**Name: Neha Antony**

**Roll No:23**

**Batch:MCA-B**

**Date:07-06-2022**

**NETWORKING & SYSTEM ADMINISTRATION LAB**

**Experiment No.: 5**

**Aim**

Installation and configuration of LAMP stack. Deploy an open source application such as phpmyadmin and Wordpress.

**Procedure**

**INSTALLING APACHE**

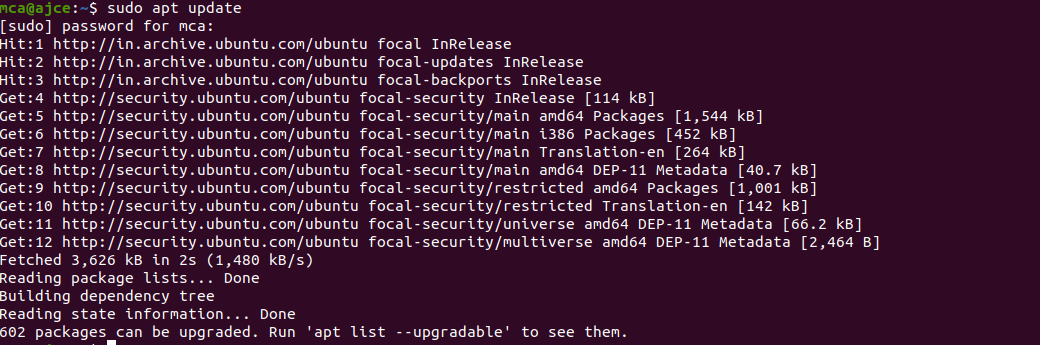
**Step 1** :Installing Apache and Updating the Firewall

The Apache web server is a popular open source web server that can be used along with PHP to host dynamic websites.

First, make sure your apt cache is updated with:

Syntax: $ sudo apt update

**Output:**



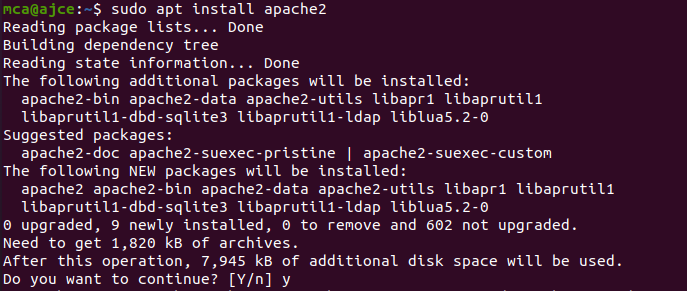
**Step2** :Install Apache 2

Once the cache has been updated, you can install Apache with:

Syntax: $ sudo apt update

Press Y and hit ENTER to confirm, and the installation will proceed.

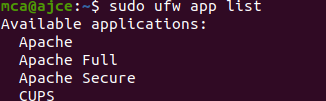
**Output:**



**Step 3** :Adjust the Firewall to Allow Web Traffic

Next, assuming that you have followed the initial server setup instructions and enabled the UFW firewall, make sure that your firewall allows HTTP and HTTPS traffic. You can check that UFW has an application profile for Apache.

**Output:**

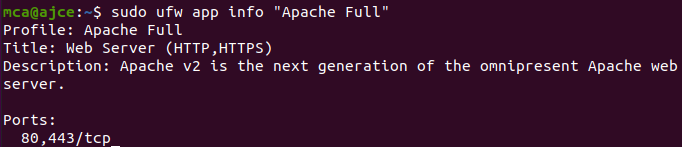
****

**Step 4** :Check Apache Full

Apache Full profile details, you’ll see that it enables traffic to ports 80 and 443:

Syntax: sudoufw app info "Apache Full"

