4/6/2025

ICT171 Assignment 2

PHISH-SHIELD: Cyber security awareness and phishing

Yididya Mekonnen
ID; 35545566

IP; 64.227.176.85

Domain name; Https://phish-shield.agency

1. Introduction

Phish Shield, as per my presentation, is a cybersecurity awareness platform designed to educate employees of CyberGuard-hub (a hypothetical cyber-security solutions company) on phishing threats through interactive content, quizzes, and real-world simulations. The project was hosted using a cloud Infrastructure-as-a-Service (IaaS) solution

(DigitalOcean), with manual server configuration, DNS linkage, and SSL/TLS implementation (for security).

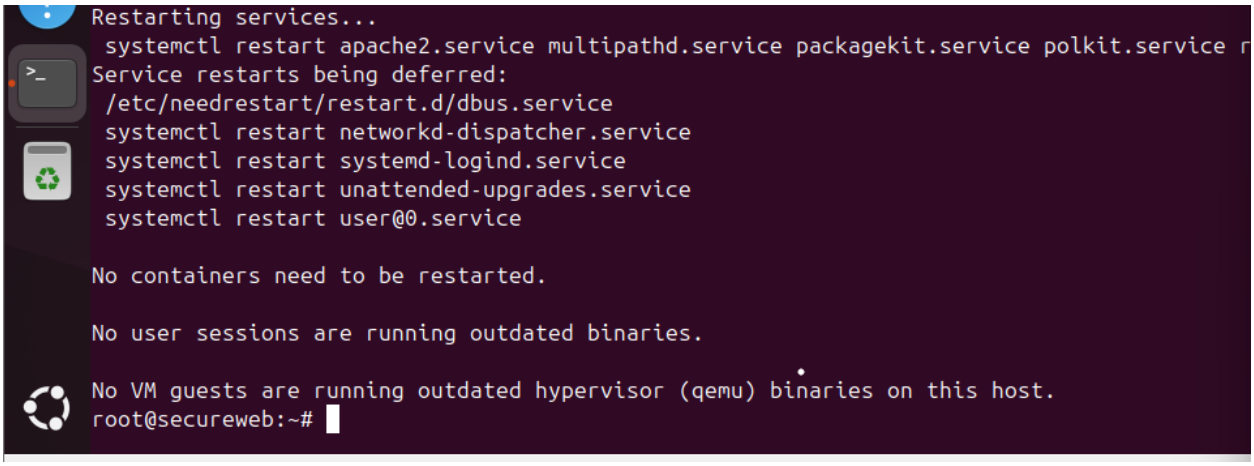
2. Server Setup and Configuration

1) Connecting to server vis ssh;

ssh root@64.227.176.85

2) Updating the server before anything else

sudo apt update and upgrade



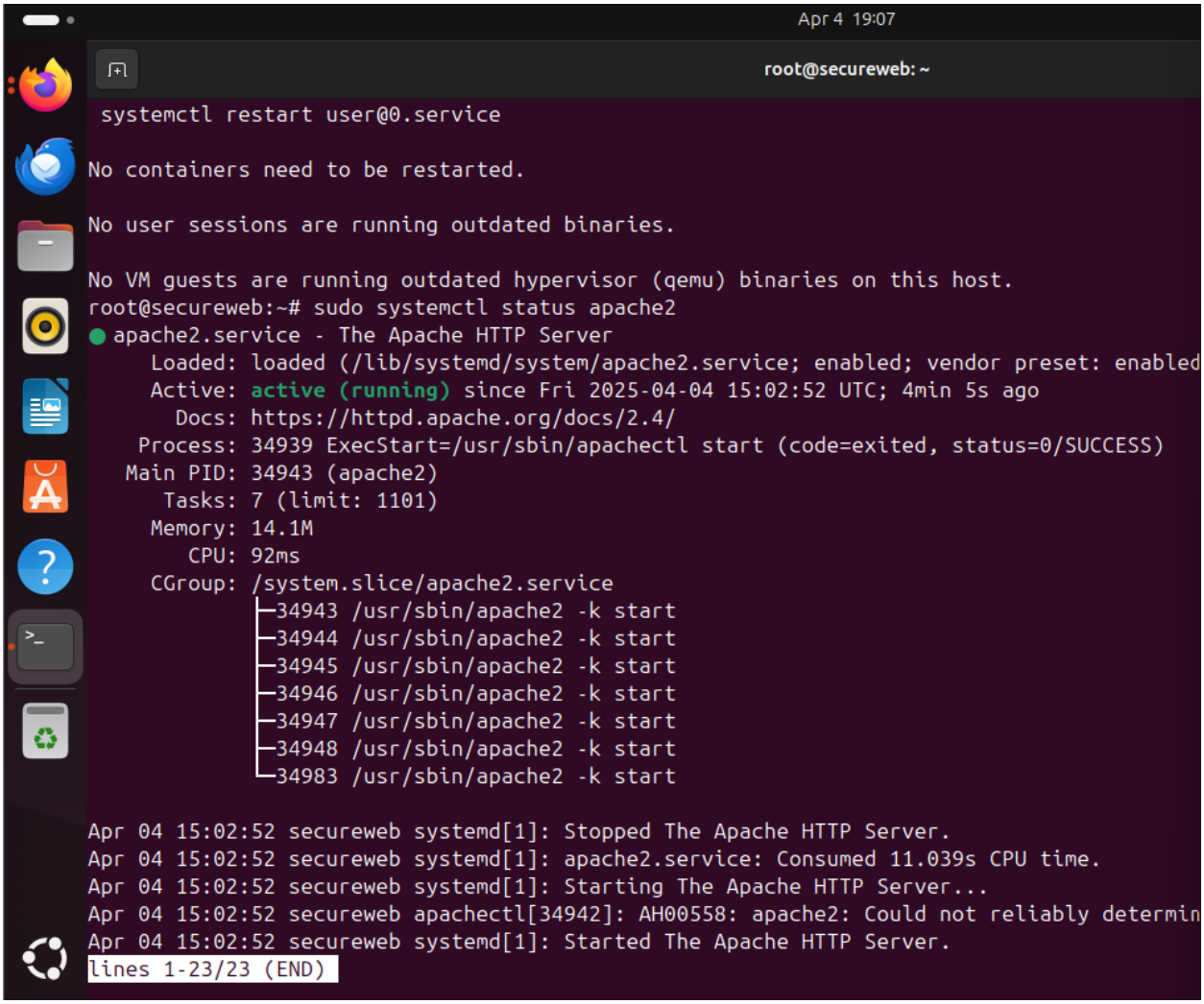
3) Install apache web services;

sudo apt install apache2 (install apache)

sudo systemctl enable apache2 (enable after installment)

sudo systemctl start apache2 (start apache)

sudo systemctl status apache2 (check if status is active after starting apache)

