

Real Estate Website with Appointment Booking and Server Monitoring

Student Name: Thiseni Perera

Student Number: 35408344

Domain Name:

<https://prabashrealestates.store>

IP Address Server: 52.63.156.214

GitHub Repo:

Video Link: [TO BE ADDED]

Project Overview

This project involved deploying a real estate website on a cloud-based virtual server. The site allows clients to view properties and book appointments online. Key technologies used include:

- Ubuntu Linux VM
- Apache web server
- WordPress + Elementor
- Simply Schedule Appointments plugin
- Custom Bash scripts for server status and visitor tracking
- Let's Encrypt for HTTPS

1.Server Setup

Cloud Provider: AWS

OS: Ubuntu Server 20.04

Web Server: Apache2

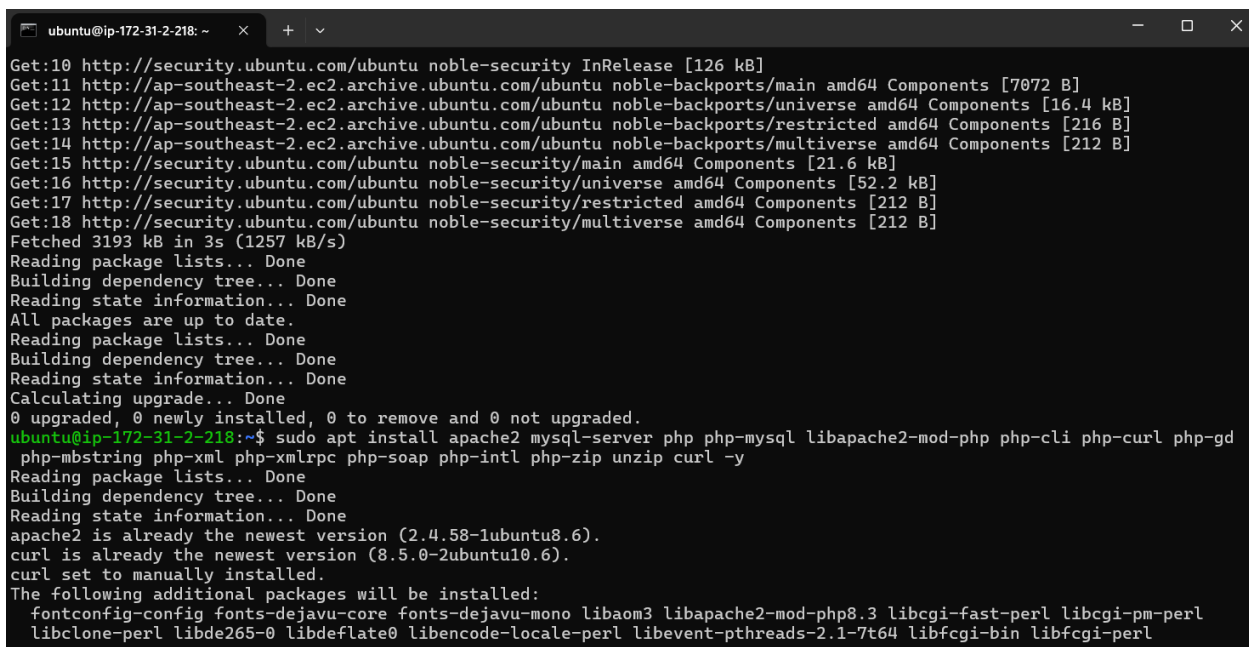
Installation Steps

```
``bash
```

```
sudo apt update
```

```
sudo apt install apache2
```

```
sudo apt install apache2 php mysql-server php-mysql
```



```
ubuntu@ip-172-31-2-218: ~  
Get:10 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]  
Get:11 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7072 B]  
Get:12 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [16.4 kB]  
Get:13 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]  
Get:14 http://ap-southeast-2.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]  
Get:15 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.6 kB]  
Get:16 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.2 kB]  
Get:17 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]  
Get:18 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]  
Fetched 3193 kB in 3s (1257 kB/s)  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
All packages are up to date.  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
Calculating upgrade... Done  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
ubuntu@ip-172-31-2-218:~$ sudo apt install apache2 mysql-server php php-mysql libapache2-mod-php php-cli php-curl php-gd  
php-mbstring php-xml php-xmlrpc php-soap php-intl php-zip unzip curl -y  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
apache2 is already the newest version (2.4.58-1ubuntu8.6).  
curl is already the newest version (8.5.0-2ubuntu10.6).  
curl set to manually installed.  
The following additional packages will be installed:  
  fontconfig-config fonts-dejavu-core fonts-dejavu-mono libaom3 libapache2-mod-php8.3 libbcgi-fast-perl libbcgi-pm-perl  
  libclone-perl libde265-0 libdeflate0 libencode-locale-perl libevent-pthreads-2.1-7t64 libfcgi-bin libfcgi-perl
```

Give all the relevant answers to set up mysql.

```
ubuntu@ip-172-31-2-218: ~  
User sessions running outdated binaries:  
ubuntu @ user manager service: systemd[218026]  
  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
ubuntu@ip-172-31-2-218:~$ sudo mysql_secure_installation  
  
Securing the MySQL server deployment.  
  
Connecting to MySQL using a blank password.  
  
VALIDATE PASSWORD COMPONENT can be used to test passwords  
and improve security. It checks the strength of password  
and allows the users to set only those passwords which are  
secure enough. Would you like to setup VALIDATE PASSWORD component?  
  
Press y|Y for Yes, any other key for No: n  
  
Skipping password set for root as authentication with auth_socket is used by default.  
If you would like to use password authentication instead, this can be done with the "ALTER_USER" command.  
See https://dev.mysql.com/doc/refman/8.0/en/alter-user.html#alter-user-password-management for more information.  
  
By default, a MySQL installation has an anonymous user,  
allowing anyone to log into MySQL without having to have  
a user account created for them. This is intended only for  
testing, and to make the installation go a bit smoother.  
You should remove them before moving into a production  
environment.  
  
Remove anonymous users? (Press y|Y for Yes, any other key for No) : y|
```

```
ubuntu@ip-172-31-2-218: ~  
Skipping password set for root as authentication with auth_socket is used by default.  
If you would like to use password authentication instead, this can be done with the "ALTER_USER" command.  
See https://dev.mysql.com/doc/refman/8.0/en/alter-user.html#alter-user-password-management for more information.  
  
By default, a MySQL installation has an anonymous user,  
allowing anyone to log into MySQL without having to have  
a user account created for them. This is intended only for  
testing, and to make the installation go a bit smoother.  
You should remove them before moving into a production  
environment.  
  
Remove anonymous users? (Press y|Y for Yes, any other key for No) : y  
Success.  
  
Normally, root should only be allowed to connect from  
'localhost'. This ensures that someone cannot guess at  
the root password from the network.  
  
Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y  
Success.  
  
By default, MySQL 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.  
  
Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y|
```

```
environment.  
  
Remove anonymous users? (Press y|Y for Yes, any other key for No) : y  
Success.  
  
Normally, root should only be allowed to connect from  
'localhost'. This ensures that someone cannot guess at  
the root password from the network.  
  
Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y  
Success.  
  
By default, MySQL 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.  
  
Remove test database and access to it? (Press y|Y for Yes, any other key for No) : 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.  
  
Reload privilege tables now? (Press y|Y for Yes, any other key for No) : |
```

```
ubuntu@ip-172-31-2-218: ~  
Normally, root should only be allowed to connect from  
'localhost'. This ensures that someone cannot guess at  
the root password from the network.  
  
Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y  
Success.  
  
By default, MySQL 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.  
  
Remove test database and access to it? (Press y|Y for Yes, any other key for No) : 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.  
  
Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y  
Success.  
  
All done!  
ubuntu@ip-172-31-2-218:~$ |
```

