單元名稱 (LAB I2605)

Name : 劉泰佑
Student ID : C100E030

1.1 Objective

To be able to setup a new raspberry pi from empty to fully functional system. Then create some users beside the original “pi” user. After that make it to be able to connect to ap router and access it through its ip address. And is to enable the raspberry pi to be able to run localhost in the system and access it through ip address.

1.2 Working theory

Raspberry pi is mainly used as the brain in the iot devices. In this case we try to do some basic setup where we are able to setup the device, connect it to the wifi network, and make it able to run localhost. For the first setup we just need to use raspberry pi official “os flasher” application to flash the os into the sdcard, next up is to create some users by using terminal using the Putty, then we connect it to wifi network, and finally we install the apache, mysql, and php into the raspberry pi.

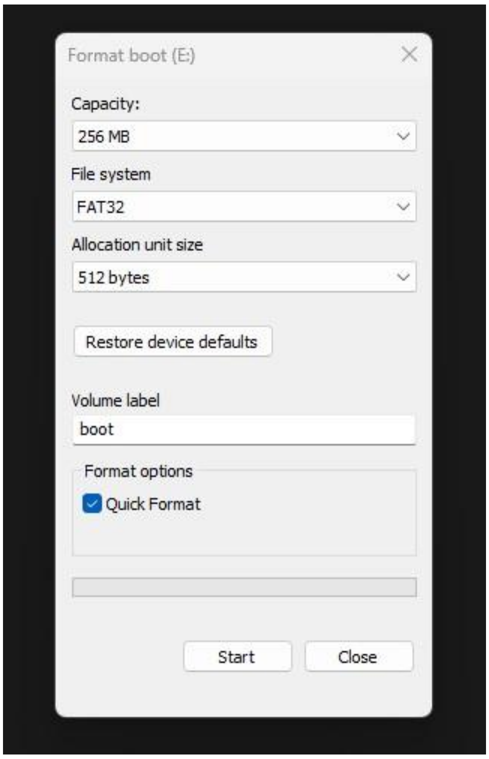
1.3 Experimental device and components)

- Raspberry Pi
- Laptop
- Card Reader
- Micro SD
- Access Point
- Putty
- Raspberry Pi Imager

1.4 Procedure

Step 1

First, we need to format the micro SD to make it empty. So that we can install the operating system freely.



Format boot (E:)

Capacity:
256 MB

File system
FAT32

Allocation unit size
512 bytes

Restore device defaults

Volume label
boot

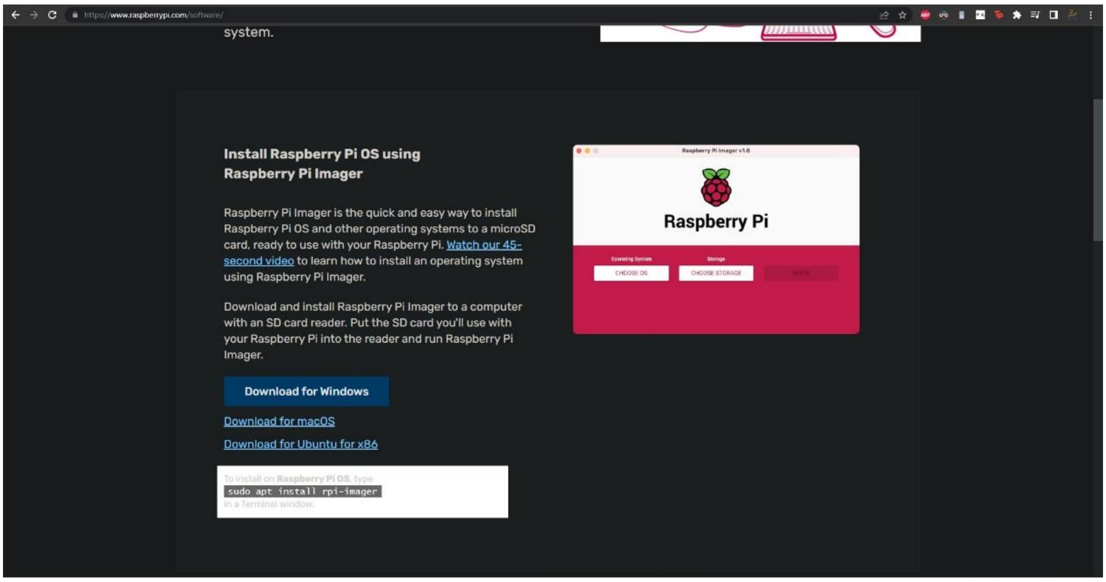
Format options
☑ Quick Format

Start Close

Fig. 1.4-1

Step 2

And then we need to download Raspberry Pi Imager from its official website to install the operating systems.



https://www.raspberrypi.com/software/

system.

Install Raspberry Pi OS using Raspberry Pi Imager

Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi. Watch our 45-second video to learn how to install an operating system using Raspberry Pi Imager.

Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.

