



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

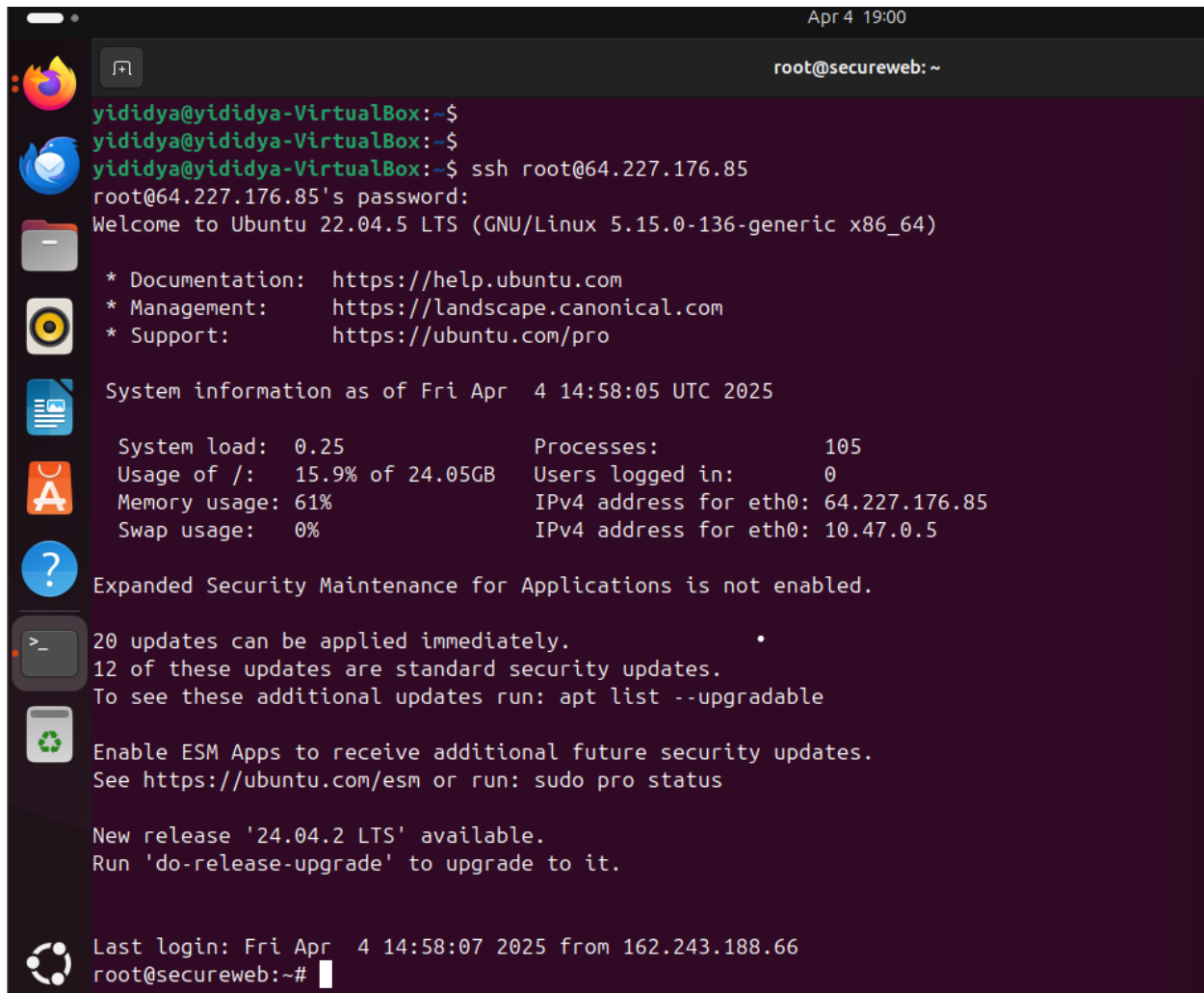
Phish Shield, as per my presentation, is a cybersecurity awareness platform designed to educate employees of CyberGuard-hub (a hypothetical cyber-security solutions company) on phishing threats through interactive content, quizzes, and real-world simulations. The project was hosted using a cloud Infrastructure-as-a-Service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



The image is a screenshot of a terminal window. At the top right, it says 'Apr 4 19:00'. The terminal title bar shows 'root@secureweb: ~'. The user 'yididya' is at a 'yididya-VirtualBox' machine. They enter the command 'ssh root@64.227.176.85'. The terminal shows the password prompt and a successful login to 'root@64.227.176.85's password:'. The system is 'Ubuntu 22.04.5 LTS (GNU/Linux 5.15.0-136-generic x86\_64)'. It displays system information: 'System load: 0.25', 'Processes: 105', 'Usage of /: 15.9% of 24.05GB', 'Users logged in: 0', 'Memory usage: 61%', 'IPv4 address for eth0: 64.227.176.85', and 'Swap usage: 0%'. It also shows 'IPv4 address for eth0: 10.47.0.5'. A message states 'Expanded Security Maintenance for Applications is not enabled.' and '20 updates can be applied immediately. 12 of these updates are standard security updates. To see these additional updates run: apt list --upgradable'. It suggests enabling ESM Apps and provides a link 'https://ubuntu.com/esm' and a command 'sudo pro status'. It also mentions a new release '24.04.2 LTS' is available and suggests running 'do-release-upgrade'. At the bottom, it shows the last login: 'Last login: Fri Apr 4 14:58:07 2025 from 162.243.188.66' and the prompt 'root@secureweb:~#'.

```
Apr 4 19:00
root@secureweb: ~
yididya@yididya-VirtualBox:~$
yididya@yididya-VirtualBox:~$
yididya@yididya-VirtualBox:~$ ssh root@64.227.176.85
root@64.227.176.85's password:
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 5.15.0-136-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Fri Apr  4 14:58:05 UTC 2025

System load:  0.25                       Processes:            105
Usage of /:   15.9% of 24.05GB           Users logged in:     0
Memory usage: 61%                       IPv4 address for eth0: 64.227.176.85
Swap usage:   0%                        IPv4 address for eth0: 10.47.0.5

Expanded Security Maintenance for Applications is not enabled.

20 updates can be applied immediately.
12 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

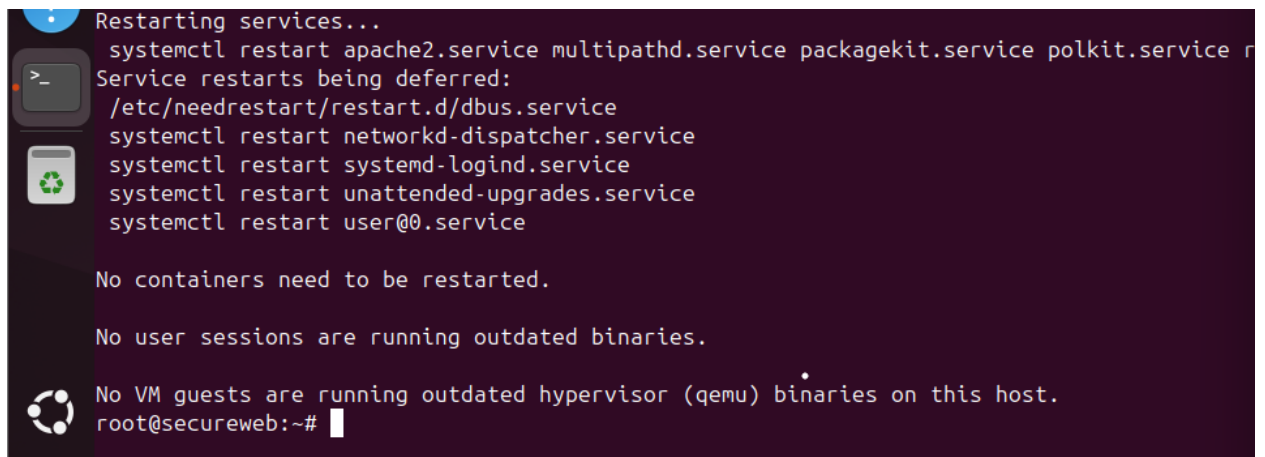
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '24.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Apr  4 14:58:07 2025 from 162.243.188.66
root@secureweb:~#
```

2) Updating the server before anything else

**sudo apt update and upgrade**



```
Restarting services...
systemctl restart apache2.service multipathd.service packagekit.service polkit.service r
Service restarts being deferred:
/etc/needrestart/restart.d/dbus.service
systemctl restart networkd-dispatcher.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service
systemctl restart user@0.service

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@secureweb:~#
```