**Output:**



To allow incoming HTTP and HTTPS traffic for this server, run

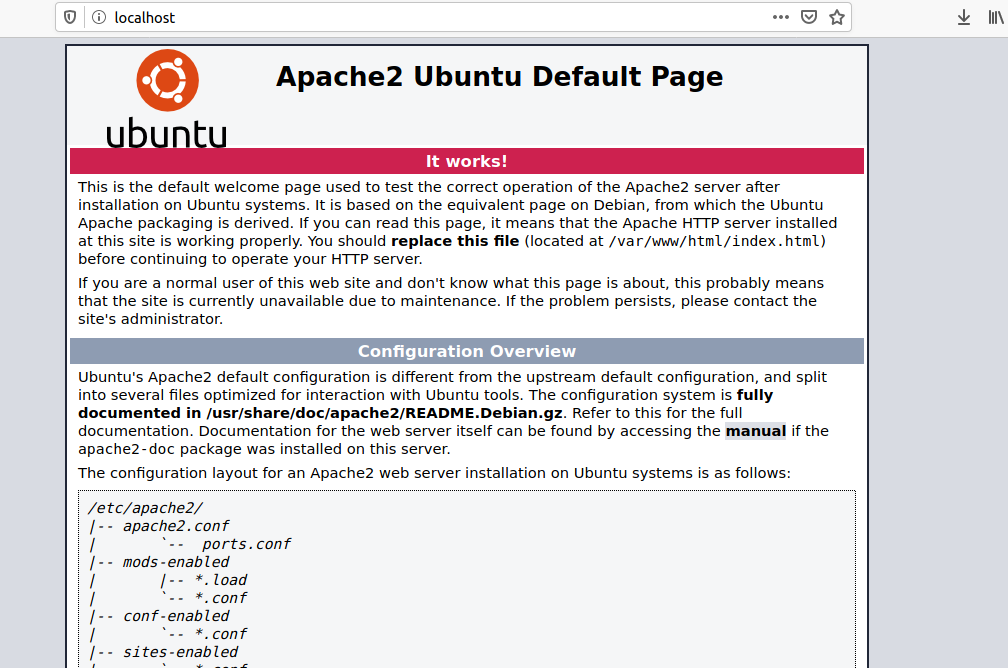
Syntax: sudoufw allow "Apache Full"

**Output:**

**C:\Users\Student\AppData\Local\Temp\Rar$DI62.500\5.png**

**Step 5**:A spot check right away to verify that everything went as planned by visiting your server’s public IP address in your web browser

**Output:**

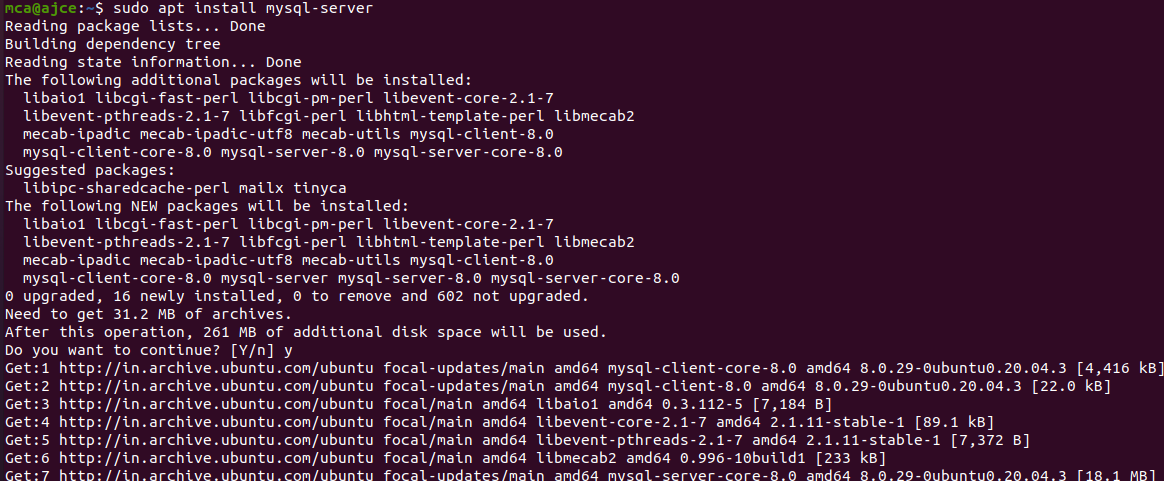
****

**Installing MySQL**

**Step 1**:In this case, you do not have to run sudo apt update prior to the command. This is because you recently ran it in the commands above to install Apache. The package index on your computer should already be up-to-date.

