**Placement Empowerment Program**

*Cloud Computing and DevOps Centre*

Host a Static Website Locally-

Set Up a Local Server Apache and Host a Simple HTML page with your name

Name: Monisha JR                    Department : CSE

# Introduction and Overview :

- In this POC, we will learn how to host a static website locally using the Apache HTTP Server.
- This process involves setting up a local web server, configuring it correctly, and hosting a simple HTML page.
- By following these steps, you'll get hands-on experience with configuring and running a local Apache server, which is a foundational skill for web hosting and server management.

# Objective :

The goal of this project is to:

1. Set up a local web server using Apache.

2. Configure the server to host static files.

3. Create and host a simple HTML page displaying your name.

# Importance of Local Hosting :

Local hosting is an essential skill for developers, as it allows them to test and experiment with web applications in a controlled environment. It offers several advantages, such as:

- ➤ **Hands-On Learning:** Gain practical experience with server setup and configuration.
- ➤ **Testing Ground:** Safely test and debug websites before deploying them to a live server.
- ➤ **Offline Development:** Work on web projects without requiring an active internet connection.
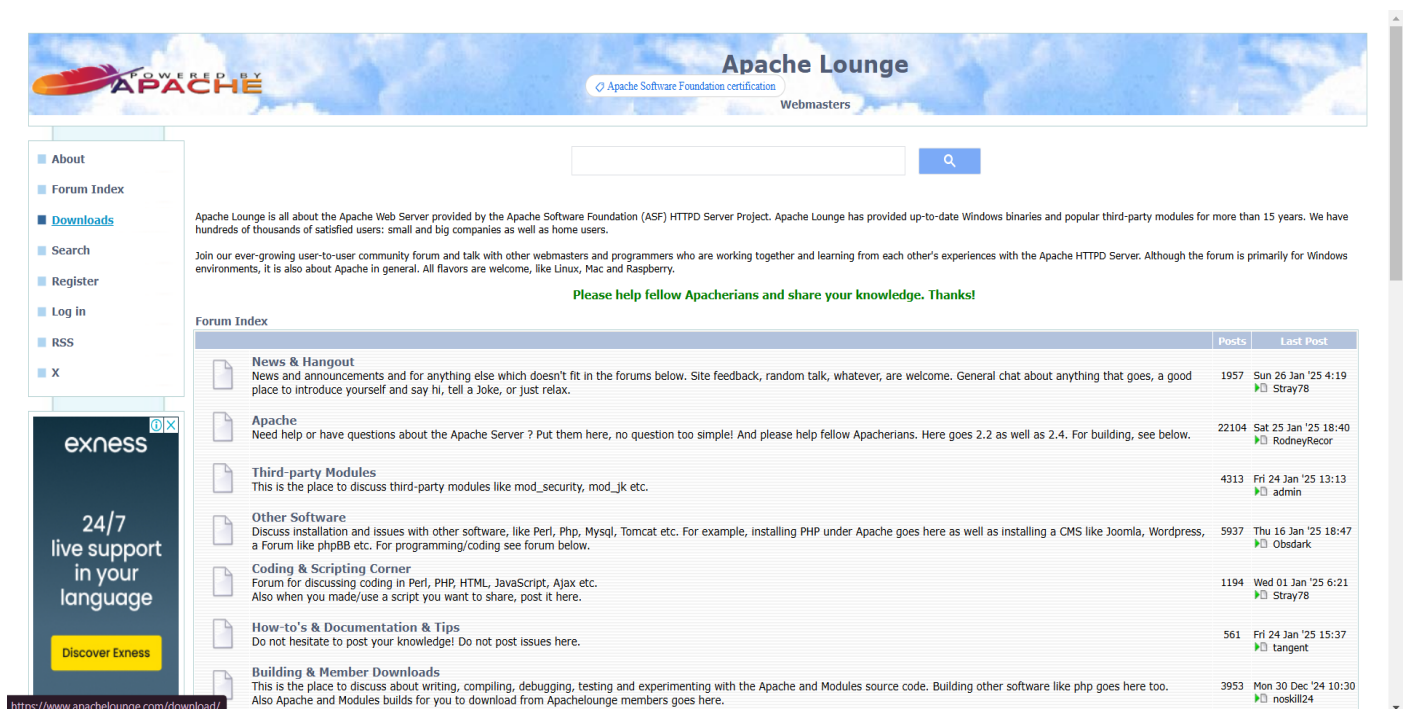
# Step-by-Step Overview :