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)

```
Apr 4 19:07
root@secureweb: ~

systemctl restart user@0.service

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@secureweb:~# sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2025-04-04 15:02:52 UTC; 4min 5s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 34939 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 34943 (apache2)
    Tasks: 7 (limit: 1101)
   Memory: 14.1M
      CPU: 92ms
   CGroup: /system.slice/apache2.service
           └─34943 /usr/sbin/apache2 -k start
             └─34944 /usr/sbin/apache2 -k start
               └─34945 /usr/sbin/apache2 -k start
                 └─34946 /usr/sbin/apache2 -k start
                   └─34947 /usr/sbin/apache2 -k start
                     └─34948 /usr/sbin/apache2 -k start
                       └─34983 /usr/sbin/apache2 -k start

Apr 04 15:02:52 secureweb systemd[1]: Stopped The Apache HTTP Server.
Apr 04 15:02:52 secureweb systemd[1]: apache2.service: Consumed 11.039s CPU time.
Apr 04 15:02:52 secureweb systemd[1]: Starting The Apache HTTP Server...
Apr 04 15:02:52 secureweb apachectl[34942]: AH00558: apache2: Could not reliably determine
Apr 04 15:02:52 secureweb systemd[1]: Started The Apache HTTP Server.
lines 1-23/23 (END)
```

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)

```
Apr 4 19:11
root@secureweb: ~

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/pro

System information as of Fri Apr  4 15:11:14 UTC 2025

System load:  0.0                       Processes:            102
Usage of /:   15.9% of 24.05GB          Users logged in:     0
Memory usage: 60%                      IPv4 address for eth0: 64.227.176.85
Swap usage:   0%                      IPv4 address for eth0: 10.47.0.5

Expanded Security Maintenance for Applications is not enabled.

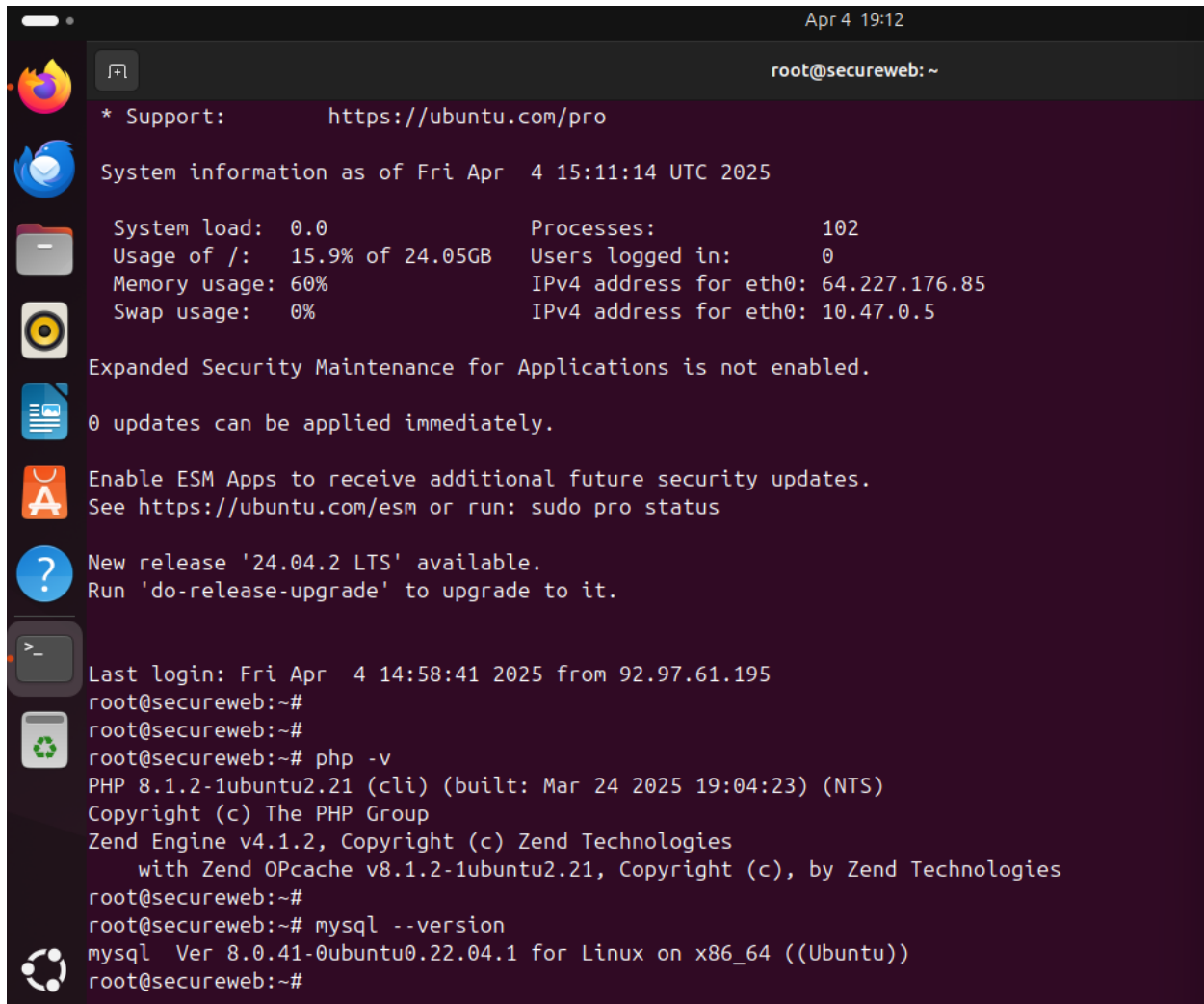
0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '24.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Apr  4 14:58:41 2025 from 92.97.61.195
root@secureweb:~#
root@secureweb:~#
root@secureweb:~# php -v
PHP 8.1.2-1ubuntu2.21 (cli) (built: Mar 24 2025 19:04:23) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
    with Zend OPcache v8.1.2-1ubuntu2.21, Copyright (c), by Zend Technologies
root@secureweb:~#
```

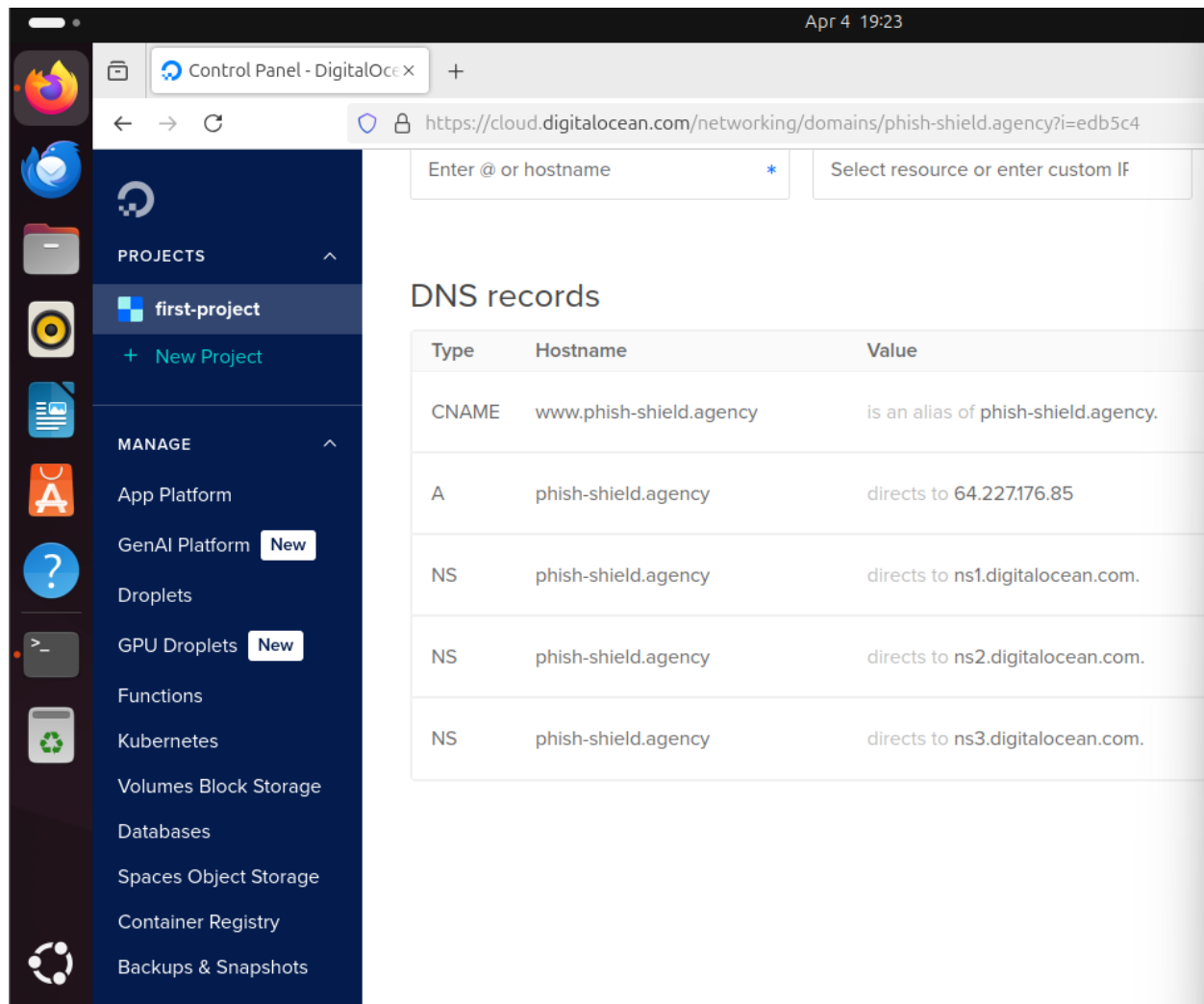
**mysql -version** (check if mysql was installed)

