

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: Shana R S

Department : CSE

Introduction:

In today's digital world, understanding how web servers work is essential for anyone interested in web development or cloud computing. This task focuses on setting up a local server using Apache HTTP Server, one of the most widely used open-source web servers. By doing this, I learned how websites are hosted, how servers process requests, and how to troubleshoot common configuration issues.

Why Is This Task Important?

When developing websites, it's inefficient to deploy changes to a live server every time. A local server allows developers to test and modify their websites before making them publicly available. Apache provides a simple way to set up such a server on a personal computer.

Steps Followed

1. **Downloaded Apache** – Obtained the latest Apache version to set up the server.
2. **Extracted Files** – Placed Apache files in a suitable directory for installation.
3. **Installed Apache** – Used `httpd.exe -k install` to register Apache as a Windows service.
4. **Configured Apache** – Edited the `httpd.conf` file to define settings like the document root and server port.
5. **Started the Server** – Ran the command `httpd.exe -k start` to launch Apache.
6. **Hosted the Page** – Placed the HTML file in the `htdocs` folder and accessed it via `http://localhost:`

Where Can This Be Applied?

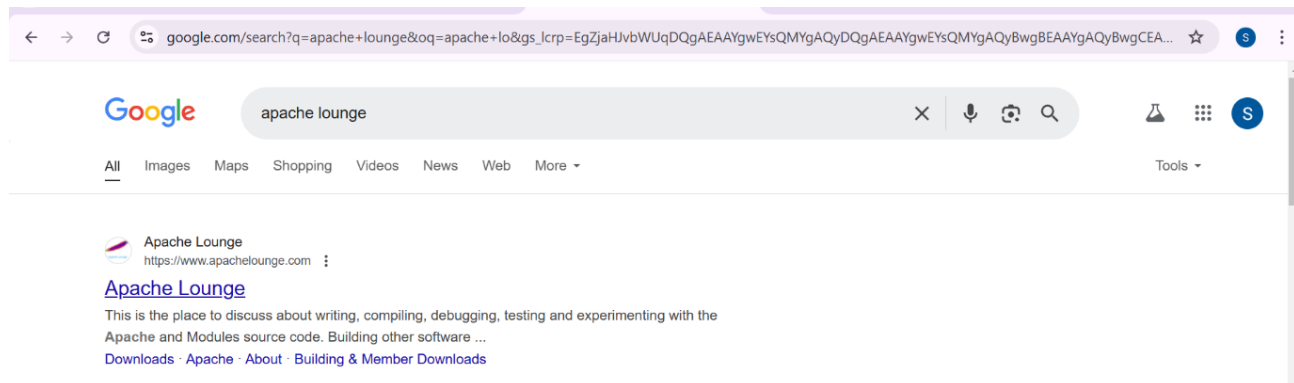
- **Web Development:** Developers use local servers to test their websites before deployment.
- **Cloud Computing & Hosting:** Understanding Apache helps when working with cloud-based servers.
- **Cybersecurity Testing:** Security professionals use local servers to analyze vulnerabilities.
- **Personal Projects & Prototyping:** Great for hosting small projects before making them live.

Step-by-Step Overview

Step1: Browse the Apache Website

✓ Instruction:

- Open your web browser (Chrome, Edge, Firefox).
- Go to the **official Apache website** (<https://www.apachelounge.com/download/>).