## Step1:

Search for "Apache Lounge" on Google and click the first link to access the official website.
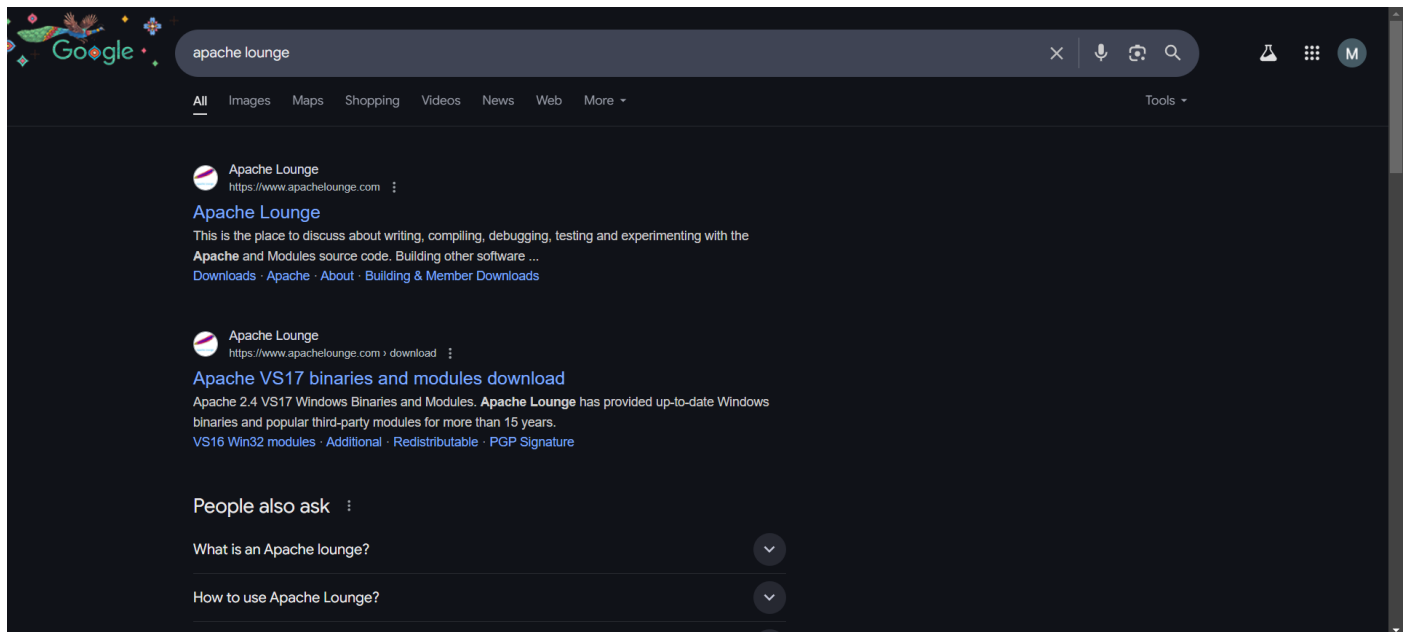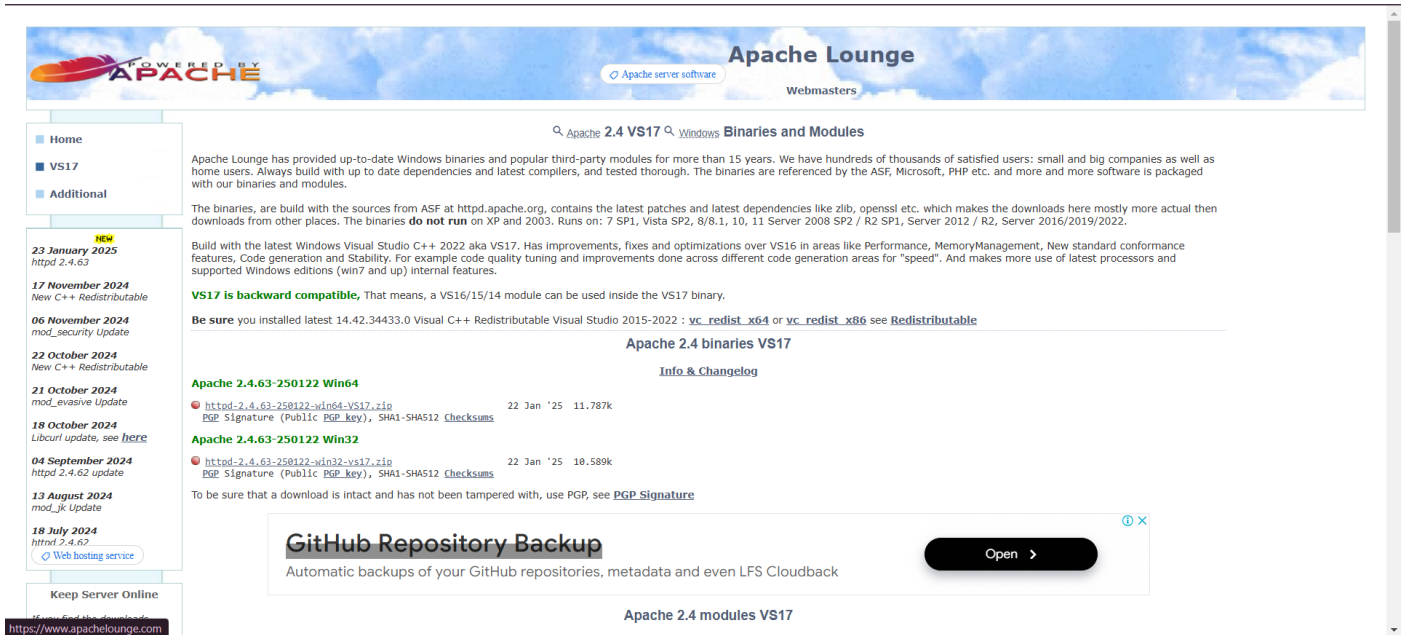