To install on Raspberry Pi OS, type

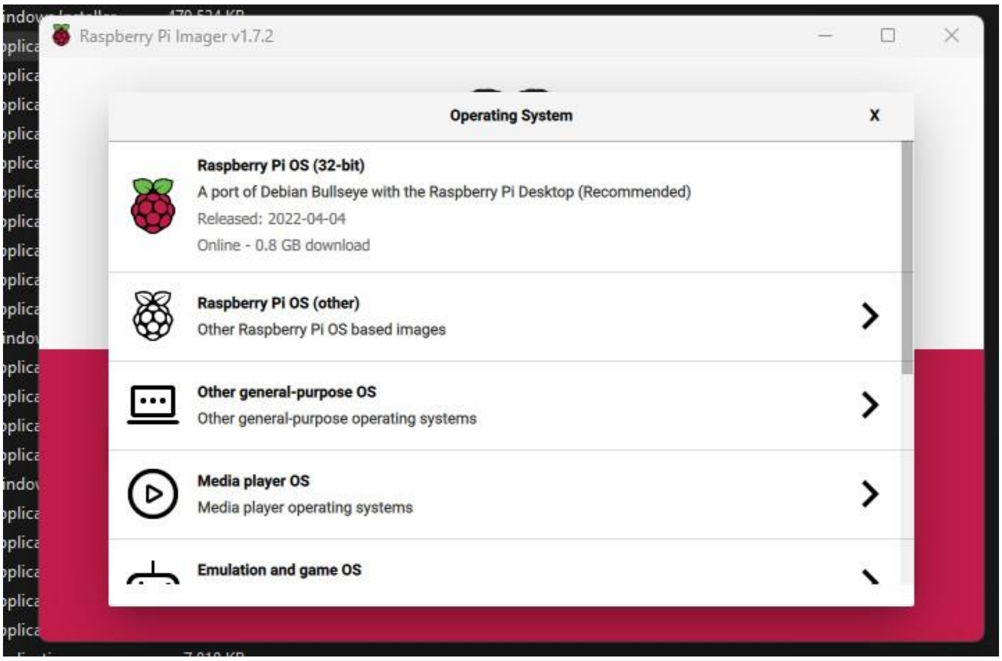
```
sudo apt install rpi-imager
```

in a Terminal window.

Fig. 1.4-2

Step 3

After that, in the app we need to select what operating system we want to install, in this case we choose Raspberry OS.



Raspberry Pi OS (32-bit)
A port of Debian Bullseye with the Raspberry Pi Desktop (Recommended)
Released: 2022-04-04
Online - 0.8 GB download

Raspberry Pi OS (other)
Other Raspberry Pi OS based images

Other general-purpose OS
Other general-purpose operating systems

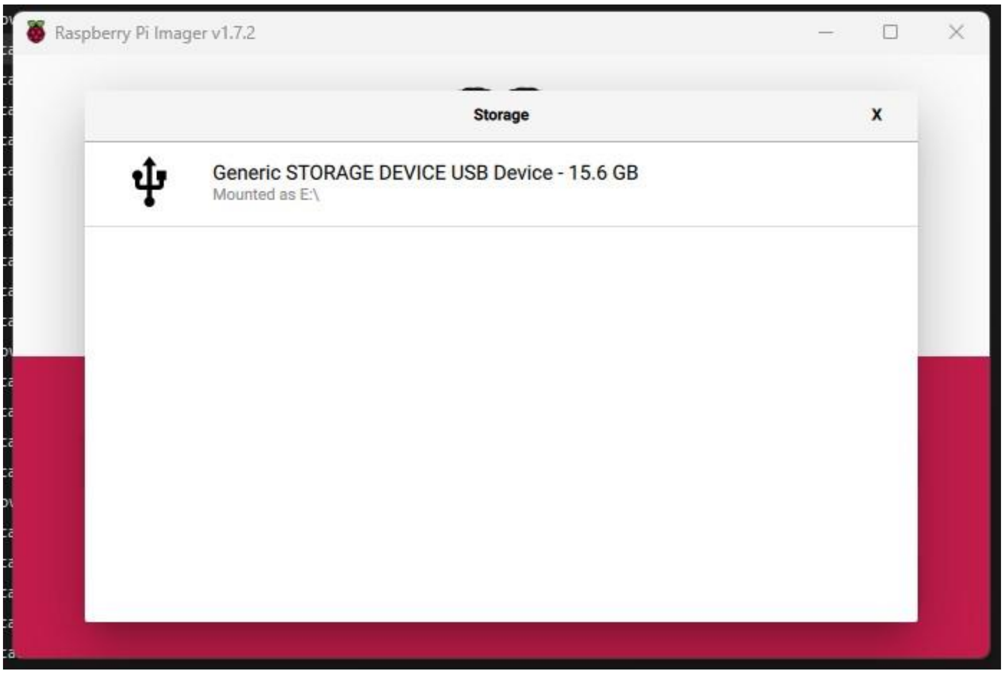
Media player OS
Media player operating systems

Emulation and game OS

Fig. 1.4-3

Step 4

Next up is we need to choose the sdcard that we want to flash.