2. setting up wordpress

Next install wordpress by typing:

```
sudo apt install wordpress
```

```
ubuntu@ip-172-31-2-218:~$ cd /tmp
curl -O https://wordpress.org/latest.tar.gz
tar -xzf latest.tar.gz
sudo mv wordpress /var/www/html/
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %             %         Dload  Upload  Total  Spent    Left     Speed
24 25.6M    24 6322k    0     0  147k      0  0:02:58  0:00:42  0:02:16  150k|
```

Then set it up by creating the database and configuring it.

```
- 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.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
ubuntu@ip-172-31-2-218:~$ sudo mysql -u root -p
CREATE DATABASE wordpress;
CREATE USER 'wpuser'@'localhost' IDENTIFIED BY 'yourpassword';
GRANT ALL PRIVILEGES ON wordpress.* TO 'wpuser'@'localhost';
FLUSH PRIVILEGES;
EXIT;
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.42-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2025, 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 wordpress;
Query OK, 1 row affected (0.02 sec)

mysql> CREATE USER 'wpuser'@'localhost' IDENTIFIED BY 'Mycatisgay';
Query OK, 0 rows affected (0.04 sec)

mysql> |
```


Create an account for the account and user.

```
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> CREATE DATABASE wordpress;  
Query OK, 1 row affected (0.02 sec)  
  
mysql> CREATE USER 'wpuser'@'localhost' IDENTIFIED BY 'Mycatisgay';  
Query OK, 0 rows affected (0.04 sec)  
  
mysql> GRANT ALL PRIVILEGES ON wordpress.* TO 'wpuser'@'localhost';|
```

Copied WordPress to Apache directory:

```
sudo cp -r /usr/share/wordpress/* /var/www/html/
```

```
sudo chown -R www-data:www-data /var/www/html
```

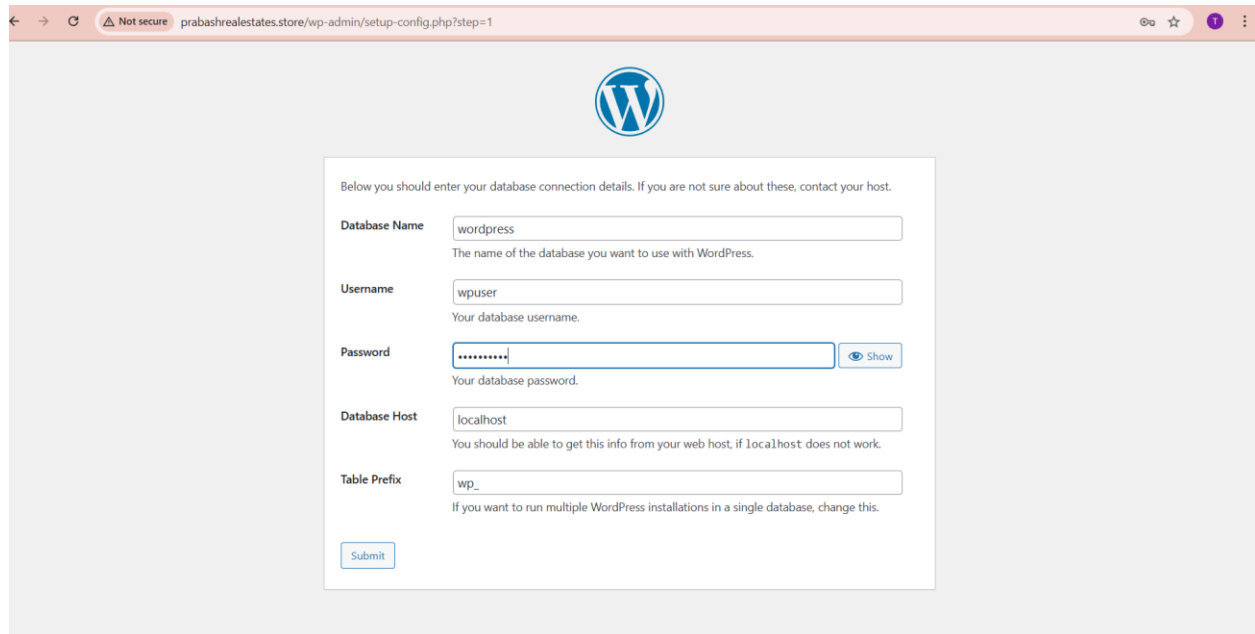
```
ubuntu@ip-172-31-2-218:/tmp$ ls /var/www/html/wordpress  
index.php    wp-activate.php    wp-comments-post.php  wp-cron.php          wp-load.php    wp-settings.php    xmlrpc.php  
license.txt  wp-admin           wp-config-sample.php  wp-includes          wp-login.php   wp-signup.php  
readme.html  wp-blog-header.php wp-content            wp-links-opml.php    wp-mail.php    wp-trackback.php  
ubuntu@ip-172-31-2-218:/tmp$ sudo chown -R www-data:www-data /var/www/html/wordpress  
sudo chmod -R 755 /var/www/html/wordpress  
ubuntu@ip-172-31-2-218:/tmp$ |
```

Restart apache

```
sudo rmdir /var/www/html/wordpress  
ubuntu@ip-172-31-2-218:/tmp$ sudo chown -R www-data:www-data /var/www/html  
sudo chmod -R 755 /var/www/html  
ubuntu@ip-172-31-2-218:/tmp$ sudo systemctl restart apache2  
ubuntu@ip-172-31-2-218:/tmp$ |
```

Configured WordPress in browser at:

<http://52.63.156.214>



The screenshot shows the WordPress installation configuration page in a web browser. The URL bar indicates the page is at `prabashrealestates.store/wp-admin/setup-config.php?step=1`. The page features the WordPress logo at the top center. Below it, a white box contains the database configuration form. The form includes fields for Database Name (wordpress), Username (wpuser), Password (masked with dots), Database Host (localhost), and Table Prefix (wp_). Each field has a descriptive label and a 'Show' button for the password. A 'Submit' button is at the bottom of the form.

Below you should enter your database connection details. If you are not sure about these, contact your host.

Database Name
The name of the database you want to use with WordPress.

Username
Your database username.

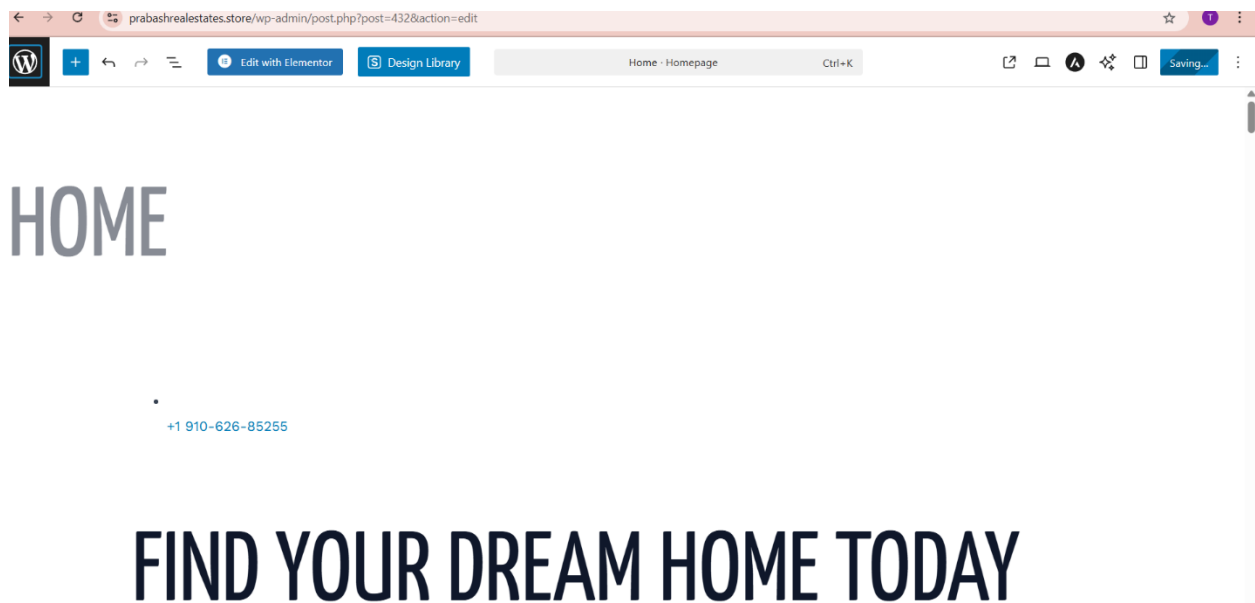
Password [Show](#)
Your database password.