Syntax: $ sudo apt install mysql-server

**Output:**

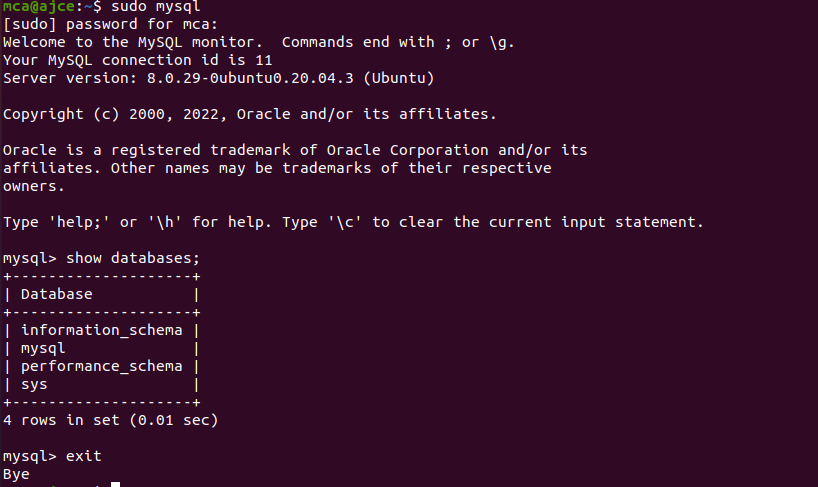


**Step 2** :

This will connect to the MySQL server as the administrative database user root, which is inferred by the use of sudo when running this command.

Syntax: $ sudomysql

**Output:**

****

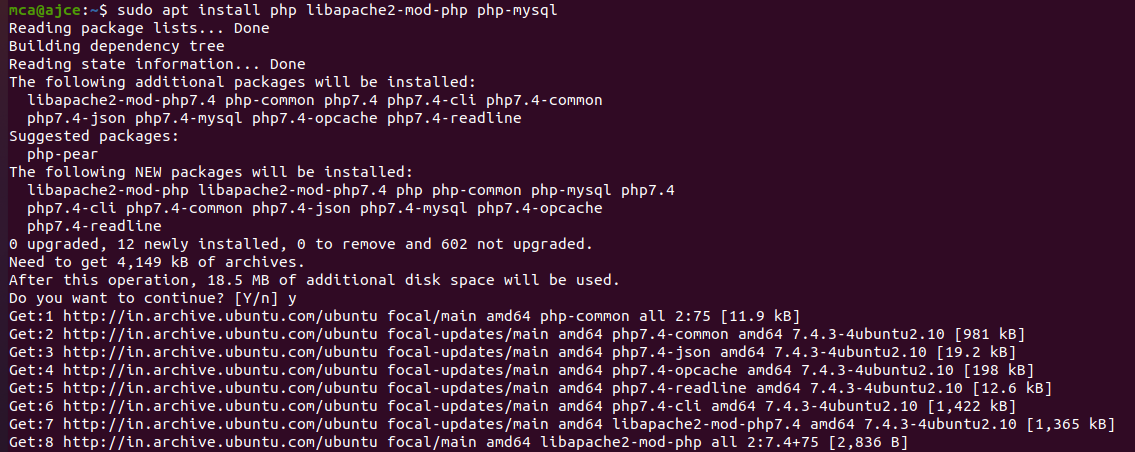
**Installing PHP**

In addition to the php package, you’ll also need libapache2-mod-php to integrate PHP into Apache, and the php-mysql package to allow PHP to connect to MySQL databases. Run the following command to install all three packages and their dependencies.