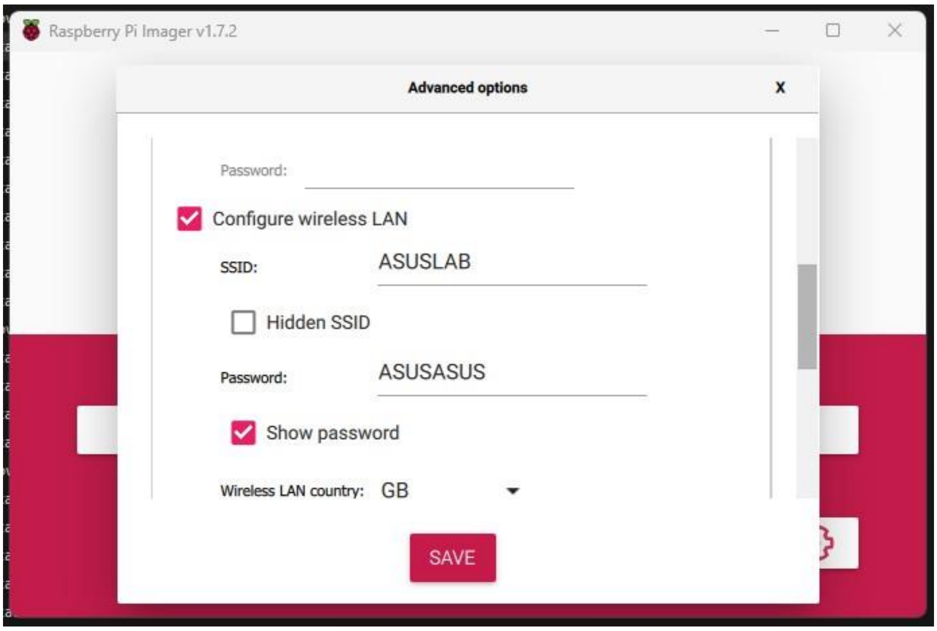


Generic STORAGE DEVICE USB Device - 15.6 GB
Mounted as E:\

Fig. 1.4-4

Step 5

Last one is to do some setting where it will by default connect to the wifi network.



Password:

☑ Configure wireless LAN

SSID: ASUSLAB

□ Hidden SSID

Password: ASUSASUS

☑ Show password

Wireless LAN country: GB

Fig. 1.4-5

Step 6

And then press the write button to flash the sdcard with the Raspberry OS.

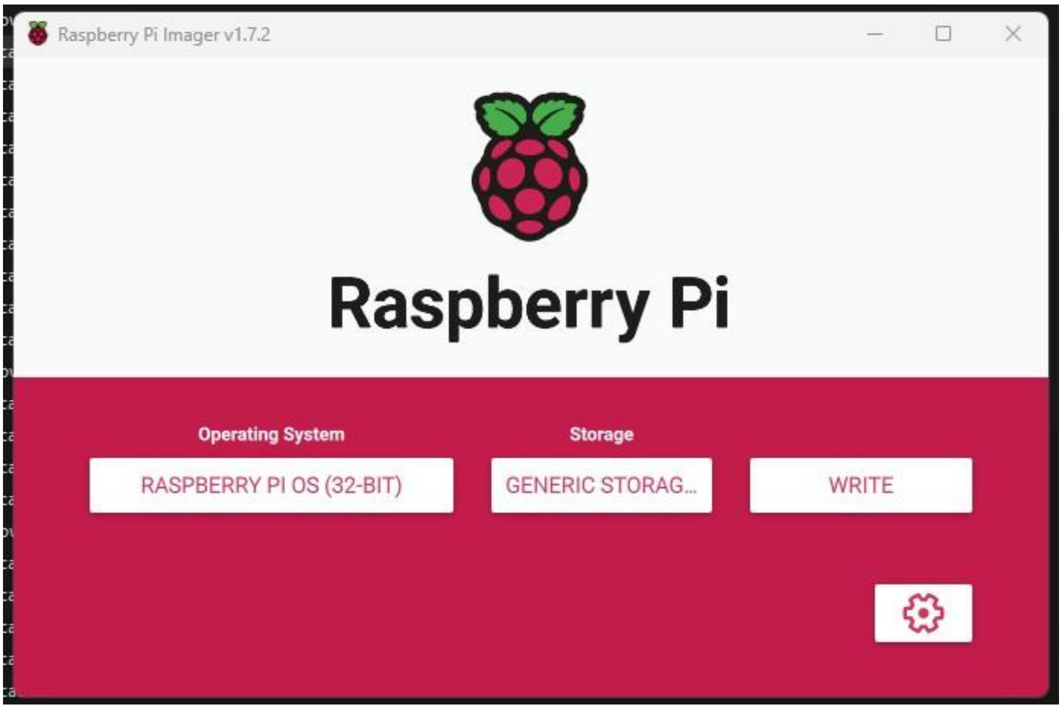


Fig. 1.4-6

Step 7

And in the bottom of the app, it will show the install progress bar and wait until it finish.