## Step 2 :

Click on the "Downloads" option located on the left-hand side of the Apache Lounge website.

# Step 3 :

Click on the link **"Apache 2.4.62-240904 Win64"** (Windows version), download the file, and extract all its contents.



# Step 4 :

Open Command Prompt as Administrator (Windows + R, type cmd, right-click and select 'Run as Administrator') and use the command cd C:\path\to\apache\bin to set the path to the Apache bin folder.

Then Run the installation command :

**httpd.exe -k install**

# Step 5 :

Navigate to the Apache folder you downloaded, go to the **conf** folder, and right-click on the httpd.conf file; select 'Edit with Notepad'

(Apache/conf/httpd.conf)
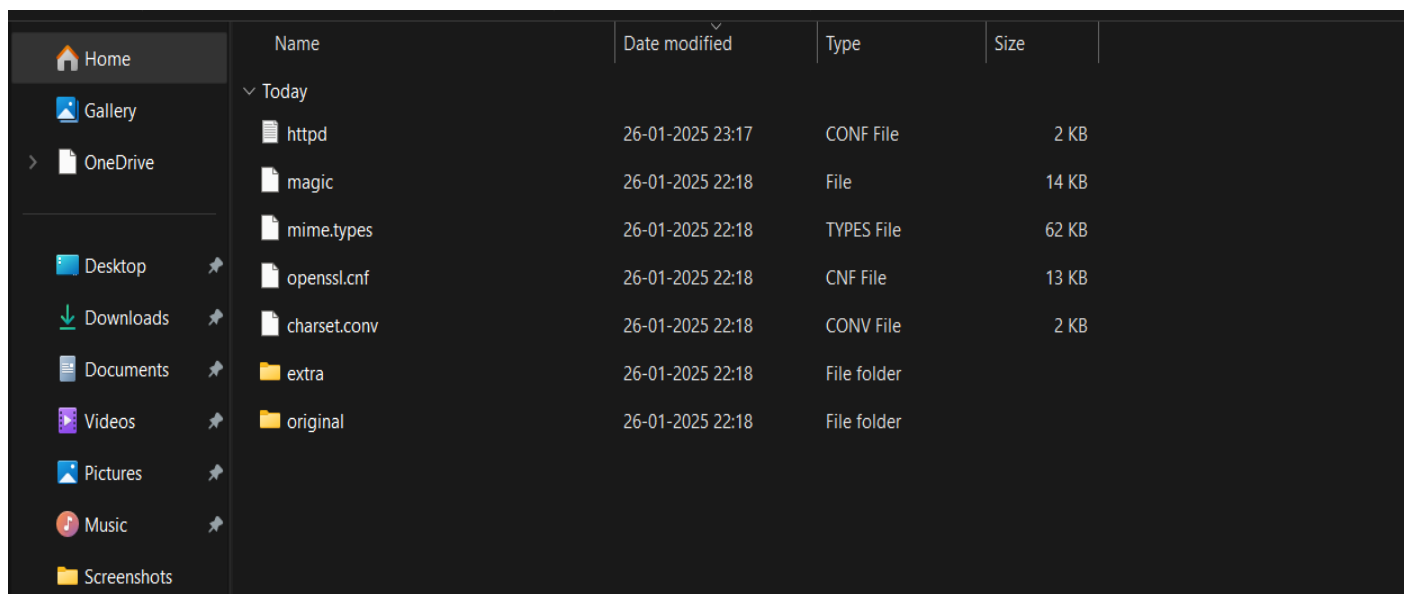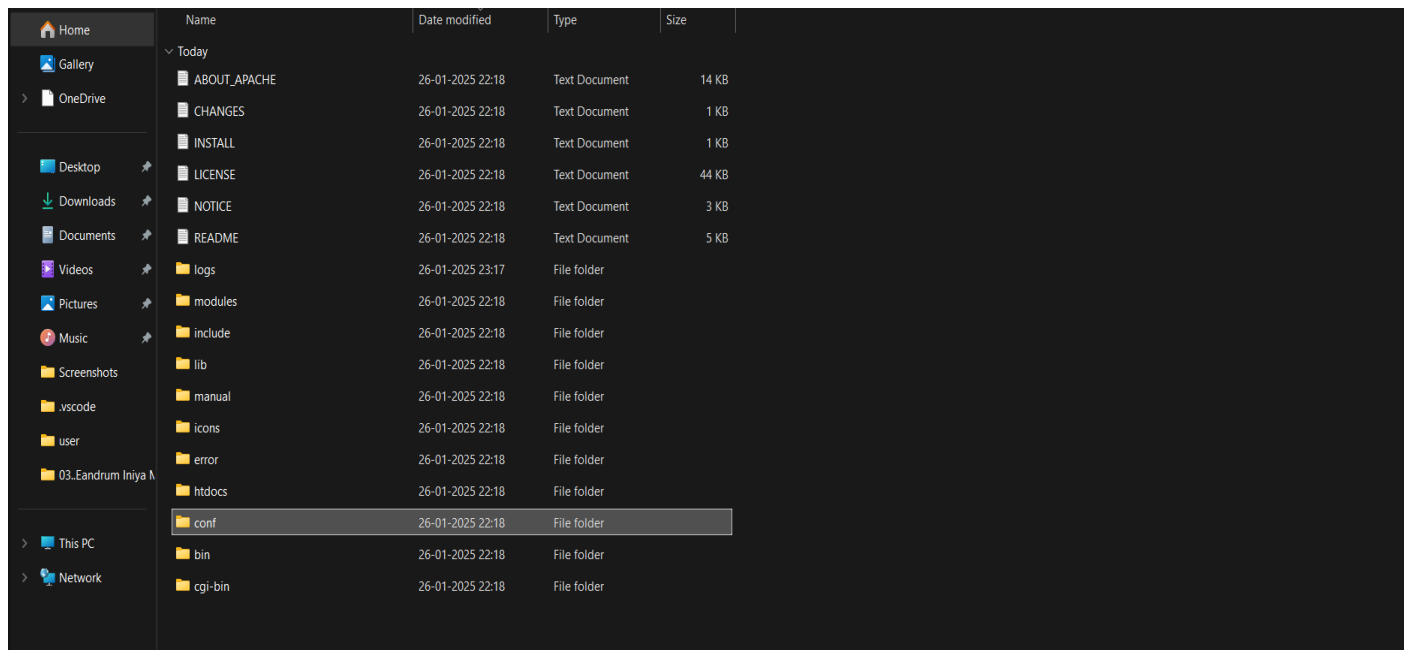




