

How to build a local database server?

1. Open **terminal**. Make sure your Raspberry Pi is fully updated by entering the following command.

```
sudo apt-get update  
  
sudo apt-get upgrade
```

2. Install Apache, PHP and package needed.

```
sudo apt-get install apache2  
  
sudo apt-get install libapache2-mod-php  
  
sudo apt-get install php
```

Check if Apache is installed.

```
apache2 -v
```

You should be able to see the following:

```
pi@raspberrypi:~ $ apache2 -v  
Server version: Apache/2.4.38 (Raspbian)  
Server built: 2019-10-15T19:53:42
```

Check if PHP is installed.

```
php -v
```

You should be able to see the following:

```
pi@raspberrypi:~ $ php -v  
PHP 7.3.14-1~deb10u1 (cli) (built: Feb 16 2020 15:07:23) ( NTS )  
Copyright (c) 1997-2018 The PHP Group  
Zend Engine v3.3.14, Copyright (c) 1998-2018 Zend Technologies  
with Zend OPcache v7.3.14-1~deb10u1, Copyright (c) 1999-2018, by Zend Technologies
```

3. Edit the file.

```
sudo nano /etc/apache2/sites-enabled/000-default.conf
```

4. Find the line `DocumentRoot /var/www/html` and change to `DocumentRoot /var/www`.

```
GNU nano 3.2 /etc/apache2/sites-enabled/000-default.conf Modified
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    #<VirtualHost>
    #</VirtualHost>

    # vim: syntax=apache ts=4 sw=4 sts=4 sr noet
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Also, find the line `#Include conf-available/serve-cgi-bin.conf`, delete the #.

```
GNU nano 3.2 /etc/apache2/sites-enabled/000-default.conf Modified
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
</VirtualHost>

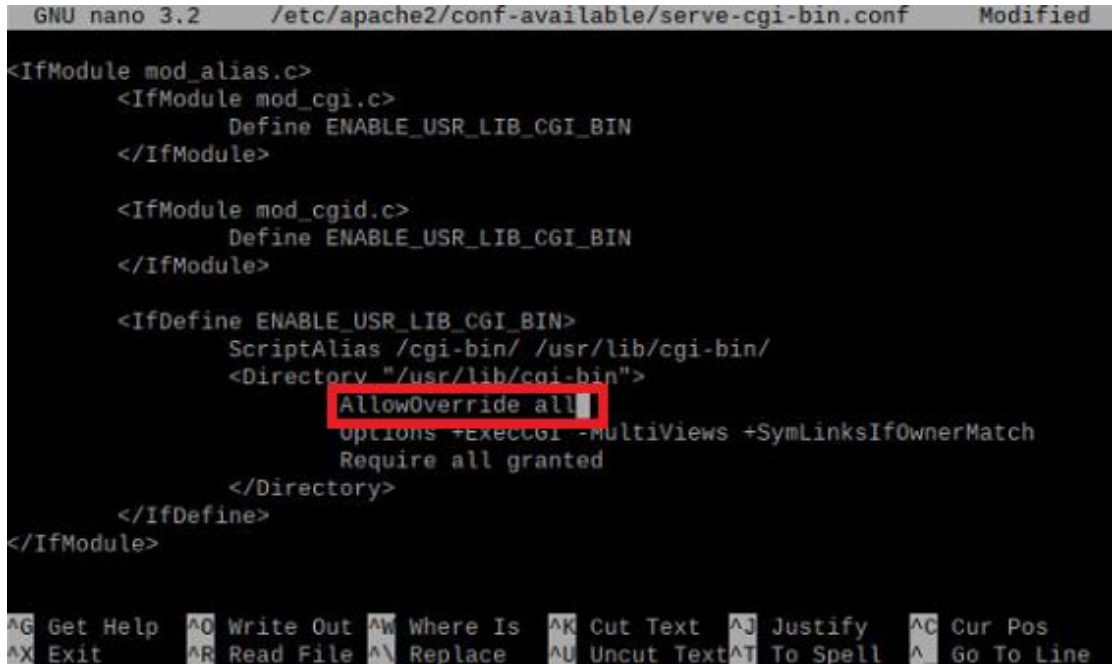
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

To exit the text editor and save the file, press **Ctrl + X → Y → Enter**.

5. Edit the file.

```
sudo nano /etc/apache2/conf-available/serve-cgi-bin.conf
```

6. Find the line `AllowOverride None` and change to `AllowOverride all`.



```
GNU nano 3.2 /etc/apache2/conf-available/serve-cgi-bin.conf Modified
<IfModule mod_alias.c>
  <IfModule mod_cgi.c>
    Define ENABLE_USR_LIB_CGI_BIN
  </IfModule>

  <IfModule mod_cgid.c>
    Define ENABLE_USR_LIB_CGI_BIN
  </IfModule>

  <IfDefine ENABLE_USR_LIB_CGI_BIN>
    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
    <Directory "/usr/lib/cgi-bin">
      AllowOverride all
      Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
      Require all granted
    </Directory>
  </IfDefine>
</IfModule>

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```

To exit the text editor and save the file, press **Ctrl + X** → **Y** → **Enter**.

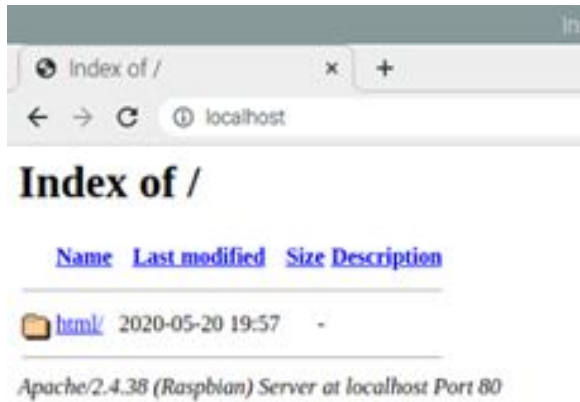
7. Restart Apache service.