A terminal window titled 'root@secureweb: ~' with a dark purple background. The window shows system information as of Friday, April 4, 2025, at 15:11:14 UTC. It lists system load (0.0), memory usage (60%), swap usage (0%), and processes (102). It also shows the IPv4 address for eth0 as 64.227.176.85 and 10.47.0.5. A message indicates that Expanded Security Maintenance for Applications is not enabled and that 0 updates can be applied immediately. It suggests enabling ESM Apps for additional security updates and mentions a new release '24.04.2 LTS' is available. The terminal also shows the last login time and the execution of 'php -v' and 'mysql --version' commands.

```
Apr 4 19:12
root@secureweb: ~
* Support: https://ubuntu.com/pro
System information as of Fri Apr 4 15:11:14 UTC 2025
System load: 0.0          Processes: 102
Usage of /: 15.9% of 24.05GB Users logged in: 0
Memory usage: 60%        IPv4 address for eth0: 64.227.176.85
Swap usage: 0%           IPv4 address for eth0: 10.47.0.5
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
New release '24.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
Last login: Fri Apr 4 14:58:41 2025 from 92.97.61.195
root@secureweb:~#
root@secureweb:~#
root@secureweb:~# php -v
PHP 8.1.2-1ubuntu2.21 (cli) (built: Mar 24 2025 19:04:23) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
    with Zend OPcache v8.1.2-1ubuntu2.21, Copyright (c), by Zend Technologies
root@secureweb:~#
root@secureweb:~# mysql --version
mysql Ver 8.0.41-0ubuntu0.22.04.1 for Linux on x86_64 ((Ubuntu))
root@secureweb:~#
```

## 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.

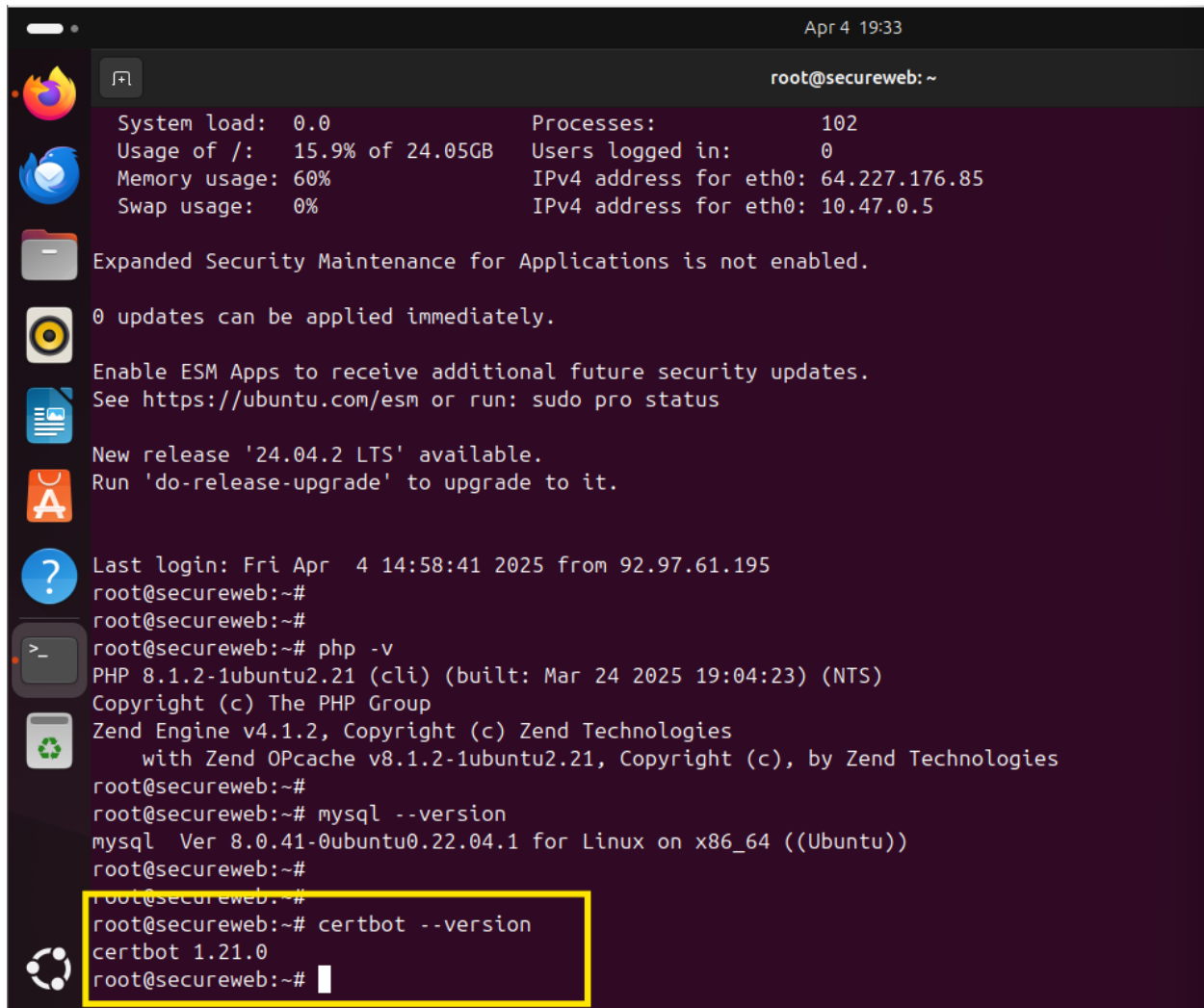
6) certbot installation and Apache plugin;

**sudo apt install certbot python3-certbot-apache**

**sudo certbot --apache** (installed ssl certificate)

**certbot --version** (check if certbot was installed)





A terminal window titled 'root@secureweb: ~' showing system status and command execution. The window has a dark background with a sidebar on the left containing icons for various applications. The terminal output includes system load, memory usage, and network information. It also displays security maintenance status and update information. The user has executed several commands, including 'php -v', 'mysql --version', and 'certbot --version'. The 'certbot --version' command output is highlighted with a yellow box.

```
Apr 4 19:33
root@secureweb: ~

System load: 0.0          Processes: 102
Usage of /: 15.9% of 24.05GB Users logged in: 0
Memory usage: 60%        IPv4 address for eth0: 64.227.176.85
Swap usage: 0%           IPv4 address for eth0: 10.47.0.5

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

New release '24.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

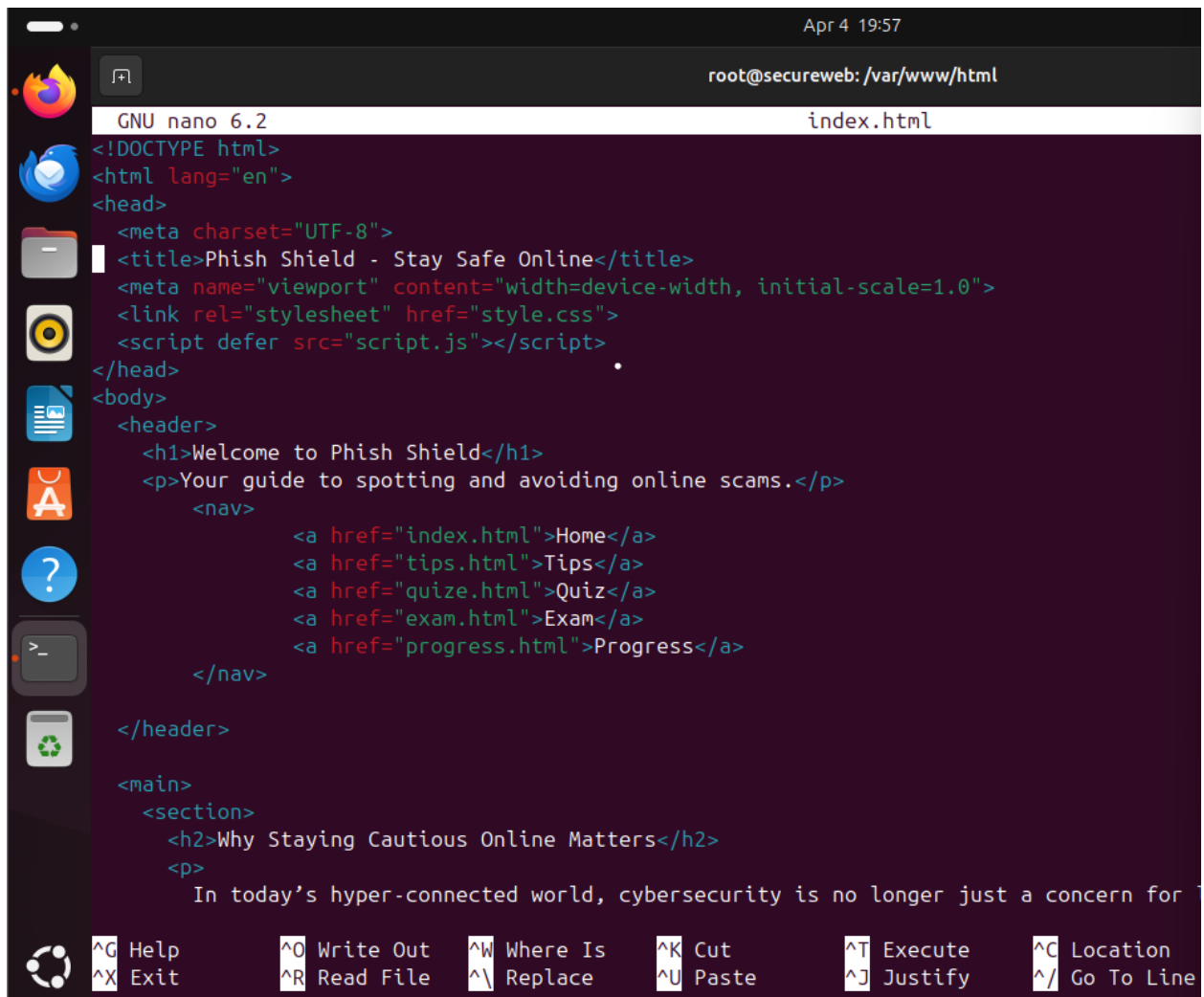
Last login: Fri Apr 4 14:58:41 2025 from 92.97.61.195
root@secureweb:~#
root@secureweb:~#
root@secureweb:~# php -v
PHP 8.1.2-1ubuntu2.21 (cli) (built: Mar 24 2025 19:04:23) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
    with Zend OPcache v8.1.2-1ubuntu2.21, Copyright (c), by Zend Technologies
root@secureweb:~#
root@secureweb:~# mysql --version
mysql Ver 8.0.41-0ubuntu0.22.04.1 for Linux on x86_64 ((Ubuntu))
root@secureweb:~#
root@secureweb:~# certbot --version
certbot 1.21.0
root@secureweb:~#
```