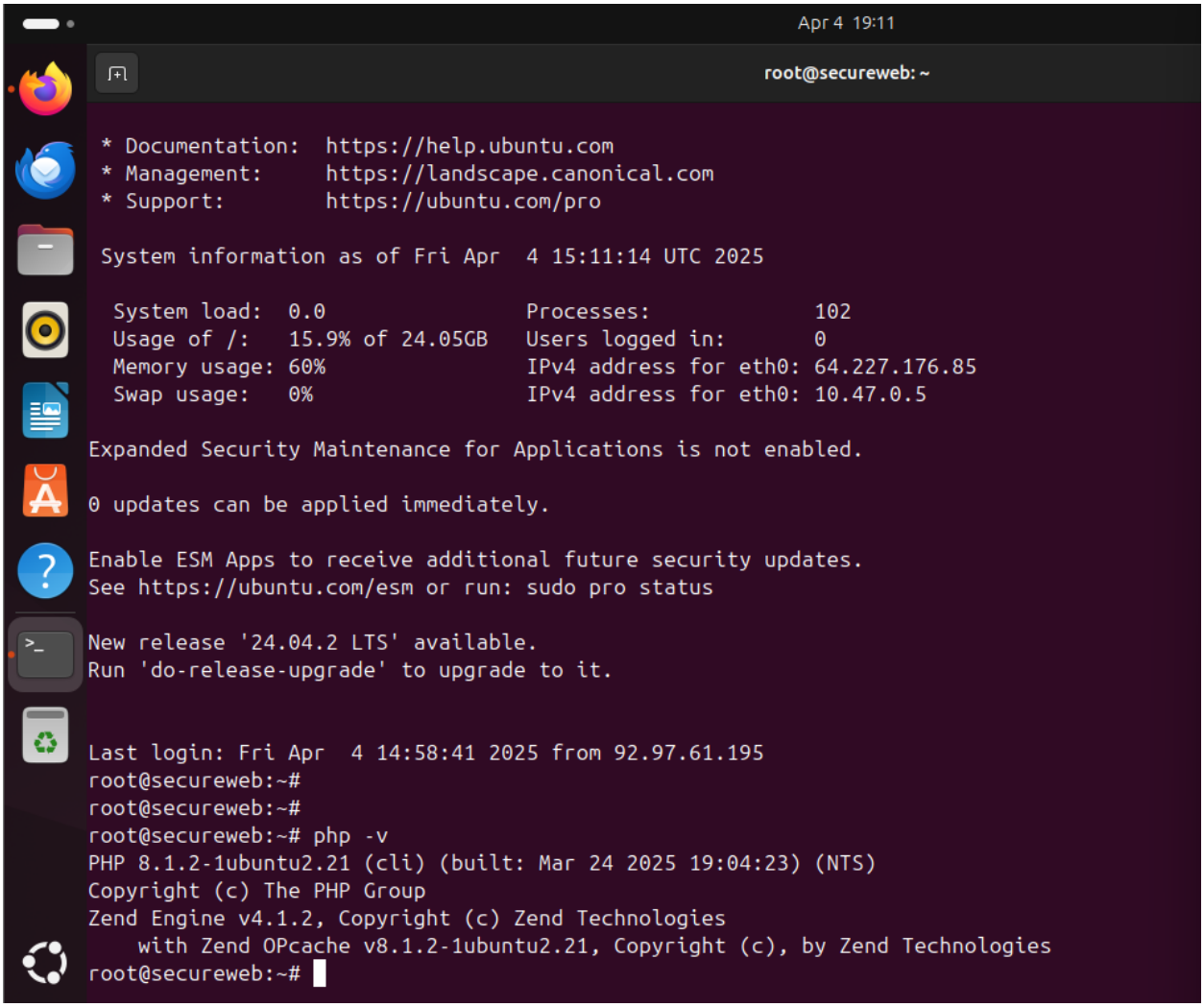


4) install php and mysql for data managment;

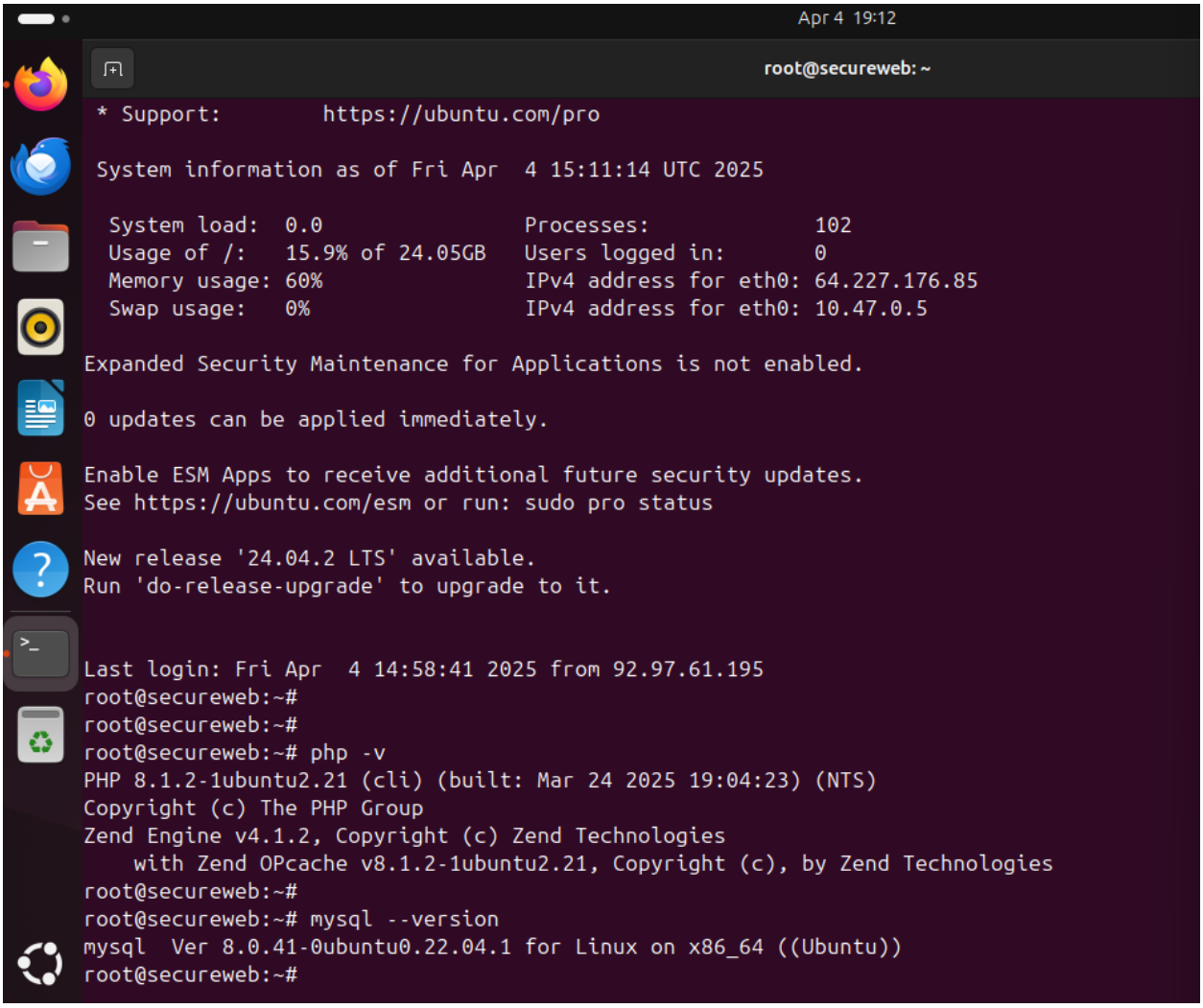
sudo apt install php libapache2-mod-php php-mysql (install php)

sudo apt install mysql-server (install mysql)

php -v (to check if php was installed)