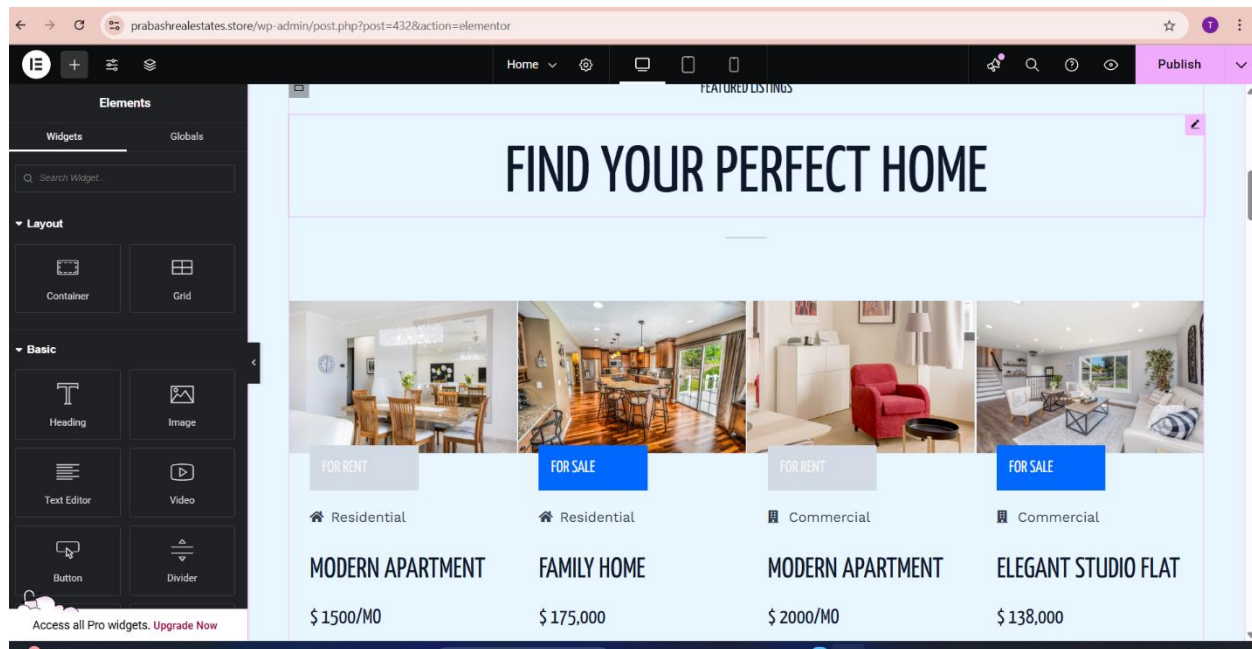
Database Host
You should be able to get this info from your web host, if localhost does not work.

Table Prefix
If you want to run multiple WordPress installations in a single database, change this.

[Submit](#)

Installed **Elementor** and **Simply Schedule Appointments** plugins





Appointment System

Plugin: Simply Schedule Appointments

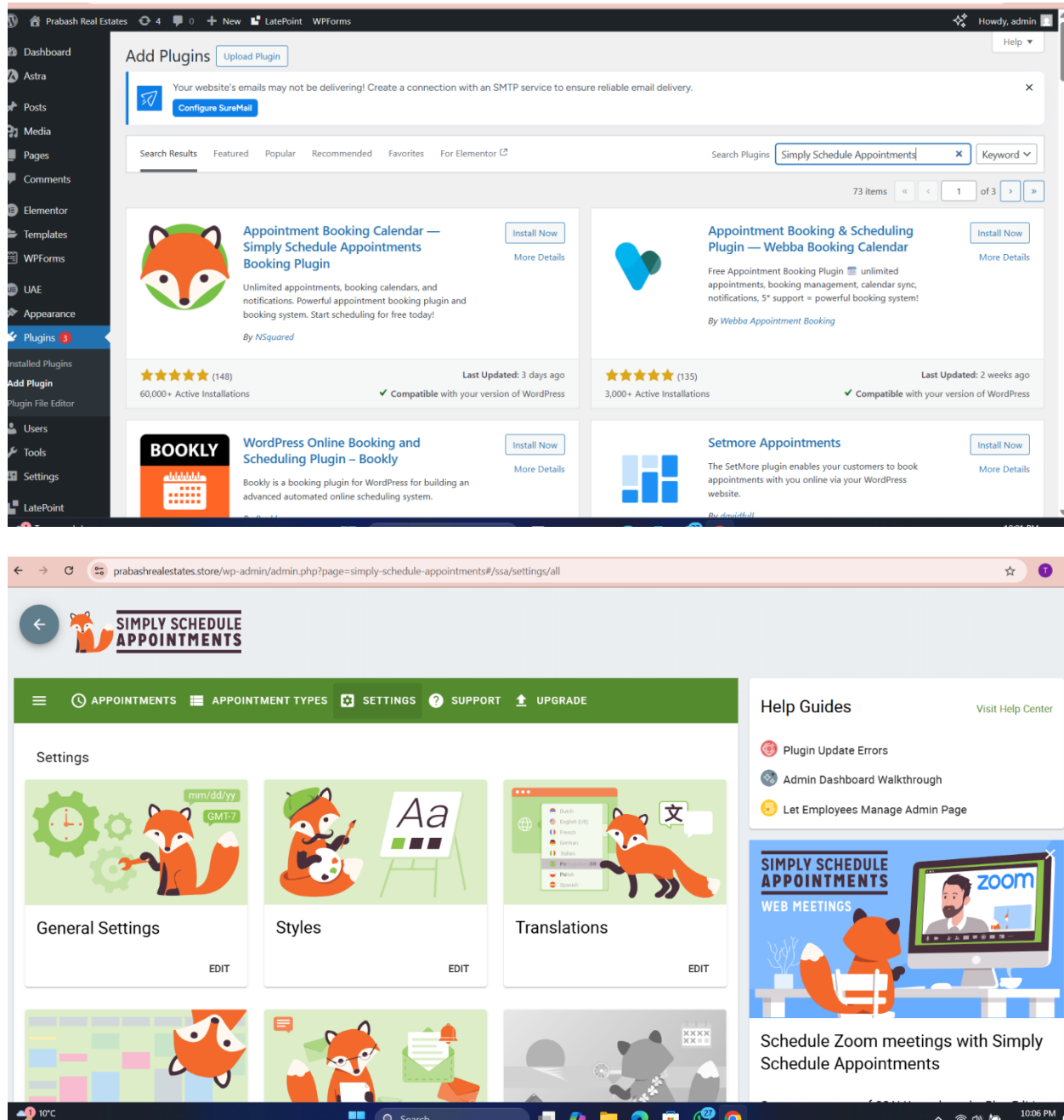
Booking Type: Property Viewing (30 mins)

Availability: Mon–Fri, 9AM–5PM

Setup steps:

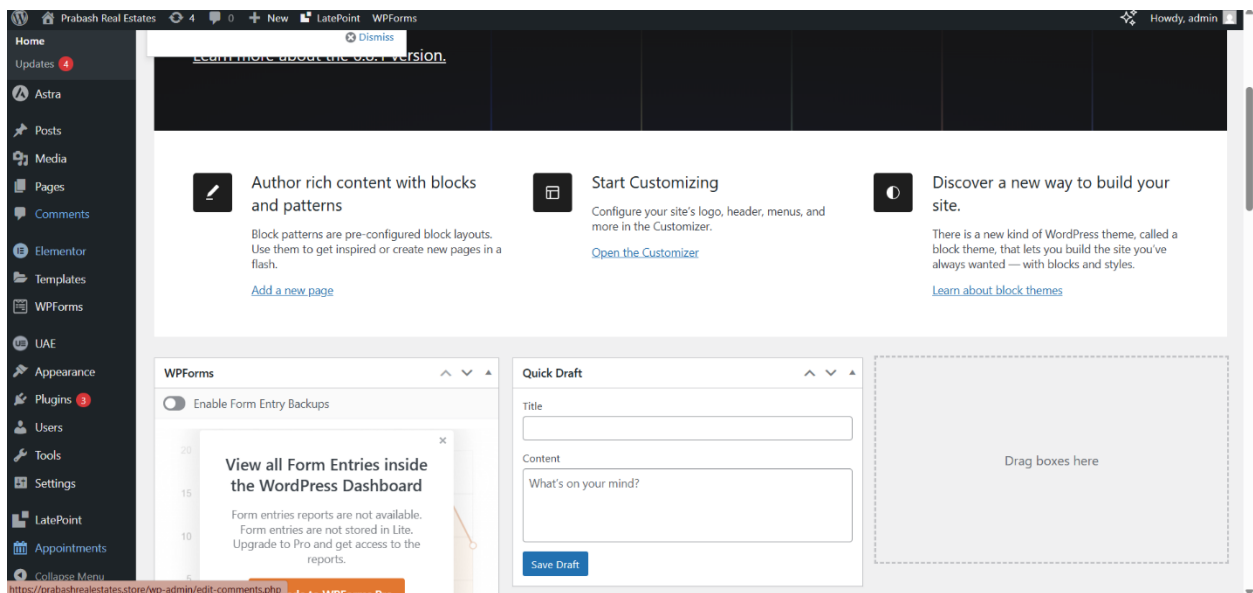
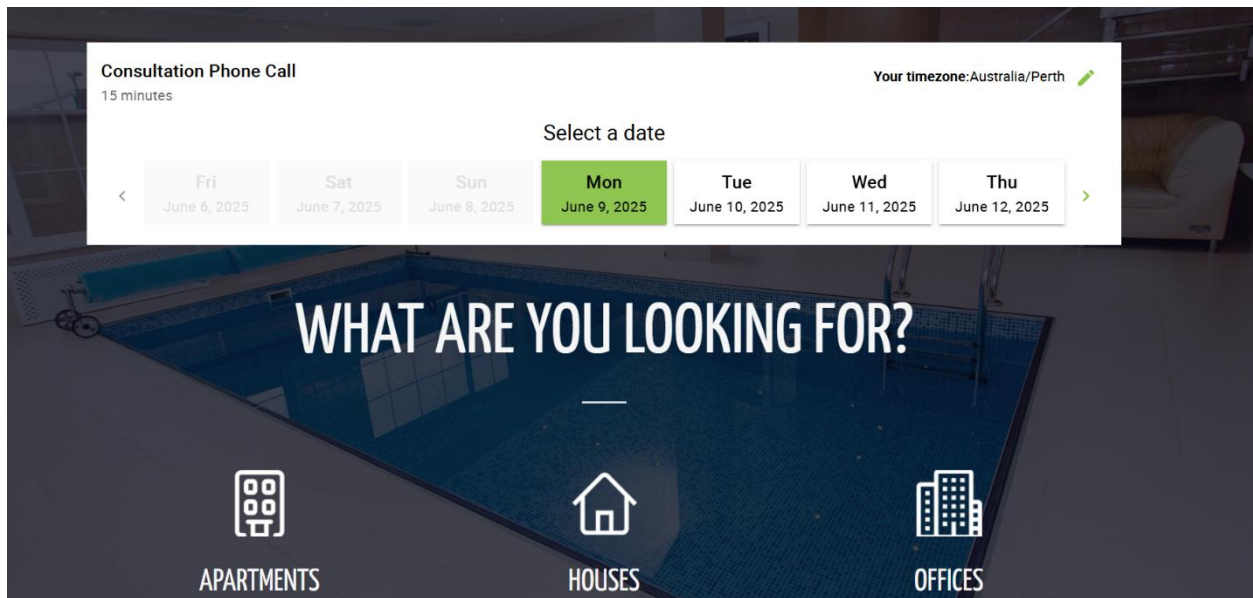
Installed via Plugins - Add New

Configured appointment type via Appointments → Appointment Types



Added booking form to home page via:

[ssa_booking]



View bookings at:

WordPress Admin → Appointments → Appointments

The screenshot shows the WordPress Appointments plugin interface. The top navigation bar is green and contains the following links: [APPOINTMENTS](#), [APPOINTMENT TYPES](#), [SETTINGS](#), [SUPPORT](#), and [UPGRADE](#). Below the navigation bar, there is a section titled "Appointment Range" with a dropdown menu set to "Upcoming Appointments". To the right of this section are three buttons: [SEARCH](#), [FILTER](#), and [EXPORT](#). The main content area displays a list of appointments. The first appointment is for "b fj" on "June 9, 2025 1:00 pm". Below the appointment name, there are icons for a person, a calendar, an envelope, a phone, and a location pin, followed by the details: "b fj", "thiseni04@gmail.com", "+61 402 668 536", and "maharagama". Below these details are two buttons: [VIEW APPOINTMENT DETAILS](#) and [CANCEL THIS APPOINTMENT](#). The second appointment is for "Meth" on "June 9, 2025 2:30 pm". Below the appointment name, there are icons for a person, a calendar, an envelope, and a phone, followed by the details: "Meth", "Uwuwwu@gmail.com", and "+61 481 258 258".

prabashrealestates.store/wp-admin/admin.php?page=simply-schedule-appointments#/ssa/appointments

[+ BOOK AN APPOINTMENT](#)

Appointment Range
Upcoming Appointments

[SEARCH](#) [FILTER](#) [EXPORT](#)

b fj
Consultation Phone Call
June 9, 2025 1:00 pm

b fj
 thiseni04@gmail.com
 +61 402 668 536
 maharagama

[VIEW APPOINTMENT DETAILS](#) [CANCEL THIS APPOINTMENT](#)

Meth
Consultation Phone Call
June 9, 2025 2:30 pm

Meth
 Uwuwwu@gmail.com
 +61 481 258 258

3. DNS Setup

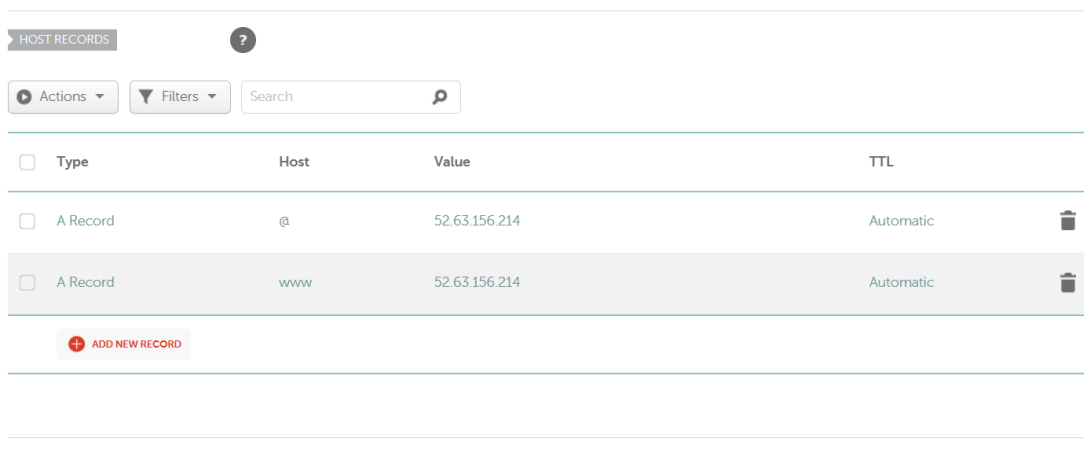
Domain: `prabashrealestates.store` (purchased on Namecheap)