```
sudo service apache2 restart
```

- Click the **web browser icon** at the top menu bar and enter “**localhost**” in the URL bar.



You should be able to see the following:



Noted that you can use **your IP address** or the word “**localhost**” or **127.0.0.1** to visit this web page.

- Install MySQL, Mariadb and the package needed.

```
sudo apt install mariadb-server

sudo apt-get install python3-dev
**default-libmysqlclient-dev**

sudo pip3 install mysqlclient
```

10. After install the MySQL, try to login.

```
sudo mysql -u root -p
```

The default password is **no password**, so simply click “**Enter**” when you are asked.

You should be able to see the following if you are login successfully.

```
pi@raspberrypi:~$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 47
Server version: 10.3.22-MariaDB-0+deb10u1 Raspbian 10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]>
```

11. Grant all the privileges to the root user, allowing root has full control over a whole database.

```
GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost'
IDENTIFIED BY '' WITH GRANT OPTION;
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' IDENTIFIED BY '' WITH GRANT OPTION;
Query OK, 0 rows affected (0.001 sec)
```

Exit the MySQL.

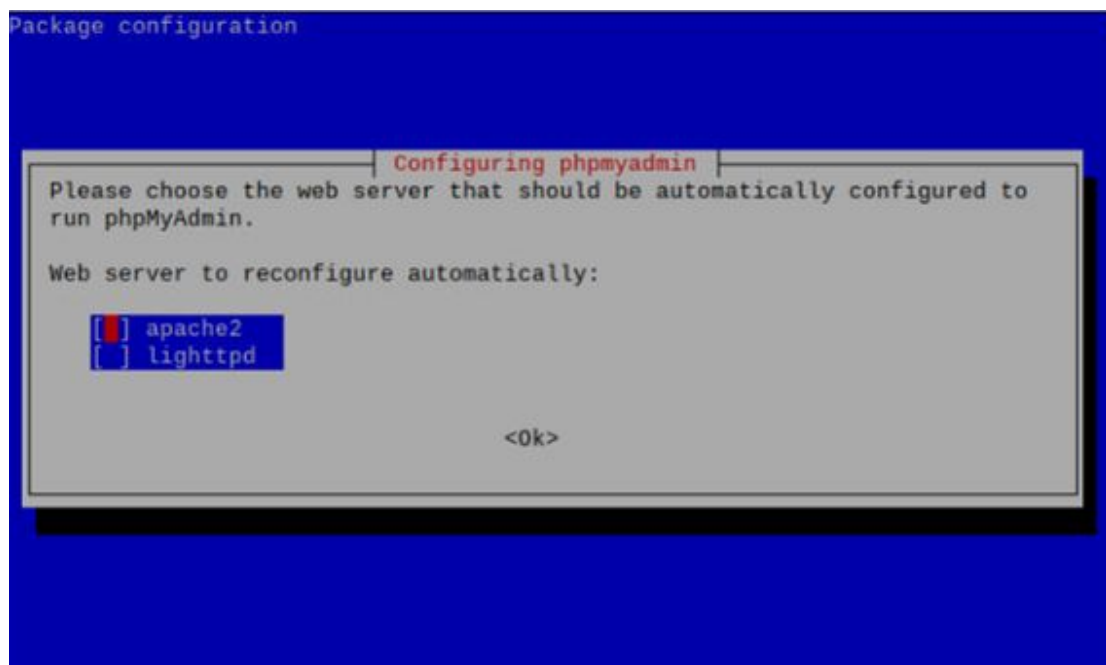
```
exit
```