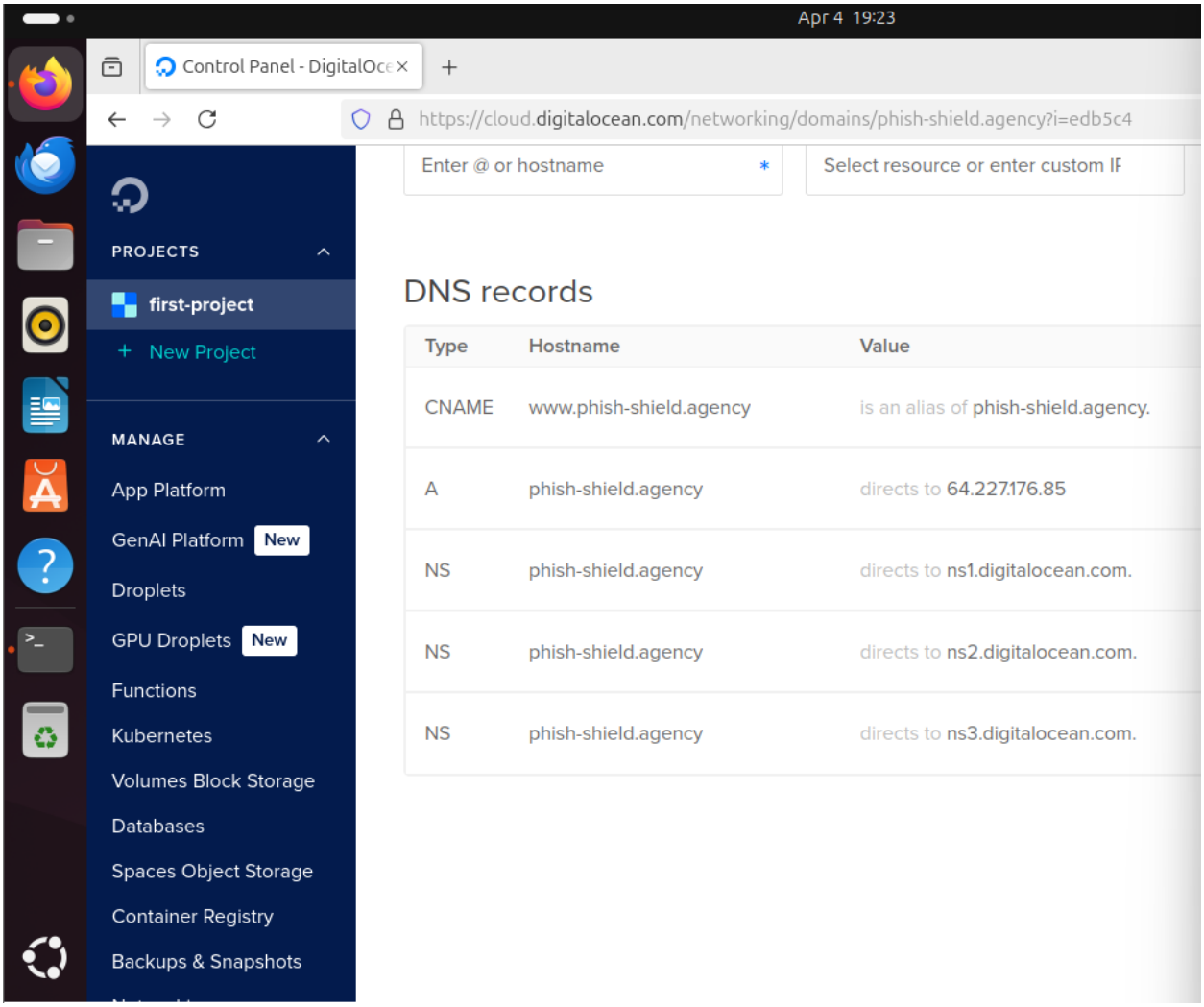


mysql –version (check if mysql was installed)



5) DNS configuration;

Before running Certbot to generate the SSL certificate, I needed to make sure that the domain name I purchased was correctly resolved to my server IP address. I did this by configuring the DNS settings inside DigitalOcean's domain management panel.



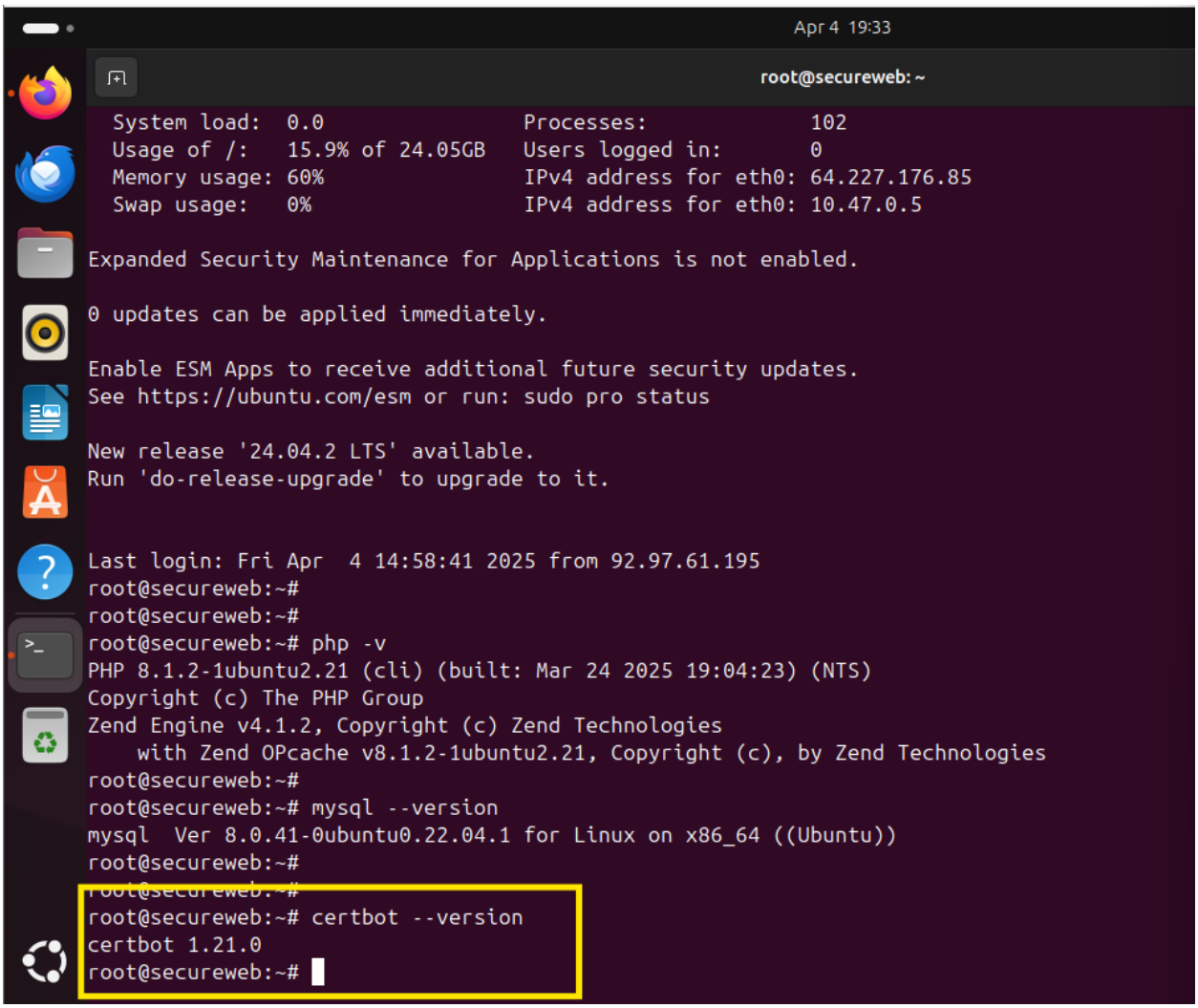
updated type A by adding my name and ipv4 IP address to make sure the domain name I purchased was linked to my IP; luckily for me, it was resolved after 2 hours, so I continued to install certbot.

6) certbot installation and Apache plugin;

sudo apt install certbot python3-certbot-apache

sudo certbot –apache (installed ssl certificate)

certbot --version (check if certbot was installed)