Used Namecheap's DNS to point domain to cloud server

Added A records:

@ - 52.63.156.214

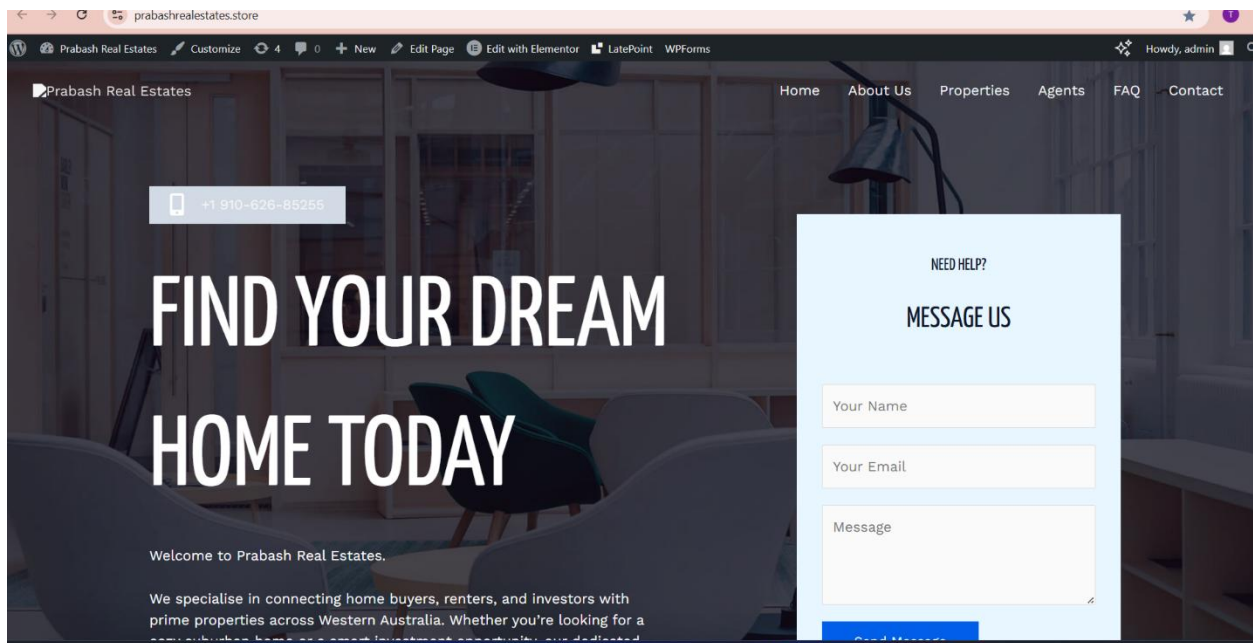
www - 52.63.156.214



The screenshot shows the 'HOST RECORDS' section of a Namecheap account. It features a table with columns for 'Type', 'Host', 'Value', and 'TTL'. Two A records are listed: one for the '@' host and one for the 'www' host, both pointing to the IP address 52.63.156.214 with an 'Automatic' TTL. There are also buttons for 'Actions', 'Filters', and 'Search' at the top, and an 'ADD NEW RECORD' button at the bottom.

Type	Host	Value	TTL
A Record	@	52.63.156.214	Automatic
A Record	www	52.63.156.214	Automatic

Tested by visiting: <http://prabashrealestates.store>



Also tested using:

nslookup prabashrealestates.store

```
ubuntu@ip-172-31-2-218:~$ nslookup prabashrealestates.store
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   prabashrealestates.store
Address: 52.63.156.214

ubuntu@ip-172-31-2-218:~$ |
```


4. SSL/TSL setup

Certbot with Apache plugin:

```
sudo apt install certbot python3-certbot-apache
```

```
sudo certbot --apache
```

stop apache momentarily if it crashes and install certbot in standalone mode.

```
ubuntu@ip-172-31-2-218: ~  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/pro  
  
System information as of Fri Jun  6 11:29:50 UTC 2025  
  
System load:  0.04          Processes:            113  
Usage of /:   50.7% of 6.71GB Users logged in:        0  
Memory usage: 66%          IPv4 address for enX0: 172.31.2.218  
Swap usage:   0%  
  
Expanded Security Maintenance for Applications is not enabled.  
  
0 updates can be applied immediately.  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
Last login: Fri Jun  6 10:52:01 2025 from 203.56.145.104  
ubuntu@ip-172-31-2-218:~$ sudo systemctl stop apache2  
ubuntu@ip-172-31-2-218:~$ sudo apt install certbot -y  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
certbot is already the newest version (2.9.0-1).  
0 upgraded, 0 newly installed, 0 to remove and 9 not upgraded.  
ubuntu@ip-172-31-2-218:~$ |
```

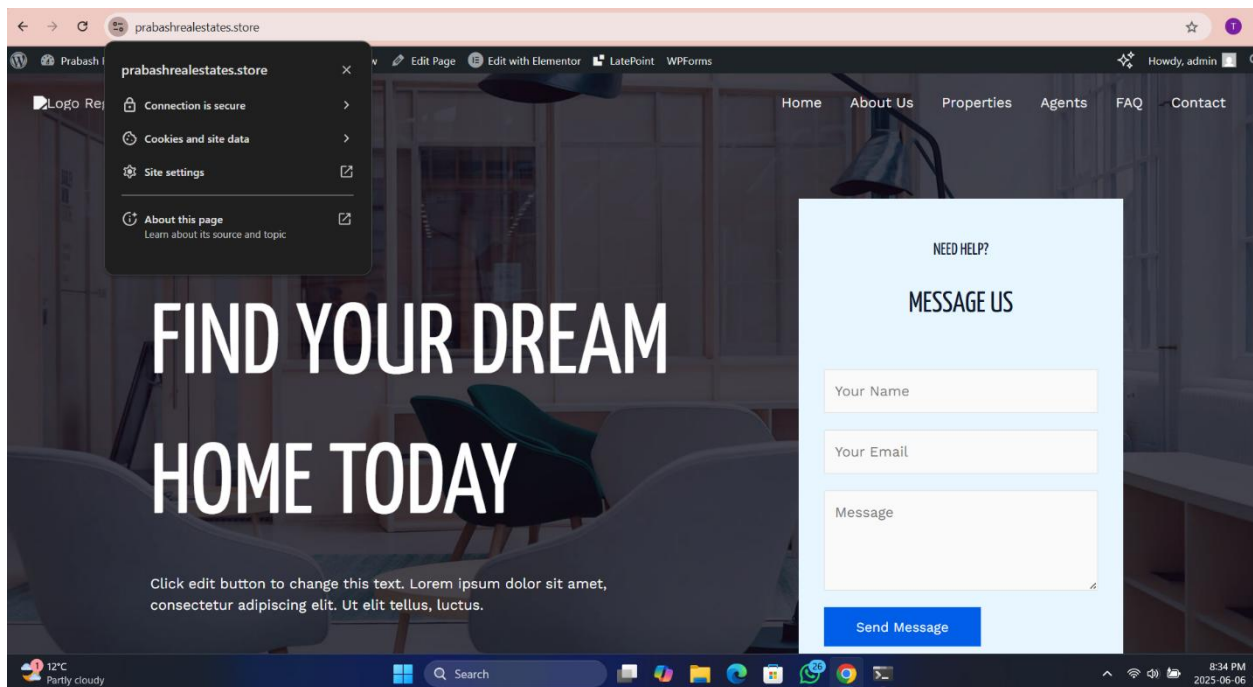
Choose to redirect all HTTP to HTTPS.

```
ypc.log or to run Certbot with --v for more details.
ubuntu@ip-172-31-2-218:~$ sudo certbot certonly --standalone -d prabashrealestates.store -d www.prabashrealestates.store
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Requesting a certificate for prabashrealestates.store and www.prabashrealestates.store

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/prabashrealestates.store/fullchain.pem
Key is saved at: /etc/letsencrypt/live/prabashrealestates.store/privkey.pem
This certificate expires on 2025-09-04.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

-----
If you like Certbot, please consider supporting our work by:
 * Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
 * Donating to EFF: https://eff.org/donate-le
-----
ubuntu@ip-172-31-2-218:~$ |
```

Check if it applies.