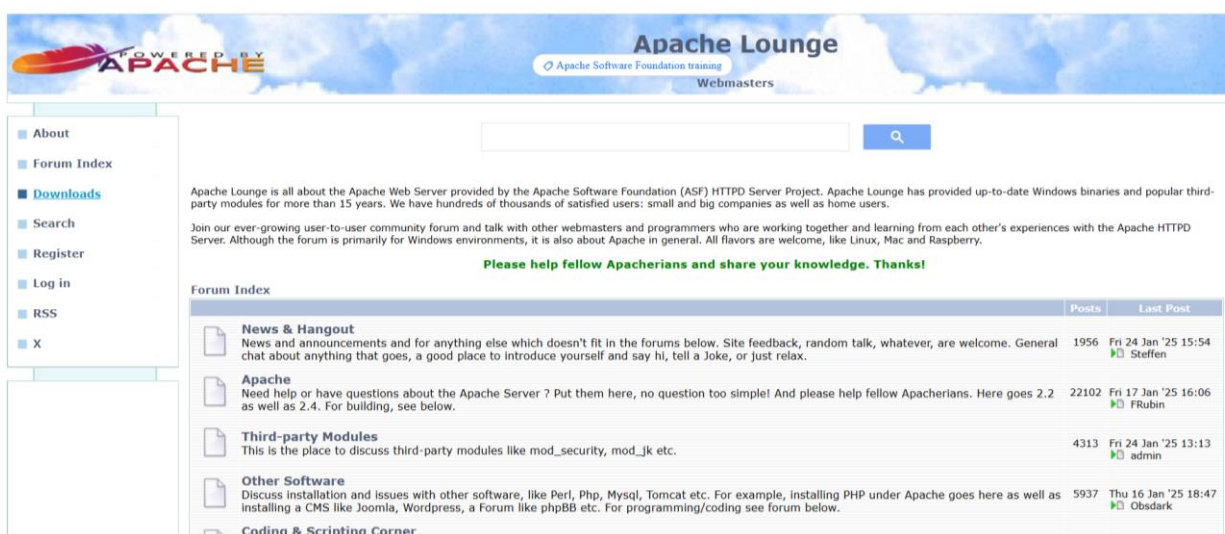


To download the **Apache HTTP Server** needed for hosting websites locally.

Step 2 : Navigate to the Downloads Section

✓ Instruction:

- Scroll down the webpage.
- Find the **Download section** for Apache.



This section contains different versions of Apache for Windows.

Step 3 : Select and Download Apache 64-bit

✓ Instruction:

- Click on the **64-bit Windows version** of Apache.
- Wait for the file to download completely.

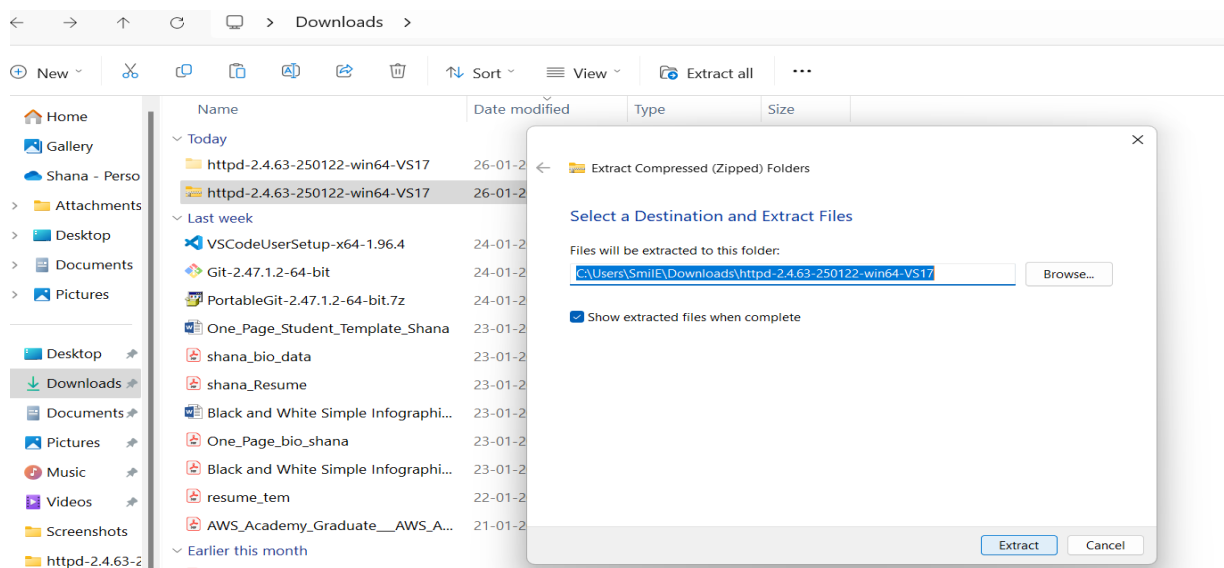


Ensure you download the correct **64-bit version** compatible with your Windows system.

