ORCHID INTERNATIONAL College

(Affiliated to Tribhuvan University)

Lab: 03

Date:

# Title: A lab report on LAMP stack on Ubuntu

Submitted By: Ankit Rai

TU Roll no: 23862/076

B.Sc. CSIT 8th Sem

Submitted To: Department of CSIT

Signature of Faculty:

## Lab 3: Linux, Apache, MySQL, PHP (LAMP) Stack on Ubuntu

### Objectives:

- To understand the components of the LAMP stack and their roles.
- To install and configure Linux, Apache, MySQL, and PHP.

### Theory:

Introduction:

A “LAMP” stack is a group of open-source software that is typically installed together in order to enable a server to host dynamic websites and web apps written in PHP. This term is an acronym which represents the Linux operating system with the Apache web server. The site data is stored in a MySQL database, and dynamic content is processed by PHP.

- Linux: The operating system (OS) that forms the foundation of the stack. Linux provides the core functionalities and services needed to run web applications.
- Apache: The web server software that handles HTTP requests from clients (such as web browsers) and serves web pages and other content over the internet. Apache is highly customizable and widely used due to its stability and performance.
- MySQL: A relational database management system (RDBMS) that stores and manages structured data. MySQL is used for storing and retrieving data in web applications, making it an essential component of the stack for handling dynamic content.
- PHP: A server-side scripting language used for developing dynamic web pages and web applications. PHP code is executed on the server, generating HTML content that is then sent to the client's web browser for display.

Setting up LAMP Stack on Ubuntu

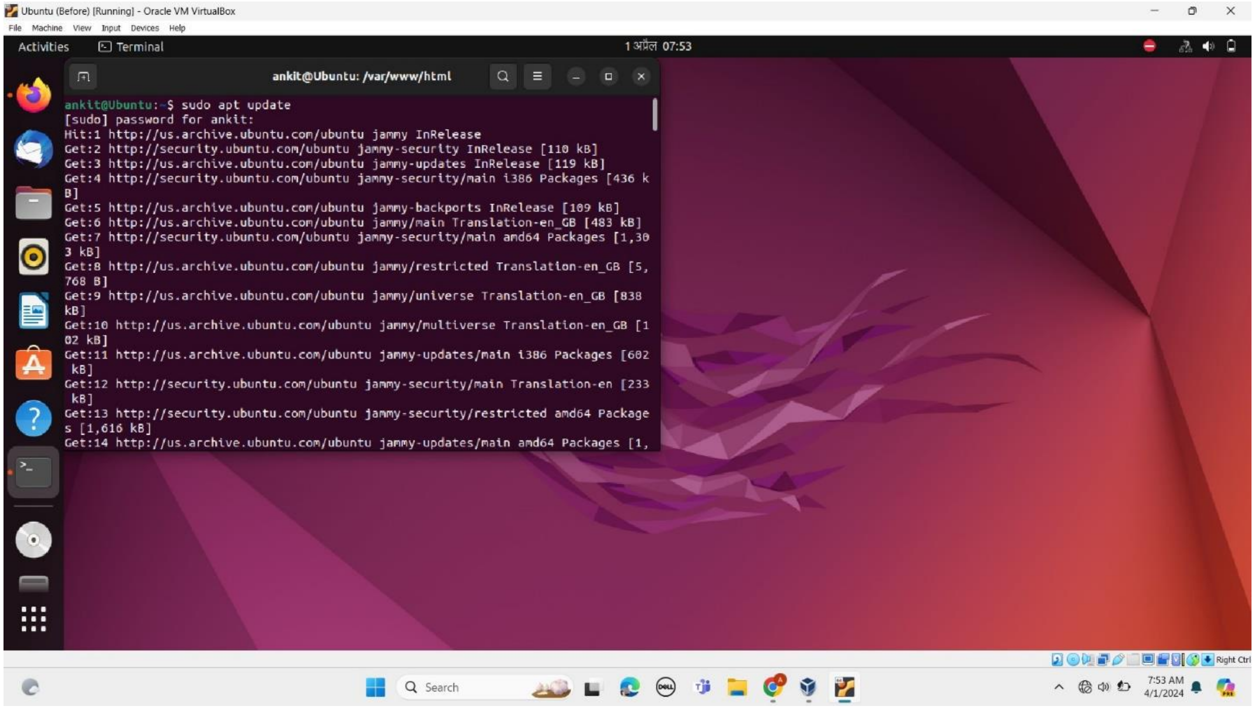
1. Install Apache & Update Firewall
2. Install MySQL Database
3. Install PHP
4. Setup Virtual Webhost
5. Test PHP Processing

6. Test Database Connection

Exploring each step as,

Step 1 — Installing Apache and Updating the Firewall

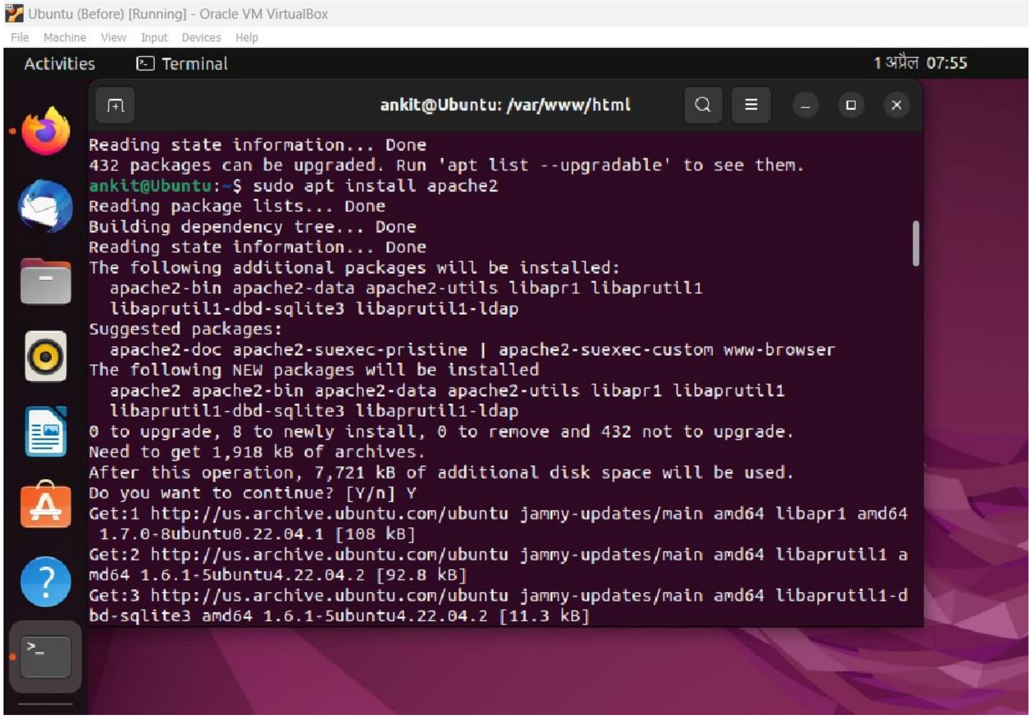
First update the package manager syntax: sudo apt update



Ubuntu (Before) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
1 अप्रैल 07:53
ankit@Ubuntu: /var/www/html

```
ankit@Ubuntu:~$ sudo apt update
[sudo] password for ankit:
Hit:1 http://us.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security/main i386 Packages [436 k
B]
Get:5 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu jammy/main Translation-en_GB [483 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1,30
3 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu jammy/restricted Translation-en_GB [5,
768 B]
Get:9 http://us.archive.ubuntu.com/ubuntu jammy/universe Translation-en_GB [838
kB]
Get:10 http://us.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en_GB [1
02 kB]
Get:11 http://us.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [602
 kB]
Get:12 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [233
 kB]
Get:13 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Package
s [1,616 kB]
Get:14 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1,
```