5. Uptime_plus_cpu.sh Script

Purpose: Log uptime, CPU, RAM, and disk info to a public file

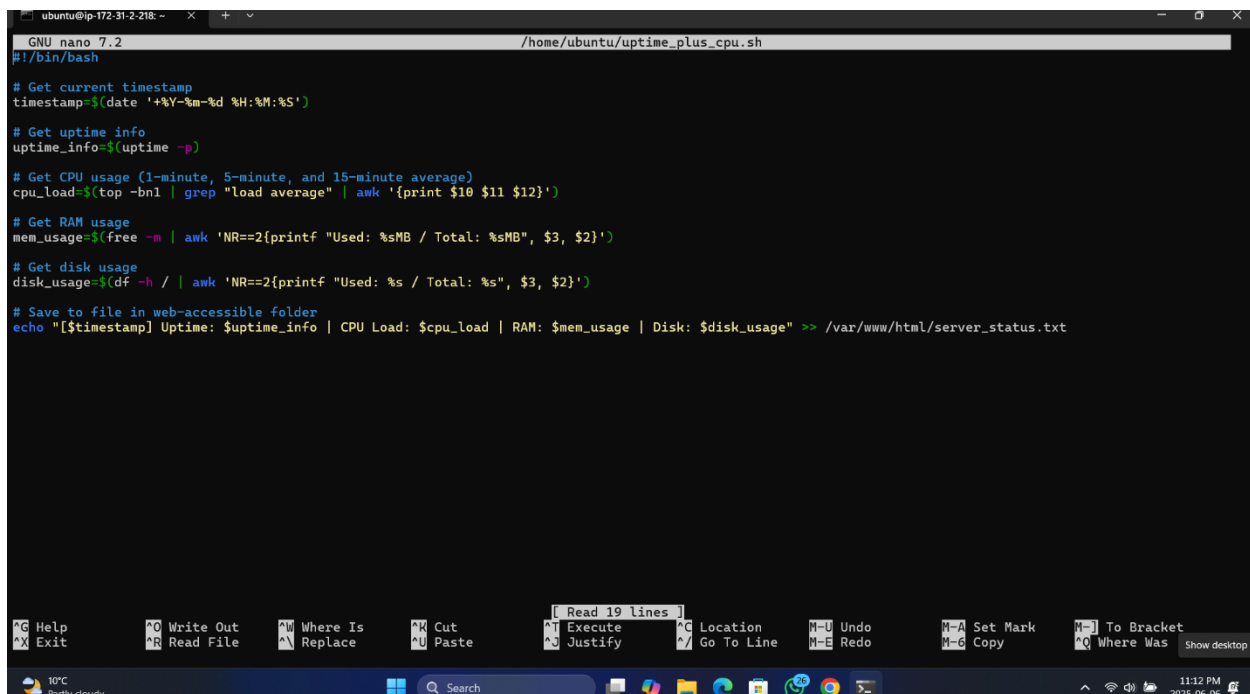
Location: /home/ubuntu/uptime_plus_cpu.sh

Create Script:

nano ~/uptime_plus_cpu.sh

```
ubuntu@ip-172-31-2-218:~$ nano ~/uptime_plus_cpu.sh
ubuntu@ip-172-31-2-218:~$ nano ~/uptime_plus_cpu.sh|
```

Script content:

A screenshot of a terminal window with a dark background. The title bar shows 'ubuntu@ip-172-31-2-218: ~' and the file path '/home/ubuntu/uptime_plus_cpu.sh'. The script content is as follows:

```
GNU nano 7.2 /home/ubuntu/uptime_plus_cpu.sh
#!/bin/bash

# Get current timestamp
timestamp=$(date '+%Y-%m-%d %H:%M:%S')

# Get uptime info
uptime_info=$(uptime -p)

# Get CPU usage (1-minute, 5-minute, and 15-minute average)
cpu_load=$(top -bn1 | grep "load average" | awk '{print $10 $11 $12}')

# Get RAM usage
mem_usage=$(free -m | awk 'NR==2{printf "Used: %sMB / Total: %sMB", $3, $2}')

# Get disk usage
disk_usage=$(df -h | awk 'NR==2{printf "Used: %s / Total: %s", $3, $2}')

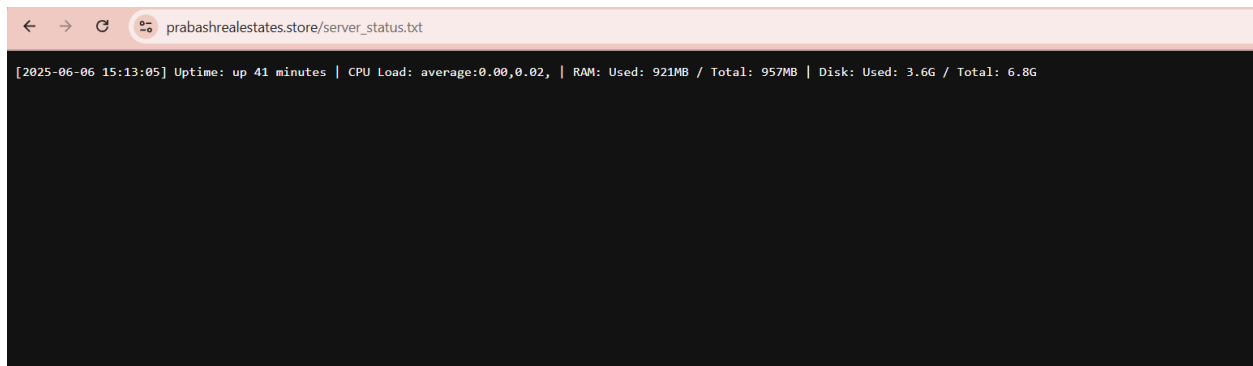
# Save to file in web-accessible folder
echo "[$timestamp] Uptime: $uptime_info | CPU Load: $cpu_load | RAM: $mem_usage | Disk: $disk_usage" >> /var/www/html/server_status.txt
```

The bottom of the window shows the nano editor's command palette with options like Help, Exit, Write Out, Read File, Where Is, Replace, Cut, Paste, Execute, Justify, Location, Go To Line, Undo, Redo, Set Mark, Copy, To Bracket, and Where Was. The system tray at the bottom indicates 10°C, 'Partly cloudy', and the time 11:12 PM on 2025-06-06.