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)

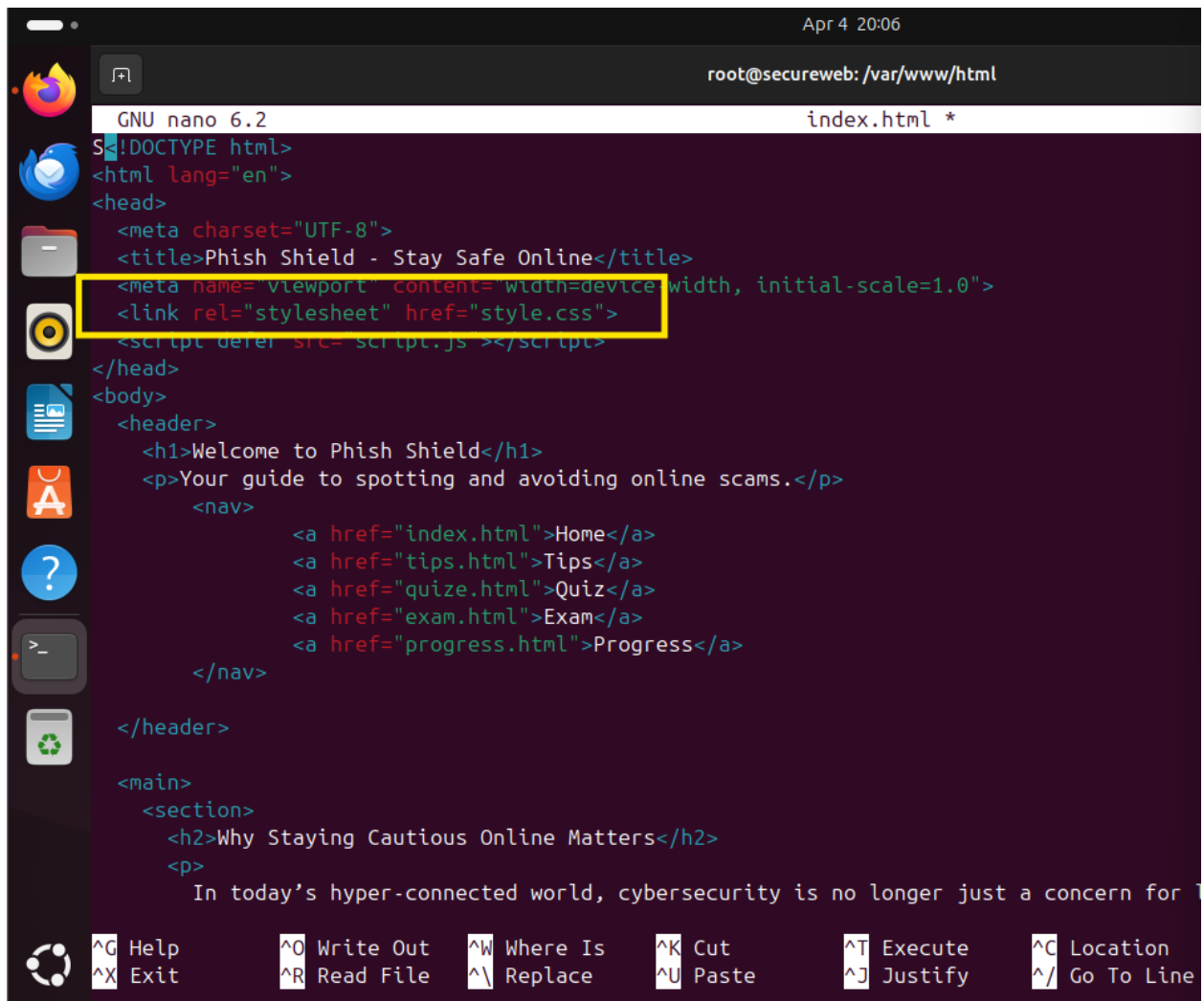
**nano index.html**; the nano command allows you to create a file if it doesn't exist or edit the content if it already exists (previously, this page had my project proposal, but now it is edited to the phish-shield official homepage).



```
GNU nano 6.2 index.html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Phish Shield - Stay Safe Online</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="style.css">
  <script defer src="script.js"></script>
</head>
<body>
  <header>
    <h1>Welcome to Phish Shield</h1>
    <p>Your guide to spotting and avoiding online scams.</p>
    <nav>
      <a href="index.html">Home</a>
      <a href="tips.html">Tips</a>
      <a href="quize.html">Quiz</a>
      <a href="exam.html">Exam</a>
      <a href="progress.html">Progress</a>
    </nav>
  </header>
  <main>
    <section>
      <h2>Why Staying Cautious Online Matters</h2>
      <p>
        In today's hyper-connected world, cybersecurity is no longer just a concern for
```

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).

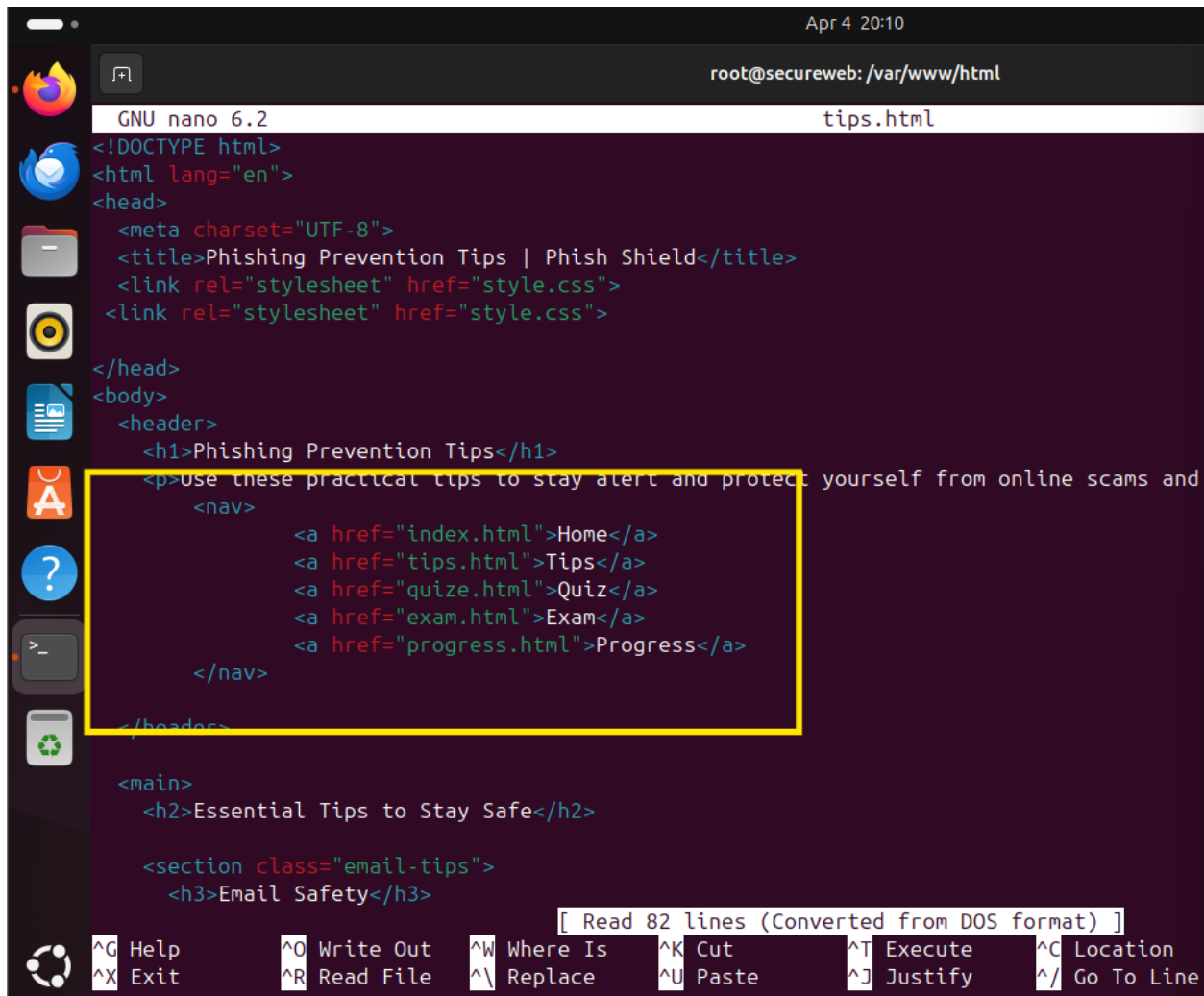
instead of using a custom HTML template for each webpage, I created a CSS file with a good theme and call that file in all my pages for easy deployment and consistency.



```
GNU nano 6.2 index.html *
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Phish Shield - Stay Safe Online</title>
  <meta name= viewport content= width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="style.css">
  <script defer src= script.js ></script>
</head>
<body>
  <header>
    <h1>Welcome to Phish Shield</h1>
    <p>Your guide to spotting and avoiding online scams.</p>
    <nav>
      <a href="index.html">Home</a>
      <a href="tips.html">Tips</a>
      <a href="quize.html">Quiz</a>
      <a href="exam.html">Exam</a>
      <a href="progress.html">Progress</a>
    </nav>
  </header>
  <main>
    <section>
      <h2>Why Staying Cautious Online Matters</h2>
      <p>
        In today's hyper-connected world, cybersecurity is no longer just a concern for
      </p>
    </section>
  </main>
</body>
</html>
```