# Step 6 :

Inside the **httpd.conf** file, replace the content with the provided configuration. Ensure you update the SRVROOT directive with your Apache installation path. This configuration defines the server's root directory, listening port, modules,

document root for serving web files, logging paths, and basic permissions, ensuring Apache serves content correctly from the specified htdocs directory.

```
#Define SRVROOT
Define SRVROOT "C:/Users/Hi/Downloads/httpd-2.4.62-240904-win64-VS17/Apache24"

#Ensure the path resolves correctly for DocumentRoot
ServerRoot "${SRVROOT}"

#Listening Port
Listen 80

#ServerName (optional, but recommended for local testing)
ServerName localhost:80

#LoadModules (essential modules)
LoadModule access_compat_module modules/mod_access_compat.so
LoadModule actions_module modules/mod_actions.so
LoadModule alias_module modules/mod_alias.so
LoadModule allowmethods_module modules/mod_allowmethods.so
LoadModule auth_basic_module modules/mod_auth_basic.so
LoadModule authn_core_module modules/mod_authn_core.so
LoadModule authz_core_module modules/mod_authz_core.so
LoadModule dir_module modules/mod_dir.so
LoadModule log_config_module modules/mod_log_config.so
LoadModule mime_module modules/mod_mime.so
LoadModule rewrite_module modules/mod_rewrite.so
LoadModule setenvif_module modules/mod_setenvif.so

#DocumentRoot and Directory configuration
DocumentRoot "${SRVROOT}/htdocs"
<Directory "${SRVROOT}/htdocs">
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

#Logs (you can adjust the paths as needed)
ErrorLog "${SRVROOT}/logs/error.log"
CustomLog "${SRVROOT}/logs/access.log" common

#Additional settings for MIME types, DirectoryIndex, etc.
<IfModule mime_module>
```