```
MariaDB [(none)]> exit
Bye
```

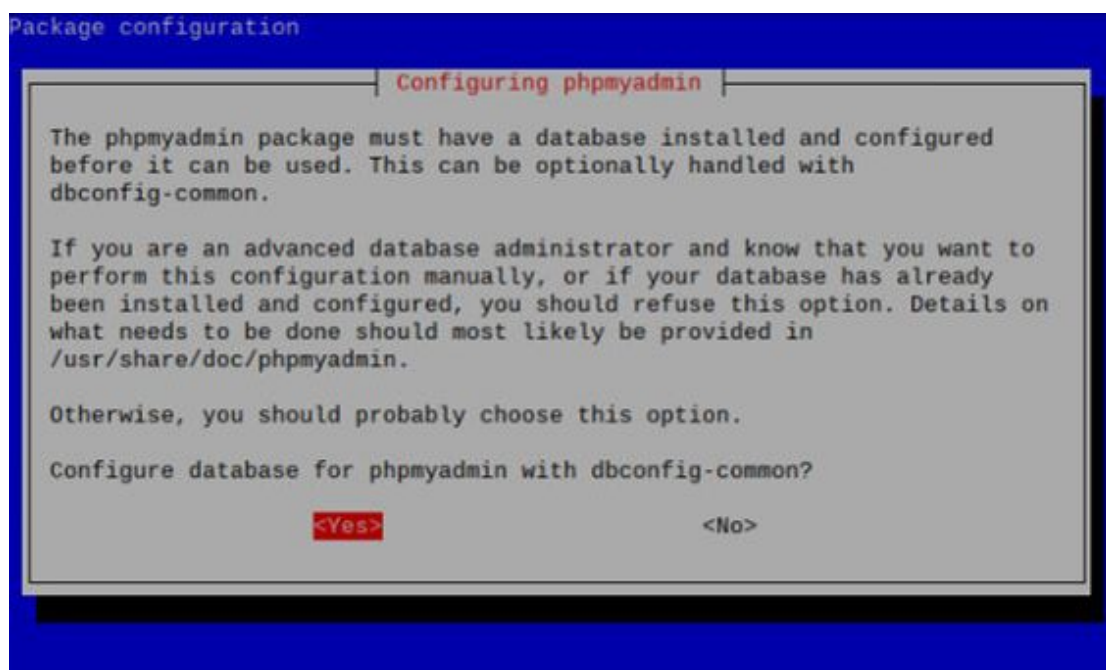
12. Install phpmyadmin.

```
sudo apt-get install phpmyadmin
```

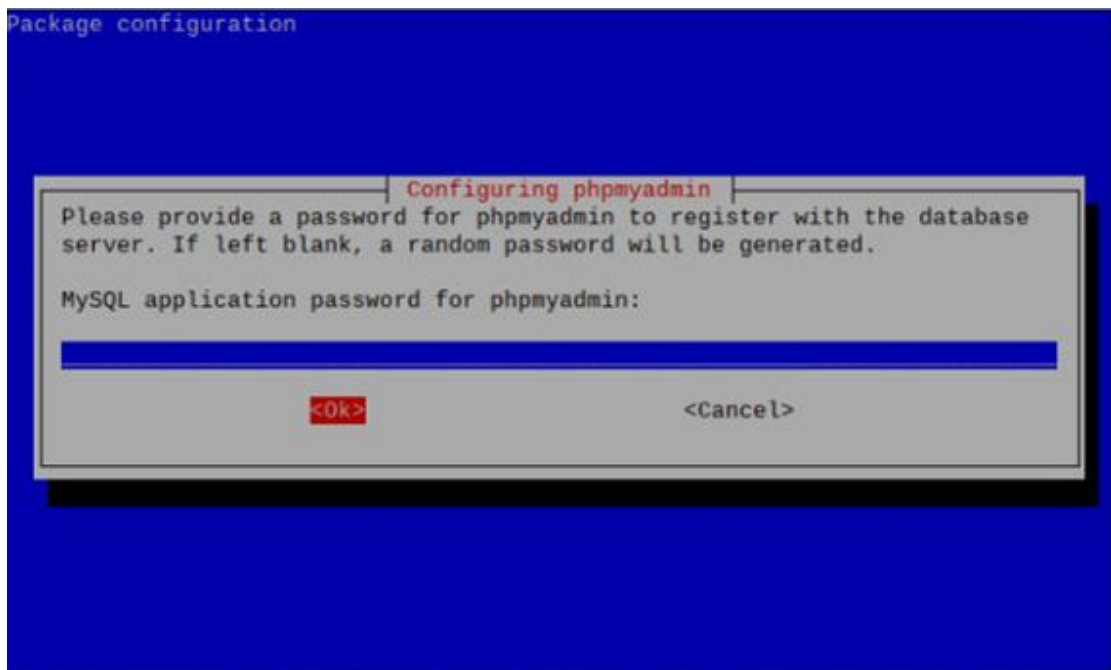
13. Choose “**apache2**” by hitting **space bar**, then “**Enter**”.



Choose “**Yes**”.



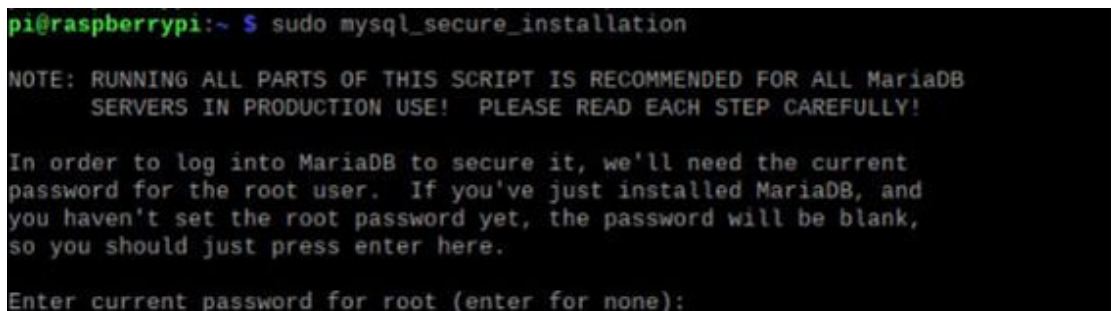
Click “OK”.



14. Change the security setting of MySQL.

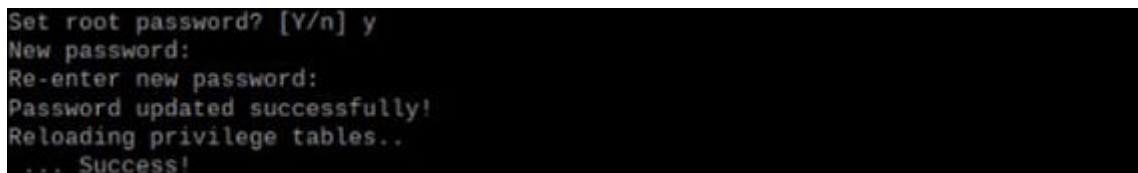
```
sudo mysql_secure_installation
```

You should be able to see the following:



“Enter” for no current password.

“Y” for set password. Enter a password, for simplicity, please use “**raspberry**”. Then, “Enter” → “**raspberry**” → “Enter”.



Noted that when you type a password in terminal it is **not echoed** to the screen. Just keep typing in the password and press Enter.

“N” → “Enter”.