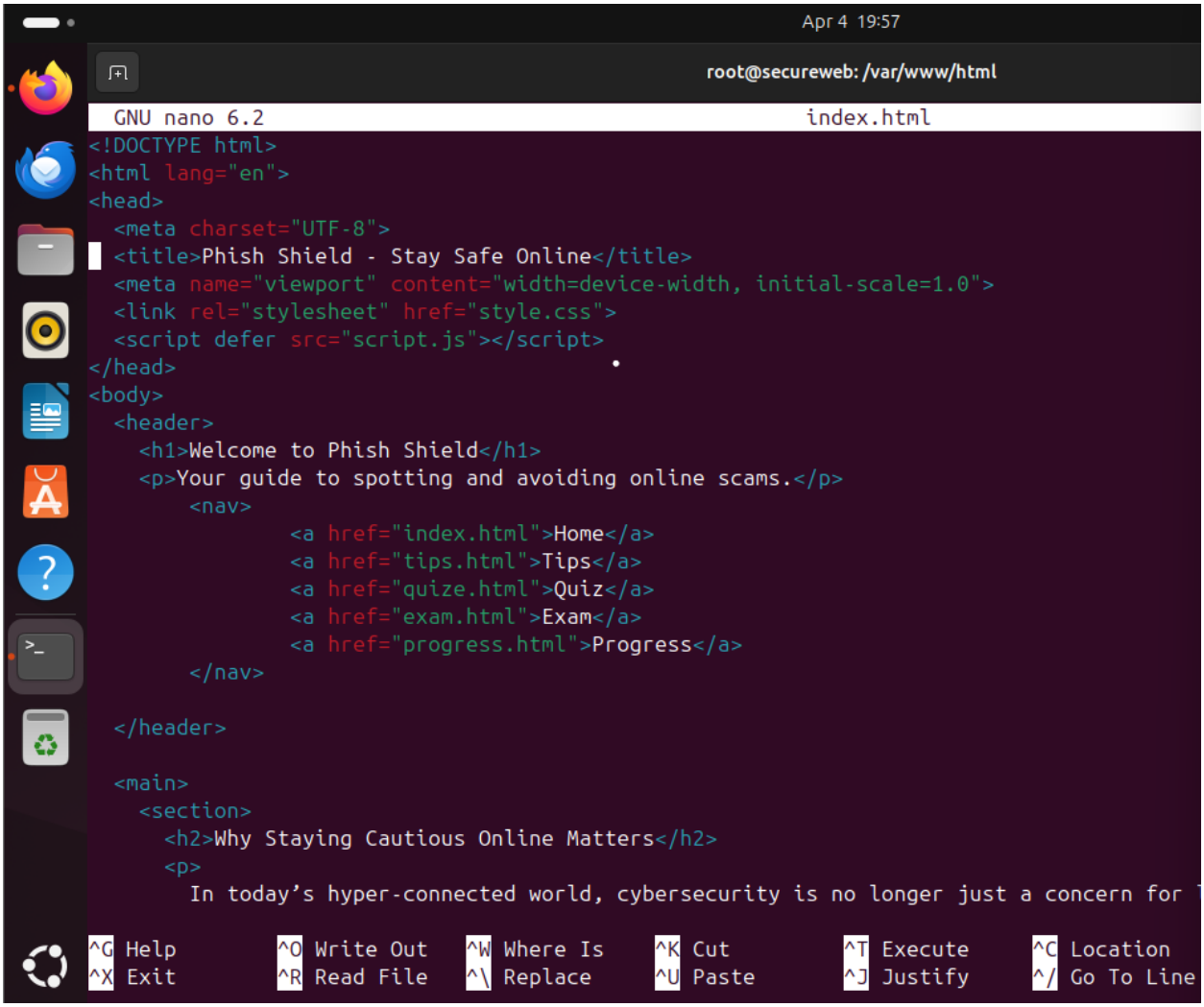


7) Deploying the website;

so the apache web root directory is /var/www/html/

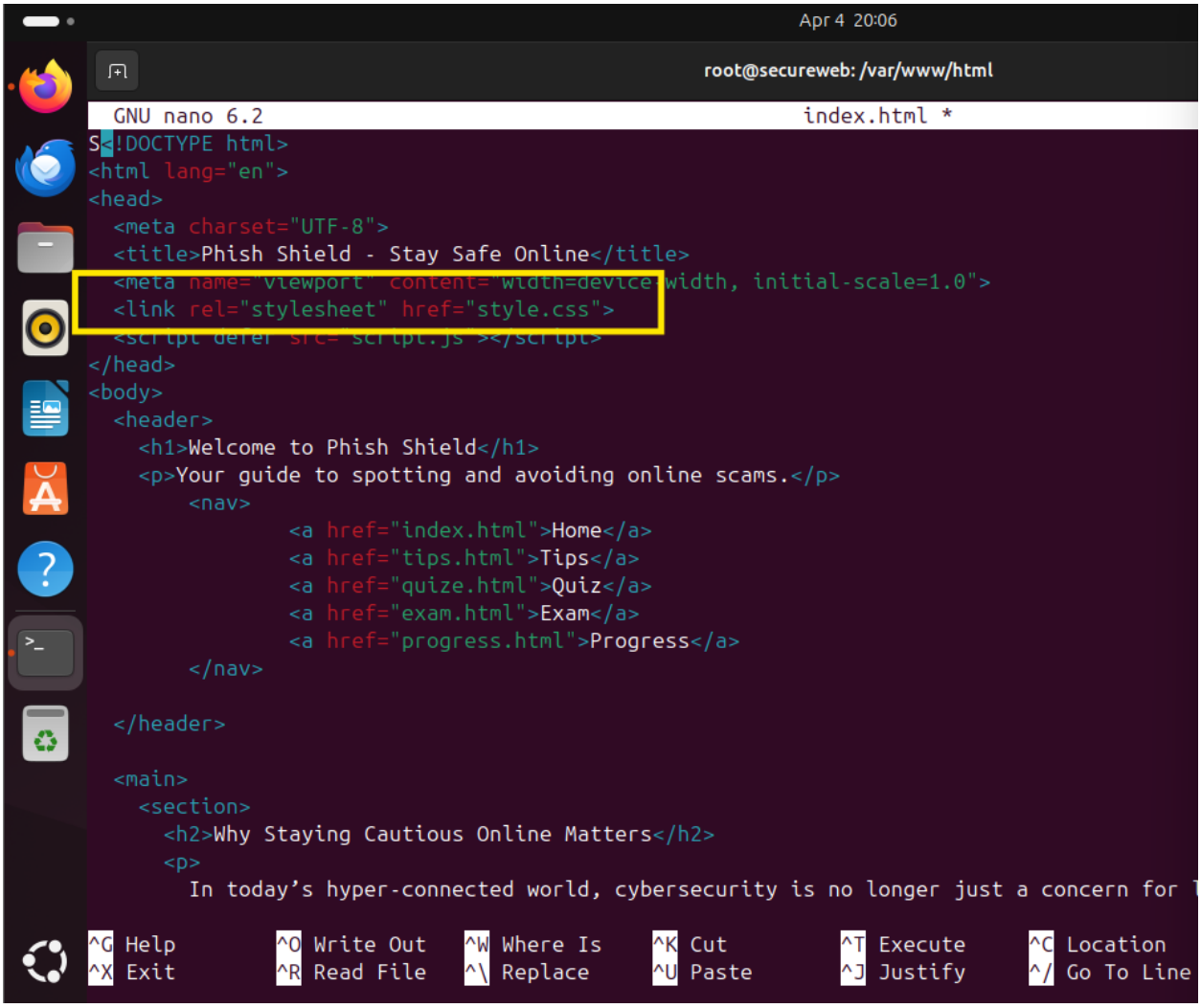
so **cd /var/www/html/** (to change the directory to it instead of typing it over and over)

nano index.html; the nano command allows you to create a file if it doesn't exist or edit the content if it already exists (previously, this page had my project proposal, but now it is edited to the phish-shield official homepage).



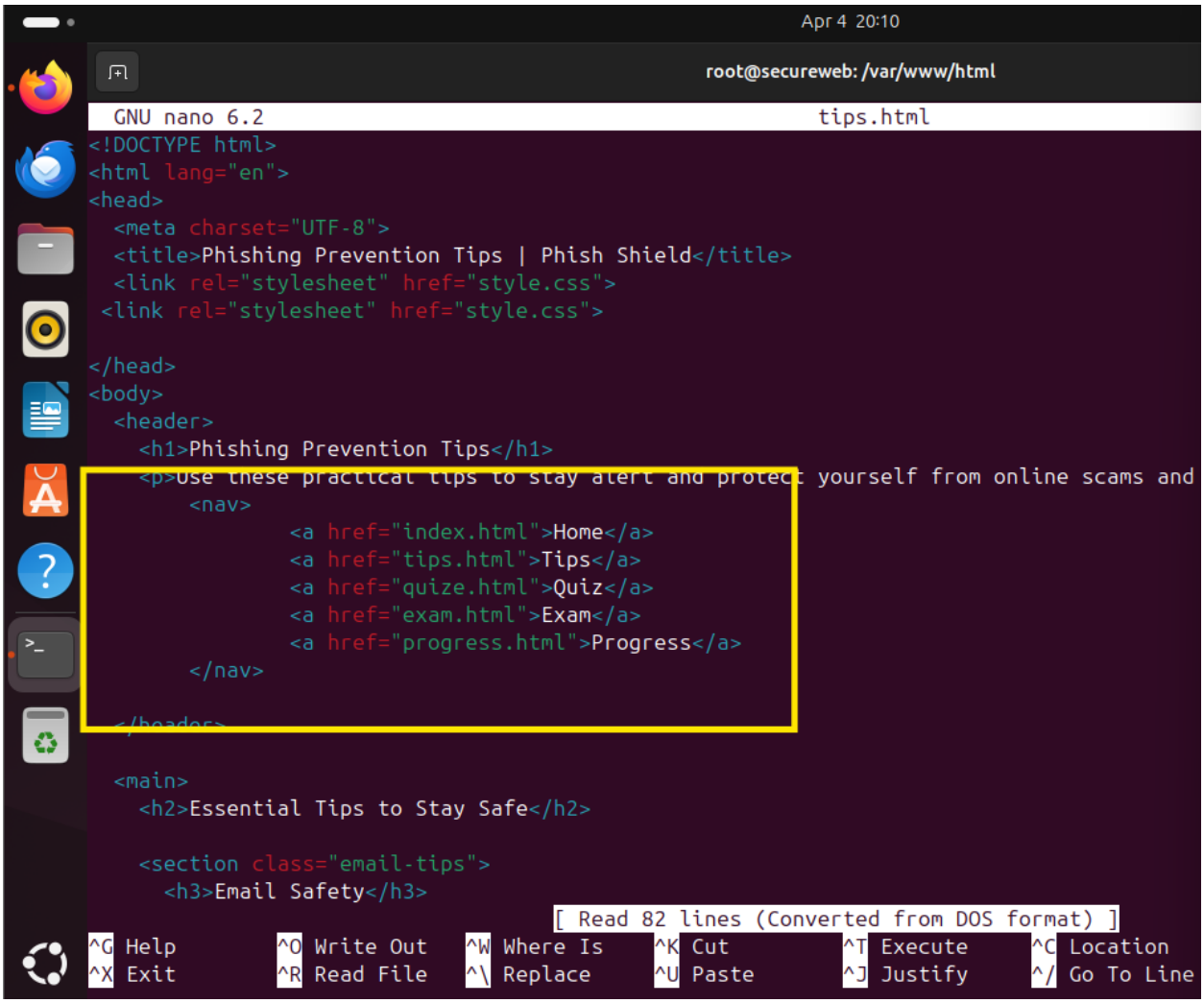
after I was done editing the homepage, save with ctrl + o and exit with the ctrl + x command, and do the same for all the other pages that are needed, like quize.html (ignore the spelling) and quize.js (functions to generate questions from the question pool I made randomly).