Then, install Apache with: sudo apt install apache2

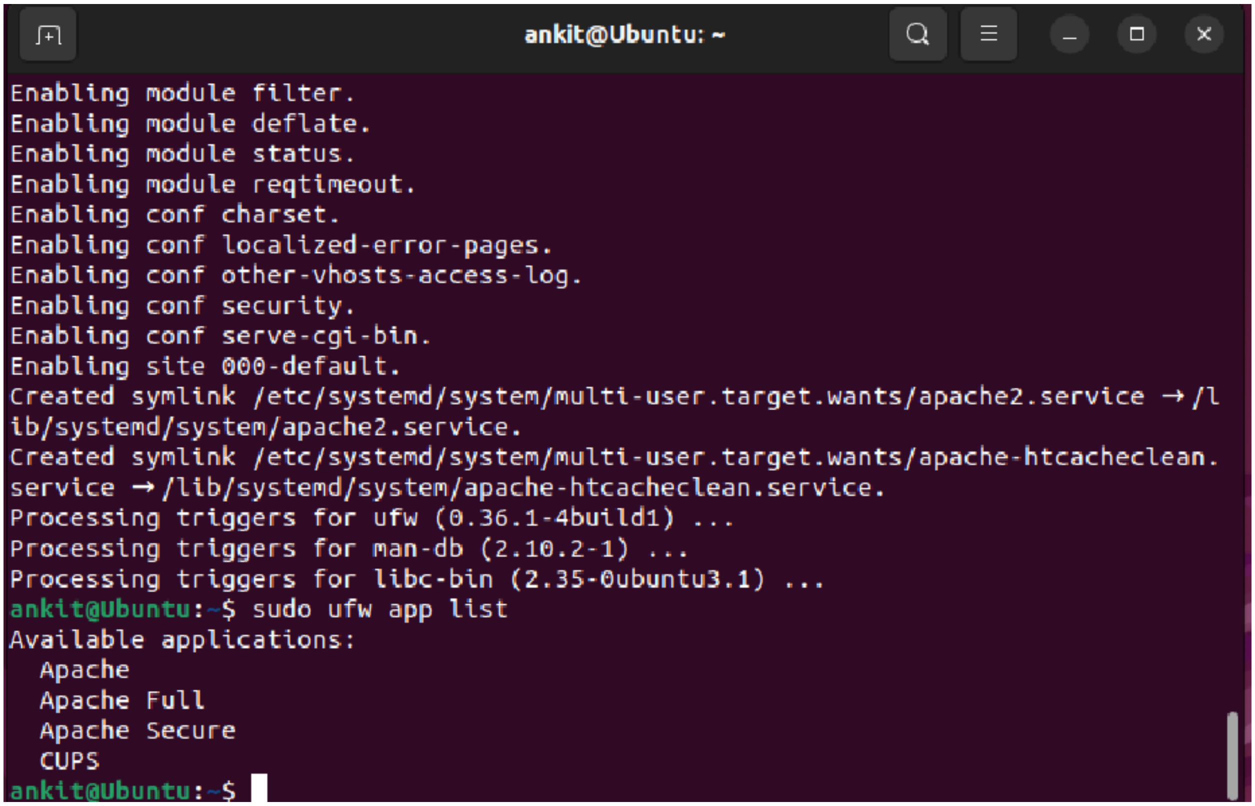


Ubuntu (Before) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
1 अप्रैल 07:55
ankit@Ubuntu: /var/www/html

```
Reading state information... Done
432 packages can be upgraded. Run 'apt list --upgradable' to see them.
ankit@Ubuntu:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap
0 to upgrade, 8 to newly install, 0 to remove and 432 not to upgrade.
Need to get 1,918 kB of archives.
After this operation, 7,721 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libapr1 amd64
 1.7.0-8ubuntu0.22.04.1 [108 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1 a
md64 1.6.1-5ubuntu4.22.04.2 [92.8 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-d
bd-sqlite3 amd64 1.6.1-5ubuntu4.22.04.2 [11.3 kB]
```

Once the installation is finished, you'll need to adjust your firewall settings to allow HTTP traffic. Ubuntu's default firewall configuration tool is called Uncomplicated Firewall (UFW). It has different application profiles that you can leverage. To list all currently available UFW application profiles, execute this command:

-$ sudo ufw app list

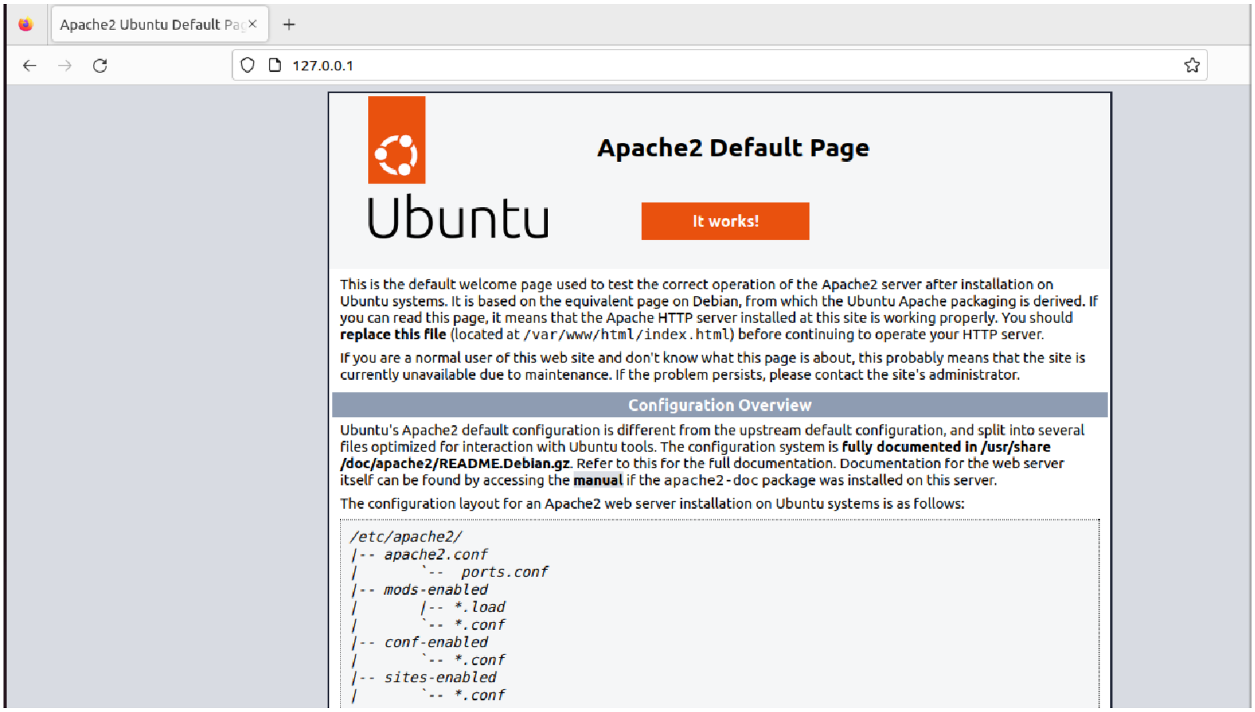


```
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /l
ib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.
service → /lib/systemd/system/apache-htcacheclean.service.
Processing triggers for ufw (0.36.1-4build1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
ankit@Ubuntu:~$ sudo ufw app list
Available applications:
  Apache
  Apache Full
  Apache Secure
  CUPS
ankit@Ubuntu:~$
```

Here's what each of these profiles mean:

- Apache: This profile opens only port 80 (normal, unencrypted web traffic).
- Apache Full: This profile opens both port 80 (normal, unencrypted web traffic) and port 443 (TLS/SSL encrypted traffic).
- Apache Secure: This profile opens only port 443 (TLS/SSL encrypted traffic).

The default Ubuntu Apache web page



**Apache2 Default Page**

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

**Configuration Overview**

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share /doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

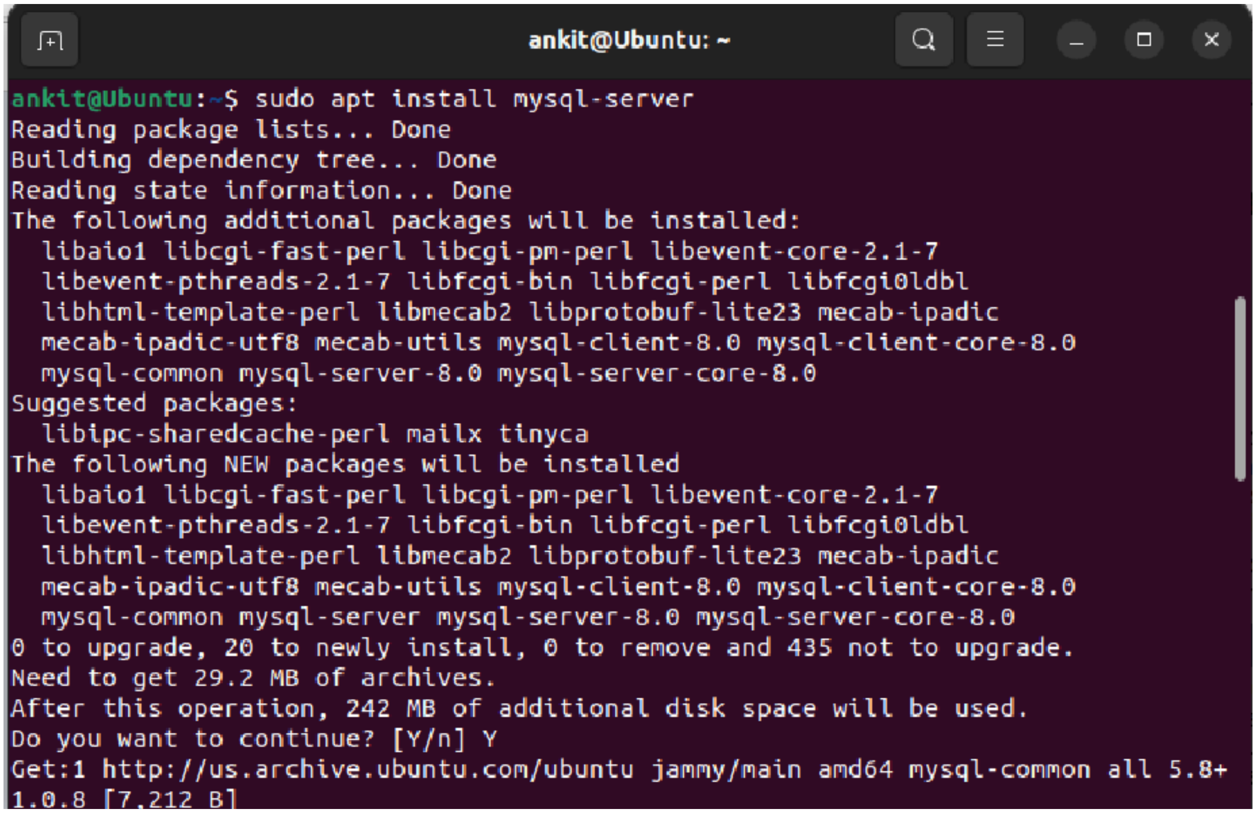
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