```
Remove anonymous users? [Y/n] n
... skipping.

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.
```

“N” → “Enter”.

```
Disallow root login remotely? [Y/n] n
... skipping.

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.
```

“Y” → “Enter”.

```
Remove test database and access to it? [Y/n] y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.
```

“Y” → “Enter”.

```
Reload privilege tables now? [Y/n] y
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

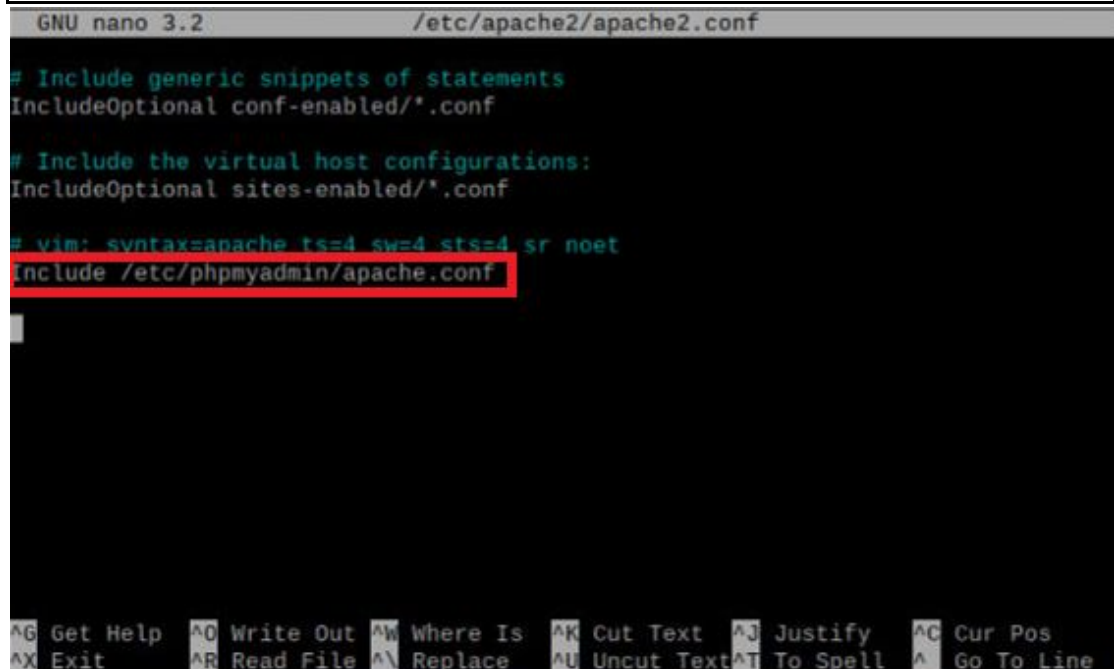
Thanks for using MariaDB!
```


15. Edit the file.

```
sudo nano /etc/apache2/apache2.conf
```

Add the following at the end of the file.