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).



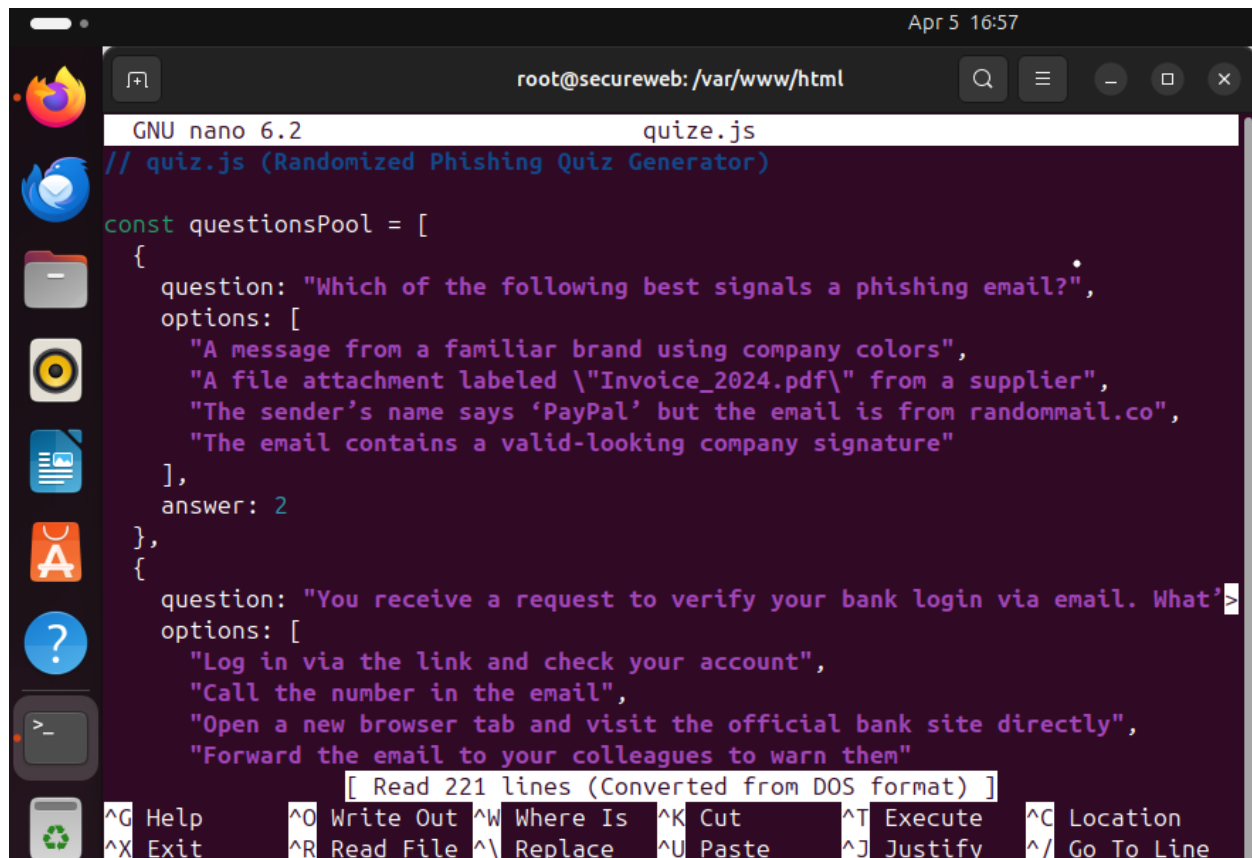
```
GNU nano 6.2 tips.html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Phishing Prevention Tips | Phish Shield</title>
  <link rel="stylesheet" href="style.css">
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <header>
    <h1>Phishing Prevention Tips</h1>
    <p>Use these practical tips to stay alert and protect yourself from online scams and
      <nav>
        <a href="index.html">Home</a>
        <a href="tips.html">Tips</a>
        <a href="quize.html">Quiz</a>
        <a href="exam.html">Exam</a>
        <a href="progress.html">Progress</a>
      </nav>
    </p>
  </header>
  <main>
    <h2>Essential Tips to Stay Safe</h2>
    <section class="email-tips">
      <h3>Email Safety</h3>
```

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;**



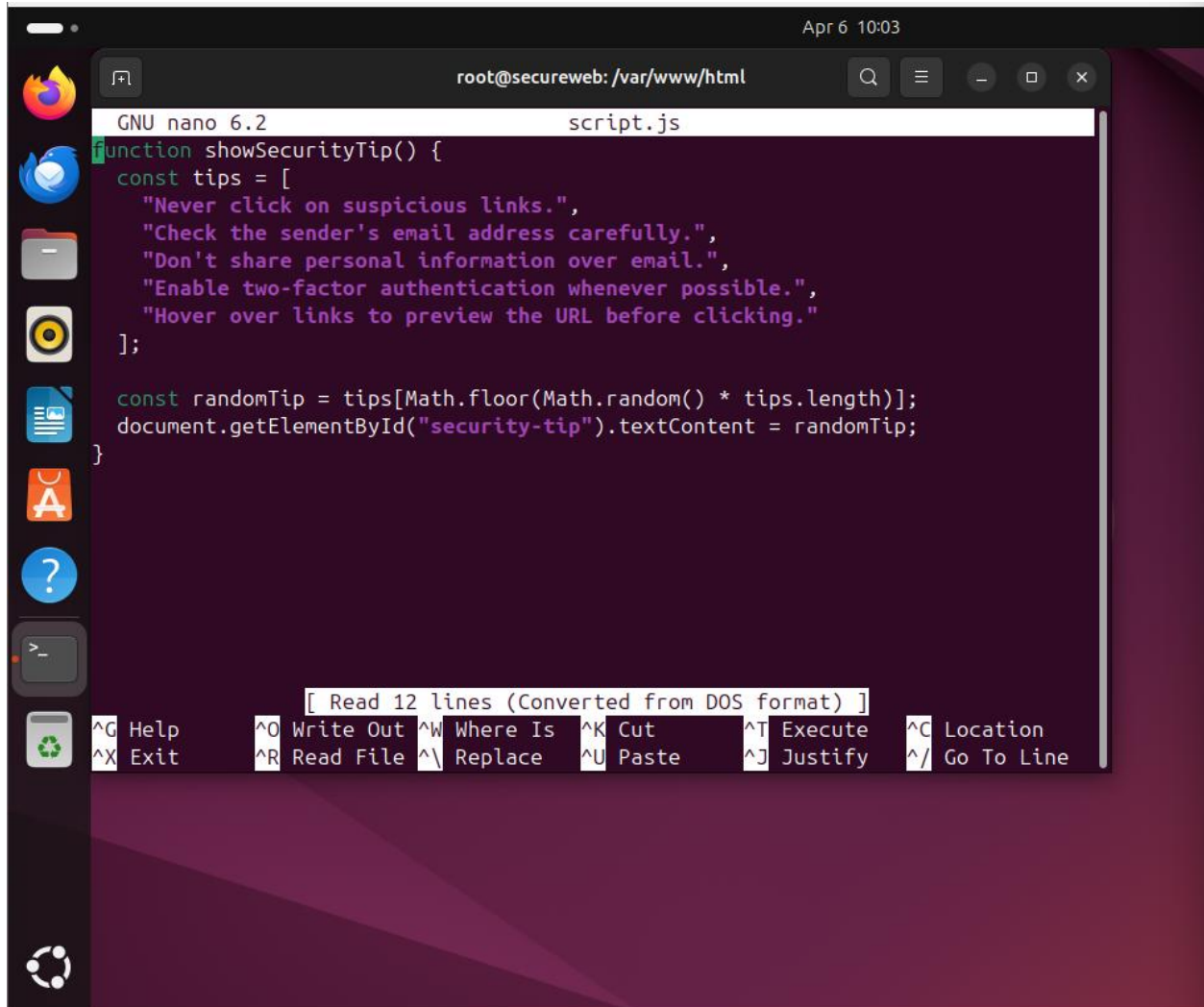
```
root@secureweb: /var/www/html
GNU nano 6.2 quiz.js
// quiz.js (Randomized Phishing Quiz Generator)

const questionsPool = [
  {
    question: "Which of the following best signals a phishing email?",
    options: [
      "A message from a familiar brand using company colors",
      "A file attachment labeled \"Invoice_2024.pdf\" from a supplier",
      "The sender's name says 'PayPal' but the email is from randommail.co",
      "The email contains a valid-looking company signature"
    ],
    answer: 2
  },
  {
    question: "You receive a request to verify your bank login via email. What?",
    options: [
      "Log in via the link and check your account",
      "Call the number in the email",
      "Open a new browser tab and visit the official bank site directly",
      "Forward the email to your colleagues to warn them"
    ]
  }
]

[ Read 221 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
```