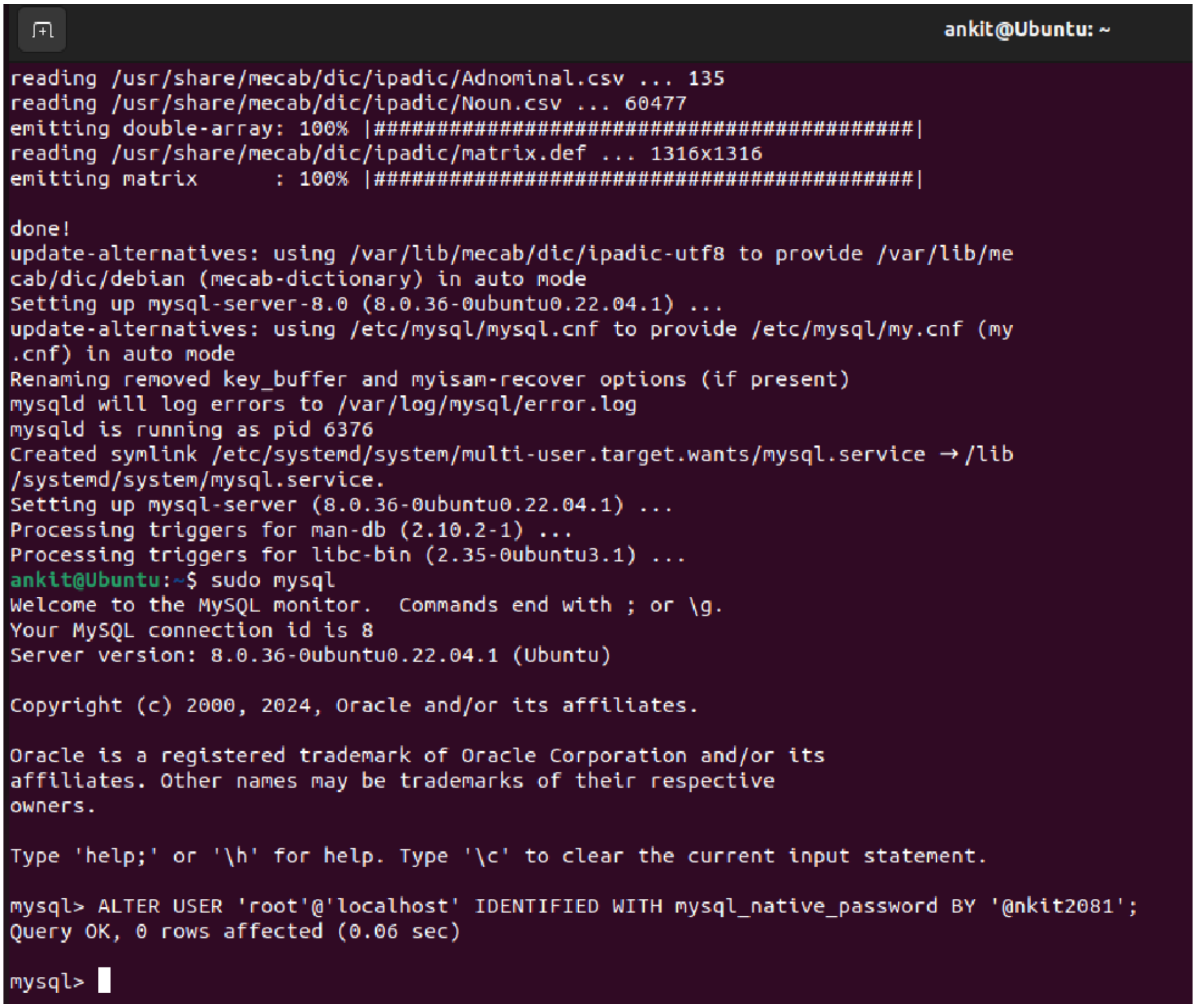
Step 2 — Installing MySQL

Again, use apt to acquire and install this software:

Command: sudo apt install mysql-server



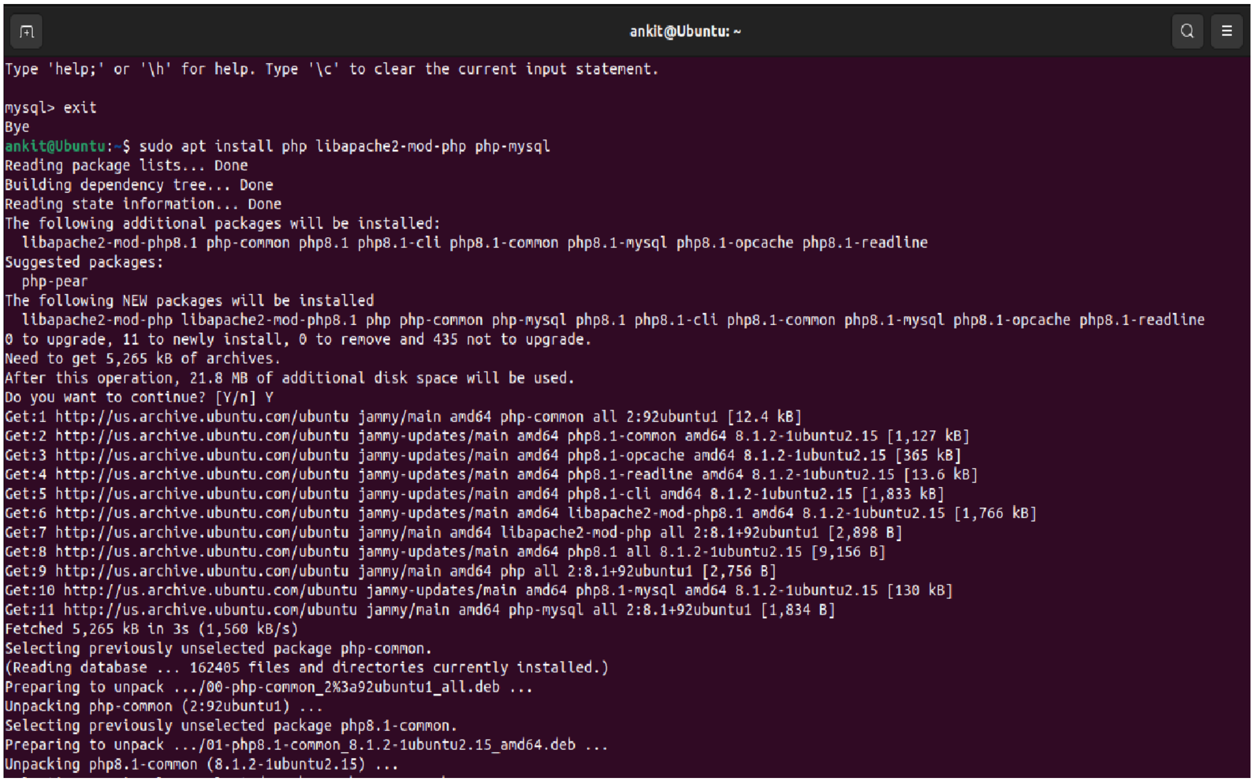
```
ankit@Ubuntu:~$ sudo apt install mysql-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libevent-core-2.1-7
  libevent-pthreads-2.1-7 libfcgi-bin libfcgi-perl libfcgi0ldbl
  libhtml-template-perl libmecab2 libprotobuf-lite23 mecab-ipadic
  mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0
  mysql-common mysql-server-8.0 mysql-server-core-8.0
Suggested packages:
  libipc-sharedcache-perl mailx tinyca
The following NEW packages will be installed
  libaio1 libcgi-fast-perl libcgi-pm-perl libevent-core-2.1-7
  libevent-pthreads-2.1-7 libfcgi-bin libfcgi-perl libfcgi0ldbl
  libhtml-template-perl libmecab2 libprotobuf-lite23 mecab-ipadic
  mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0
  mysql-common mysql-server mysql-server-8.0 mysql-server-core-8.0
0 to upgrade, 20 to newly install, 0 to remove and 435 not to upgrade.
Need to get 29.2 MB of archives.
After this operation, 242 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 mysql-common all 5.8+
1.0.8 [7,212 B]
```