```
Include /etc/phpmyadmin/apache.conf
```



```
GNU nano 3.2 /etc/apache2/apache2.conf

# Include generic snippets of statements
IncludeOptional conf-enabled/*.conf

# Include the virtual host configurations:
IncludeOptional sites-enabled/*.conf

# vim: svntax=apache ts=4 sw=4 sts=4 sr noet
Include /etc/phpmyadmin/apache.conf
```

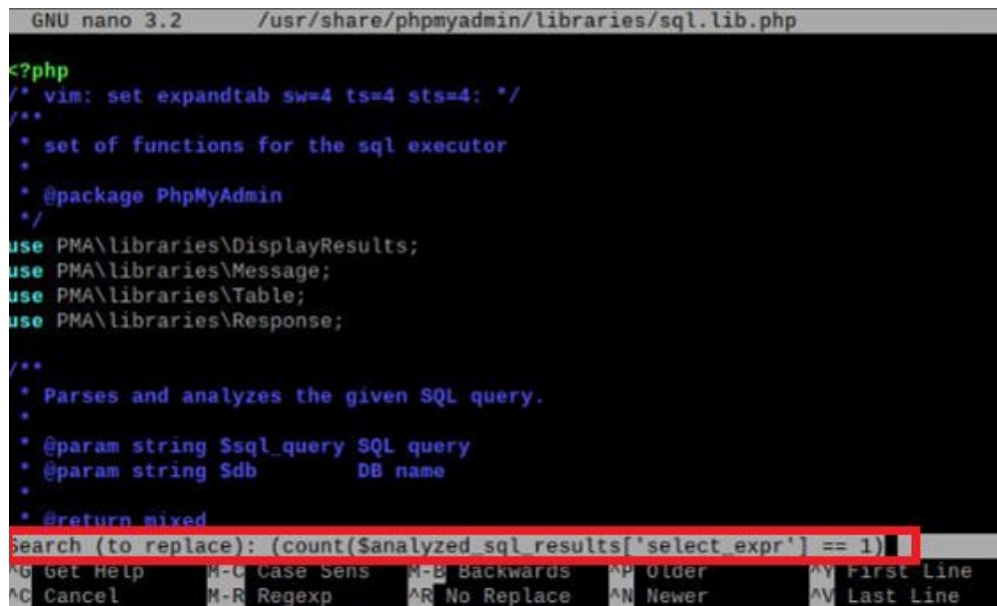
To exit the text editor and save the file, press **Ctrl + X** → **Y** → **Enter**.

16. Edit the file.

```
sudo pico /usr/share/phpmyadmin/libraries/sql.lib.php
```

**Ctrl + ** → paste this command

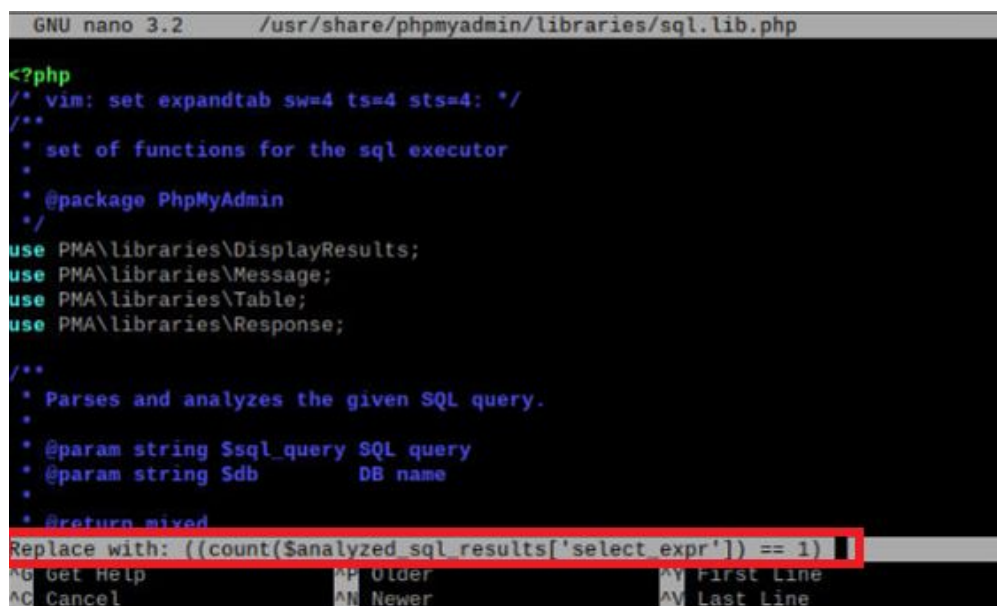
(count(\$analyzed_sql_results['select_expr'] == 1) on the search bar as shown below:



The screenshot shows the nano text editor interface with the file `/usr/share/phpmyadmin/libraries/sql.lib.php` open. The editor contains PHP code with comments. A search bar at the bottom is active, displaying the search string `(count($analyzed_sql_results['select_expr'] == 1)`. The search bar is highlighted with a red box. The bottom status bar shows navigation options: `^G Get Help`, `^M Case Sens`, `^B Backwards`, `^O Older`, `^Y First Line`, `^C Cancel`, `^R Regexp`, `^R No Replace`, `^N Newer`, and `^V Last Line`.

“Enter” → paste this command

((count(\$analyzed_sql_results['select_expr'])) == 1) on the the bar as shown below:



The screenshot shows the nano text editor interface with the same file open. A replace bar at the bottom is active, displaying the replacement string `((count($analyzed_sql_results['select_expr'])) == 1)`. The replace bar is highlighted with a red box. The bottom status bar shows navigation options: `^G Get Help`, `^M Older`, `^Y First Line`, `^C Cancel`, `^N Newer`, and `^V Last Line`.

“Enter” → **“Y”** to confirm the change. To exit the text editor and save the file, press **Ctrl + X** → **Y** → **Enter**.

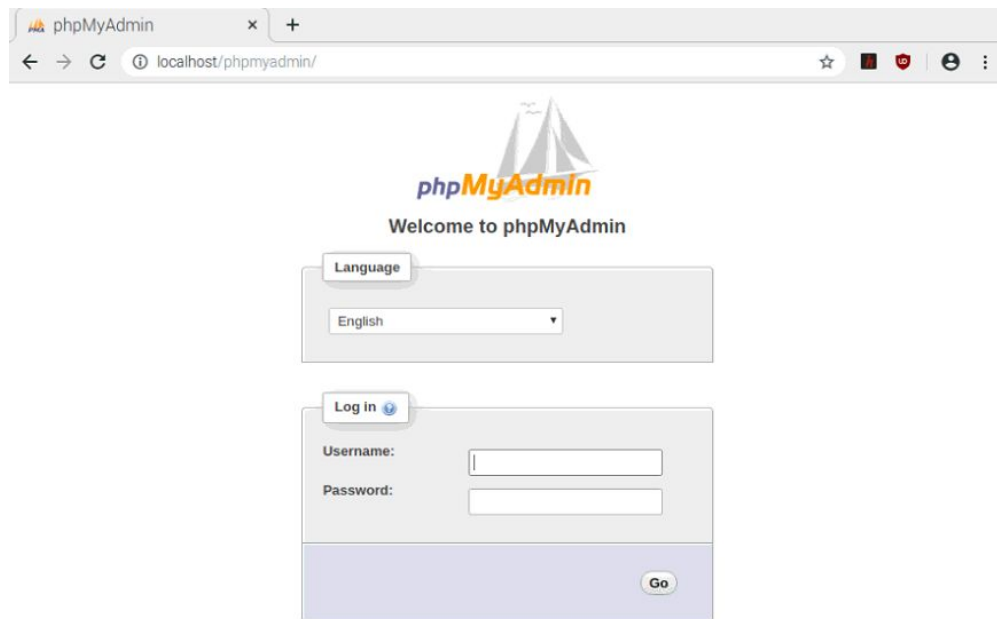
17. Restart the Apache.