# Step 7 :

Open Command Prompt and type the command **httpd.exe -t** to test the configuration file. If the configuration is correct, you should see **'Syntax OK'**.

```
C:\Users\user\Downloads\httpd-2.4.63-250122-win64-VS17\Apache24\bin>httpd.exe -t
Syntax OK

C:\Users\user\Downloads\httpd-2.4.63-250122-win64-VS17\Apache24\bin>
```
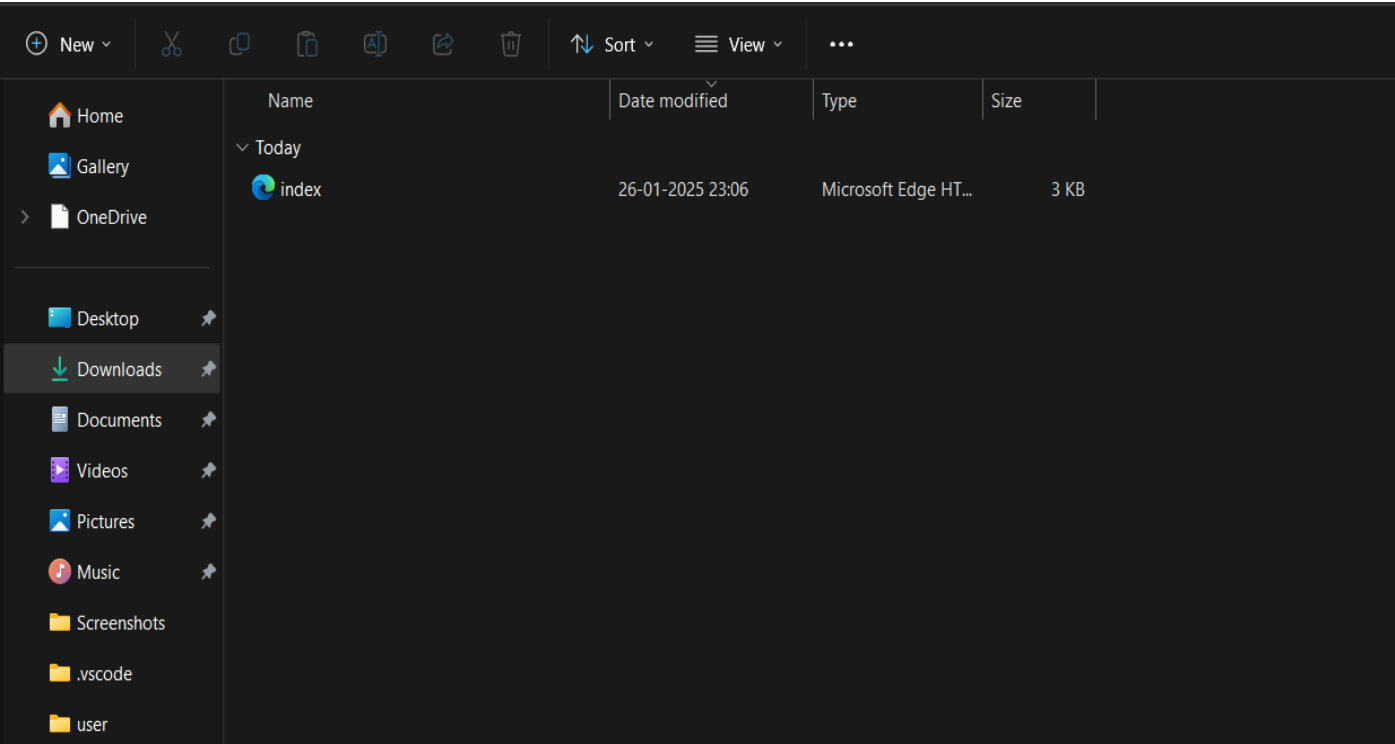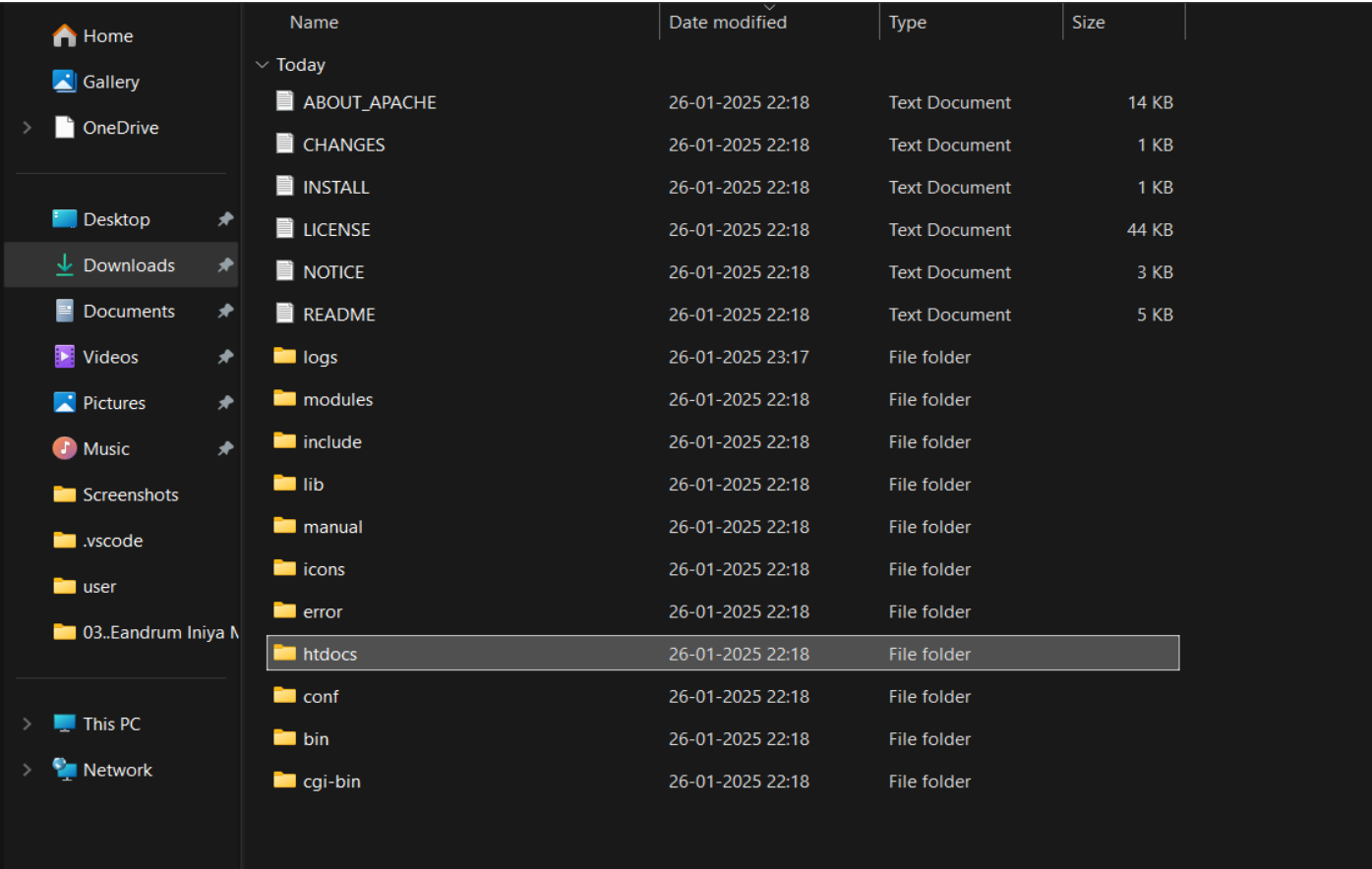
# Step 8 :