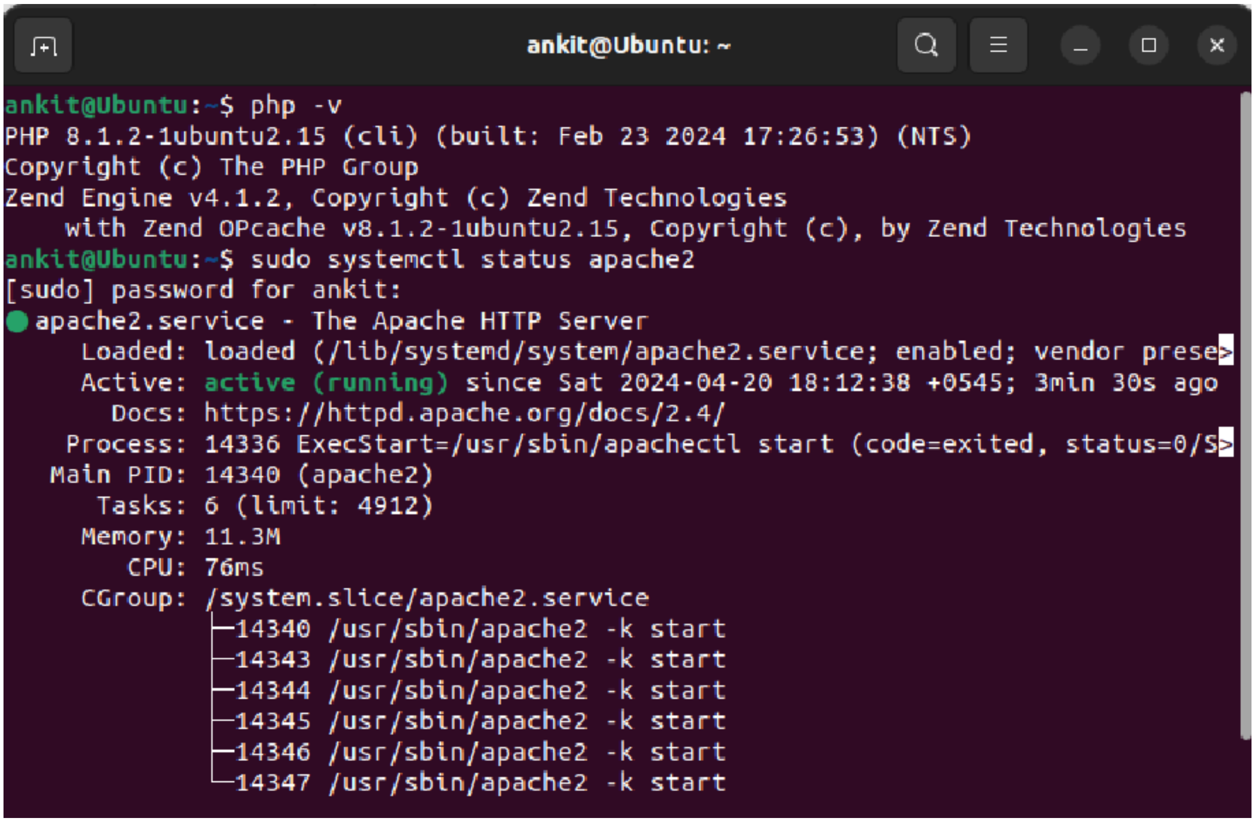
```
reading /usr/share/mecab/dic/ipadic/Adnominal.csv ... 135
reading /usr/share/mecab/dic/ipadic/Noun.csv ... 60477
emitting double-array: 100% |###########################################|
reading /usr/share/mecab/dic/ipadic/matrix.def ... 1316x1316
emitting matrix      : 100% |###########################################|

done!
update-alternatives: using /var/lib/mecab/dic/ipadic-utf8 to provide /var/lib/me
cab/dic/debian (mecab-dictionary) in auto mode
Setting up mysql-server-8.0 (8.0.36-0ubuntu0.22.04.1) ...
update-alternatives: using /etc/mysql/mysql.cnf to provide /etc/mysql/my.cnf (my
.cnf) in auto mode
Renaming removed key_buffer and myisam-recover options (if present)
mysqld will log errors to /var/log/mysql/error.log
mysqld is running as pid 6376
Created symlink /etc/systemd/system/multi-user.target.wants/mysql.service → /lib
/systemd/system/mysql.service.
Setting up mysql-server (8.0.36-0ubuntu0.22.04.1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
ankit@Ubuntu:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.36-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

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> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '@nkit2081';
Query OK, 0 rows affected (0.06 sec)

mysql> 
```

## Step 3 — Installing PHP

Command: sudo apt install php libapache2-mod-php php-mysql



```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> exit
Bye
ankit@Ubuntu:~$ sudo apt install php libapache2-mod-php php-mysql
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libapache2-mod-php8.1 php-common php8.1 php8.1-cli php8.1-common php8.1-mysql php8.1-opcache php8.1-readline
Suggested packages:
  php-pear
The following NEW packages will be installed
  libapache2-mod-php libapache2-mod-php8.1 php php-common php-mysql php8.1 php8.1-cli php8.1-common php8.1-mysql php8.1-opcache php8.1-readline
0 to upgrade, 11 to newly install, 0 to remove and 435 not to upgrade.
Need to get 5,265 kB of archives.
After this operation, 21.8 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 php-common all 2:92ubuntu1 [12.4 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-common amd64 8.1.2-1ubuntu2.15 [1,127 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-opcache amd64 8.1.2-1ubuntu2.15 [365 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-readline amd64 8.1.2-1ubuntu2.15 [13.6 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-cli amd64 8.1.2-1ubuntu2.15 [1,833 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libapache2-mod-php8.1 amd64 8.1.2-1ubuntu2.15 [1,766 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libapache2-mod-php all 2:8.1+92ubuntu1 [2,898 B]
Get:8 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1 all 8.1.2-1ubuntu2.15 [9,156 B]
Get:9 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 php all 2:8.1+92ubuntu1 [2,756 B]
Get:10 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-mysql amd64 8.1.2-1ubuntu2.15 [130 kB]
Get:11 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 php-mysql all 2:8.1+92ubuntu1 [1,834 B]
Fetched 5,265 kB in 3s (1,560 kB/s)
Selecting previously unselected package php-common.
(Reading database ... 162405 files and directories currently installed.)
Preparing to unpack .../00-php-common_2%3a92ubuntu1_all.deb ...
Unpacking php-common (2:92ubuntu1) ...
Selecting previously unselected package php8.1-common.
Preparing to unpack .../01-php8.1-common_8.1.2-1ubuntu2.15_amd64.deb ...
Unpacking php8.1-common (8.1.2-1ubuntu2.15) ...
```



```
ankit@Ubuntu:~$ php -v
PHP 8.1.2-1ubuntu2.15 (cli) (built: Feb 23 2024 17:26:53) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
    with Zend OPcache v8.1.2-1ubuntu2.15, Copyright (c), by Zend Technologies
ankit@Ubuntu:~$ sudo systemctl status apache2
[sudo] password for ankit:
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese>
     Active: active (running) since Sat 2024-04-20 18:12:38 +0545; 3min 30s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 14336 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/S>
   Main PID: 14340 (apache2)
      Tasks: 6 (limit: 4912)
     Memory: 11.3M
        CPU: 76ms
     CGroup: /system.slice/apache2.service
             ├─14340 /usr/sbin/apache2 -k start
             ├─14343 /usr/sbin/apache2 -k start
             ├─14344 /usr/sbin/apache2 -k start
             ├─14345 /usr/sbin/apache2 -k start
             ├─14346 /usr/sbin/apache2 -k start
             └─14347 /usr/sbin/apache2 -k start
```

Step 4 — Creating a Virtual Host for your Website

When using the Apache web server, you can create virtual hosts (similar to server blocks in Nginx) to encapsulate configuration details and host more than one domain from a single server. In this guide, we’ll set up a domain called your_domain, but you should replace this with your own domain name.

Apache on Ubuntu has one virtual host enabled by default that is configured to serve documents from the /var/www/html directory. While this works well for a single site, it can become unwieldy if you are hosting multiple sites. Instead of modifying /var/www/html, we’ll create a directory structure within /var/www for the your_domain site, leaving /var/www/html in place as the default directory to be served if a client request doesn’t match any other sites.