```
sudo /etc/init.d/apache2 restart
```

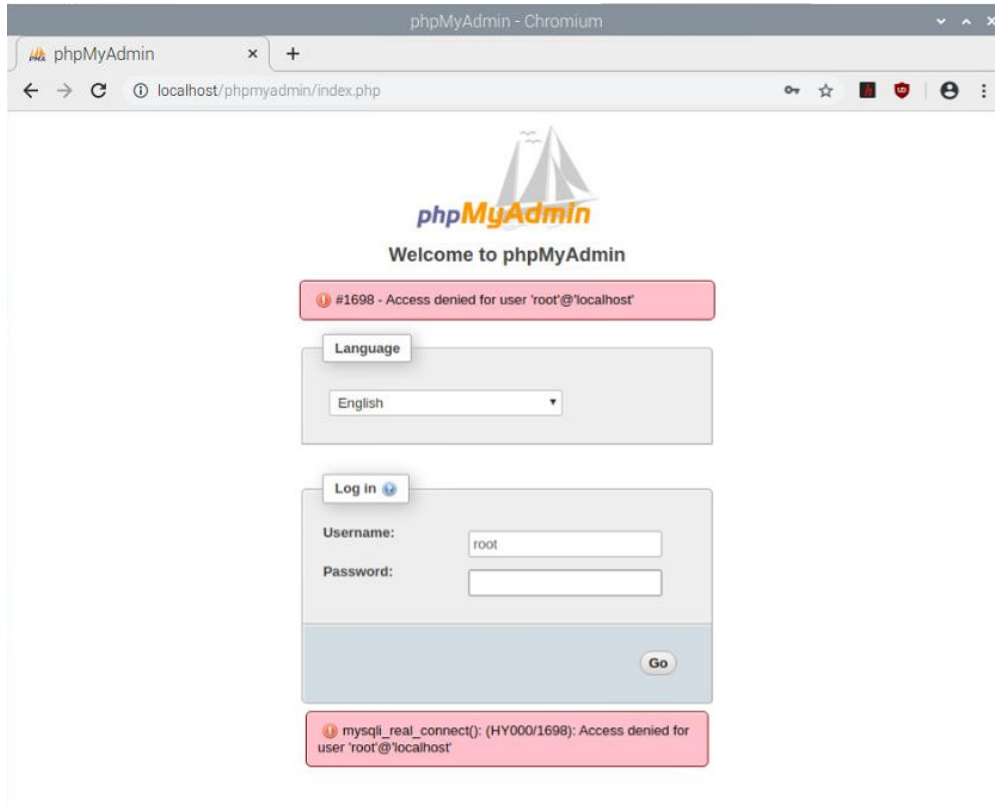
You should be able to see the following:

```
pi@raspberrypi:~$ sudo /etc/init.d/apache2 restart
[ ok ] Restarting apache2 (via systemctl): apache2.service.
```

18. After completed all the steps above, test your server. Open the **web browser** of Raspberry Pi and enter “**localhost/phpmyadmin**”. It should show the following.



19. You can login with the username “**root**” and the password “**raspberry**”. You may find the following happen.



In this case, you need to perform the following **step 20-23**.

20. Back to terminal and login to MySQL with the password you set before.