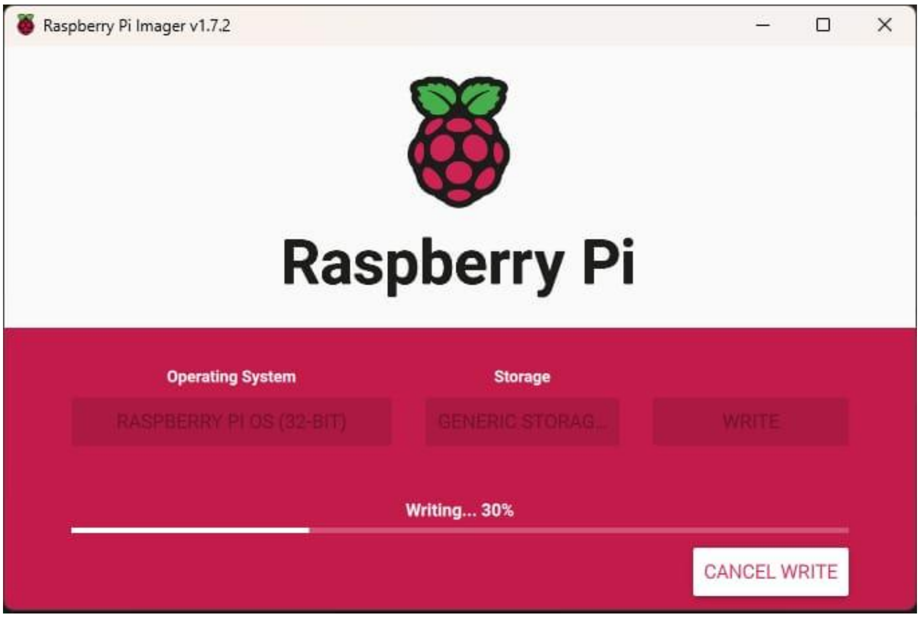
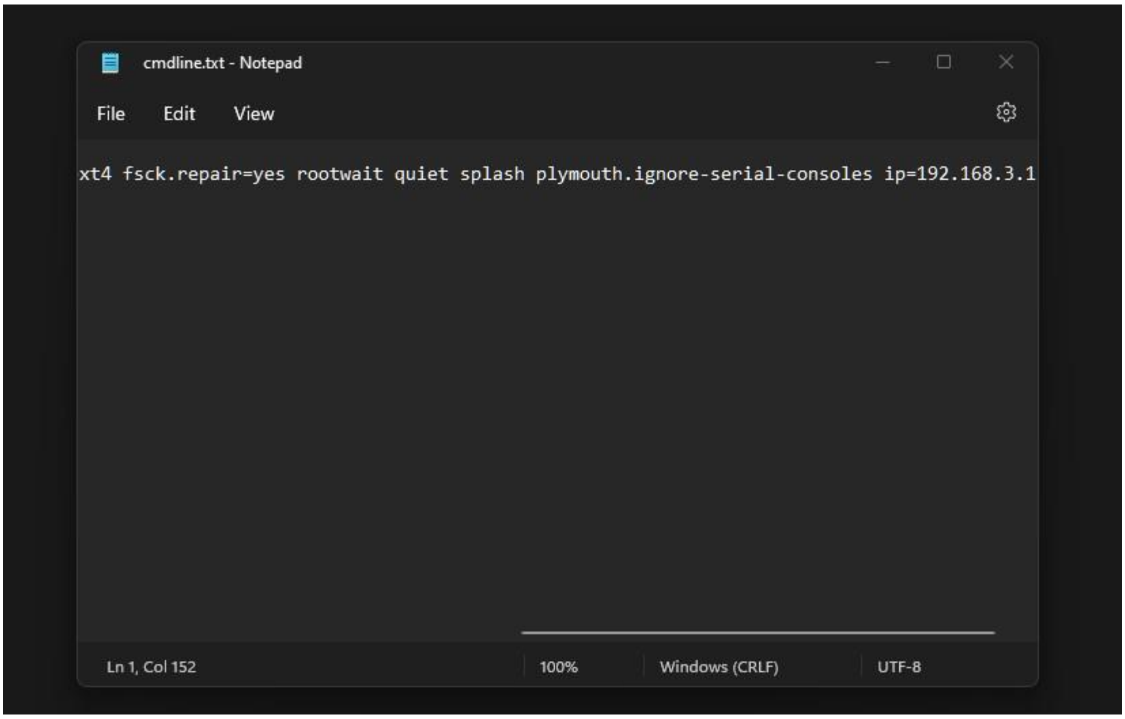


Fig. 1.4-7

Step 8

After done installing, then we open the “cmdline.txt” file and we add “ip=192.168.3.1” in the end of the line.

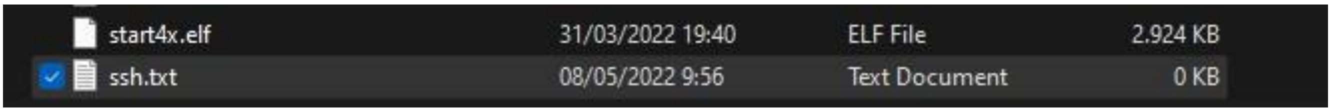


```
xt4 fsck.repair=yes rootwait quiet splash plymouth.ignore-serial-consoles ip=192.168.3.1
```

Fig. 1.4-8

Step 9

Also we add new file name “ssh.txt” and keep it empty.

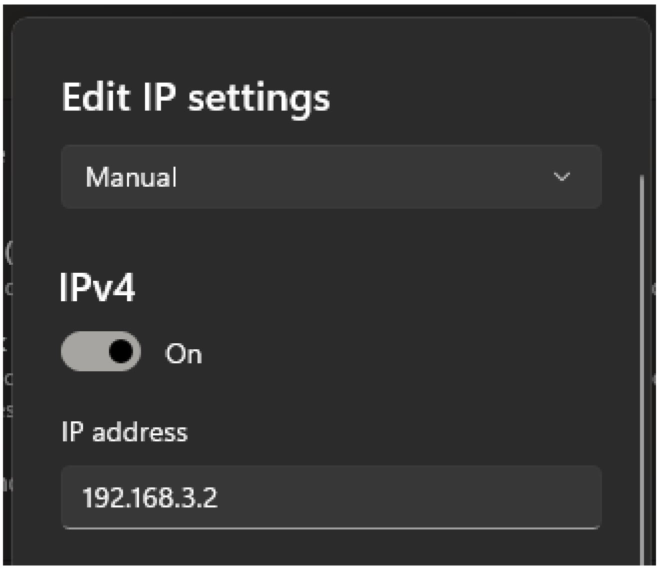


start4x.elf	31/03/2022 19:40	ELF File	2.924 KB
☑ ssh.txt	08/05/2022 9:56	Text Document	0 KB

Fig. 1.4-9

Step 10

After that we can put the sdcard back into the raspberry pi, then connect to the laptop using the lan cable. But before we can access the device using putty, we need to change the laptop's ethernet ip address to 192.168.3.2.