Command used to run:

sudo ./uptime_plus_cpu.sh

https://prabashrealestates.store/server_status.txt



6. Script 2: visitor_logger.sh

Purpose: Log top 5 visitor IPs from Apache logs

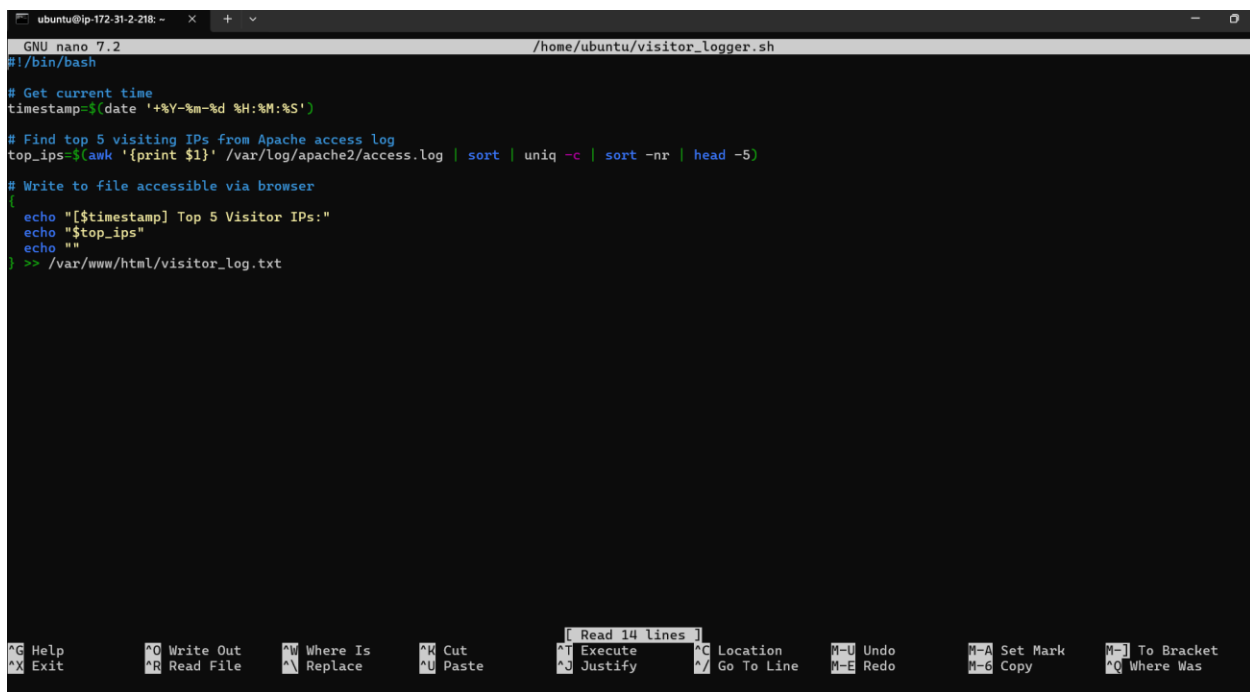
Location: /home/ubuntu/visitor_logger.sh

Create Script:

nano ~/visitor_logger.sh

```
Last login: Fri Jun  6 14:36:12 2025 from 13.239.158.5
ubuntu@ip-172-31-2-218:~$ nano visitor_logger.sh
ubuntu@ip-172-31-2-218:~$ chmod +x visitor_logger.sh
sudo ./visitor_logger.sh
ubuntu@ip-172-31-2-218:~$ |
```

Script content:



```
GNU nano 7.2 /home/ubuntu/visitor_logger.sh
#!/bin/bash

# Get current time
timestamp=$(date '+%Y-%m-%d %H:%M:%S')

# Find top 5 visiting IPs from Apache access log
top_ips=$(awk '{print $1}' /var/log/apache2/access.log | sort | uniq -c | sort -nr | head -5)

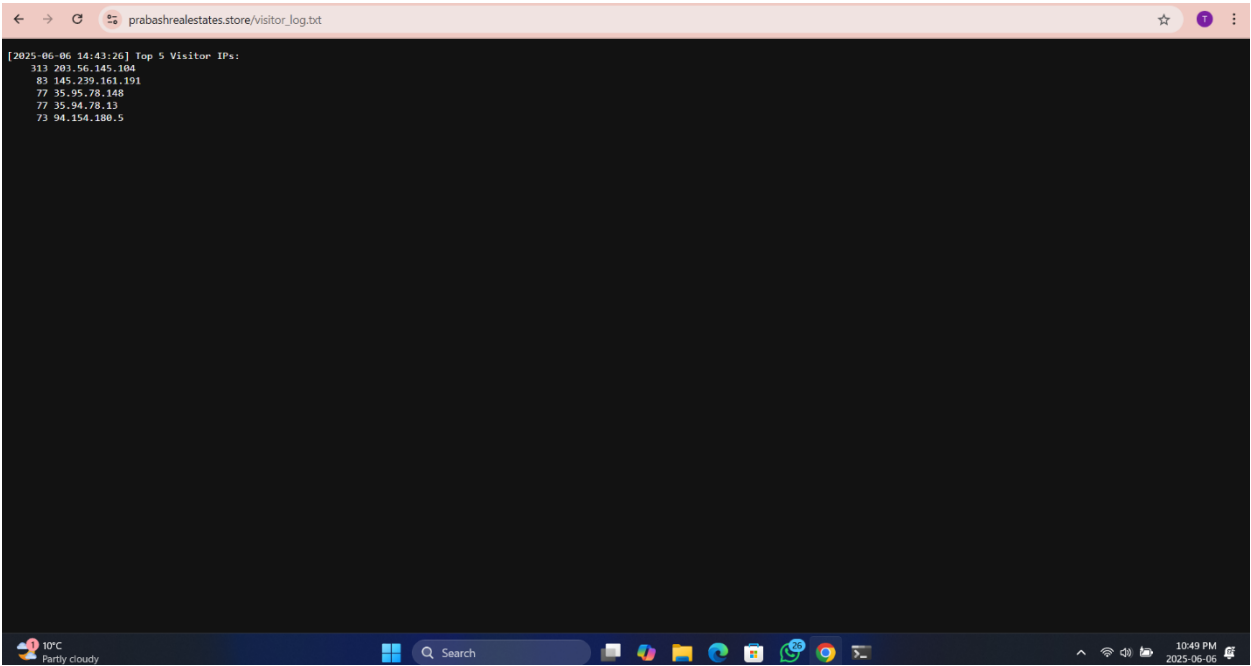
# Write to file accessible via browser
{
    echo "[${timestamp}] Top 5 Visitor IPs:"
    echo "$top_ips"
    echo ""
} >> /var/www/html/visitor_log.txt
```

Command used to run and execute it:

chmod +x visitor_logger.sh

sudo ./visitor_logger.sh

test it at: https://prabashrealestates.store/visitor_log.txt



7. Folder and File Management

Created script folder:

```
mkdir ~/ict171_files
```

```
buntu@ip-172-31-2-218:~$ sudo ./uptime_plus_cpu.sh
buntu@ip-172-31-2-218:~$ nano ~/visitor_logger.sh
buntu@ip-172-31-2-218:~$ mkdir ~/ict171_files
```

Copied files:

```
cp ~/uptime_plus_cpu.sh ~/ict171_files/
```

```
cp ~/visitor_logger.sh ~/ict171_files/
```

```
sudo cp /var/www/html/visitor_log.txt ~/ict171_files/
```

```
sudo cp /var/www/html/server_status.txt ~/ict171_files/
```

```
cp ~/visitor_logger.sh ~/ict171_files/
ubuntu@ip-172-31-2-218:~$ sudo cp /var/www/html/server_status.txt ~/ict171_files/
sudo cp /var/www/html/visitor_log.txt ~/ict171_files/
ubuntu@ip-172-31-2-218:~$ ls ~/ict171_files
server_status.txt  uptime_plus_cpu.sh  visitor_log.txt  visitor_logger.sh
ubuntu@ip-172-31-2-218:~$
```

Zipped folder (Install it):