```
sudo mysql -u root -p
```

Enter the following:

```
SET PASSWORD FOR root@localhost=PASSWORD('');
```

```
MariaDB [(none)]> SET PASSWORD FOR root@localhost=PASSWORD('');  
Query OK, 0 rows affected, 1 warning (0.001 sec)
```

```
exit
```

```
MariaDB [(none)]> exit  
Bye
```

21. Re-install phpmyadmin.

```
sudo dpkg-reconfigure phpmyadmin
```

**“OK” → “YES” → “TCP/IP” → “localhost” → Enter → Enter → “OK” → Enter
→ Enter → Enter → “OK” → Enter → Enter → “apache2” → Enter**

Noted that if you are unable to press “Enter” and go to next step, try to click **rightwards arrow** then click **Enter**.

22. Redo the security setting of MySQL.

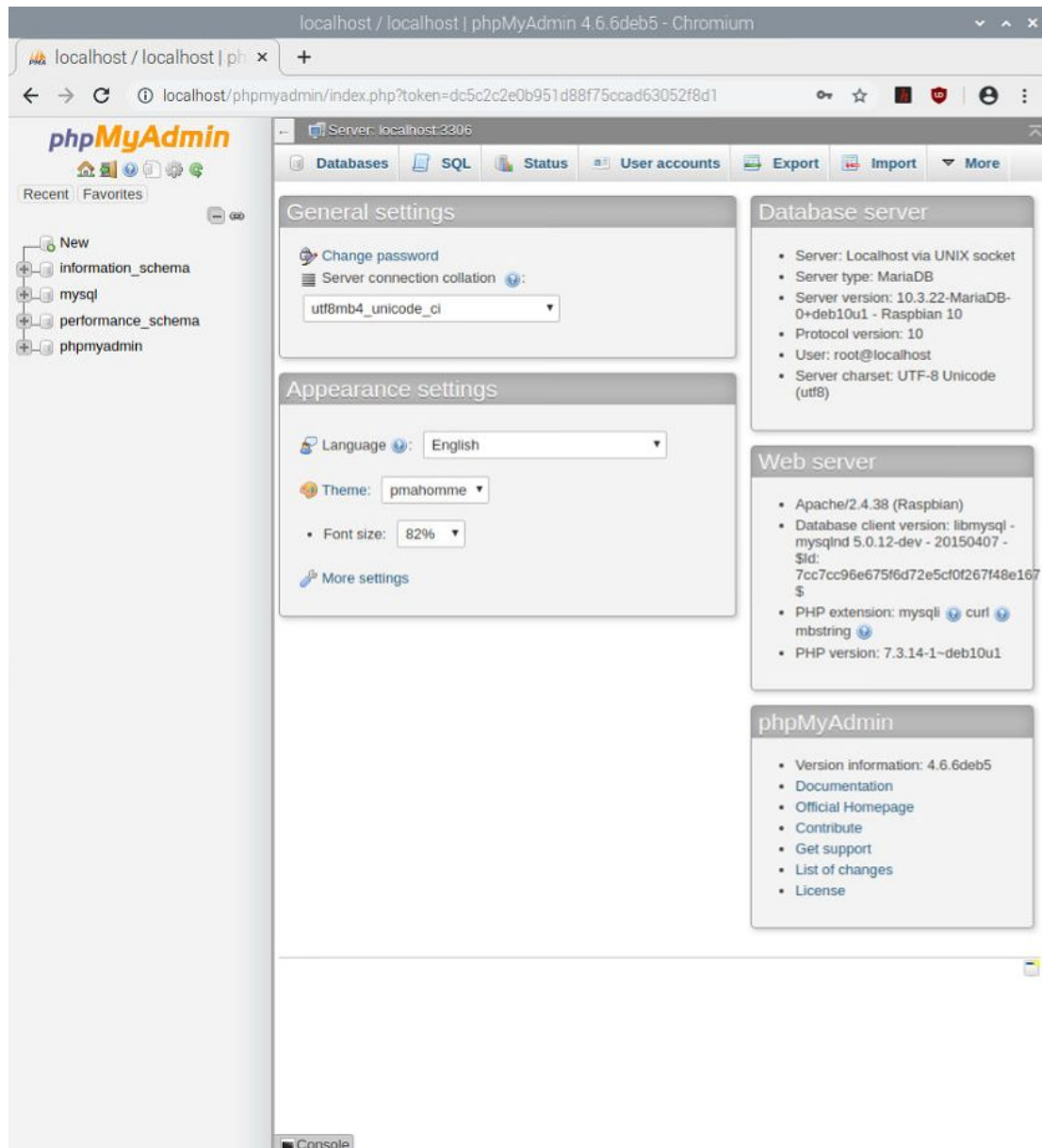
```
sudo mysql_secure_installation
```

Enter → Y → raspberry → raspberry → N → N → Y → Y

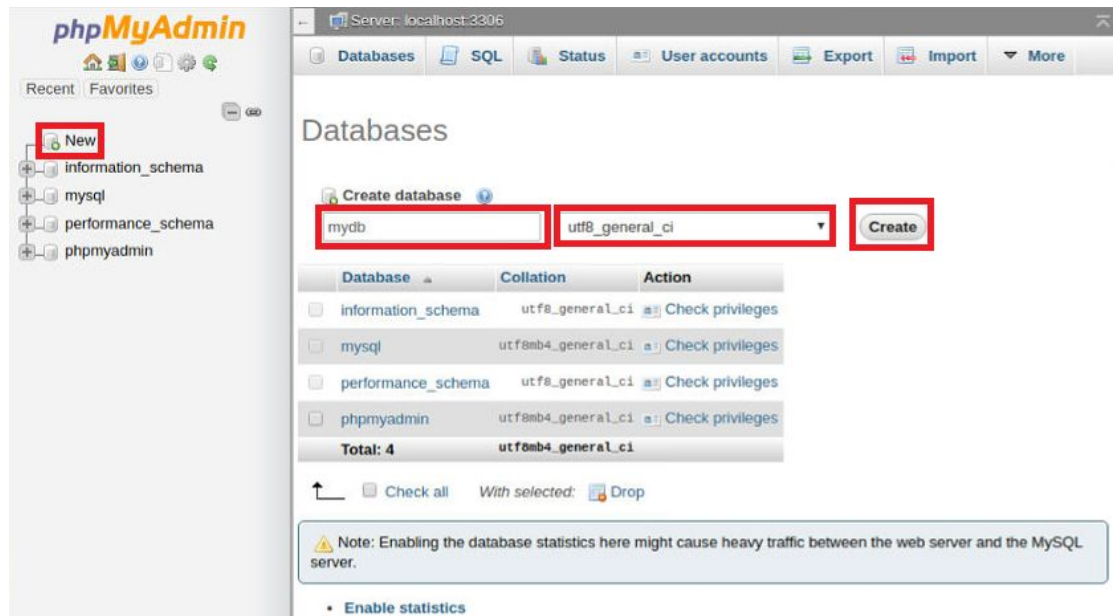
23. Restart the Apache.