Edit IP settings

Manual

IPv4

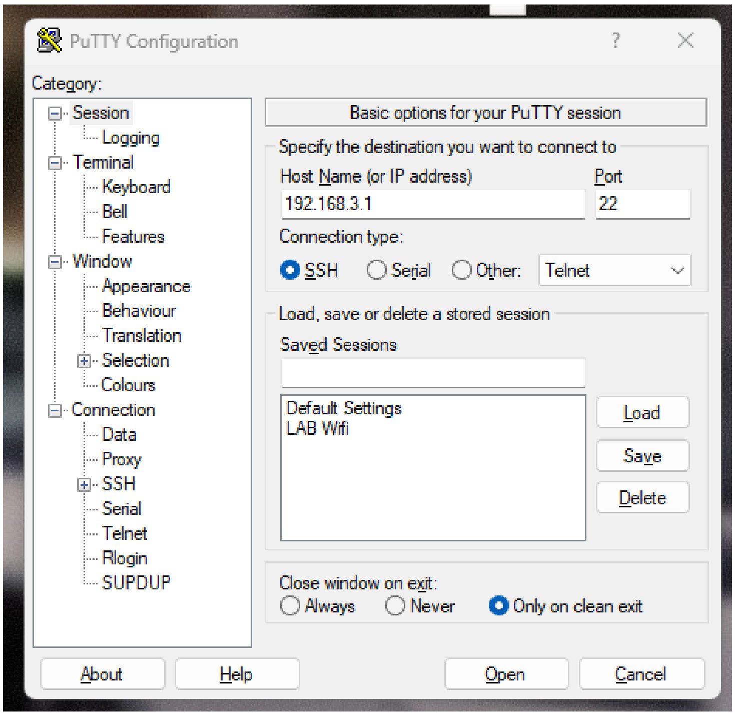
On

IP address: 192.168.3.2

Fig. 1.4-10

Step 11

Next we try to access the device using putty, use 192.168.3.1 in the ip address, SSH connection type, and press open.



PuTTY Configuration

Category:

- Session
  - Logging
- Terminal
  - Keyboard
  - Bell
  - Features
- Window
  - Appearance
  - Behaviour
  - Translation
  - Selection
  - Colours
- Connection
  - Data
  - Proxy
  - SSH
  - Serial
  - Telnet
  - Rlogin
  - SUPDUP

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 192.168.3.1

Port: 22

Connection type: ● SSH ○ Serial ○ Other: Telnet

Load, save or delete a stored session

Saved Sessions

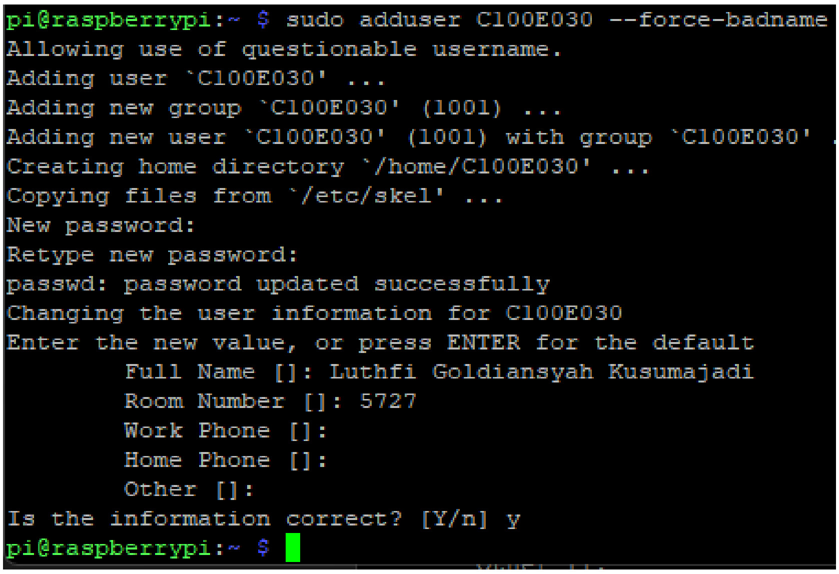
Default Settings
LAB Wifi

Close window on exit: ○ Always ○ Never ● Only on clean exit

Fig. 1.4-11

Step 12

Next, what we need to do is to create new user.