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 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 = `

${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

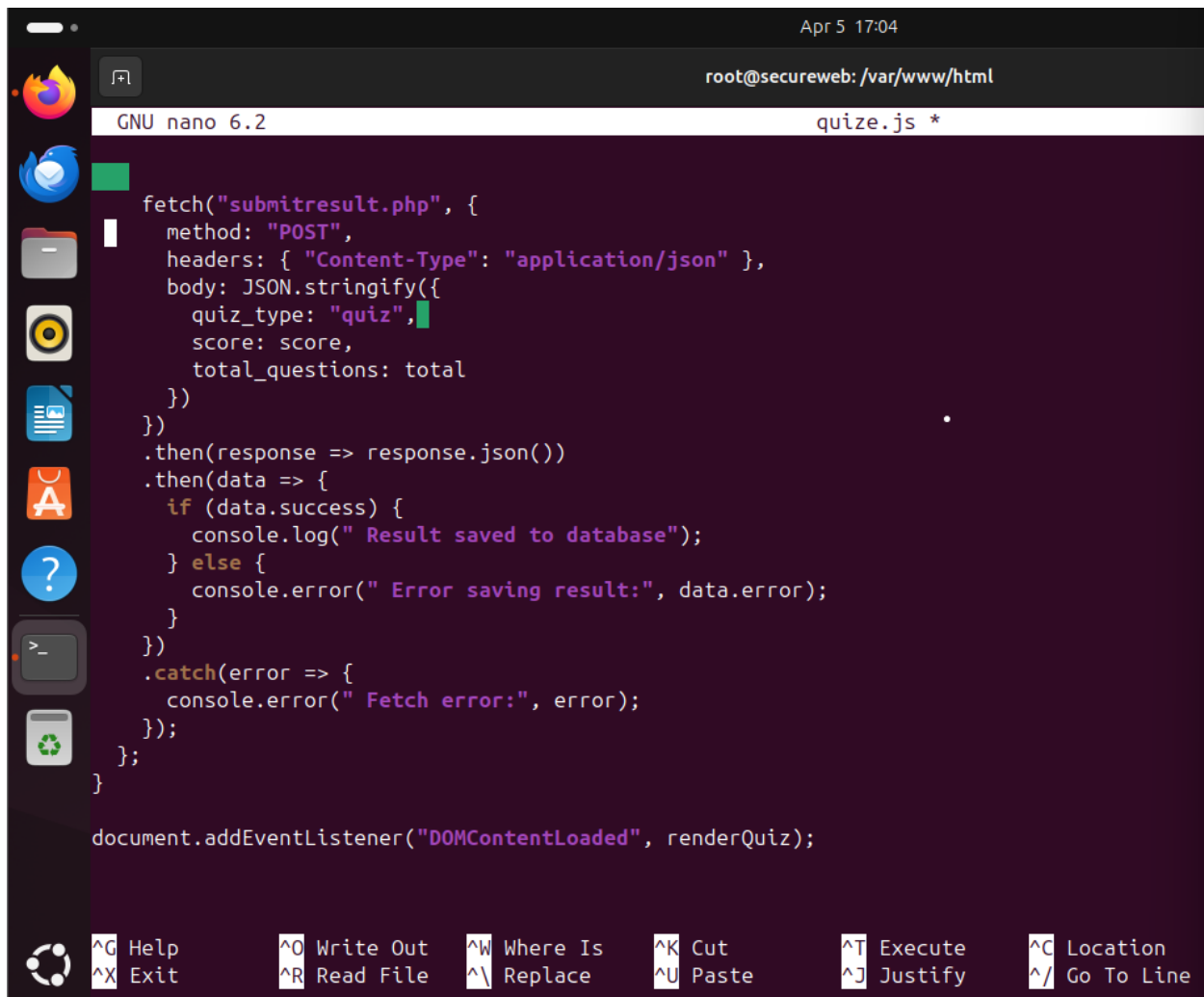
^R Read File

^\_ Replace

^U Paste

^J Justify

^\_ Go To Line



```
root@secureweb: /var/www/html
GNU nano 6.2 quiz.js *

fetch("submitresult.php", {
  method: "POST",
  headers: { "Content-Type": "application/json" },
  body: JSON.stringify({
    quiz_type: "quiz",
    score: score,
    total_questions: total
  })
})
.then(response => response.json())
.then(data => {
  if (data.success) {
    console.log(" Result saved to database");
  } else {
    console.error(" Error saving result:", data.error);
  }
})
.catch(error => {
  console.error(" Fetch error:", error);
});
});

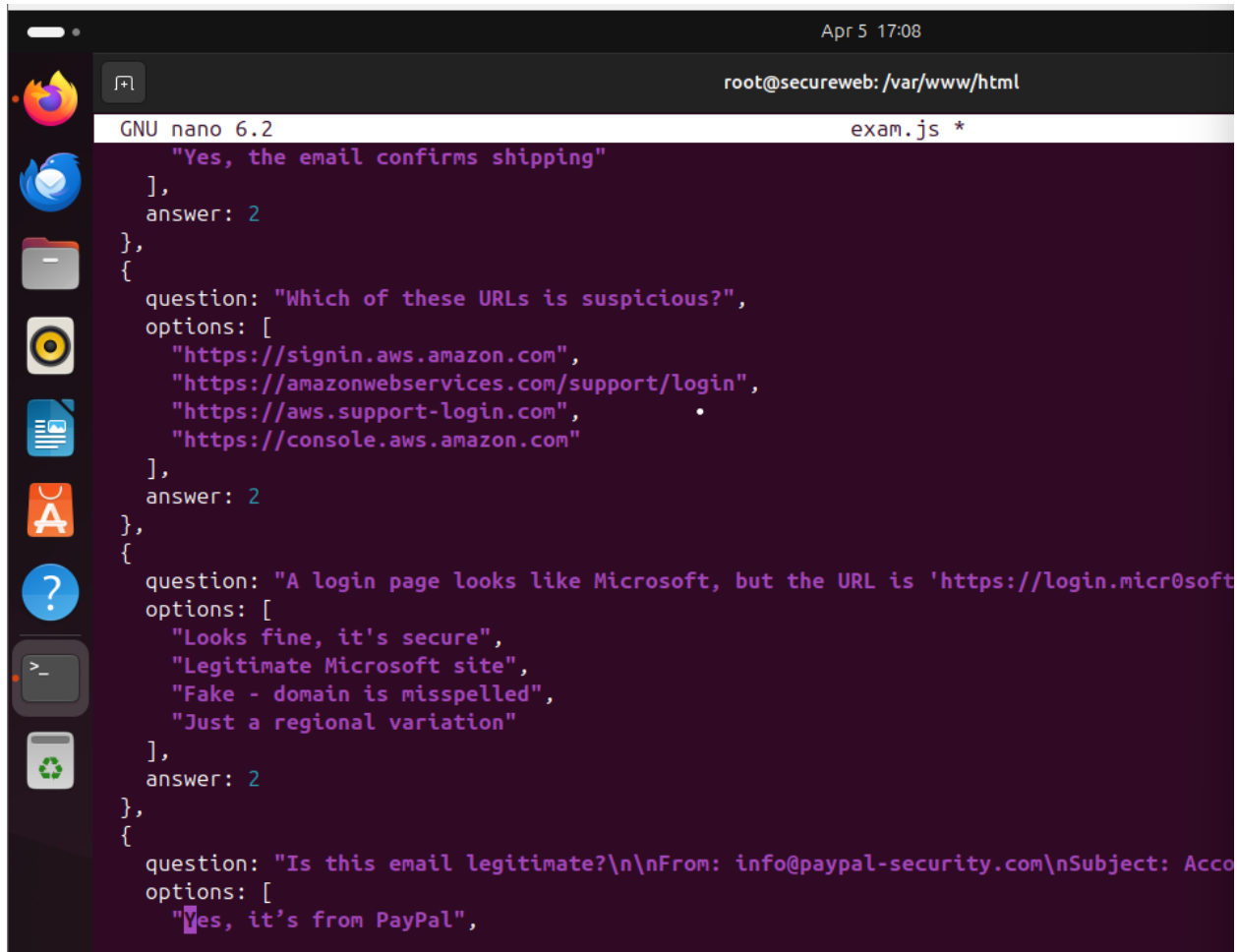
document.addEventListener("DOMContentLoaded", renderQuiz);

^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



A screenshot of a terminal window with a dark background. The top bar shows the date and time 'Apr 5 17:08' and the user and location 'root@secureweb: /var/www/html'. The terminal is running GNU nano 6.2, editing a file named 'exam.js'. The script contains a series of quiz questions and answers. The first question is 'Yes, the email confirms shipping' with answer 2. The second question is 'Which of these URLs is suspicious?' with four options: 'https://signin.aws.amazon.com', 'https://amazonwebervices.com/support/login', 'https://aws.support-login.com', and 'https://console.aws.amazon.com', with answer 2. The third question is 'A login page looks like Microsoft, but the URL is https://login.microsoft.com' with four options: 'Looks fine, it's secure', 'Legitimate Microsoft site', 'Fake - domain is misspelled', and 'Just a regional variation', with answer 2. The fourth question is 'Is this email legitimate?' with a long email header and one option 'Yes, it's from PayPal', with answer 2. The terminal window has a sidebar on the left with various application icons.