instead of using a custom HTML template for each webpage, I created a CSS file with a good theme and call that file in all my pages for easy deployment and consistency.



I used this command in all my html pages to give them all a consistent theme.

nano tips.html (an html page to display all the tips of how to avoid phishing and threat actors).

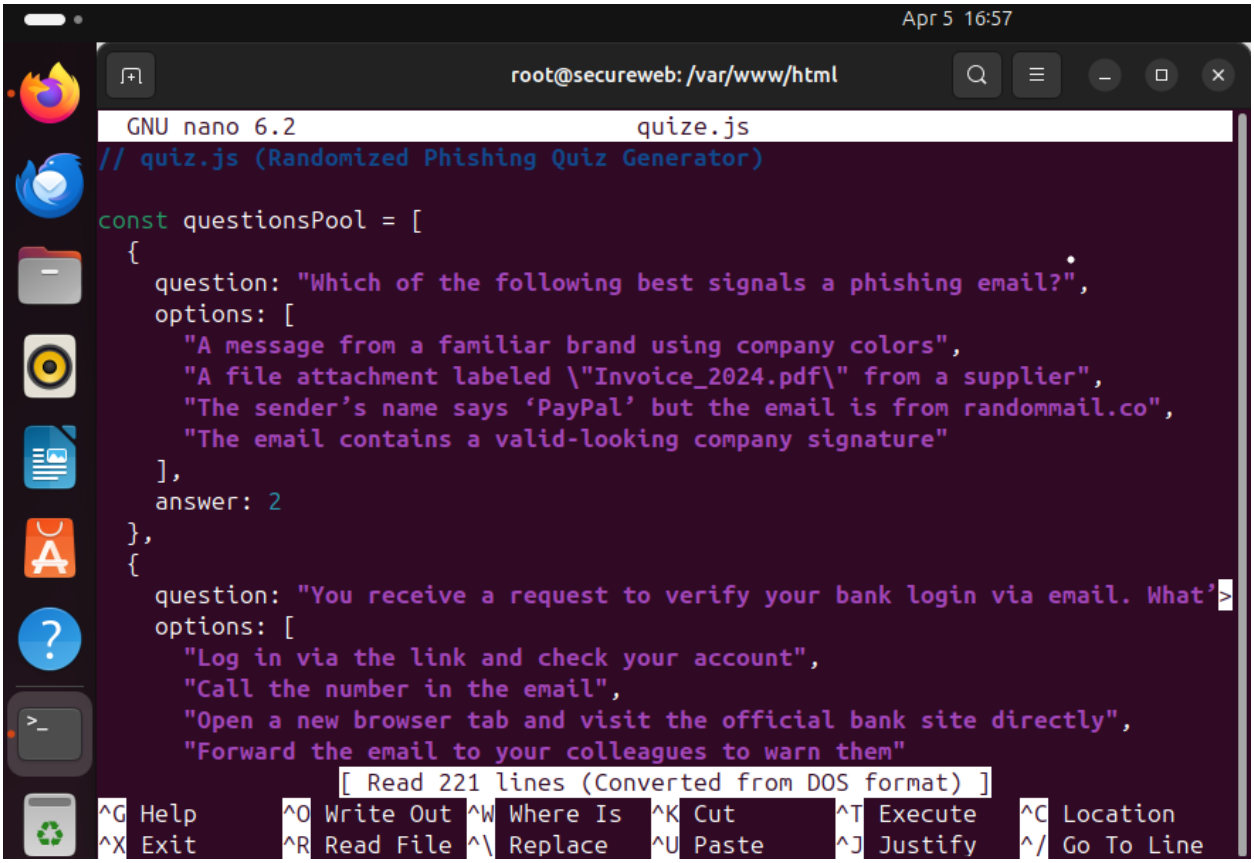


this is the tips.html page and another thing I want to point out is this highlighted navigation menu which I added for all html pages to be able to jump around from one page to another at any time instead of going back to home (index.html).

For the other html pages (exam.html, quize.html, progress.html) I did the same thing like index and tips and customized them the way I wanted them to look like.

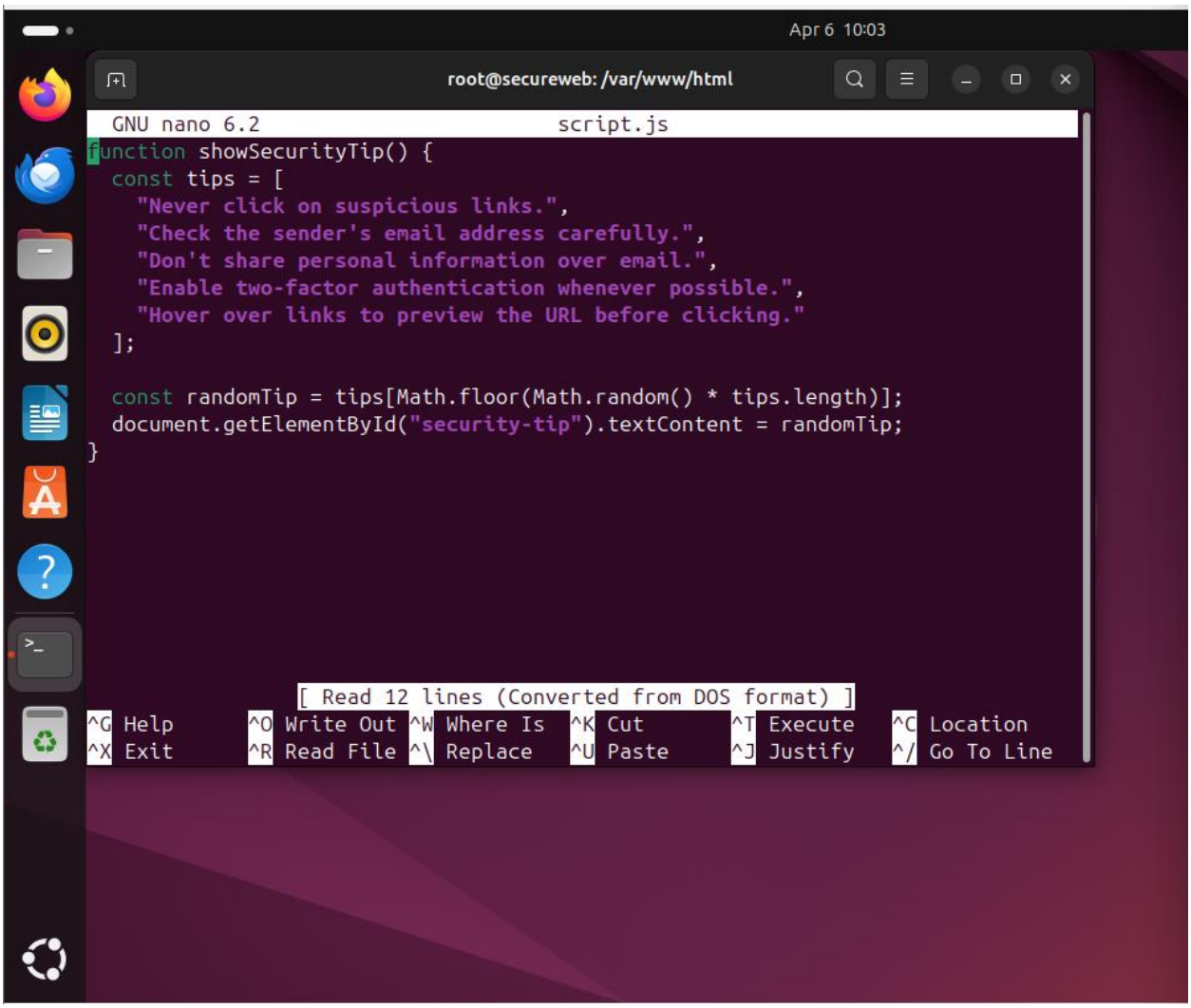
But the quize and exam pages also need Javascript. to randomly pick a question from a set of question pools stored and calculate the final score after submission and send the results to the server (getresults.php) to be saved.