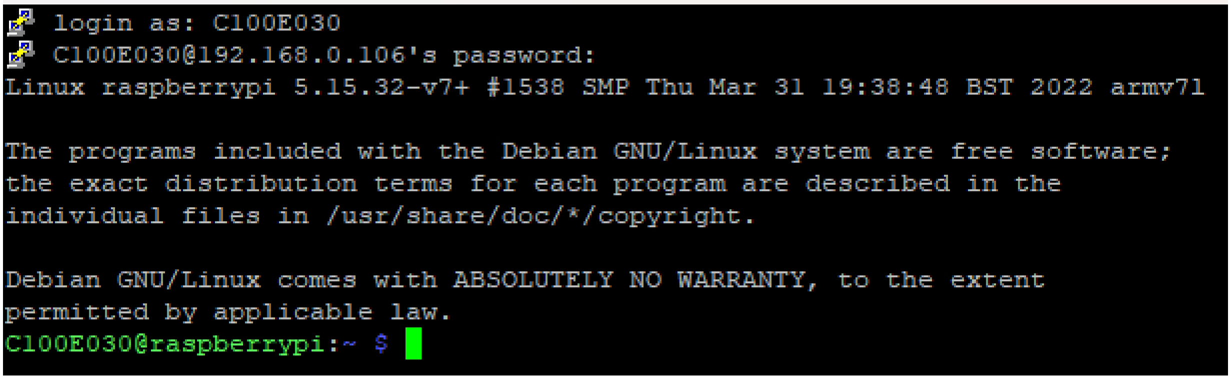


```
pi@raspberrypi:~ $ sudo adduser C100E030 --force-badname
Allowing use of questionable username.
Adding user `C100E030' ...
Adding new group `C100E030' (1001) ...
Adding new user `C100E030' (1001) with group `C100E030'
Creating home directory `/home/C100E030' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for C100E030
Enter the new value, or press ENTER for the default
        Full Name []: Luthfi Goldiansyah Kusumajadi
        Room Number []: 5727
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
pi@raspberrypi:~ $
```

Fig. 1.4.12

Step 13

Next we try to login with the new user we just created.

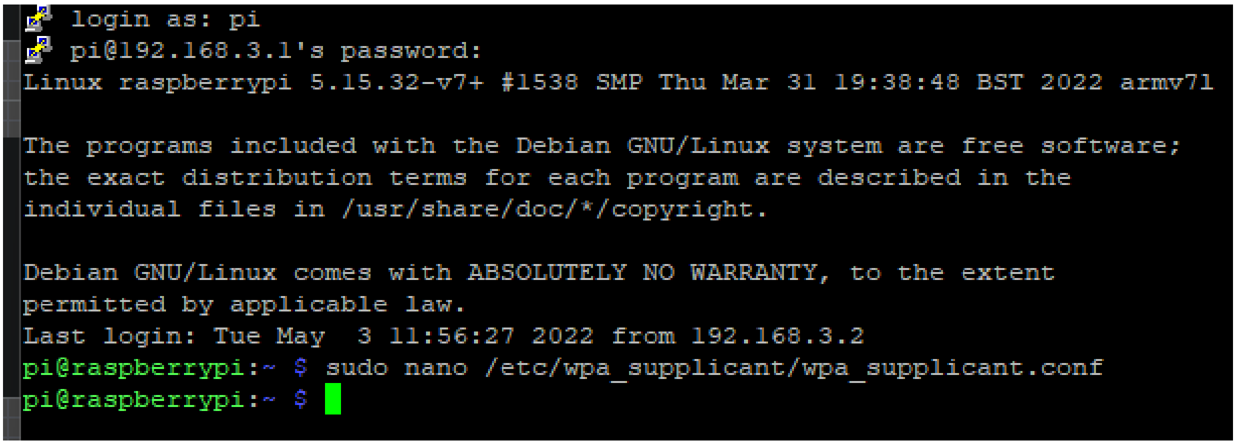


```
login as: C100E030
C100E030@192.168.0.106's password:
Linux raspberrypi 5.15.32-v7+ #1538 SMP Thu Mar 31 19:38:48 BST 2022 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
C100E030@raspberrypi:~ $
```

Fig. 1.4.13

Step 14

After that we need to connect our device to the wifi network, but because it need administrator permission, we need to login again using the pi account.

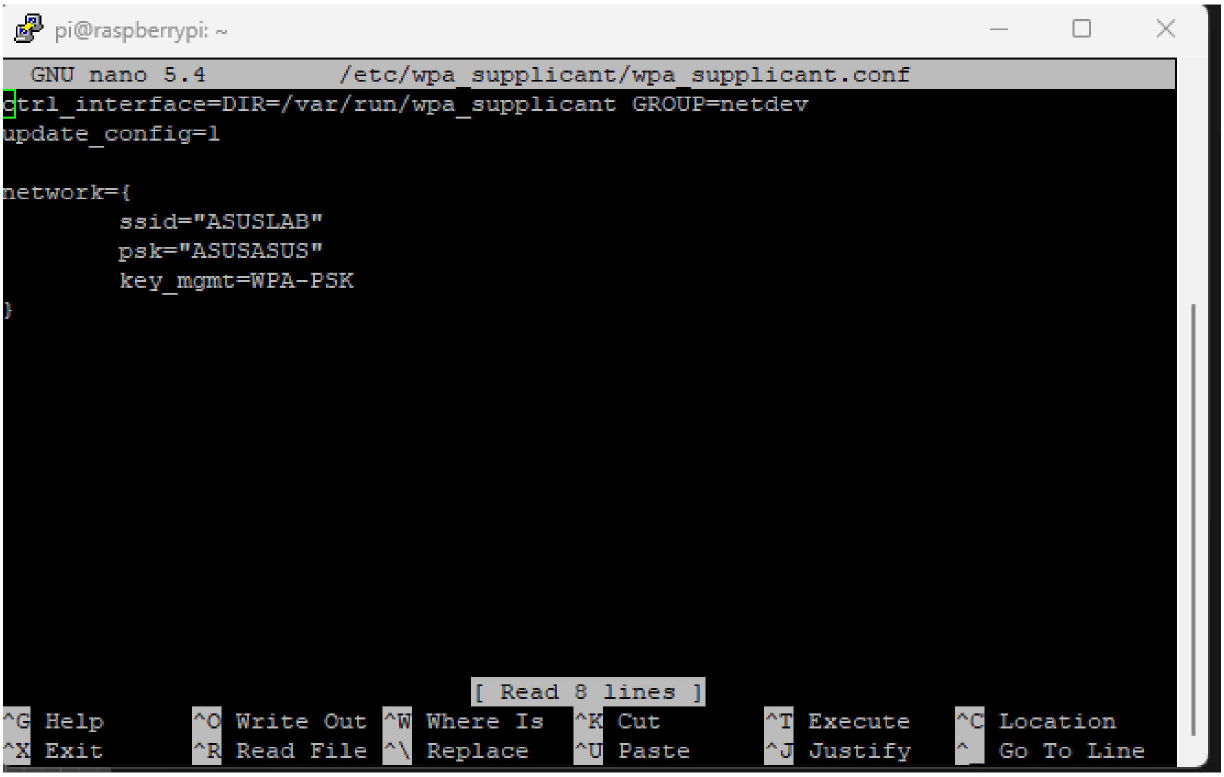


```
login as: pi
pi@192.168.3.1's password:
Linux raspberrypi 5.15.32-v7+ #1538 SMP Thu Mar 31 19:38:48 BST 2022 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue May  3 11:56:27 2022 from 192.168.3.2
pi@raspberrypi:~ $ sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
pi@raspberrypi:~ $
```