Step 4 : Extract the Apache Files

✓ Instruction:

- Locate the downloaded **ZIP** file in your **Downloads** folder.
- **Right-click** on it and select **Extract All...**
- Extract it to a suitable location, e.g., C:\Apache24.

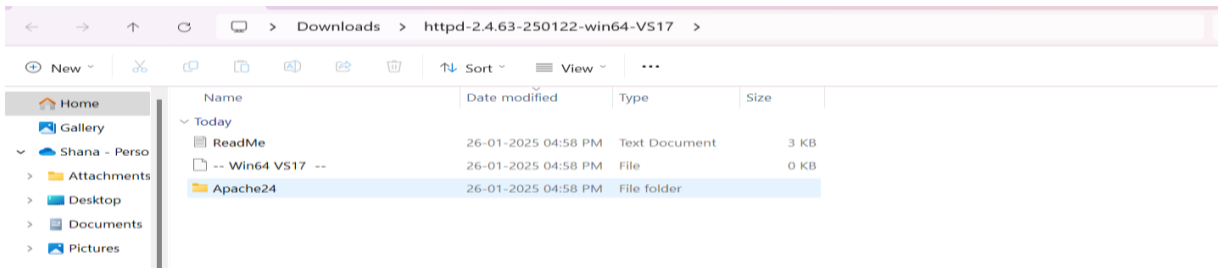


Extracting the files makes them accessible for installation.

Step 5 : Open the Apache Folder

✓ Instruction:

- Go to the location where you extracted the Apache files.
- Open the **Apache24** folder to check its contents.

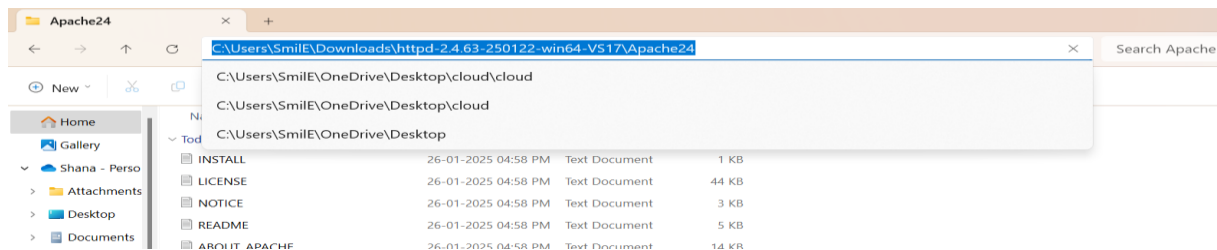


This folder contains all necessary files to run Apache.

Step 6 : Copy the Apache Path

✓ Instruction:

- Click inside the **Apache24** folder.
- Copy the **full path** (e.g., C:\Apache24).



This path is needed to navigate to the Apache directory in the command prompt.

Step 7 : Open Command Prompt and Navigate to Apache bin

✓ Instruction:

- Open **Command Prompt (cmd)** as **Administrator**.
- Type the following command and press Enter:

```
cd C:\Apache24\bin
```

```
Microsoft Windows [Version 10.0.22621.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Smile>cd C:\Users\Smile\Downloads\httpd-2.4.63-250122-win64-VS17\Apache24

C:\Users\Smile\Downloads\httpd-2.4.63-250122-win64-VS17\Apache24>cd bin
```

Navigating to the bin folder allows you to run Apache commands.