Nano quize.js;



As you can see a constant variable was declared called questionsPool which contained the questions and the answer for the questions.

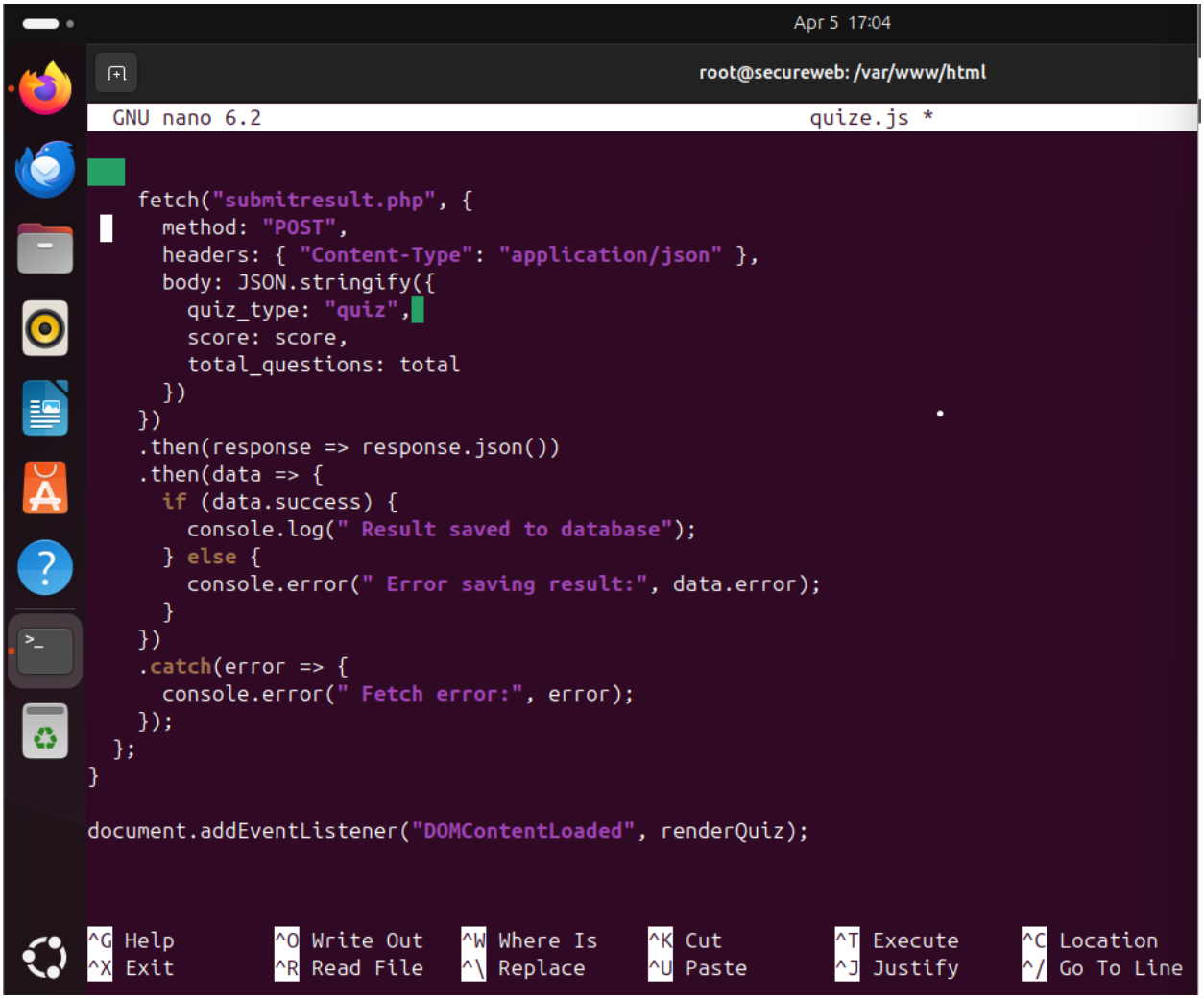
I have also created a script.js file using **nano script.js** just like the others but this one is connected to the home page to generate 1 tip randomly


Apr 6 10:03
root@secureweb: /var/www/html
GNU nano 6.2 script.js
function showSecurityTip() {
const tips = [
"Never click on suspicious links.",
"Check the sender's email address carefully.",
"Don't share personal information over email.",
"Enable two-factor authentication whenever possible.",
"Hover over links to preview the URL before clicking."
];
const randomTip = tips[Math.floor(Math.random() * tips.length)];
document.getElementById("security-tip").textContent = randomTip;
}
[Read 12 lines (Converted from DOS format)]
^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

Apr 5 16:59

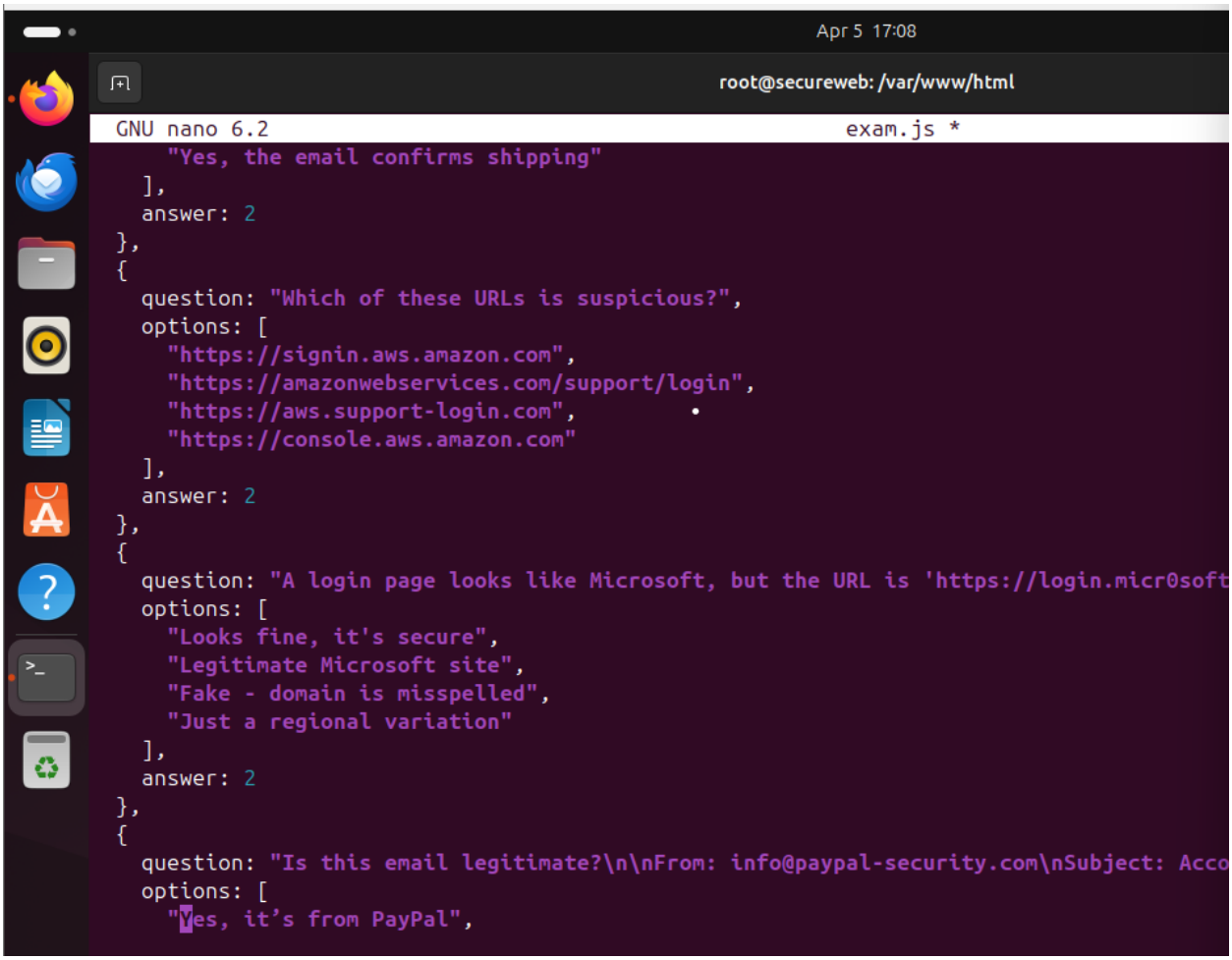
root@secureweb: /var/www/html

```
  GNU nano 6.2                          quize.js
];

function getRandomQuestions(pool, count = 10) {
  const shuffled = [...pool].sort(() => 0.5 - Math.random());
  return shuffled.slice(0, count);
}

function renderQuiz() {
  const quizForm = document.getElementById("quizForm");
  const questions = getRandomQuestions(questionsPool);
  quizForm.innerHTML = "";

  questions.forEach((q, index) => {
    const questionDiv = document.createElement("div");
    questionDiv.classList.add("question");
    questionDiv.innerHTML = `<p>${index + 1}. ${q.question}</p>` +
      q.options.map((opt, i) => `
        <label><input type="radio" name="q${index}" value="${i}"> ${opt}</label>
      `).join("");
    quizForm.appendChild(questionDiv);

^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
```