**Step 1** :Installation

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

**Output:**

****

**Step 2** :Restart

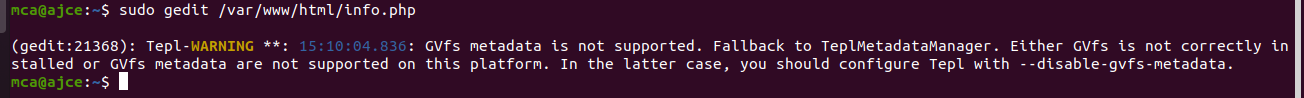
Syntax: sudosystemctl restart apache2

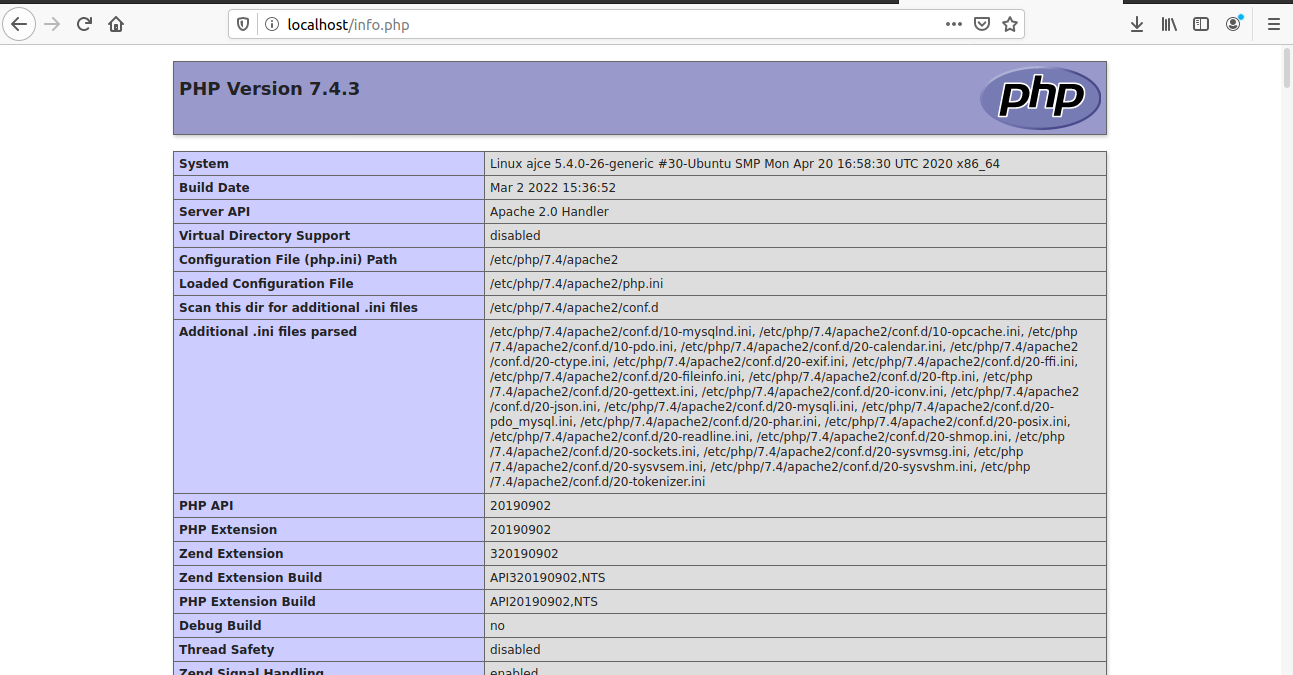
**Output:**

C:\Users\Student\AppData\Local\Temp\Rar$DI09.938\2.png

**Step 3** :Testing PHP Processing on your Web Server

**Output:**

****

****

**Install WordPress with LAMP on Ubuntu 18.04**

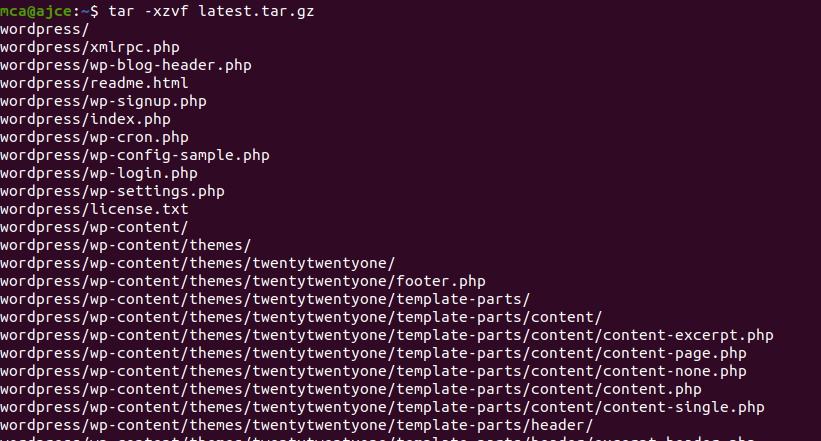
**Step 1**:Download WordPress

Syntax: $ wget -c http://wordpress.org/latest.tar.gz

$ tar -xzvf latest.tar.gz

**Output:**





**Step 2** :Creating a MySQL Database and User for WordPress

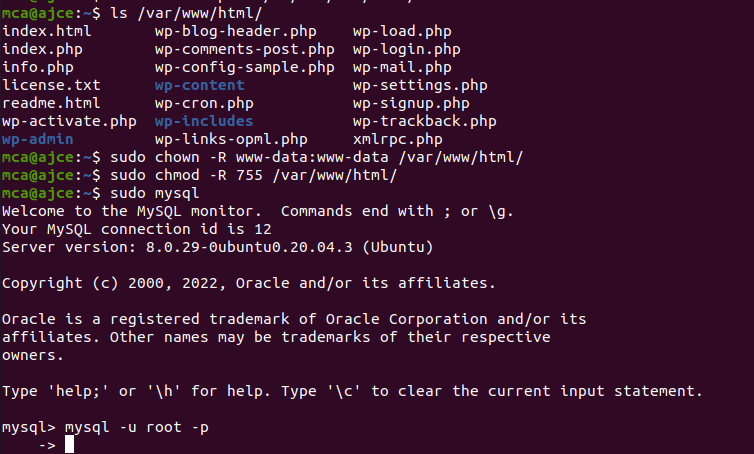
The first step you’ll take is a preparatory one. Even though MySQL is already installed, you still need to create a database to manage and store the user information for WordPress to use. To get started, log into the MySQL root (administrative) account by issuing the following command:

Syntax: $ sudomysql

You will be prompted for the password you set for the MySQL root account when you installed the software. However, if you have password authentication enabled for your root user, you can run the following command and enter your password information when prompted:

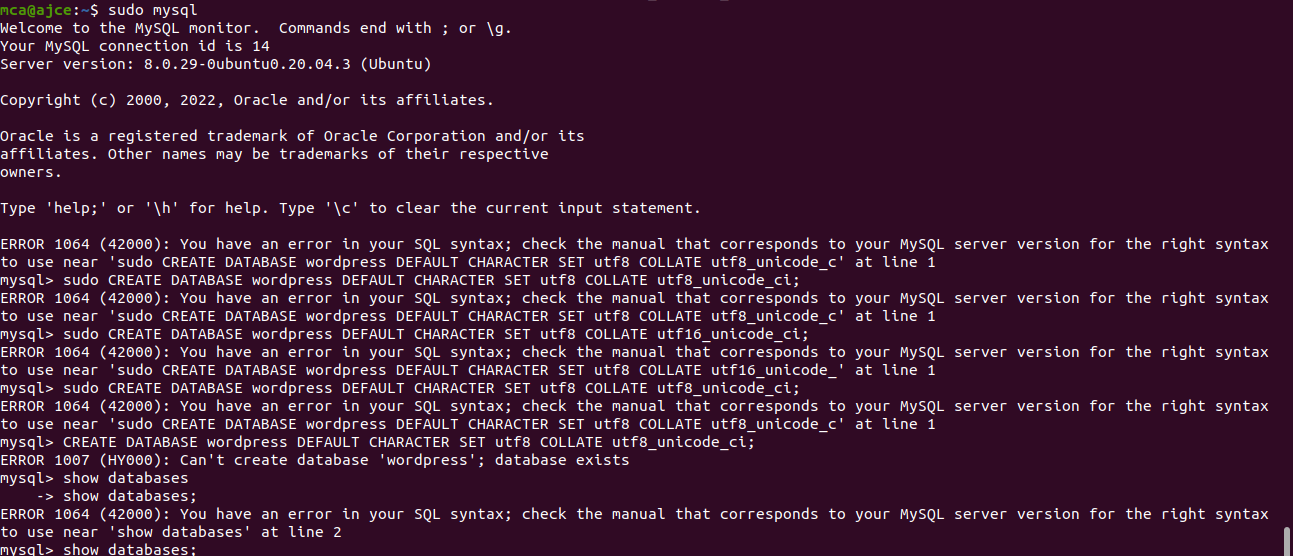
Syntax: $ mysql -u root –p

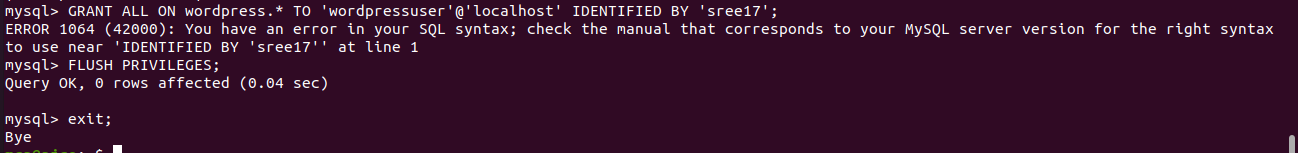
**Output:**

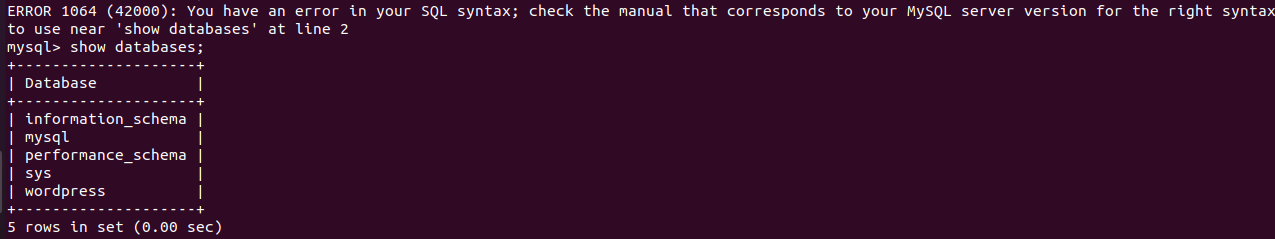


**Step 3** :Create the database for WordPress

**Output:**

****

****

****

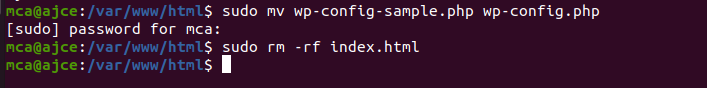
**Step 2** :

Go the /var/www/html/ directory and rename existing wp-config-sample.php to wpconfig.php. Also, make sure to remove the default Apache index page.

Syntax: sudo mv wp-config-sample.phpwp-config.php

sudorm -rf index.html

**Output:**



**Step 3** :Then update it with your database information under the MySQL settings section (refer to the highlighted boxes in the