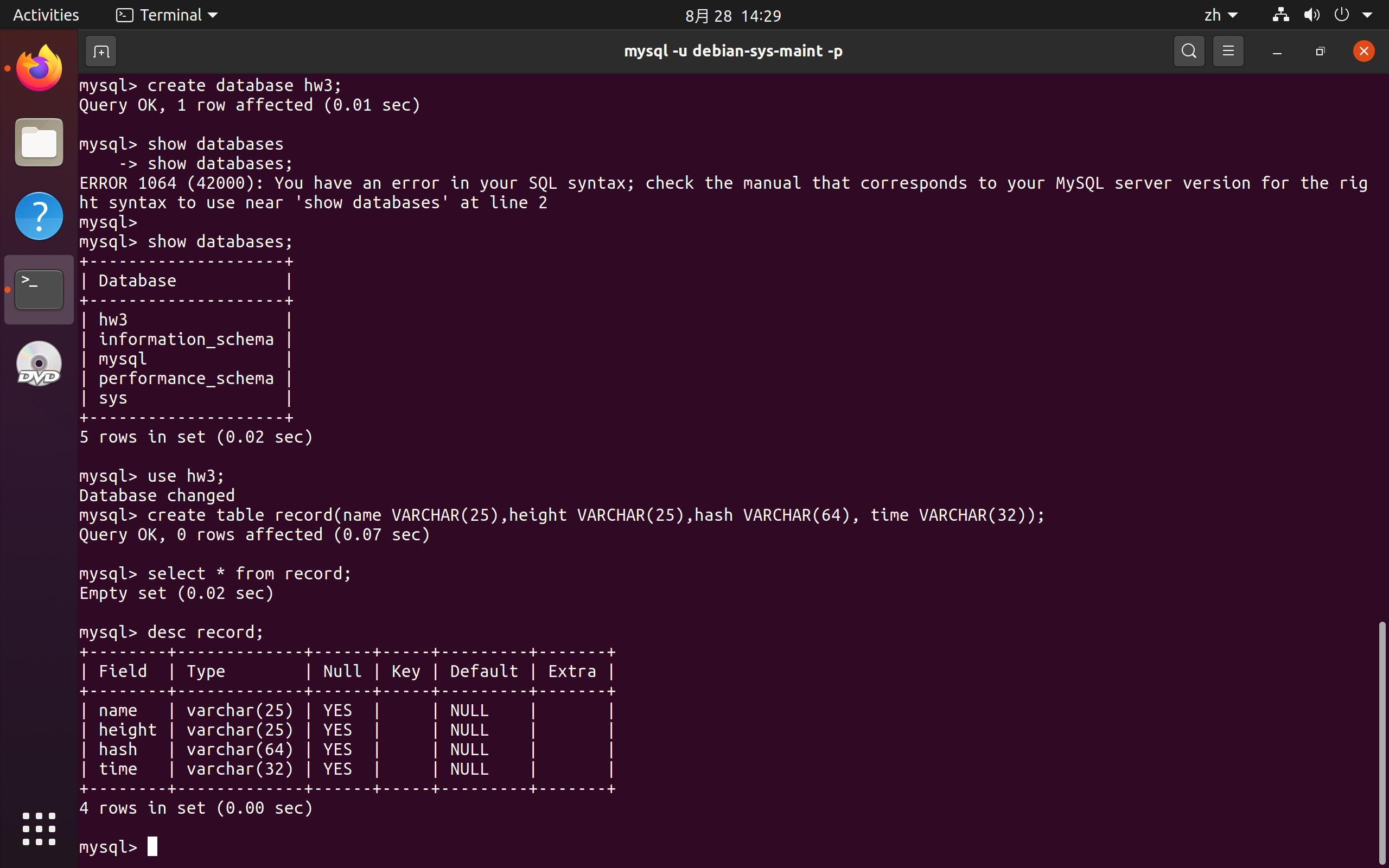
通过本次实验，我对LAMP架构的整体理解得到了提升。通过亲手搭建网站，我对各个组件（Apache、MySQL、PHP）之间的协同工作有了更直观的认识。特别是在配置和调试的过程中，我逐渐熟悉了如何让这些组件高效地配合，理解了它们各自的功能与相互依赖关系。这种实操经验让我对LAMP架构的内在逻辑有了更深的理解，也为以后处理类似任务打下了坚实的基础。