Here are the functions and variables used to display random question to display result and save the result to the database.

The logoc is very similar the major difference being, on the exam the questions are more practical and more similar to real word scenarios than the quiz. So questions like the bottom screenshot can be added using nano exam.js command



Finally, I logged in to MySQL using **sudo mysql** (since I have already installed and made sure it was working earlier) and created a database called phishshield using **CREATE DATABASE phishshield;**

and for better security I made this configurations and added a password

CREATE USER 'phishuser'@'localhost' IDENTIFIED BY 'password123';

GRANT ALL PRIVILEGES ON phishshield.* TO 'phishuser'@'localhost';

FLUSH PRIVILEGES;

To create the table, I used the following commands which was just enough to store the quiz or exam ID, name, score and date.

CREATE TABLE results (

id INT AUTO_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

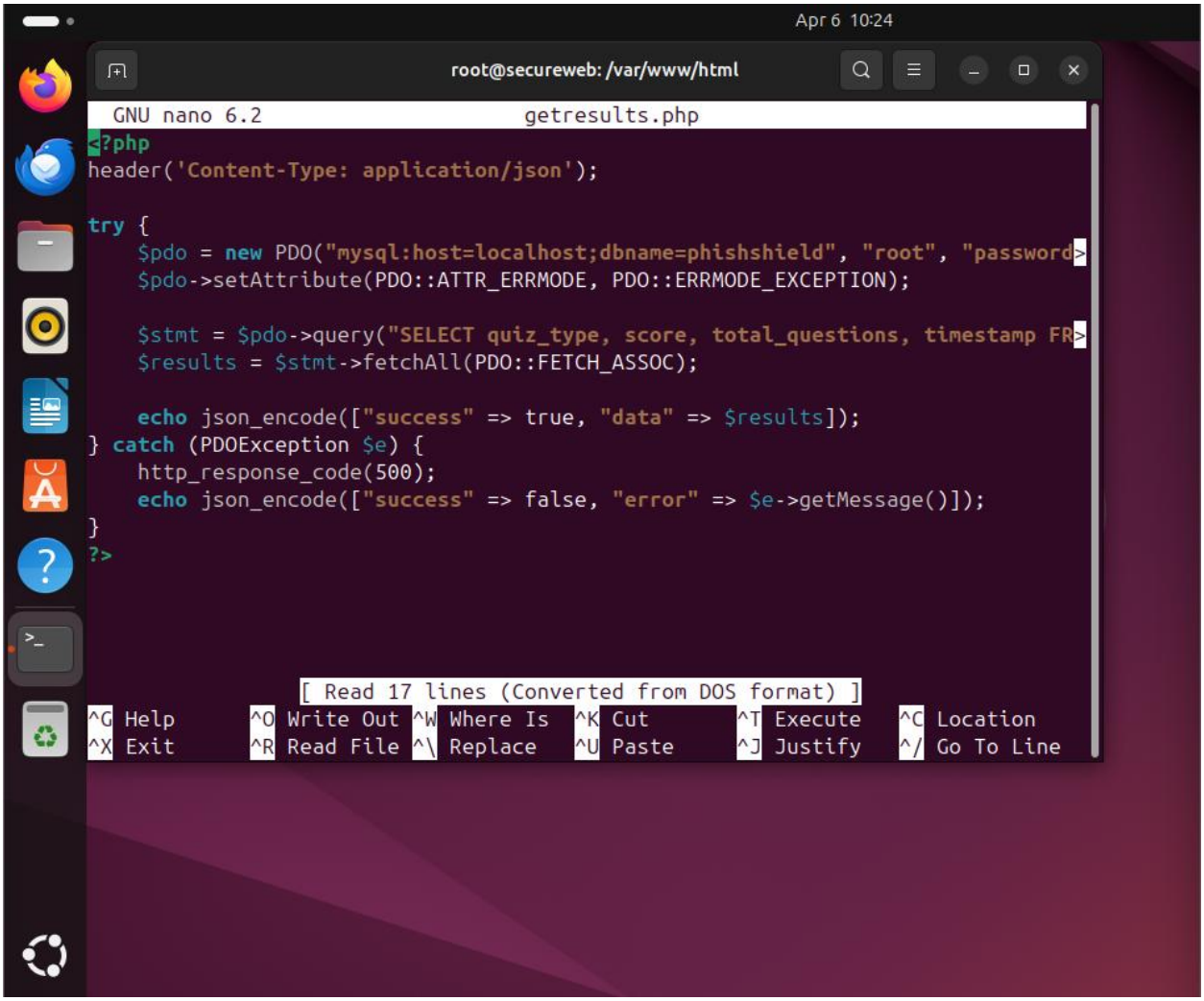
score INT NOT NULL,

date TIMESTAMP DEFAULT CURRENT_TIMESTAMP

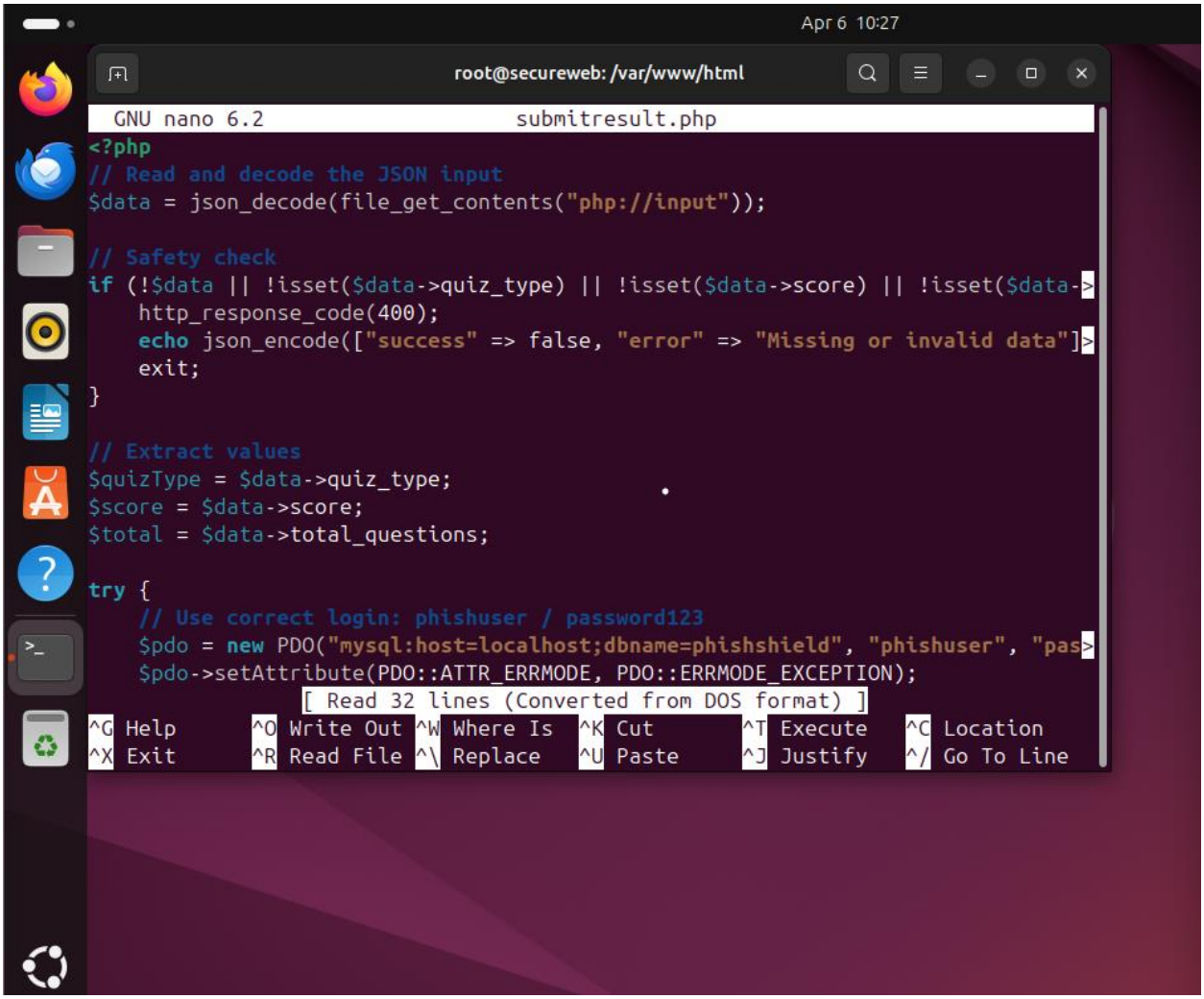
);

exit;

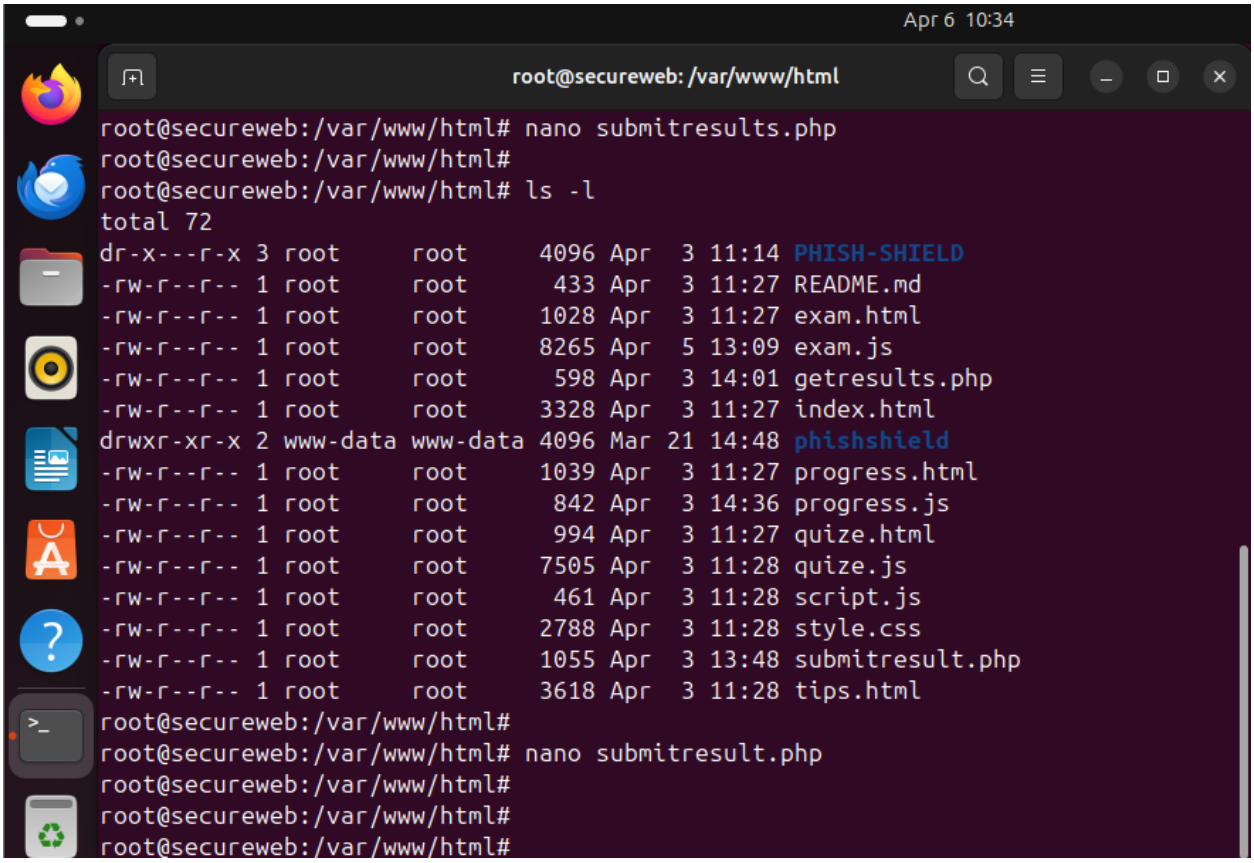
then used **nano getresults.php** to the add the following commands and specifically handle results for exam



And added the following commands to **nano submitresult.php** to handle quiz results



Everything needed for the website to work have been deployed, we can use the simple **ls -l** in the same directory to check if the files have been created and saved properly



After creating each file, it is good practice to make sure they are saved in the correct directory one by one.

Conclusion and Future Recommendations