Create the directory for your_domain as follows:

-$ sudo mkdir /var/www/your_domain

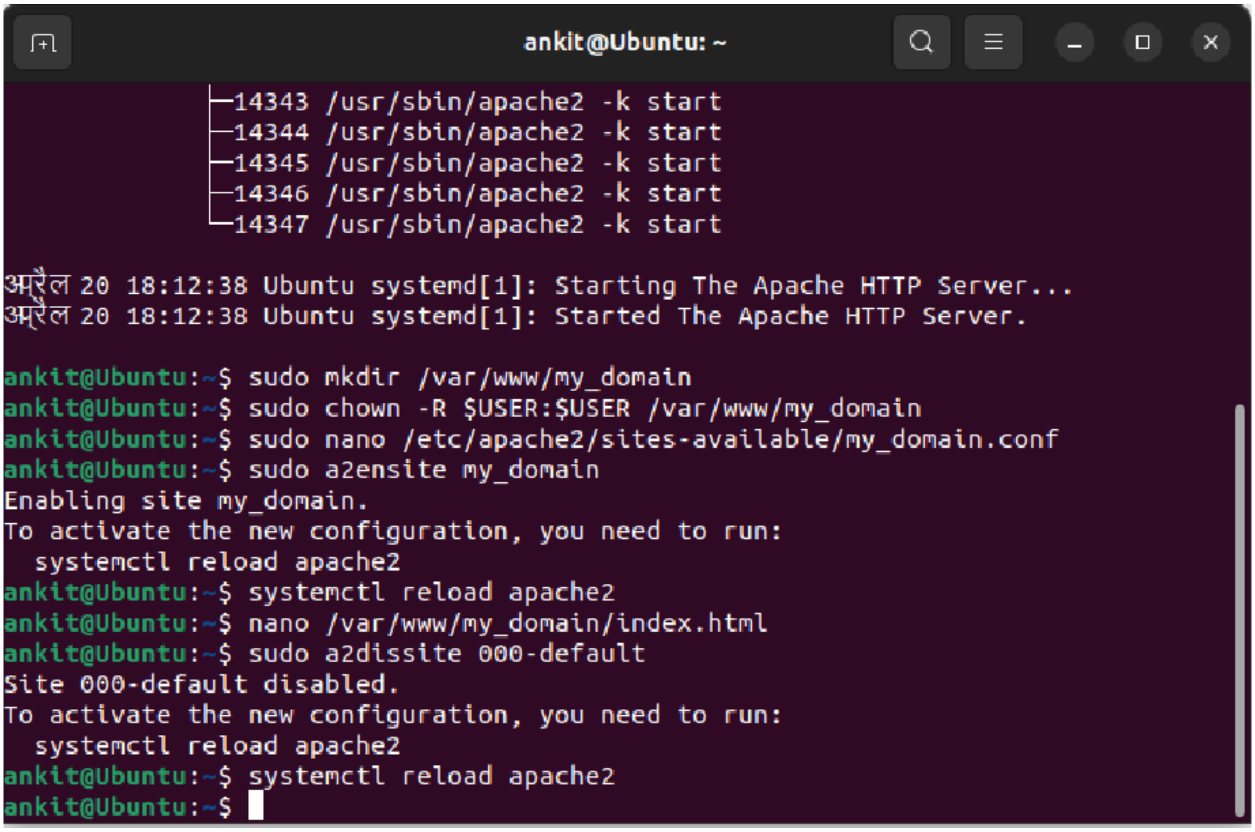
Next, assign ownership of the directory with the $USER environment variable, which will reference your current system user:

-$ sudo chown -R $USER:$USER /var/www/your_domain

Then, open a new configuration file in Apache's sites-available directory using your preferred command-line editor. Here, we'll use nano:

-$ sudo nano /etc/apache2/sites-available/your_domain.conf

This will create a new blank file. Add in the following bare-bones configuration with your own domain name:



```
             ├─14343 /usr/sbin/apache2 -k start
             ├─14344 /usr/sbin/apache2 -k start
             ├─14345 /usr/sbin/apache2 -k start
             ├─14346 /usr/sbin/apache2 -k start
             └─14347 /usr/sbin/apache2 -k start

अप्रैल 20 18:12:38 Ubuntu systemd[1]: Starting The Apache HTTP Server...
अप्रैल 20 18:12:38 Ubuntu systemd[1]: Started The Apache HTTP Server.

ankit@Ubuntu:~$ sudo mkdir /var/www/my_domain
ankit@Ubuntu:~$ sudo chown -R $USER:$USER /var/www/my_domain
ankit@Ubuntu:~$ sudo nano /etc/apache2/sites-available/my_domain.conf
ankit@Ubuntu:~$ sudo a2ensite my_domain
Enabling site my_domain.
To activate the new configuration, you need to run:
  systemctl reload apache2
ankit@Ubuntu:~$ systemctl reload apache2
ankit@Ubuntu:~$ nano /var/www/my_domain/index.html
ankit@Ubuntu:~$ sudo a2dissite 000-default
Site 000-default disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
ankit@Ubuntu:~$ systemctl reload apache2
ankit@Ubuntu:~$ 
```

Now, use a2ensite to enable the new virtual host:

sudo a2ensite your_domain

To disable Apache's default website, type:

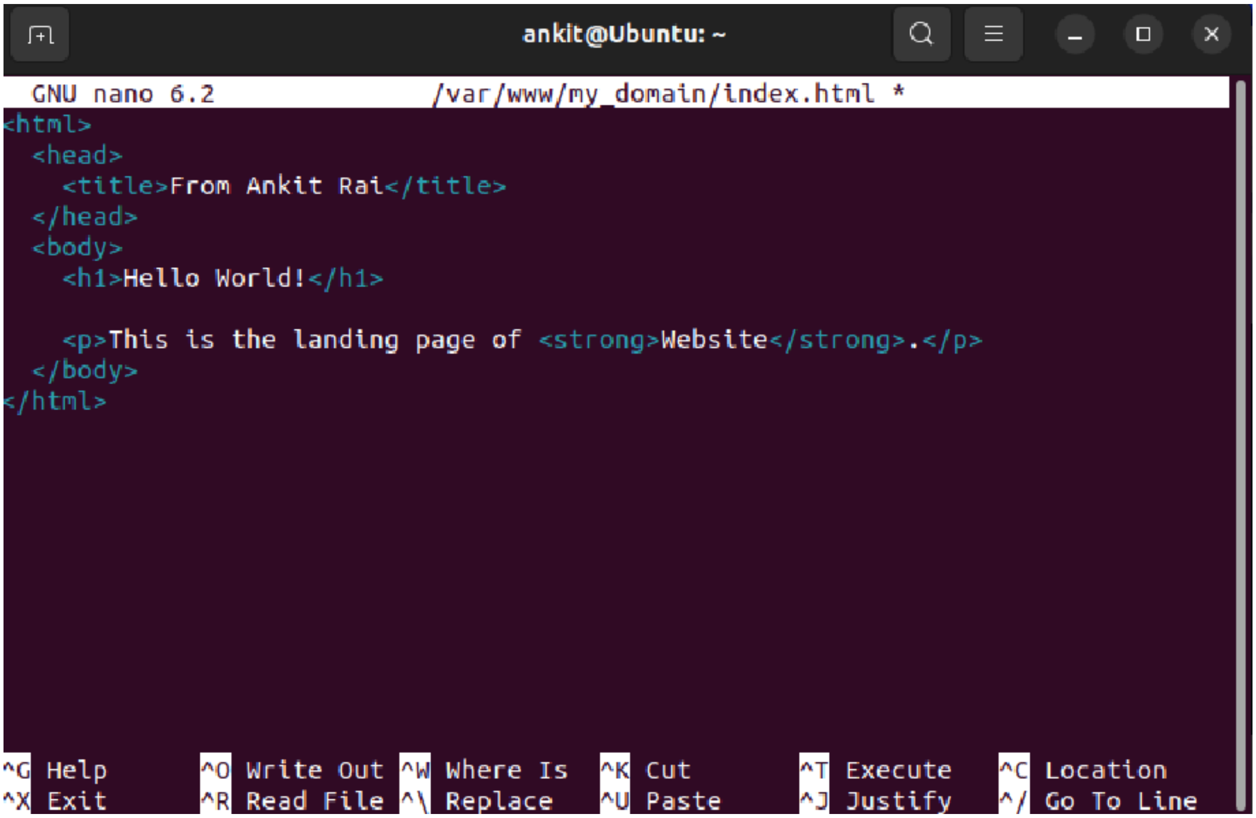
-$ sudo a2dissite 000-default

Finally, reload Apache so these changes take effect:

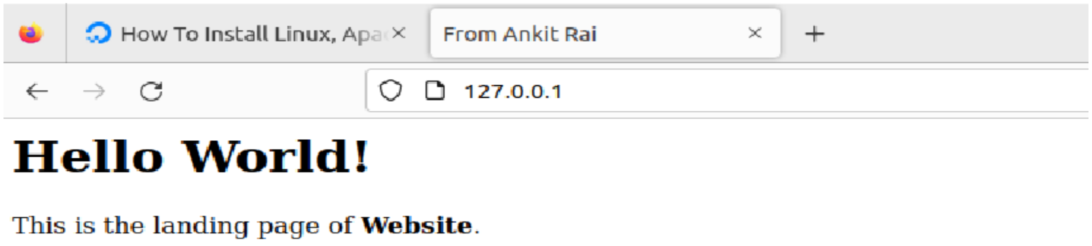
-$ sudo systemctl reload apache2

Your new website is now active, but the web root /var/www/your_domain is still empty. Create an index.html file in that location to test that the virtual host works as expected:

-$ nano /var/www/your_domain/index.html



```
  GNU nano 6.2                /var/www/my_domain/index.html *
<html>
  <head>
    <title>From Ankit Rai</title>
  </head>
  <body>
    <h1>Hello World!</h1>

    <p>This is the landing page of <strong>Website</strong>.</p>
  </body>
</html>

^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line
```



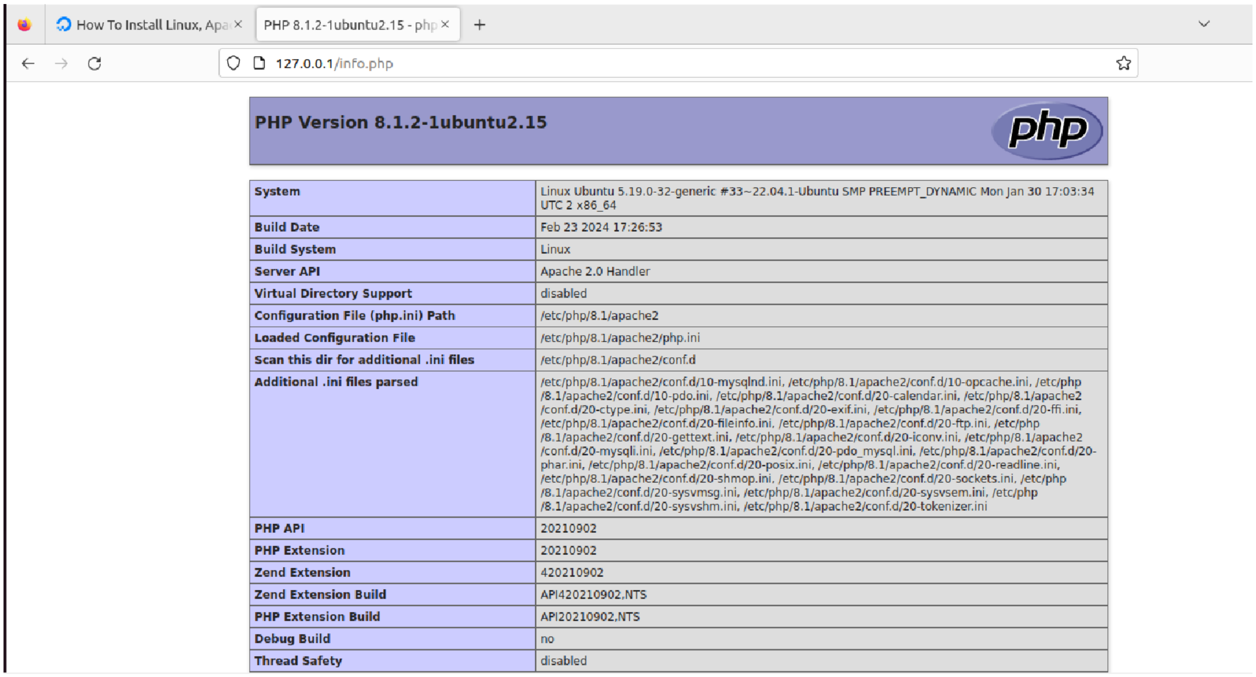
# Hello World!

This is the landing page of **Website**.

## Step 5 — Testing PHP Processing on your Web Server

Create a new file named info.php inside your custom web root folder:

-$ nano /var/www/your_domain/info.php



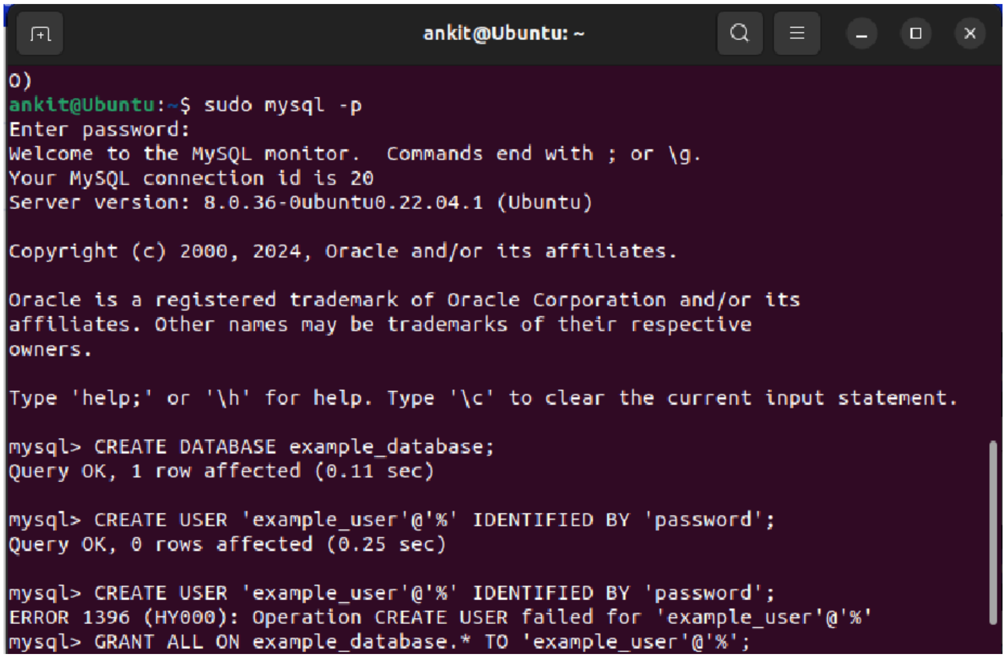
PHP Version 8.1.2-1ubuntu2.15

| | |
|---|---|
| **System** | Linux Ubuntu 5.19.0-32-generic #33~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Mon Jan 30 17:03:34 UTC 2 x86_64 |
| **Build Date** | Feb 23 2024 17:26:53 |
| **Build System** | Linux |
| **Server API** | Apache 2.0 Handler |
| **Virtual Directory Support** | disabled |
| **Configuration File (php.ini) Path** | /etc/php/8.1/apache2 |
| **Loaded Configuration File** | /etc/php/8.1/apache2/php.ini |
| **Scan this dir for additional .ini files** | /etc/php/8.1/apache2/conf.d |
| **Additional .ini files parsed** | /etc/php/8.1/apache2/conf.d/10-mysqlnd.ini, /etc/php/8.1/apache2/conf.d/10-opcache.ini, /etc/php/8.1/apache2/conf.d/10-pdo.ini, /etc/php/8.1/apache2/conf.d/20-calendar.ini, /etc/php/8.1/apache2/conf.d/20-ctype.ini, /etc/php/8.1/apache2/conf.d/20-exif.ini, /etc/php/8.1/apache2/conf.d/20-ffi.ini, /etc/php/8.1/apache2/conf.d/20-fileinfo.ini, /etc/php/8.1/apache2/conf.d/20-ftp.ini, /etc/php/8.1/apache2/conf.d/20-gettext.ini, /etc/php/8.1/apache2/conf.d/20-iconv.ini, /etc/php/8.1/apache2/conf.d/20-mysqli.ini, /etc/php/8.1/apache2/conf.d/20-pdo_mysql.ini, /etc/php/8.1/apache2/conf.d/20-phar.ini, /etc/php/8.1/apache2/conf.d/20-posix.ini, /etc/php/8.1/apache2/conf.d/20-readline.ini, /etc/php/8.1/apache2/conf.d/20-shmop.ini, /etc/php/8.1/apache2/conf.d/20-sockets.ini, /etc/php/8.1/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.1/apache2/conf.d/20-sysvsem.ini, /etc/php/8.1/apache2/conf.d/20-sysvshm.ini, /etc/php/8.1/apache2/conf.d/20-tokenizer.ini |
| **PHP API** | 20210902 |
| **PHP Extension** | 20210902 |
| **Zend Extension** | 420210902 |
| **Zend Extension Build** | API420210902,NTS |
| **PHP Extension Build** | API20210902,NTS |
| **Debug Build** | no |
| **Thread Safety** | disabled |

## Step 6 — Testing Database Connection from PHP

Create a database named example_database and a user named example_user. You can replace these names with different values.