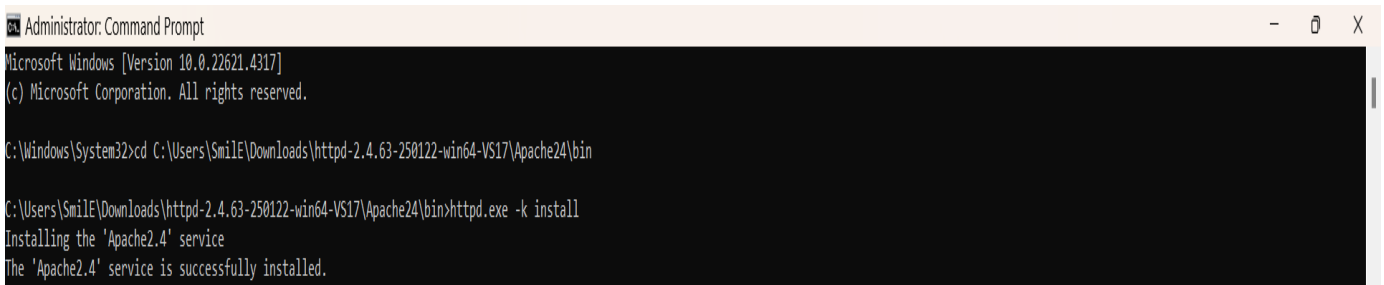
Step 8 : Install Apache as a Service

✓ Instruction:

- Run the command:

```
httpd.exe -k install
```

- If installed successfully, you should see:
“The 'Apache2.4' service is successfully installed.”



```
Administrator: Command Prompt

Microsoft Windows [Version 10.0.22621.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>cd C:\Users\Smile\Downloads\httpd-2.4.63-250122-win64-VS17\Apache24\bin

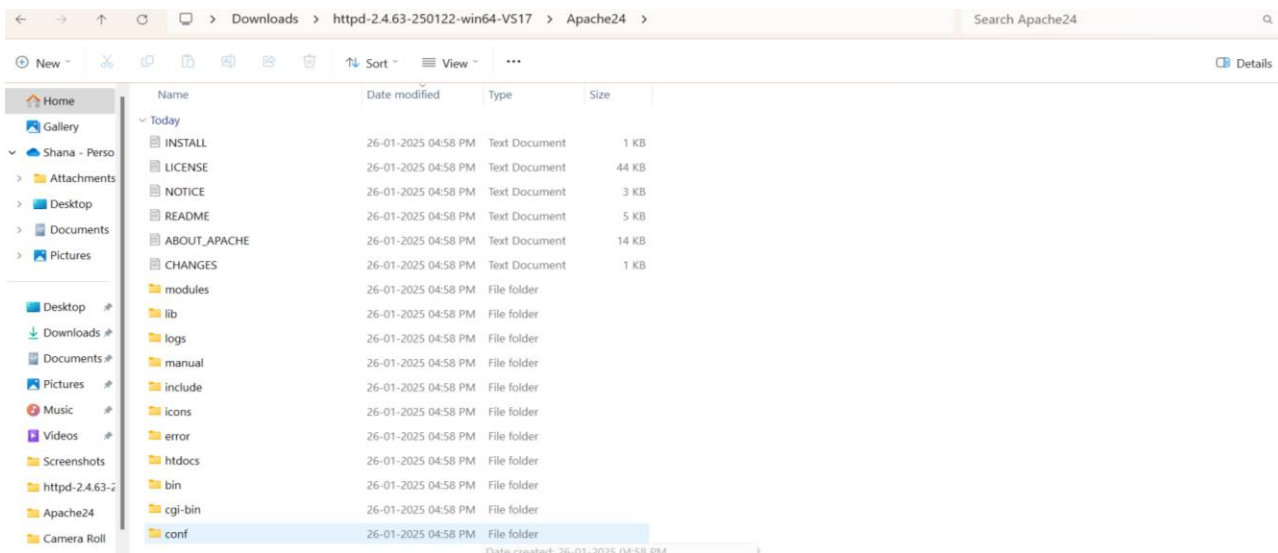
C:\Users\Smile\Downloads\httpd-2.4.63-250122-win64-VS17\Apache24\bin>httpd.exe -k install
Installing the 'Apache2.4' service
The 'Apache2.4' service is successfully installed.
```

Installing Apache as a service allows it to start automatically.

Step 9 : Open the Apache Configuration Folder

✓ Instruction:

- Go to the **Apache24** folder.
- Open the **conf** folder inside it.