在实验中，我学习了如何配置Apache服务器，尤其是在处理虚拟主机时，学会了如何根据实际需求设置多个站点的运行环境。在数据库操作方面，我通过使用PHP与MySQL进行交互，巩固了数据库操作的基本技能。

同时，我还学会了如何通过API访问与数据库交互，实现了基础的交易系统功能。这让我对API的设计与实现有了更深入的理解，也让我认识到，API不仅仅是前后端交互的桥梁，更是系统安全性和稳定性的关键。

* 1. **实验记录（实验过程中关键步骤截图记录及文字描述）**

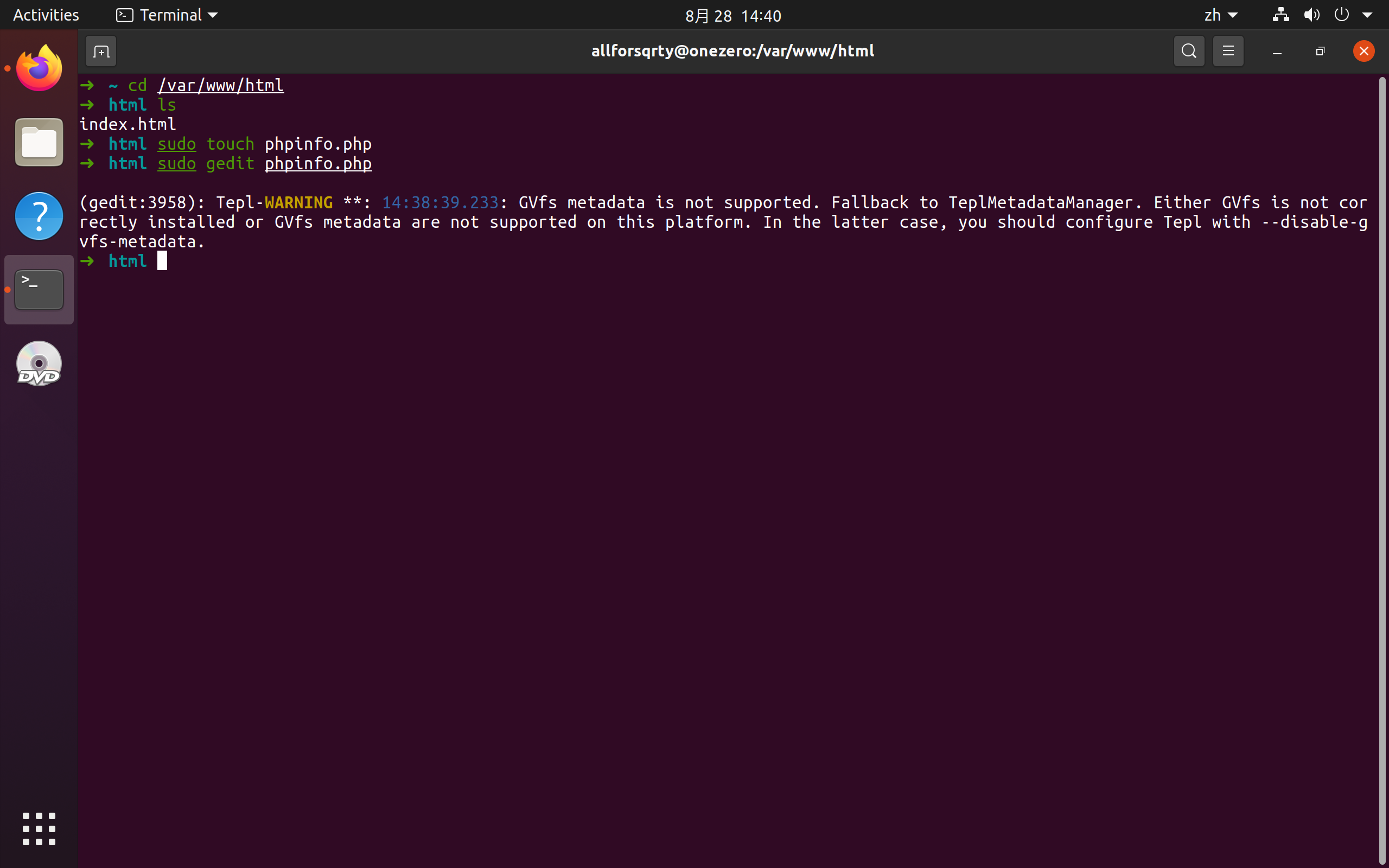
1. **环境配置**



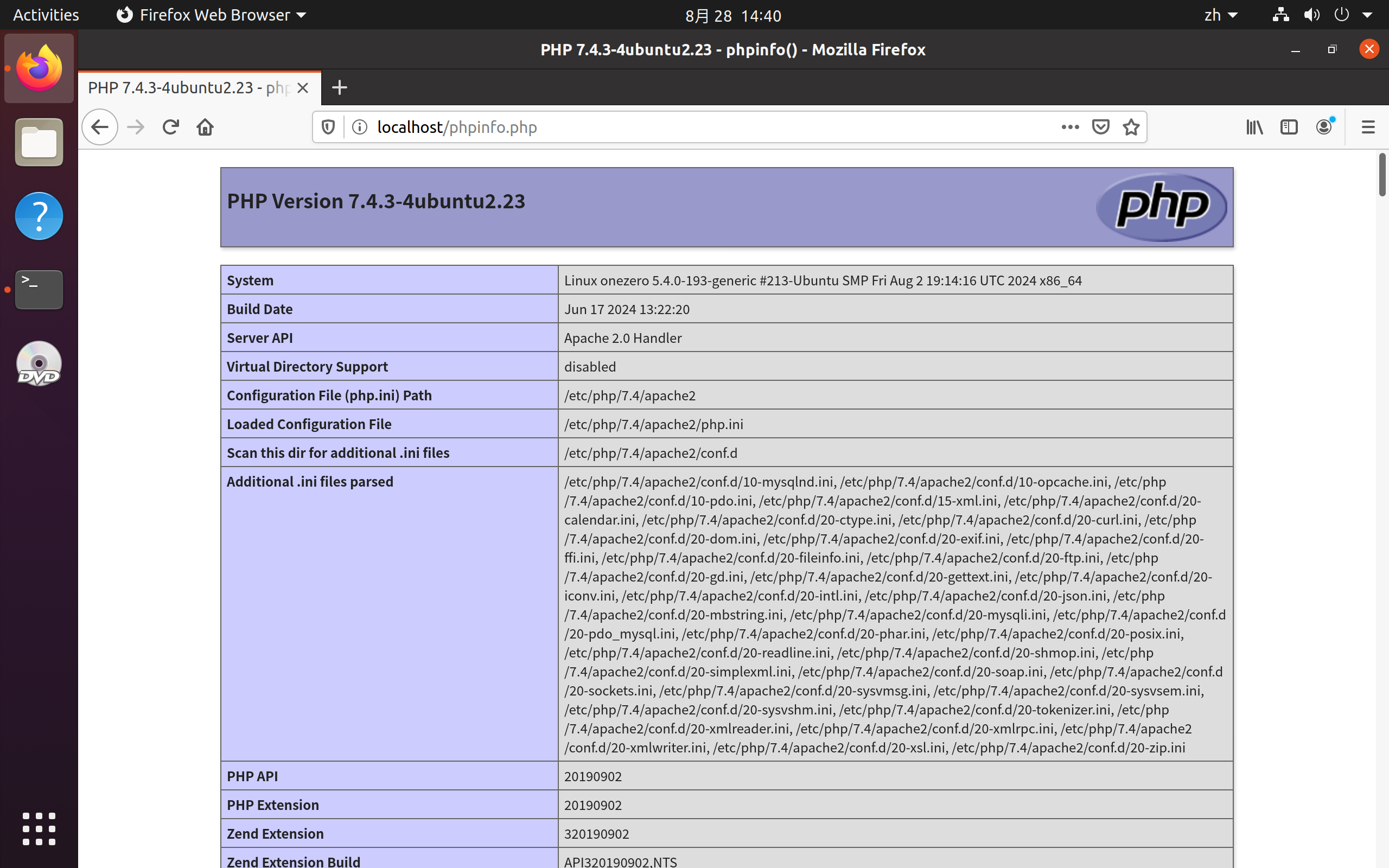
上图为创建的数据库和数据表。

1. **简单的php网站搭建**

（1）示例



通过上图命令，添加phpinfo.php文件，文件中执行phpinfo()命令，在浏览器中打开后结果如下。



输出了php文件的环境信息。

1. 搭建网页

通过下面的php及html代码，我搭建了一个简易的网页。该网页的功能是四则运算的计算，输入两个数并选择运算符，可以计算出结果。

**代码：**

<!DOCTYPE html>

<body>

<h1>PHP Calculator</h1>

<form method="post" action="">

<input type="number" name="num1">

<select name="oper">

<option value="add">+</option>

<option value="sub">-</option>

<option value="mul">\*</option>

<option value="div">/</option>

</select>

<input type="number" name="num2">

<input type="submit" value="Calculate">

</form>

<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$num1 = $\_POST['num1'];

$num2 = $\_POST['num2'];

$oper = $\_POST['oper'];

$result = 0;

switch ($oper) {

case "add":

$result = $num1 + $num2;

break;

case "sub":

$result = $num1 - $num2;

break;

case "mul":

$result = $num1 \* $num2;

break;

case "div":

if ($num2 != 0) {

$result = $num1 / $num2;

} else {

$result = "Error: Division by zero";

}

break;

}

echo "result = $result";