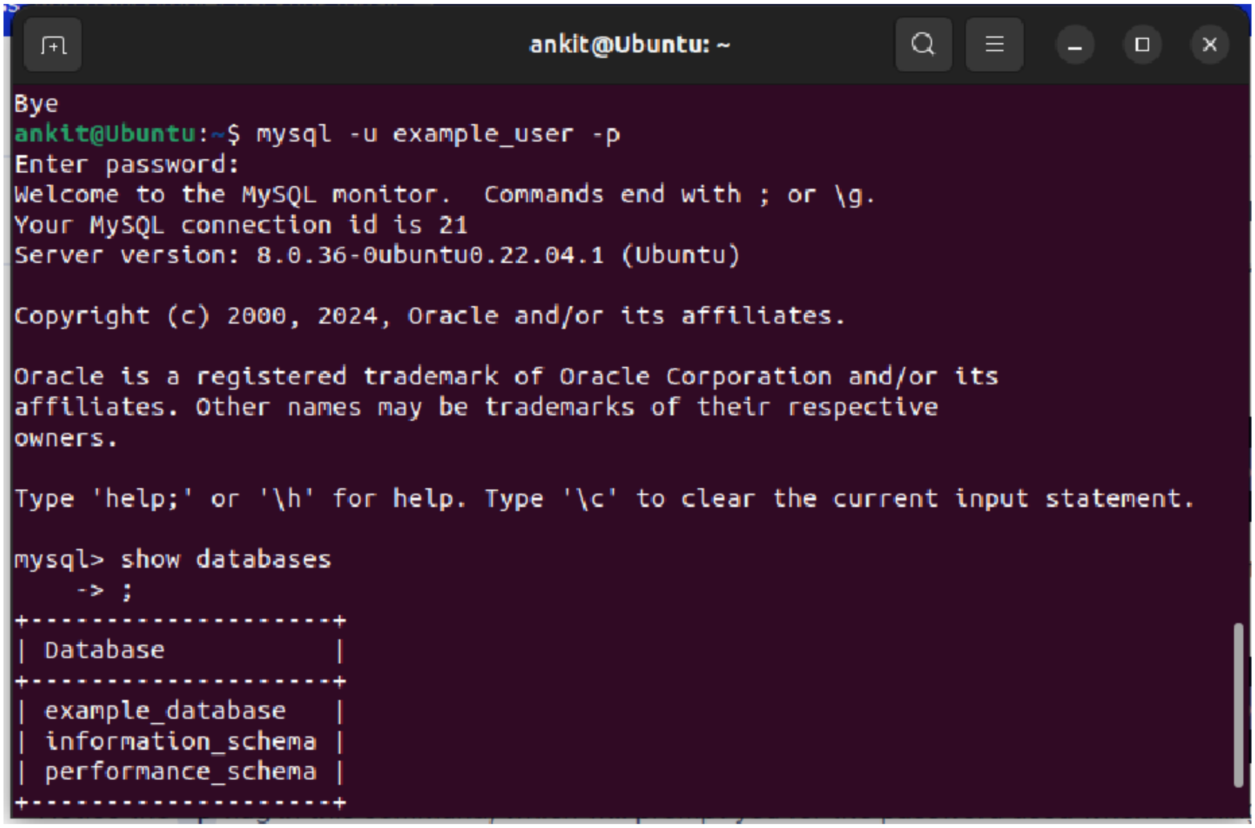
First, connect to the MySQL console using the root account and create a new database:



```
0)
ankit@Ubuntu:~$ sudo mysql -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 20
Server version: 8.0.36-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

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> CREATE DATABASE example_database;
Query OK, 1 row affected (0.11 sec)

mysql> CREATE USER 'example_user'@'%' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.25 sec)

mysql> CREATE USER 'example_user'@'%' IDENTIFIED BY 'password';
ERROR 1396 (HY000): Operation CREATE USER failed for 'example_user'@'%'
mysql> GRANT ALL ON example_database.* TO 'example_user'@'%';
```

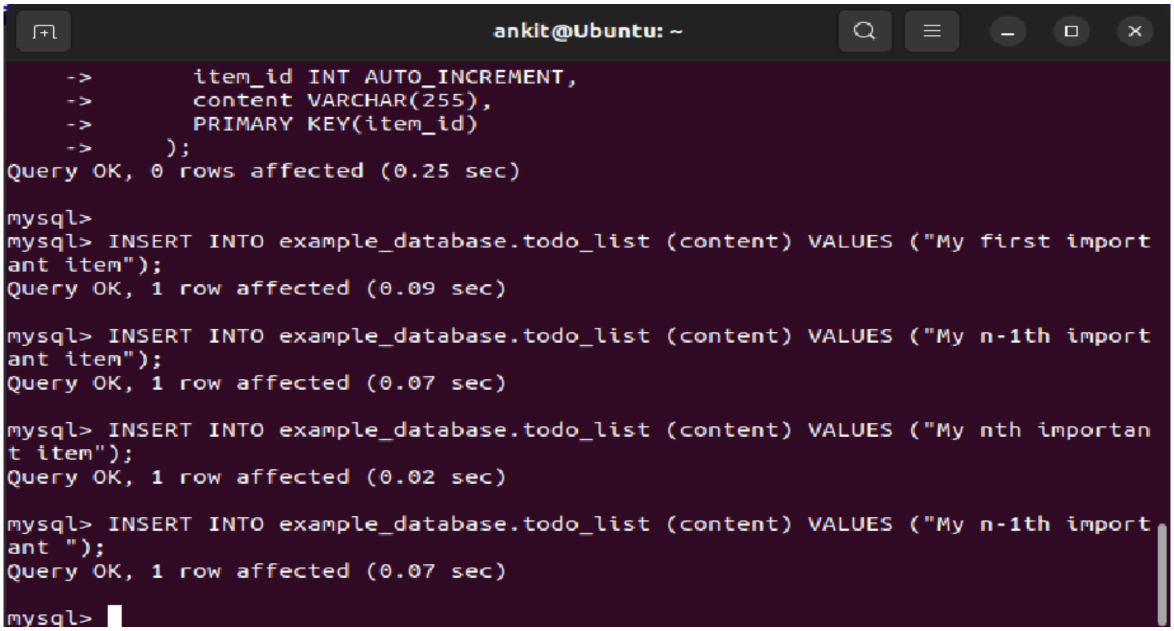
Now create a new user and grant them full privileges on the custom database you've just created.

Now give this user permission over the example_database database as shown in above figure.

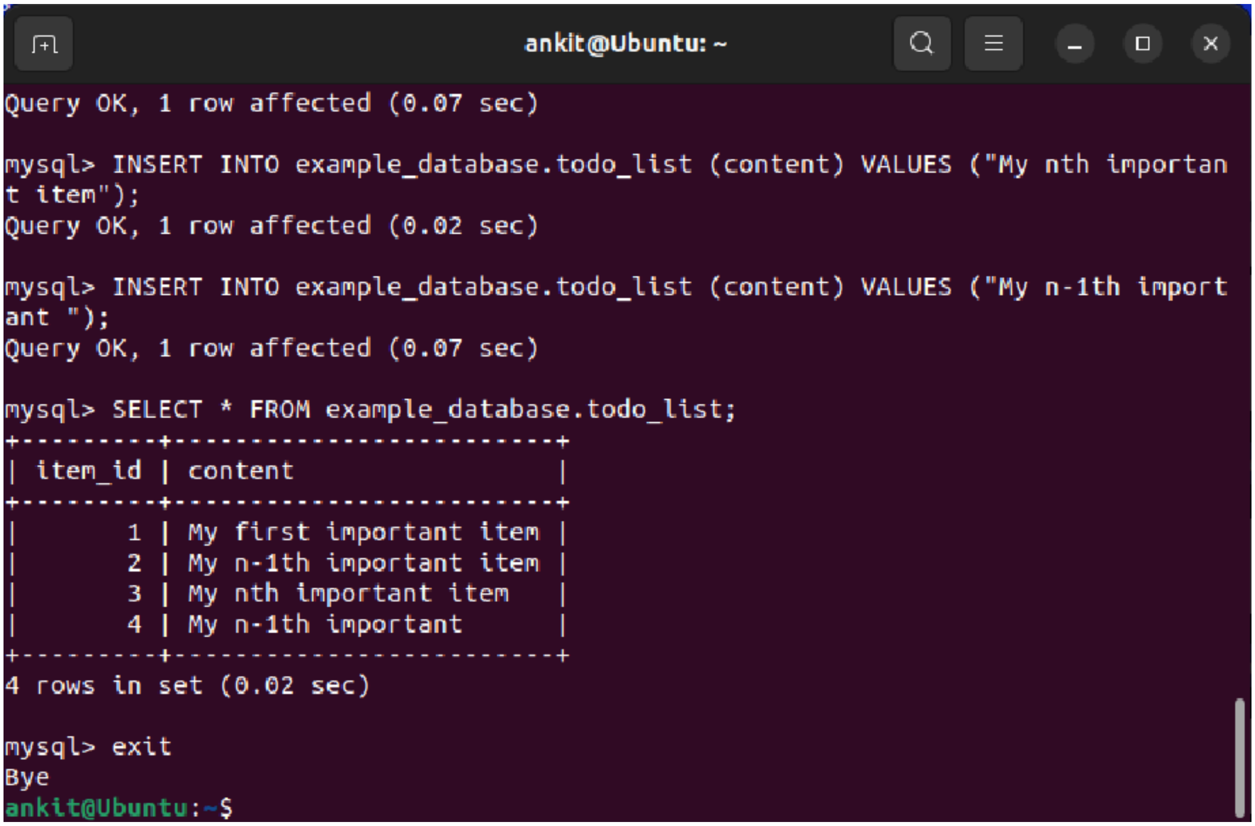


```
Bye
ankit@Ubuntu:~$ mysql -u example_user -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 21
Server version: 8.0.36-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

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> show databases
    -> ;
+--------------------+
| Database           |
+--------------------+
| example_database   |
| information_schema |
| performance_schema |
+--------------------+
```