Run the command **httpd.exe -k** start to start the Apache server.

```
C:\Users\user\Downloads\httpd-2.4.63-250122-win64-VS17\Apache24\bin>httpd.exe -k start
```

# Step 9 :

Go to the Apache folder, navigate to the **htdocs** folder, and find the **index.html** file. Right-click on it and select 'Edit with Notepad'.

# Step 10 :

Create a simple model to display your name in HTML (you may optionally add CSS for styling).

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Hello, Monisha JR!</title>
    <style>
        body {
            margin: 0;
            height: 100vh;
            display: flex;
            align-items: center;
            justify-content: center;
            font-family: 'Courier New', monospace;
            background: radial-gradient(circle at center, #0f0f0f, #000000);
            color: #00ffcc;
            overflow: hidden;
        }
        .container {
            text-align: center;
            padding: 20px 40px;
            border-radius: 20px;
            background: rgba(0, 0, 0, 0.7);
            box-shadow: 0 0 15px #00ffcc, 0 0 30px #00ffcc;
            animation: pulseGlow 2s infinite;
        }
        .greeting {
            font-size: 2.5rem;
            font-weight: bold;
            text-shadow: 0 0 10px #00ffcc, 0 0 20px #00ffcc;
            white-space: nowrap; /* Ensures the text stays on one line */
        }
        .info {
            font-size: 1rem;
            color: #e0e0e0;
            margin-top: 10px;
            opacity: 0.85;
        }
        @keyframes pulseGlow {
            0%, 100% {