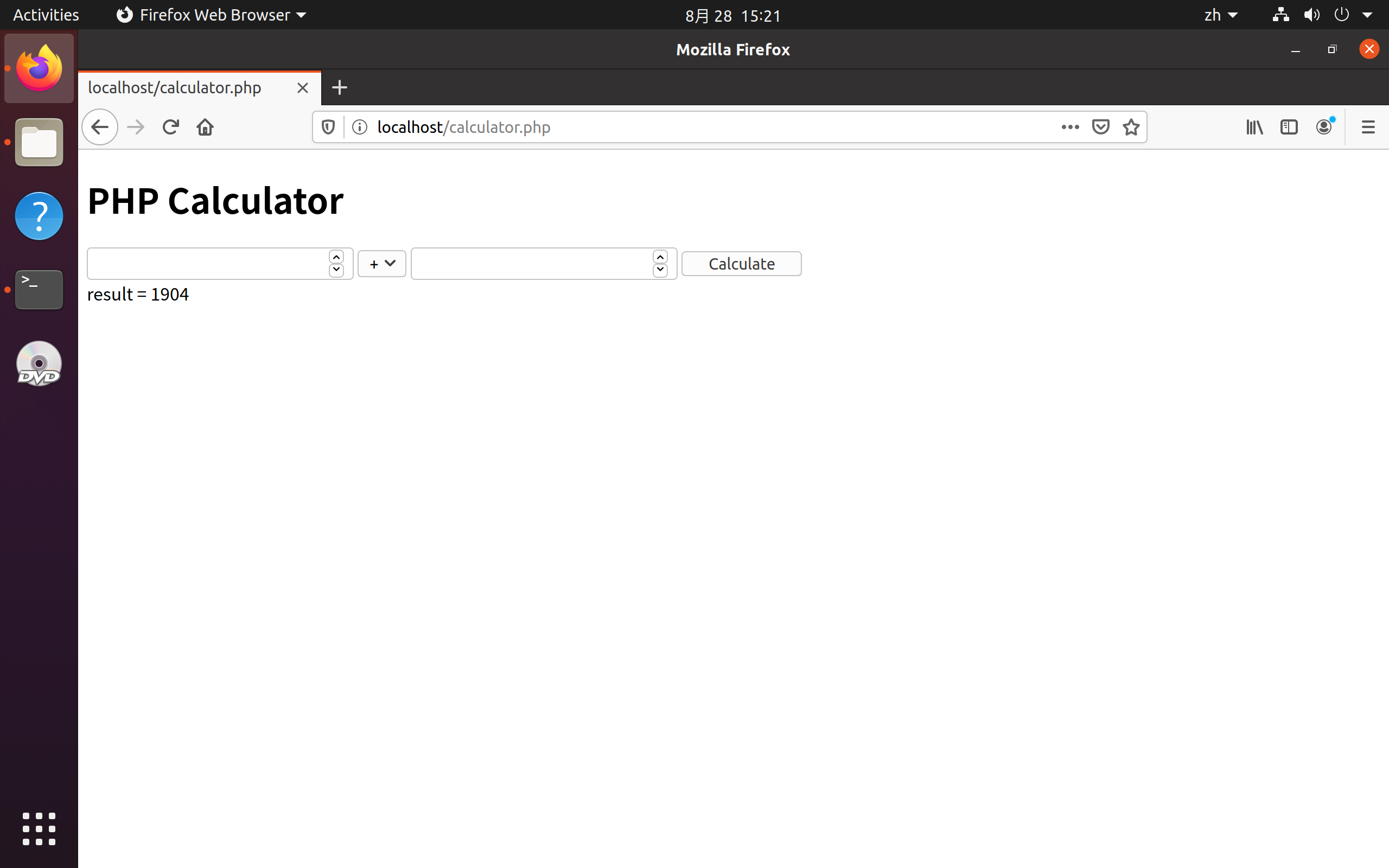