image below): This setting can be added after the database connection settings, or anywhere else in the file:.

Syntax:

define('DB\_NAME', 'wordpress');

/\*\* MySQL database username \*/

define('DB\_USER', 'wordpressuser');

/\*\* MySQL database password \*/

define('DB\_PASSWORD', 'password');

. . .

define('FS\_METHOD', 'direct');

**Output:**

****

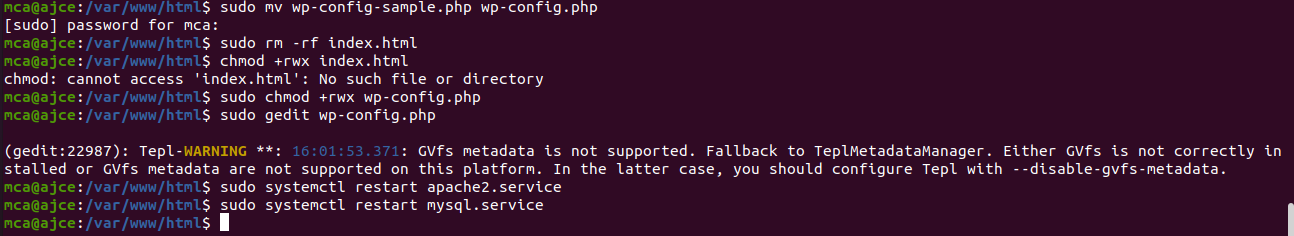
Save and close the file when you are finished.

**Step 4**: Restart the web server and mysql service

Syntax: $sudosystemctl restart apache2.service

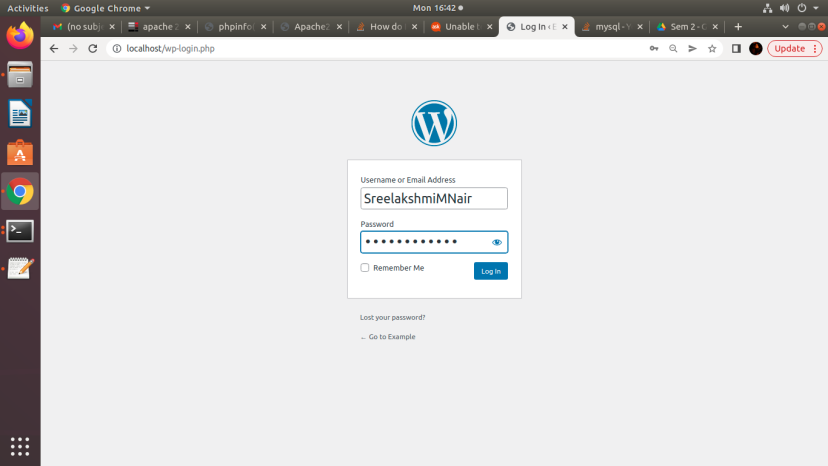
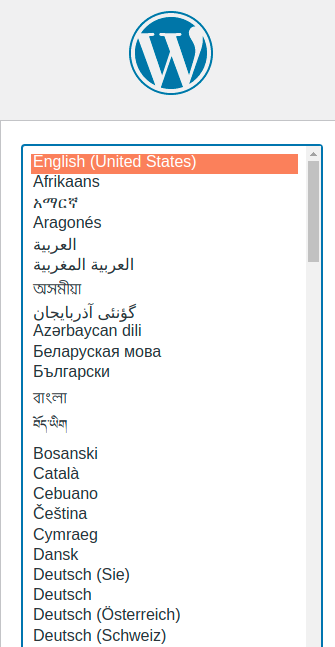
$ sudosystemctl restart mysql.service