```
Apr 5 17:08
root@secureweb: /var/www/html
GNU nano 6.2                                exam.js *
    "Yes, the email confirms shipping"
  ],
  answer: 2
},
{
  question: "Which of these URLs is suspicious?",
  options: [
    "https://signin.aws.amazon.com",
    "https://amazonwebervices.com/support/login",
    "https://aws.support-login.com",
    "https://console.aws.amazon.com"
  ],
  answer: 2
},
{
  question: "A login page looks like Microsoft, but the URL is 'https://login.microsoft.com'",
  options: [
    "Looks fine, it's secure",
    "Legitimate Microsoft site",
    "Fake - domain is misspelled",
    "Just a regional variation"
  ],
  answer: 2
},
{
  question: "Is this email legitimate?\n\nFrom: info@paypal-security.com\nSubject: Account verification\n\nContent: Please verify your account by clicking the link below.\n\nLink: https://paypal-security.com/verify\n\nThank you for using PayPal.",
  options: [
    "Yes, it's from PayPal",
  ],
  answer: 2
},
{
  question: "Is this email legitimate?\n\nFrom: info@paypal-security.com\nSubject: Account verification\n\nContent: Please verify your account by clicking the link below.\n\nLink: https://paypal-security.com/verify\n\nThank you for using PayPal.",
  options: [
    "Yes, it's from PayPal",
  ],
  answer: 2
},
{
  question: "Is this email legitimate?\n\nFrom: info@paypal-security.com\nSubject: Account verification\n\nContent: Please verify your account by clicking the link below.\n\nLink: https://paypal-security.com/verify\n\nThank you for using PayPal.",
  options: [
    "Yes, it's from PayPal",
  ],
  answer: 2
},
}
```

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,
```