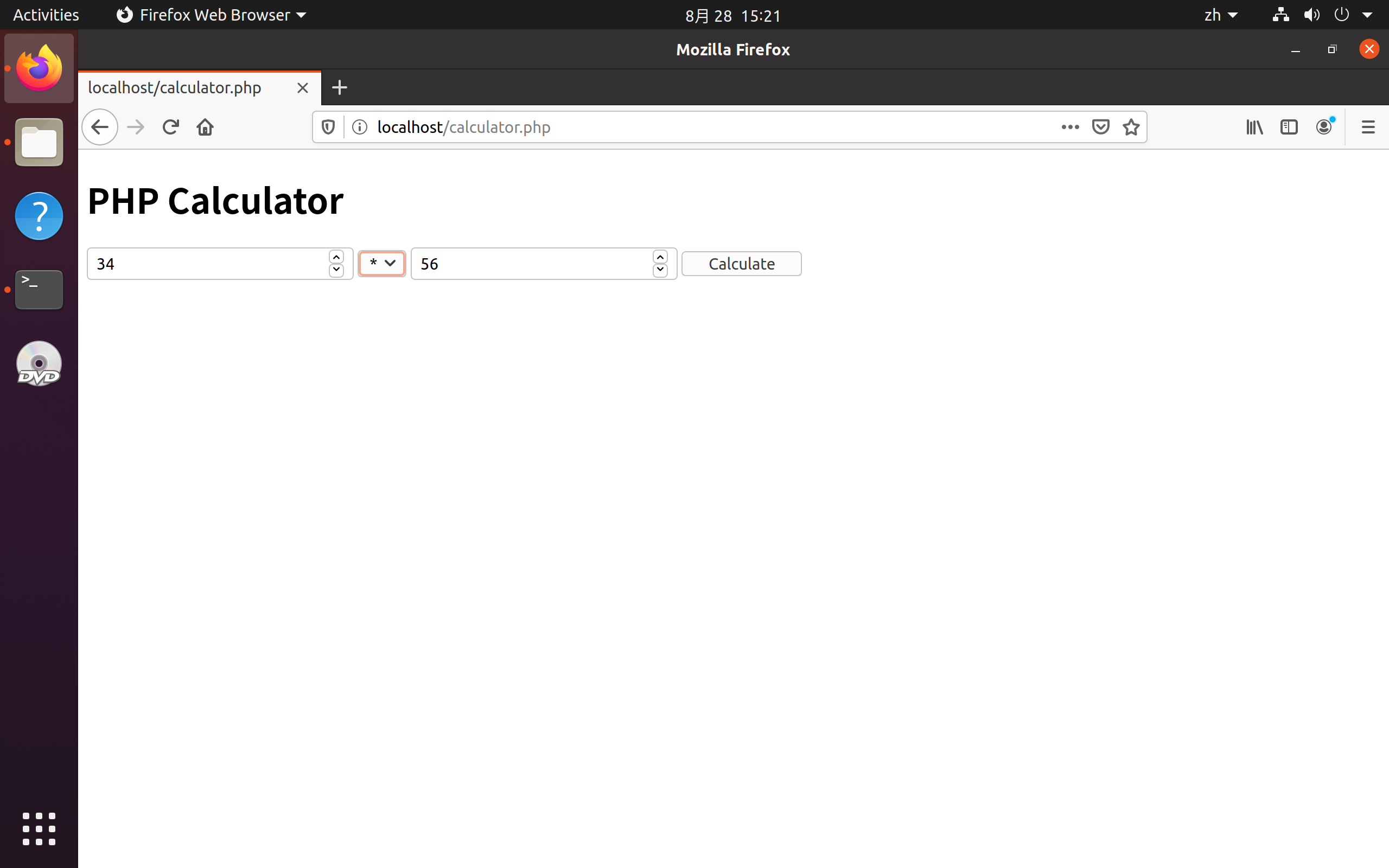
}