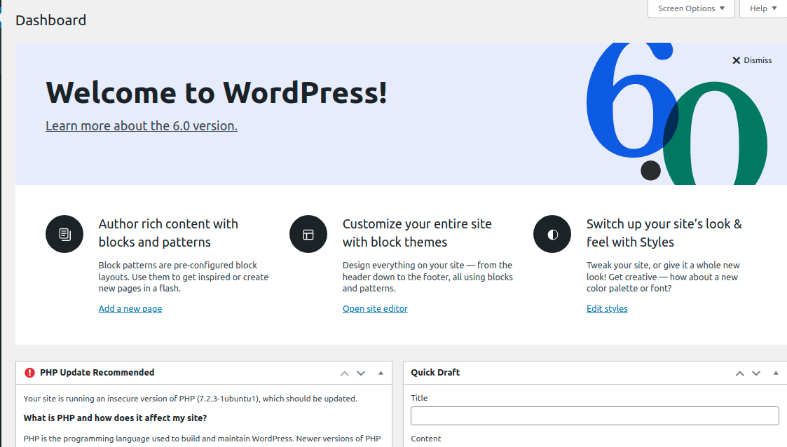
**Output:**

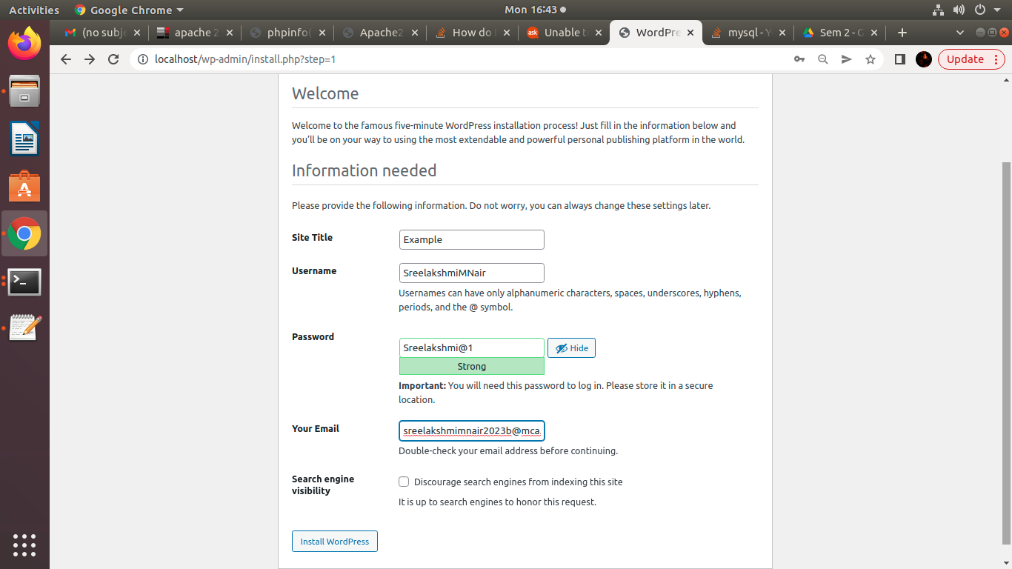
**Step 5**: Completing the Installation Through the Web Interface

The server configuration is complete, you can complete the installation through the web interface.In your web browser, navigate to your server’s domain name or public IP address

**Output:**

****

****

****