        @keyframes pulseGlow {
            0%, 100% {
                box-shadow: 0 0 15px #00ffcc, 0 0 30px #00ffcc;
            }
            50% {
                box-shadow: 0 0 25px #00ffcc, 0 0 40px #00ffcc;
            }
        }
        .cyber-grid {
            position: absolute;
            top: 0;
            left: 0;
            width: 100%;
            height: 100%;
            background: repeating-linear-gradient(
                0deg,
                rgba(0, 255, 204, 0.1),
                rgba(0, 255, 204, 0.1) 1px,
                transparent 1px,
                transparent 50px
            ),
            repeating-linear-gradient(
                90deg,
                rgba(0, 255, 204, 0.1),
                rgba(0, 255, 204, 0.1) 1px,
                transparent 1px,
                transparent 50px
            );
            pointer-events: none;
        }
    </style>
</head>
<body>
    <div class="cyber-grid"></div>
    <div class="container">
        <div class="greeting">Hello, Monisha JR!</div>
        <div class="info">This website is locally hosted on Apache</div>
    </div>
</body>
</html>
```

# Step 11 :

Open the Chrome browser and type **localhost/index.html** in the address bar. You should be able to see the website hosted successfully.
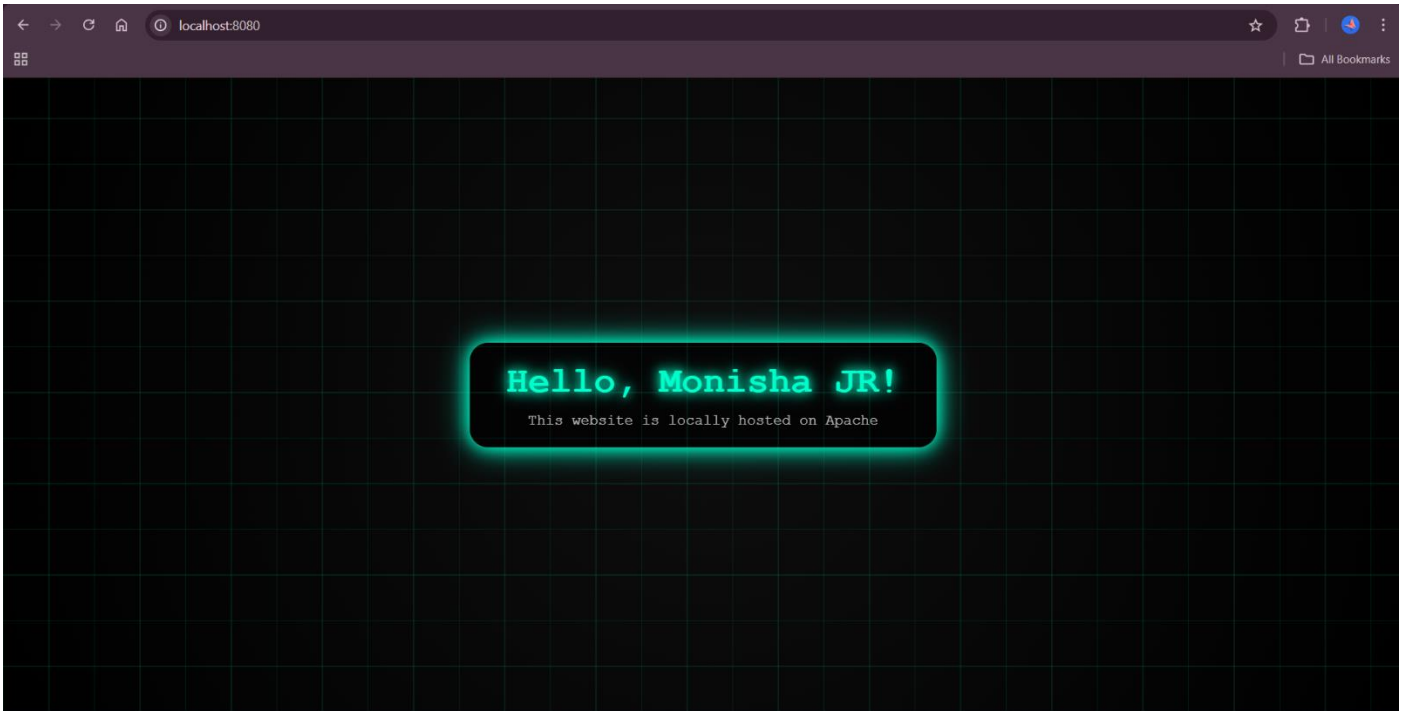


# **Outcome :**

By completing this POC, you will:

    1. Successfully configure and run an Apache server locally.

    2. Host a static HTML website that displays your name.

    3. Understand the basics of web server configuration and file hosting.