?>

</body>

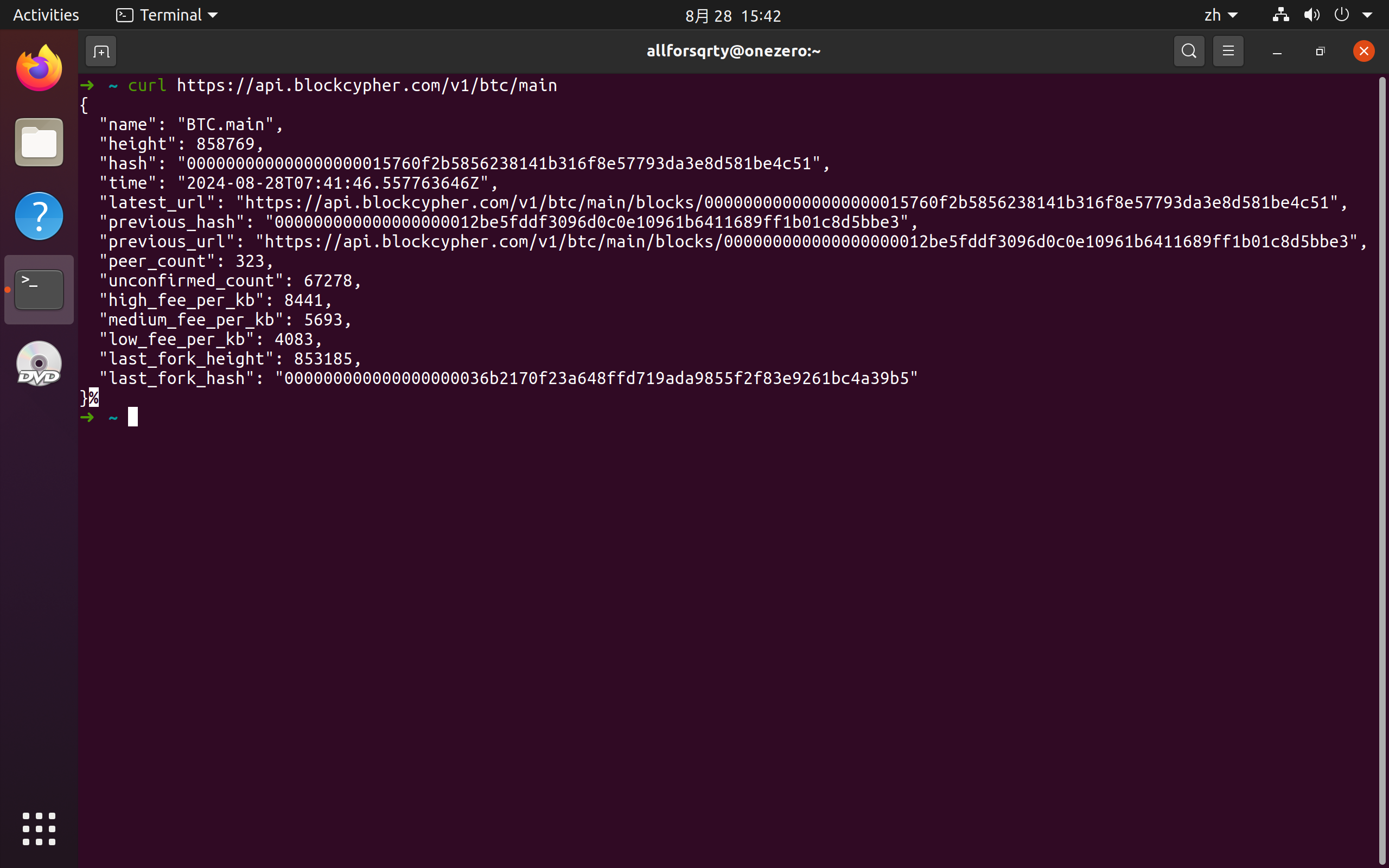
</html>

**效果：**



1. **交易系统API访问及数据库操作**
2. 测试

如下图，输入比特币链API进行测试，返回比特币的区块高度、时间、哈希值等相关信息



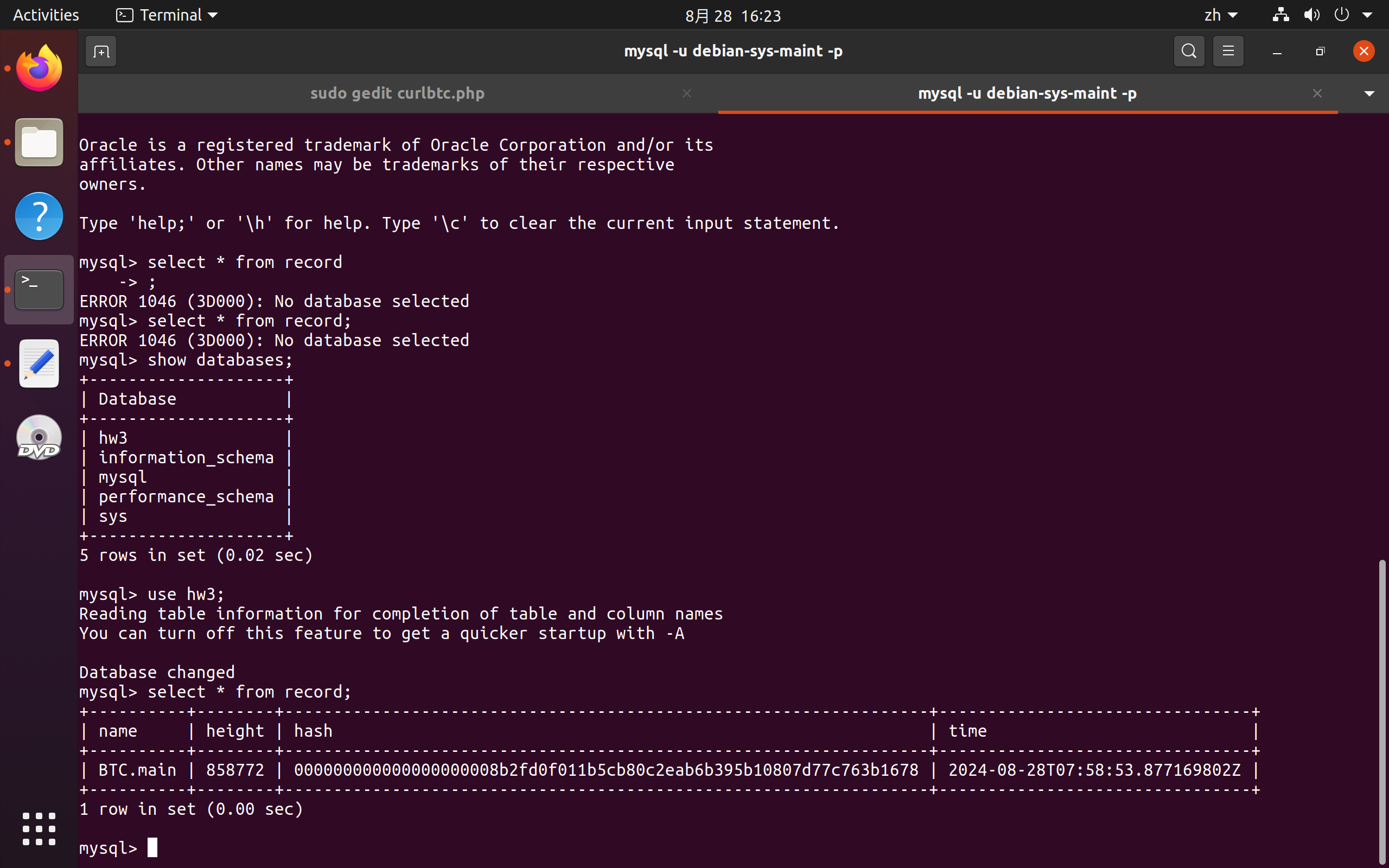
1. API数据网页回显

通过输入的php代码，浏览器中的网页显示如下。该php文件首先请求curl结果返回并打印在网站上，之后通过json解码为数组的形式。最后制成表格。



1. 数据库存储

通过添加文档中的php代码，可以将网页内容存到数据库中，如下图。



通过添加下面的代码，判断记录是否已经存在，若存在，则输出信息，并不添加到数据库中。

**代码：**

$hash = $str['hash'];