This folder contains the Apache configuration file (httpd.conf).

Step10: Open and Edit httpd.conf File

✓ Instruction:

- Locate and open httpd.conf in Notepad or any text editor.
- Find and modify the following lines:

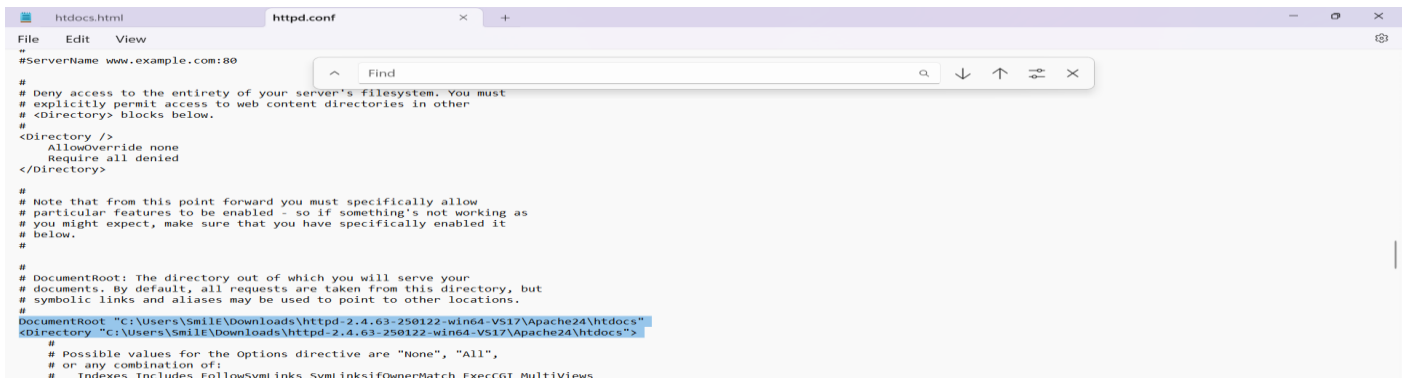
Change ServerRoot:

ServerRoot "C:/Apache24"

Change DocumentRoot:

DocumentRoot "C:/Apache24/htdocs"

```
htdocs.html httpd.conf
File Edit View
# Configuration and logfile names: If the filenames you specify for many
# of the server's control files begin with "/" (or "drive:/" for Win32), the
# server will use that explicit path.  If the filenames do "not" begin
# with "/", the value of ServerRoot is prepended -- so "logs/access_log"
# with ServerRoot set to "/usr/local/apache2" will be interpreted by the
# server as "/usr/local/apache2/logs/access_log", whereas "/logs/access_log"
# will be interpreted as '/logs/access_log'.
#
# NOTE: Where filenames are specified, you must use forward slashes
# instead of backslashes (e.g., "c:/apache" instead of "c:\apache").
# If a drive letter is omitted, the drive on which httpd.exe is located
# will be used by default.  It is recommended that you always supply
# an explicit drive letter in absolute paths to avoid confusion.
#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# Do not add a slash at the end of the directory path.  If you point
# ServerRoot at a non-local disk, be sure to specify a local disk on the
# Mutex directive, if file-based mutexes are used.  If you wish to share the
# same ServerRoot for multiple httpd daemons, you will need to change at
# least PidFile.
#
Define SRVROOT "c:/Apache24"
ServerRoot "C:/Users/Smile/Downloads/httpd-2.4.63-250122-win64-V517/Apache24/conf"
#
# Mutex: Allows you to set the mutex mechanism and mutex file directory
# for individual mutexes, or change the global defaults
```