```
sudo /etc/init.d/apache2 restart
```

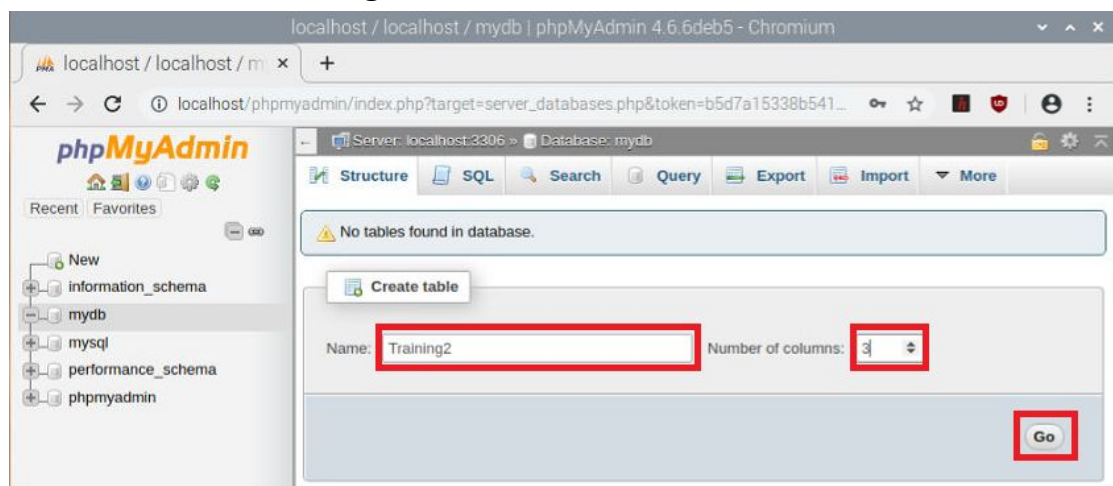
24. Open the **web browser** of Raspberry Pi and enter “**localhost/phpmyadmin**”. Login with “**root**” and “**raspberry**”. This time you should be able to login.



25. Create a new database name “**mydb**”. Select “**utf8_general_ci**” for Collation. Then click “**Create**”.



26. Create table name “**Training2**”. Number of column is “**3**” then “**Go**”.



27. Try to make a Temperature Recorder. Enter the structure according to the following table:

Name	Type	Length/Value	Index	A_I
sr_no	INT		PRIMARY	✓
rec_time	TIMESTAMP		INDEX	
rec_temp	FLOAT	6,3	---	

Table name: Add column(s)

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A_I
<input type="text" value="sr_no"/> <small>Pick from Central Columns</small>	INT	<input type="text"/>	None			<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>
<input type="text" value="rec_time"/> <small>Pick from Central Columns</small>	TIMESTAMP	<input type="text"/>	None			<input type="checkbox"/>	INDEX	<input type="checkbox"/>
<input type="text" value="rec_temp"/> <small>Pick from Central Columns</small>	FLOAT	6,3	None			<input type="checkbox"/>	---	<input type="checkbox"/>

Noted that A_I stands for **Auto-increment**, allows a unique number to be generated automatically when a new record is inserted into a table.

Click “**Save**”. You should be able to see the following:

Server: localhost:3306 » Database: mydb » Table: Training2

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	sr_no	int(11)			No	None		AUTO_INCREMENT
2	rec_time	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()
3	rec_temp	float(6,3)			No	None		