$checkrecord = "SELECT \* FROM record WHERE hash = '$hash'";

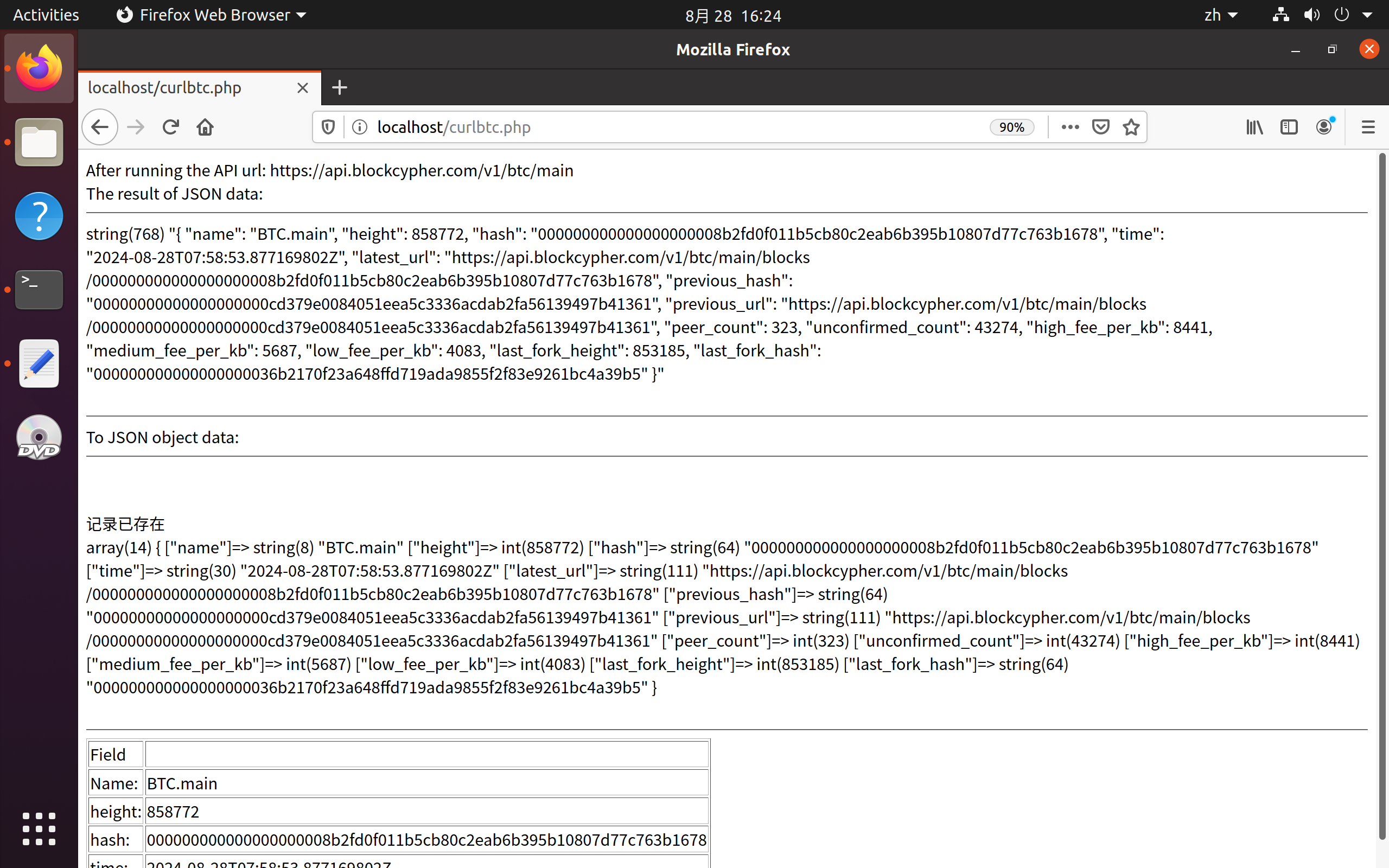
$exist = $conn->query($checkrecord);

if($exist->num\_rows > 0){

echo "记录已存在";

}

**结果：**



同时，若记录不存在，则添加该记录至数据库中，结果如下。