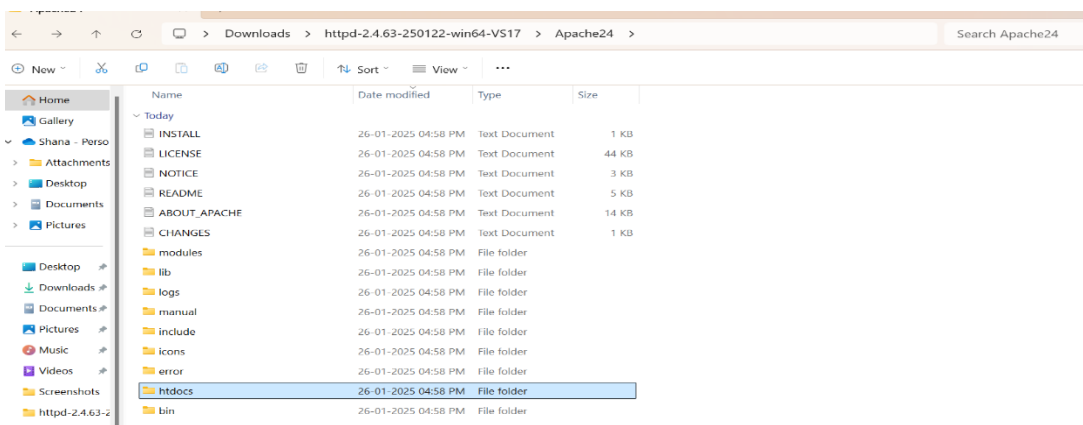
```
# httdocs.html httpd.conf
#
# ServerName www.example.com:80
#
#
# Deny access to the entirety of your server's filesystem. You must
# explicitly permit access to web content directories in other
# <Directory> blocks below.
#
<Directory />
    AllowOverride none
    Require all denied
</Directory>
#
# Note that from this point forward you must specifically allow
# particular features to be enabled - so if something's not working as
# you might expect, make sure that you have specifically enabled it
# below.
#
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "C:\Users\Smile\Downloads\httpd-2.4.63-250122-win64-VS17\Apache24\htdocs"
<Directory "C:\Users\Smile\Downloads\httpd-2.4.63-250122-win64-VS17\Apache24\htdocs">
    #
    # Possible values for the Options directive are "None", "All",
    # or any combination of:
    #   Indexes Includes FollowSvmlinks SvmlinksifOwnerMatch ExecCGI MultiViews
    #
```

This ensures Apache serves files from the correct location.

Step 11 : Create httdocs Folder (If Not Present)

✓ Instruction:

- Inside **C:\Apache24**, check if the **httdocs** folder exists.
- If not, **create a new folder** and name it **httdocs**.



The httdocs folder is where your website files should be stored.

Step 12 : Create an HTML File

✓ Instruction:

- Open **Notepad** or **VS Code**.
- Copy and paste the following HTML code

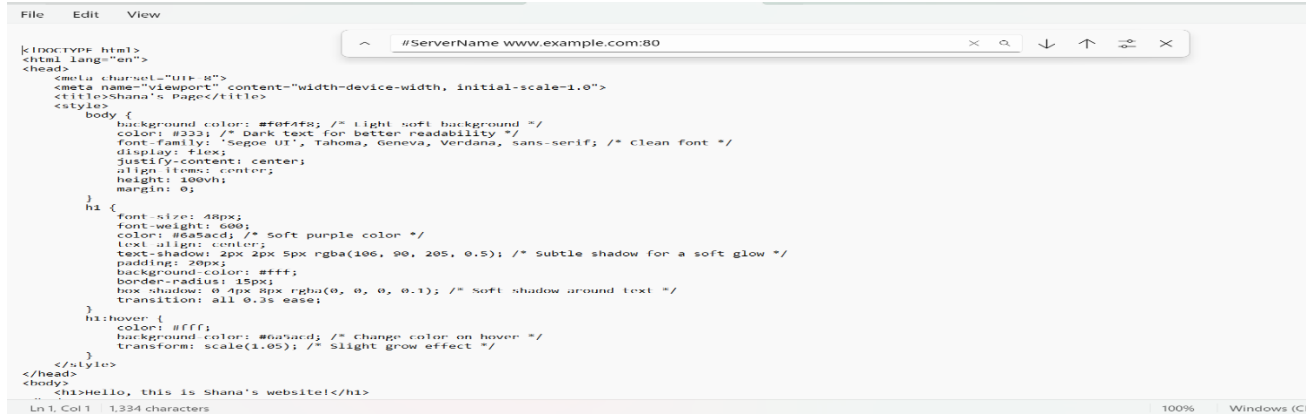
Open

Notepad

or

VS

Code.