Everything works great; the website has been secured fully with SSL/TLS encryption through the Let's Encrypt free certificate. Also, for the database and result management, I have added an extra layer of security by adding a user called phishuser with the password being 'password123', as the screenshot indicates instead of just using root. This is done for assignment purposes so everyone can take the quiz and exam and view the results on the progress page. But when it is deployed for CyberGuard-Hub, it will have a log-in page for each user of the company and a different progress tracker for everyone. Also, another recommendation is adding image-based questions and examples and, of course, keep updating the tips, the exams, and the quizzes to keep up with the latest trends and new ways of combating threat actors, especially now with the emergence of AI and new technologies day by day.

References

Certbot. (n.d.). *Certbot: Get HTTPS for your site*. Electronic Frontier Foundation. Retrieved April 5, 2025, from https://certbot.eff.org/instructions

DigitalOcean. (n.d.). *How to Set Up Apache on Ubuntu 20.04*. DigitalOcean Tutorials. Retrieved April 5, 2025, from https://www.digitalocean.com/docs/

MySQL Documentation. (n.d.). *CREATE TABLE Statement*. MySQL 8.0 Reference Manual. Retrieved April 5, 2025, from https://dev.mysql.com/doc/refman/8.0/en/create-table.html

PHP Manual. (n.d.). *PDO - PHP Data Objects*. PHP.net. Retrieved April 5, 2025, from https://www.php.net/manual/en/book.pdo.php

Stack Overflow contributors. (n.d.). *How to fetch and post data using JavaScript Fetch API*. Stack Overflow. Retrieved April 5, 2025, from https://stackoverflow.com/questions/19491336/how-to-post-data-to-php-using-javascript-fetch-api

Traversy Media. (2018, September 10). *PHP Front To Back [Video series]*. YouTube. https://www.youtube.com/watch?v=oJbfyzaA2QA

Web Dev Simplified. (2020, March 2). *Fetch API Explained [Video]*. YouTube. https://www.youtube.com/watch?v=cuEtnrL9-H0