Fig. 1.4-14

Step 15

Then the terminal will open the wifi config and we need to fill the wifi name and password.

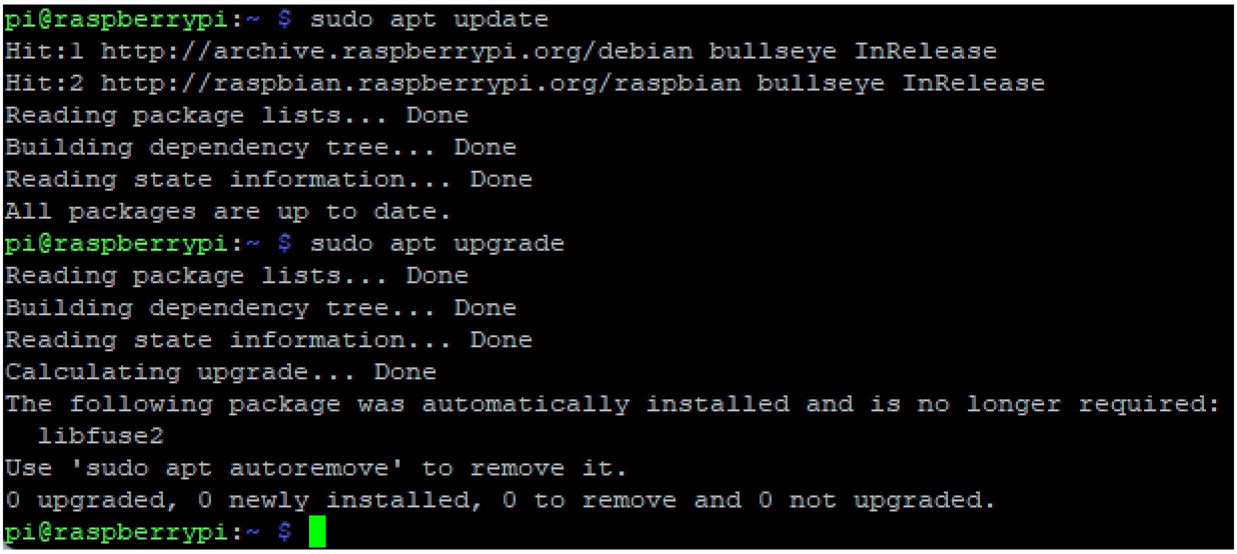


```
  GNU nano 5.4          /etc/wpa_supplicant/wpa_supplicant.conf
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
        ssid="ASUSLAB"
        psk="ASUSASUS"
        key_mgmt=WPA-PSK
}

                                   [ Read 8 lines ]
^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
```

Fig. 1.4-15

Step 16

Then we go to the next stage where we install locahost on our device, but first we need to update and upgrade our device resources.

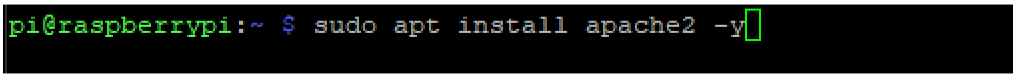


```
pi@raspberrypi:~ $ sudo apt update
Hit:1 http://archive.raspberrypi.org/debian bullseye InRelease
Hit:2 http://raspbian.raspberrypi.org/raspbian bullseye InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
pi@raspberrypi:~ $ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following package was automatically installed and is no longer required:
  libfuse2
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
pi@raspberrypi:~ $
```

Fig. 1.4-16

Step 17

Next we need to install apache.