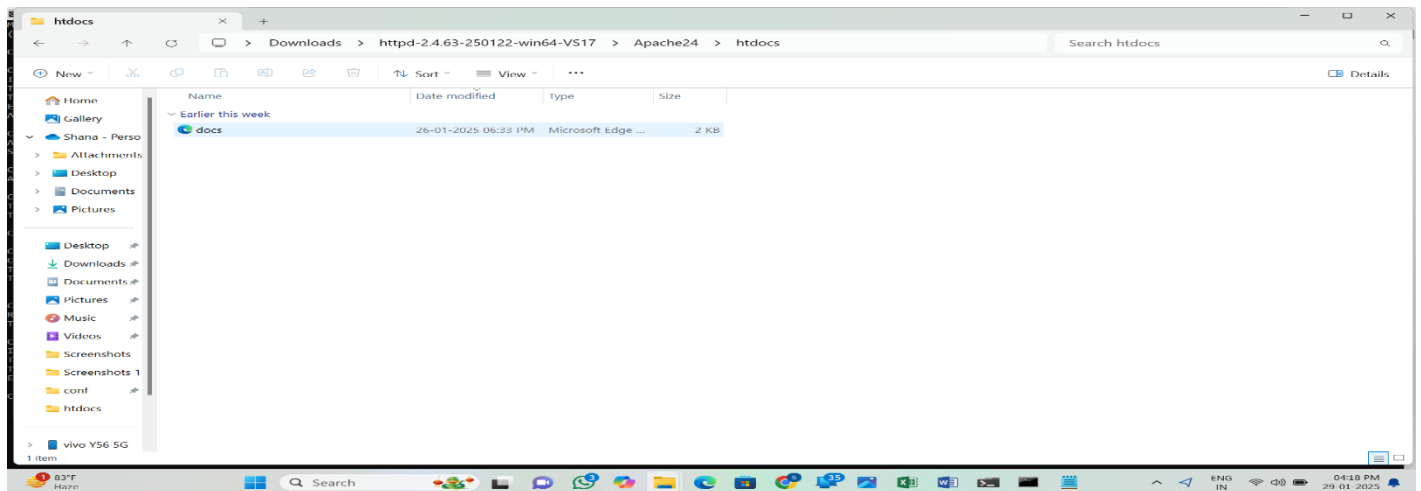


```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Shana's Page</title>
  <style>
    body {
      background-color: #000000; /* light soft background */
      color: #333; /* Dark text for better readability */
      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif; /* Clean font */
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
      margin: 0;
    }
    h1 {
      font-size: 48px;
      font-weight: 600;
      color: #66aadd; /* Soft purple color */
      text-align: center;
      padding: 20px;
      border-radius: 15px;
      background-color: #fff;
      border: 2px solid #66aadd;
      box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1); /* Soft shadow around text */
      transition: all 0.3s ease;
    }
    h1:hover {
      color: #fff;
      background-color: #66aadd; /* Change color on hover */
      transform: scale(1.05); /* Slight grow effect */
    }
  </style>
</head>
<body>
  <h1>Hello, this is Shana's website!</h1>
</body>
</html>
```

Step 13 : Save the File as index.html

✓ Instruction:

- Open **Notepad or VS Code**.
- Copy and paste the following HTML code
- Click **File** → **Save As**.
- Select **All Files** instead of .txt.
- Name the file **index.html** and save it inside C:\Apache24\htdocs.



Apache will automatically serve **INDEX.HTML** when accessed.

Step 14: Restart Apache Server

✓ Instruction:

- Open **Command Prompt as Administrator**.
- Navigate to the bin folder:

```
cd C:\Apache24\bin
```

- Restart Apache:

`httpd.exe -k restart`

- Open **Google Chrome, Edge, or Firefox**.
- Type the following in the address bar:
<http://localhost>



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.