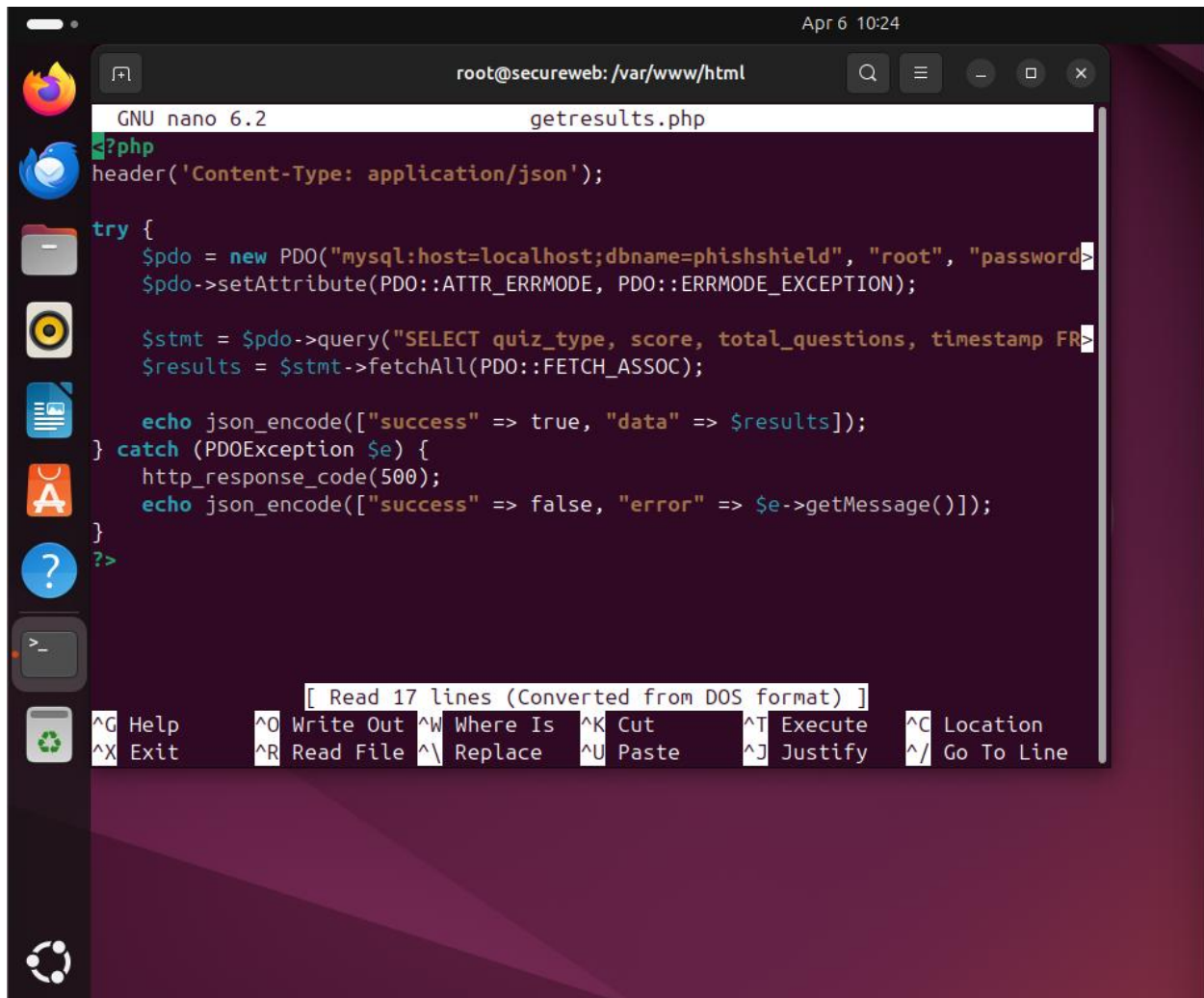
**score INT NOT NULL,**

**date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP**

**);**

**exit;**

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



```
GNU nano 6.2 getresults.php
header('Content-Type: application/json');

try {
    $pdo = new PDO("mysql:host=localhost;dbname=phishshield", "root", "password");
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

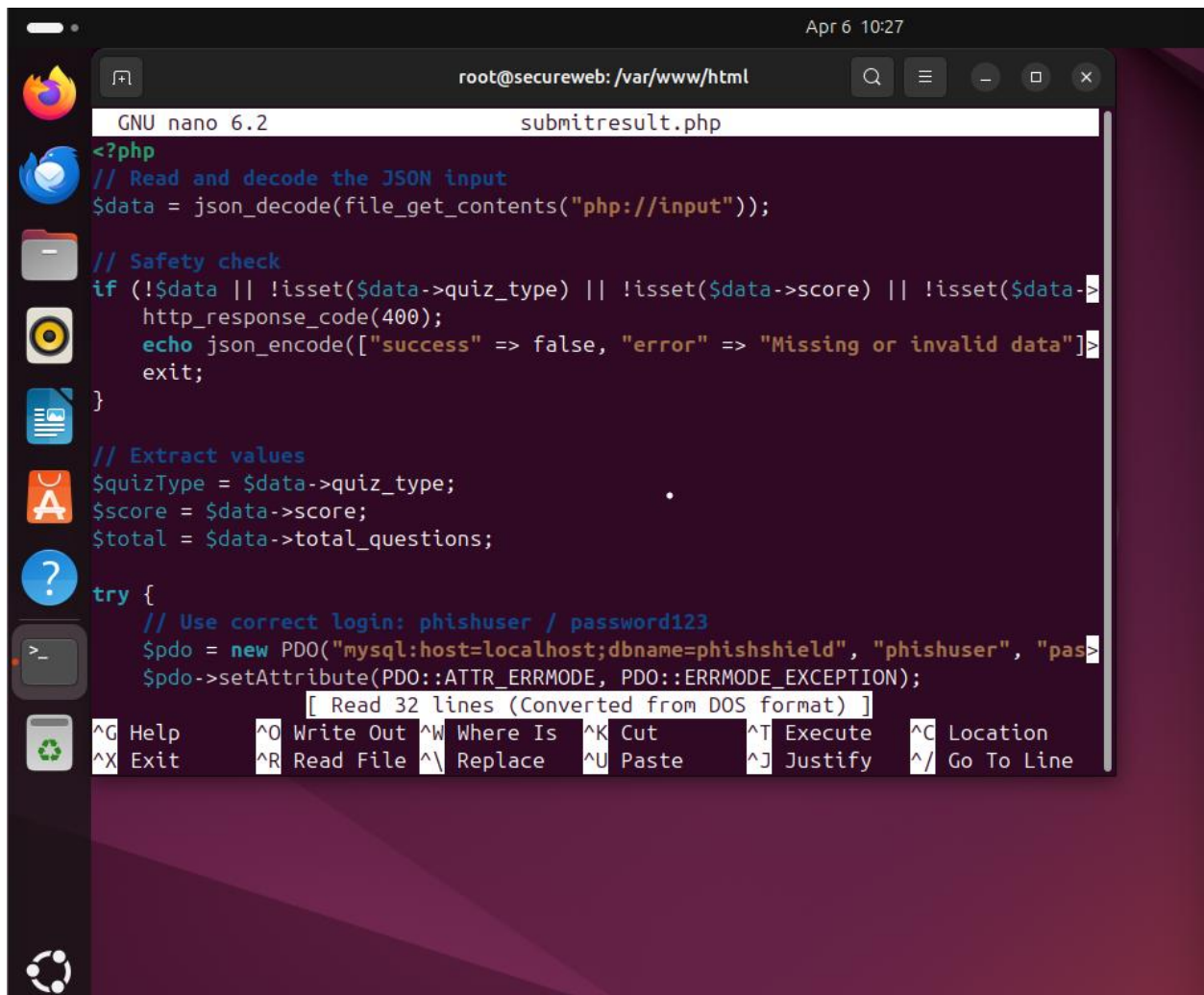
    $stmt = $pdo->query("SELECT quiz_type, score, total_questions, timestamp FROM");
    $results = $stmt->fetchAll(PDO::FETCH_ASSOC);

    echo json_encode(["success" => true, "data" => $results]);
} catch (PDOException $e) {
    http_response_code(500);
    echo json_encode(["success" => false, "error" => $e->getMessage()]);
}
?>
```

[ Read 17 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 |

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



```
root@secureweb: /var/www/html
GNU nano 6.2 submitresult.php

<?php
// Read and decode the JSON input
$data = json_decode(file_get_contents("php://input"));

// Safety check
if (!$data || !isset($data->quiz_type) || !isset($data->score) || !isset($data->total_questions)) {
    http_response_code(400);
    echo json_encode(["success" => false, "error" => "Missing or invalid data"]);
    exit;
}

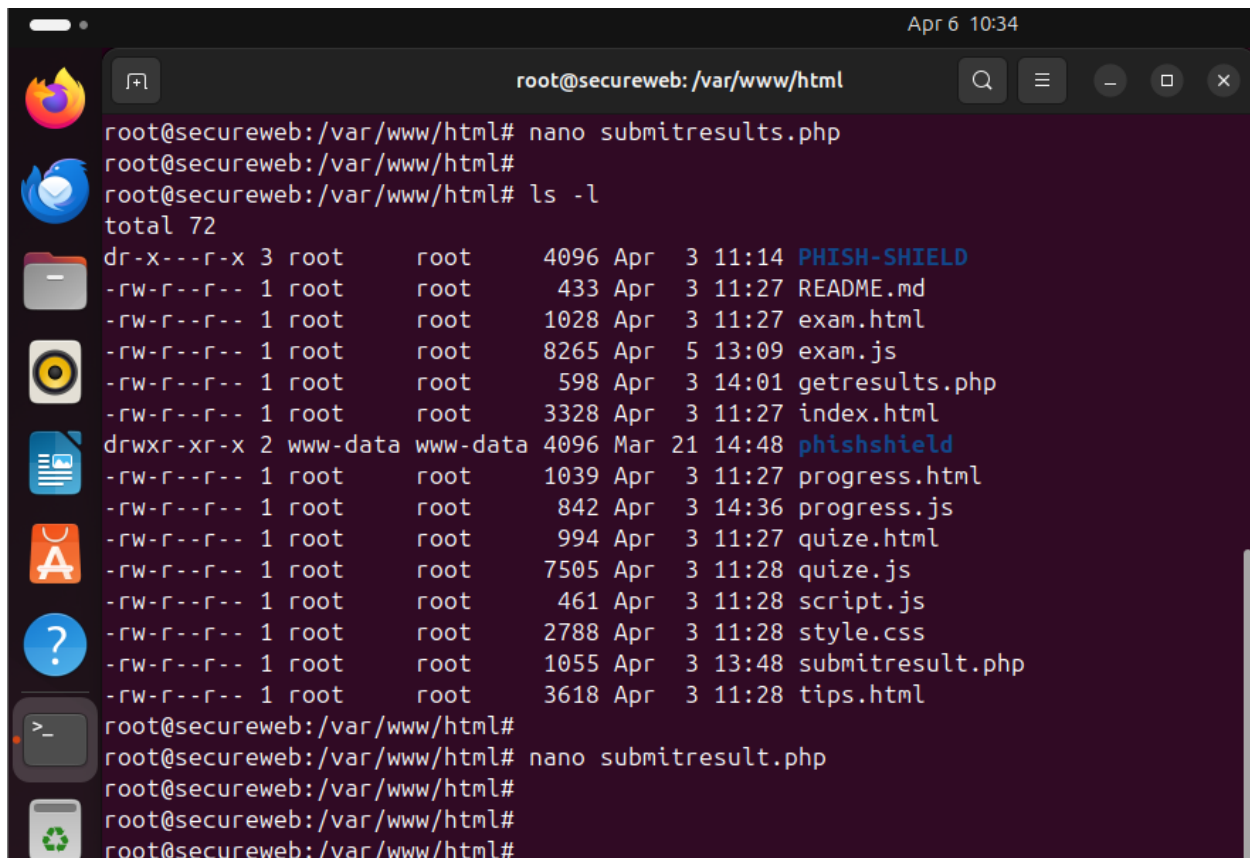
// Extract values
$quizType = $data->quiz_type;
$score = $data->score;
$total = $data->total_questions;

try {
    // Use correct login: phishuser / password123
    $pdo = new PDO("mysql:host=localhost;dbname=phishshield", "phishuser", "password123");
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    [ Read 32 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
```

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



The screenshot shows a terminal window titled 'root@secureweb: /var/www/html' with a timestamp of 'Apr 6 10:34'. The terminal displays the following commands and output:

```
root@secureweb:/var/www/html# nano submitresults.php
root@secureweb:/var/www/html#
root@secureweb:/var/www/html# ls -l
total 72
dr-x---r-x 3 root    root    4096 Apr  3 11:14 PHISH-SHIELD
-rw-r--r-- 1 root    root    433 Apr  3 11:27 README.md
-rw-r--r-- 1 root    root   1028 Apr  3 11:27 exam.html
-rw-r--r-- 1 root    root   8265 Apr  5 13:09 exam.js
-rw-r--r-- 1 root    root    598 Apr  3 14:01 getresults.php
-rw-r--r-- 1 root    root   3328 Apr  3 11:27 index.html
drwxr-xr-x 2 www-data www-data 4096 Mar 21 14:48 phishshield
-rw-r--r-- 1 root    root   1039 Apr  3 11:27 progress.html
-rw-r--r-- 1 root    root    842 Apr  3 14:36 progress.js
-rw-r--r-- 1 root    root    994 Apr  3 11:27 quize.html
-rw-r--r-- 1 root    root   7505 Apr  3 11:28 quize.js
-rw-r--r-- 1 root    root    461 Apr  3 11:28 script.js
-rw-r--r-- 1 root    root   2788 Apr  3 11:28 style.css
-rw-r--r-- 1 root    root   1055 Apr  3 13:48 submitresult.php
-rw-r--r-- 1 root    root   3618 Apr  3 11:28 tips.html
root@secureweb:/var/www/html#
root@secureweb:/var/www/html# nano submitresult.php
root@secureweb:/var/www/html#
root@secureweb:/var/www/html#
root@secureweb:/var/www/html#
```

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>