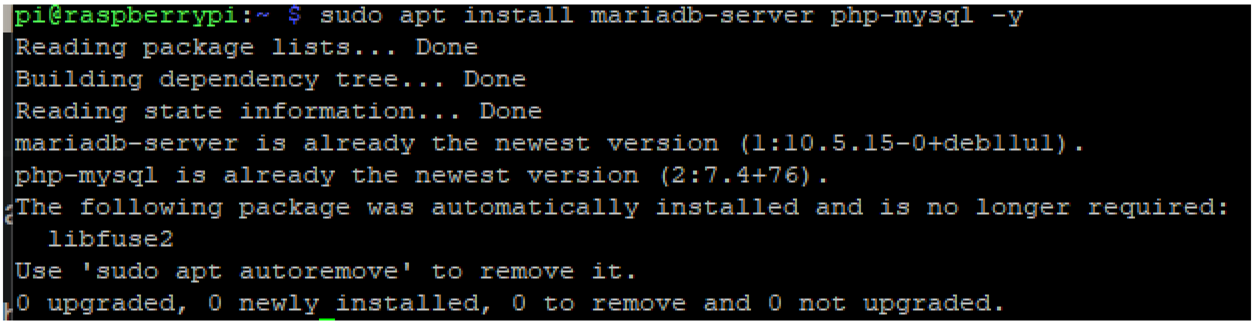


```
pi@raspberrypi:~ $ sudo apt install apache2 -y
```

Fig. 1.4-17

Step 18

Then we install mariaDB.

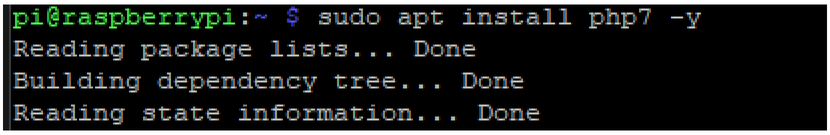


```
pi@raspberrypi:~ $ sudo apt install mariadb-server php-mysql -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
mariadb-server is already the newest version (1:10.5.15-0+deb11u1).
php-mysql is already the newest version (2:7.4+76).
The following package was automatically installed and is no longer required:
  libfuse2
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Fig. 1.4-18

Step 19

Finally we install PHP7.

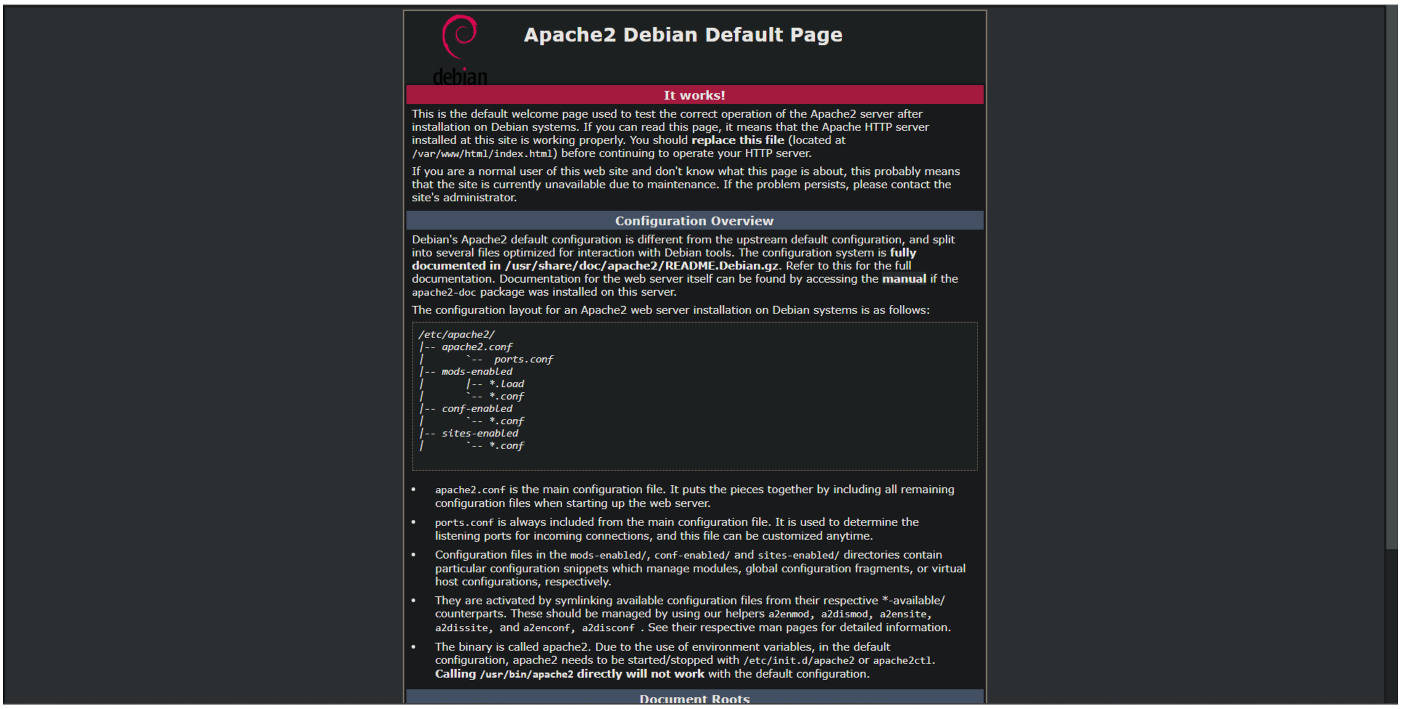


```
pi@raspberrypi:~ $ sudo apt install php7 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Fig. 1.4-19

1.5 Results

Finally we can try if our work is working or not by simply open the device ip address in the browser. If it success it should show something like the image bellow.



Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. 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

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian 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 Debian systems is as follows:

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

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf` . See their respective man pages for detailed information.
- The binary is called apache2. Due to the use of environment variables, in the default configuration, apache2 needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Document Roots

Fig. 1.5-1

1.6 Conclusion and Discussion

In conclusion, setting up the raspberry pi is pretty simple, all you need to do is do the steps above and you should be able to run localhost in your raspberry pi device.