Insert a few rows of content in the test table. Repeat the next command a few times, using different values, to populate your test table:

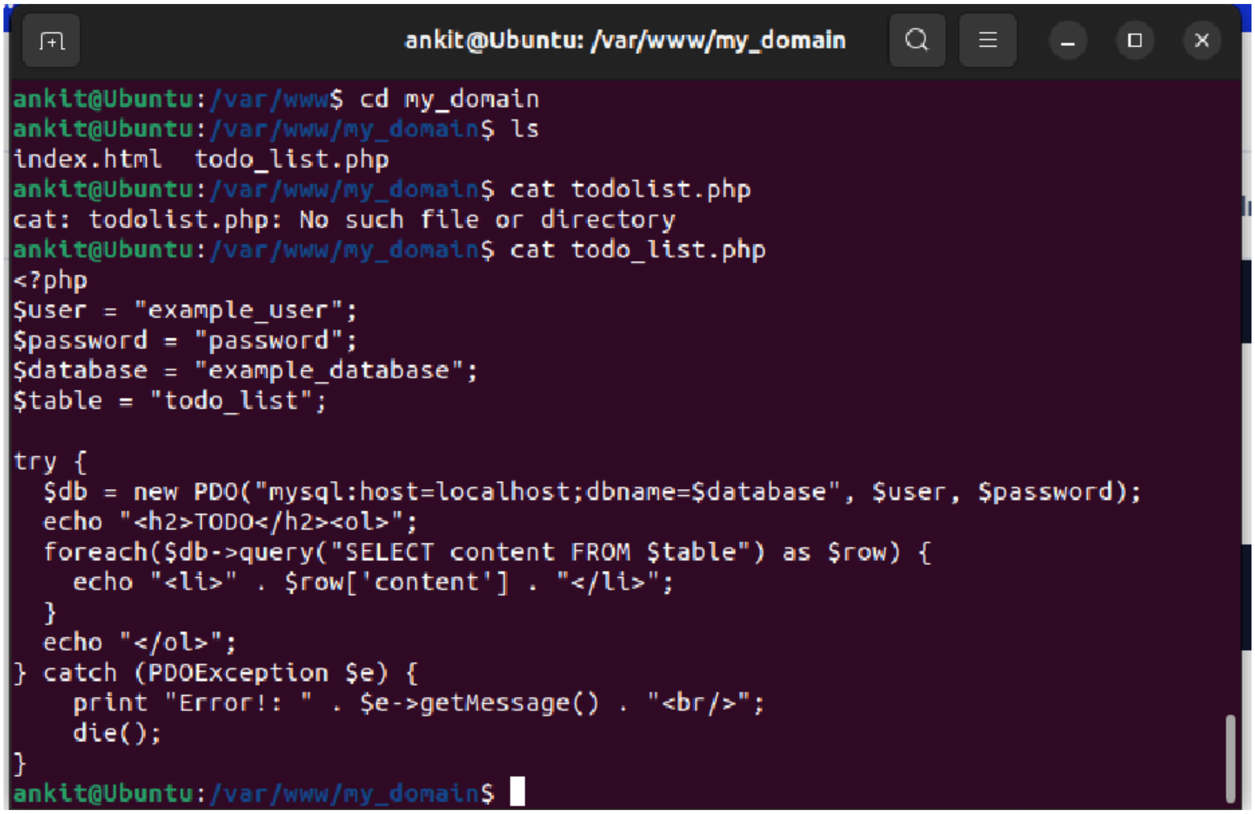


```
    ->       item_id INT AUTO_INCREMENT,
    ->       content VARCHAR(255),
    ->       PRIMARY KEY(item_id)
    ->     );
Query OK, 0 rows affected (0.25 sec)

mysql>
mysql> INSERT INTO example_database.todo_list (content) VALUES ("My first import
ant item");
Query OK, 1 row affected (0.09 sec)

mysql> INSERT INTO example_database.todo_list (content) VALUES ("My n-1th import
ant item");
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO example_database.todo_list (content) VALUES ("My nth importan
t item");
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO example_database.todo_list (content) VALUES ("My n-1th import
ant ");
Query OK, 1 row affected (0.07 sec)

mysql>
```

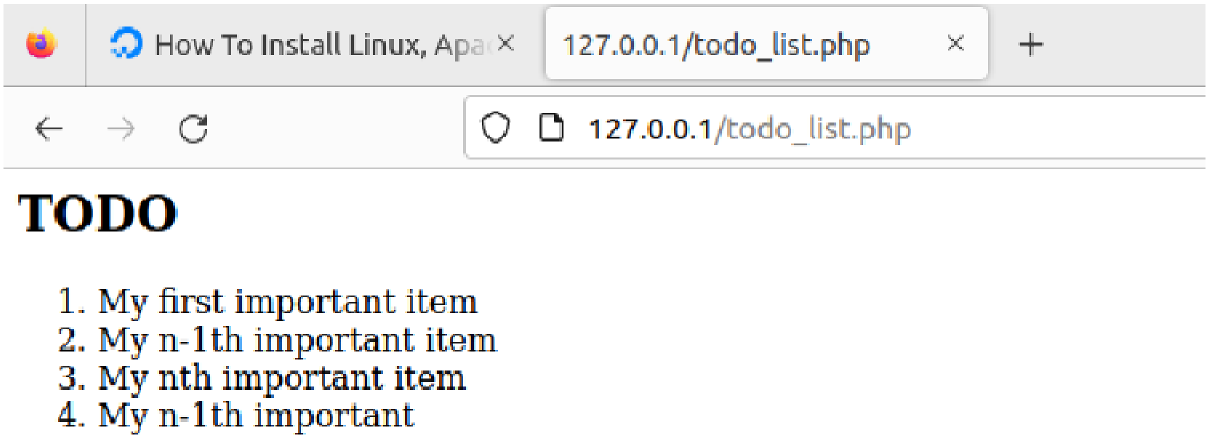


```
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO example_database.todo_list (content) VALUES ("My nth importan
t item");
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO example_database.todo_list (content) VALUES ("My n-1th import
ant ");
Query OK, 1 row affected (0.07 sec)

mysql> SELECT * FROM example_database.todo_list;
+---------+--------------------------+
| item_id | content                  |
+---------+--------------------------+
|       1 | My first important item |
|       2 | My n-1th important item |
|       3 | My nth important item   |
|       4 | My n-1th important      |
+---------+--------------------------+
4 rows in set (0.02 sec)

mysql> exit
Bye
ankit@Ubuntu:~$
```

The following PHP script connects to the MySQL database and queries for the content of the todo_list table, exhibiting the results in a list. If there’s a problem with the database connection, it will throw an exception.



```
ankit@Ubuntu:/var/www$ cd my_domain
ankit@Ubuntu:/var/www/my_domain$ ls
index.html  todo_list.php
ankit@Ubuntu:/var/www/my_domain$ cat todolist.php
cat: todolist.php: No such file or directory
ankit@Ubuntu:/var/www/my_domain$ cat todo_list.php
<?php
$user = "example_user";
$password = "password";
$database = "example_database";
$table = "todo_list";

try {
  $db = new PDO("mysql:host=localhost;dbname=$database", $user, $password);
  echo "<h2>TODO</h2><ol>";
  foreach($db->query("SELECT content FROM $table") as $row) {
    echo "<li>" . $row['content'] . "</li>";
  }
  echo "</ol>";
} catch (PDOException $e) {
    print "Error!: " . $e->getMessage() . "<br/>";
    die();
}
ankit@Ubuntu:/var/www/my_domain$
```



# TODO

1. My first important item
2. My n-1th important item
3. My nth important item
4. My n-1th important

**Conclusion:**

In conclusion, the LAMP stack lab provided a comprehensive hands-on experience with setting up a web server environment using Linux, Apache, MySQL, and PHP.