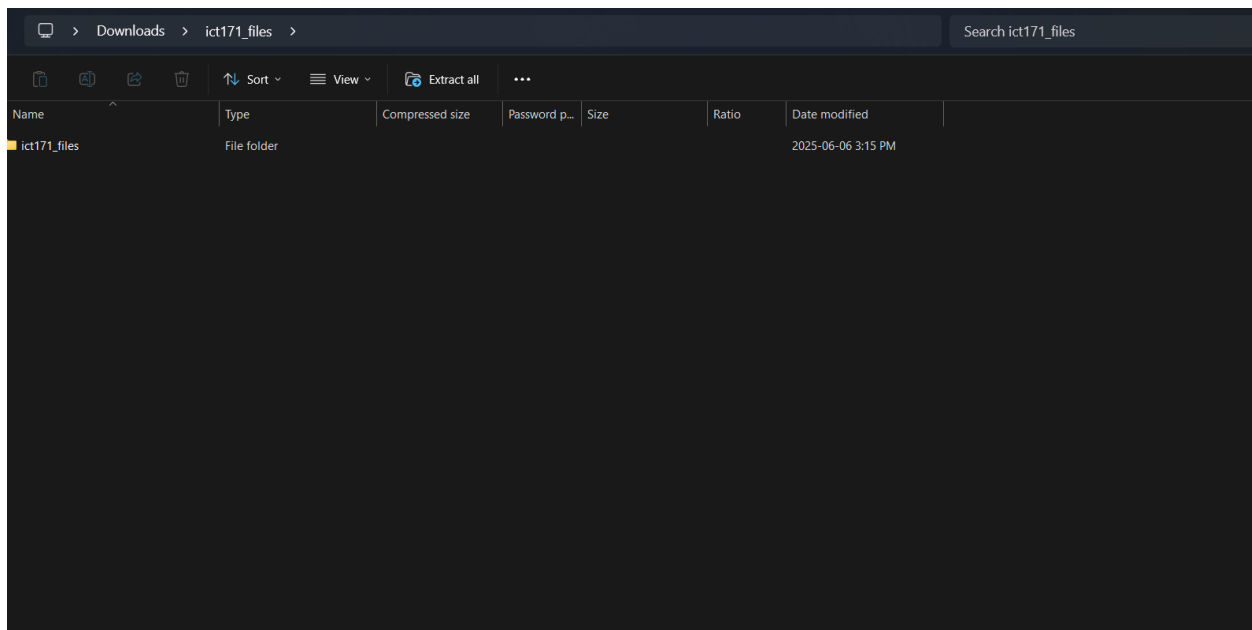
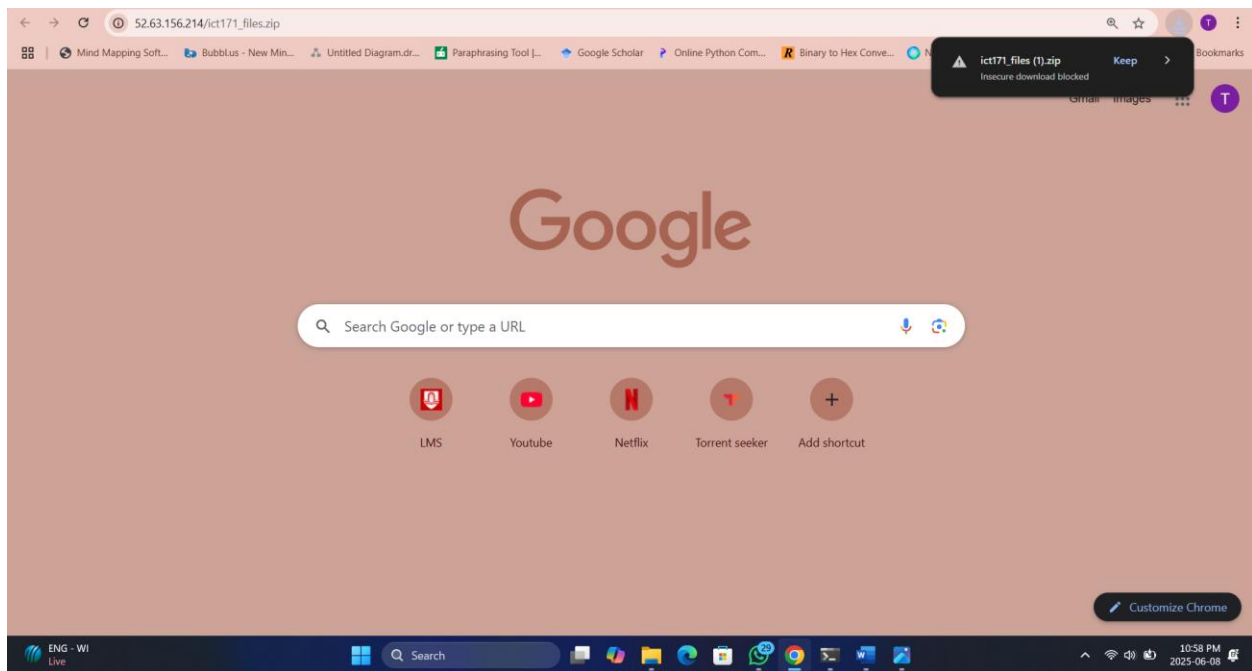
```
ubuntu@ip-172-31-2-218:~$ cd
ubuntu@ip-172-31-2-218:~$ zip -r ict171_files.zip ict171_files
Command 'zip' not found, but can be installed with:
sudo apt install zip
ubuntu@ip-172-31-2-218:~$ sudo apt install zip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done

ubuntu@ip-172-31-2-218:~$ sudo ./uptime_plus_cpu.sh
ubuntu@ip-172-31-2-218:~$ nano ~/visitor_logger.sh
ubuntu@ip-172-31-2-218:~$ cd ~
ubuntu@ip-172-31-2-218:~$ zip -r ict171_files.zip ict171_files
```

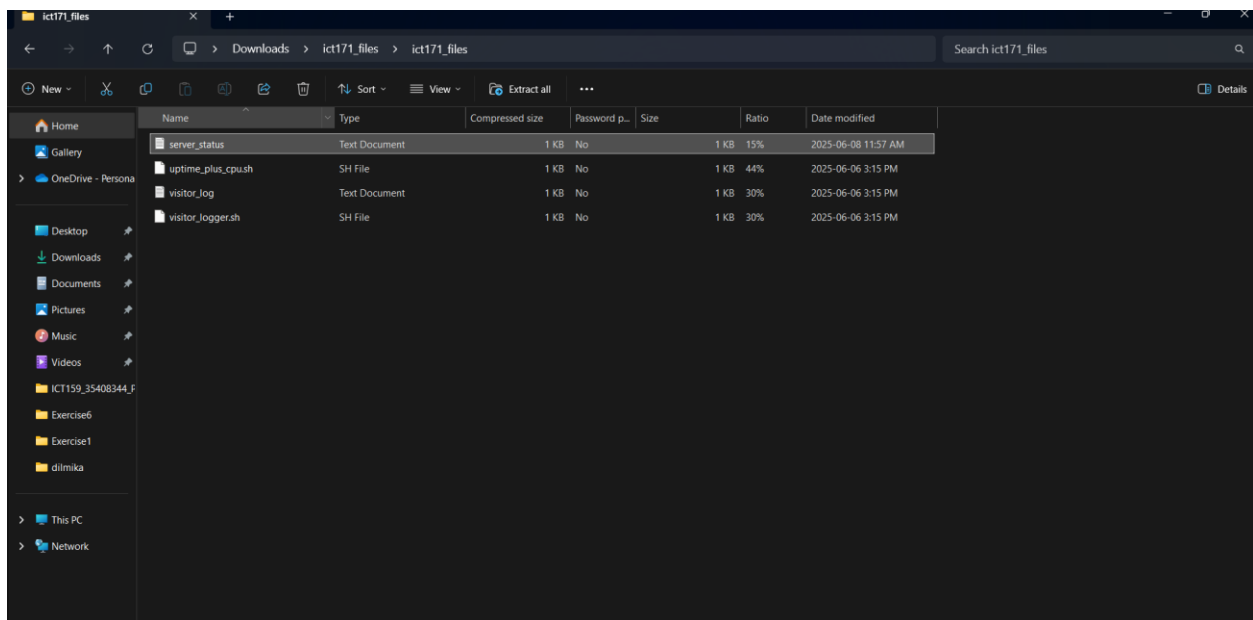
Moved ZIP to public directory:

```
sudo cp ~/ict171_files.zip /var/www/html/
```

Downloaded from: http://52.63.156.214/ict171_files.zip



Open the zipped file:



View the contents of the file:

