

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: AVINASH S

Department : CSE

Introduction

In this POC, we will explore the process of hosting a static website locally using the Apache HTTP Server. Hosting a website locally involves setting up a local web server, configuring it properly, and deploying a basic HTML page. By following these steps, you will gain hands-on experience with:

1. **Installing and configuring Apache** – Understanding how to set up a popular web server on your machine.
2. **Hosting static content** – Deploying and serving a simple static HTML page from the local server.
3. **Foundational web hosting skills** – Learning how web servers function and preparing for more advanced server management tasks.

Project Goal

The goal of this project is to set up and configure a local web server using the Apache HTTP Server to host static content. By the end of this project, you will have gained valuable skills in web server management, including:

1. Setting up a local web server using Apache – Learning how to install and configure Apache on your local machine to serve web content.
2. Configuring the server to host static files – Understanding the process of setting up Apache to handle static files such as HTML, CSS, and JavaScript, and configuring the server for optimal performance.
3. Creating and hosting a simple HTML page – Developing a basic HTML page that displays your name and hosting it on your local server, demonstrating your ability to manage web content.

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.

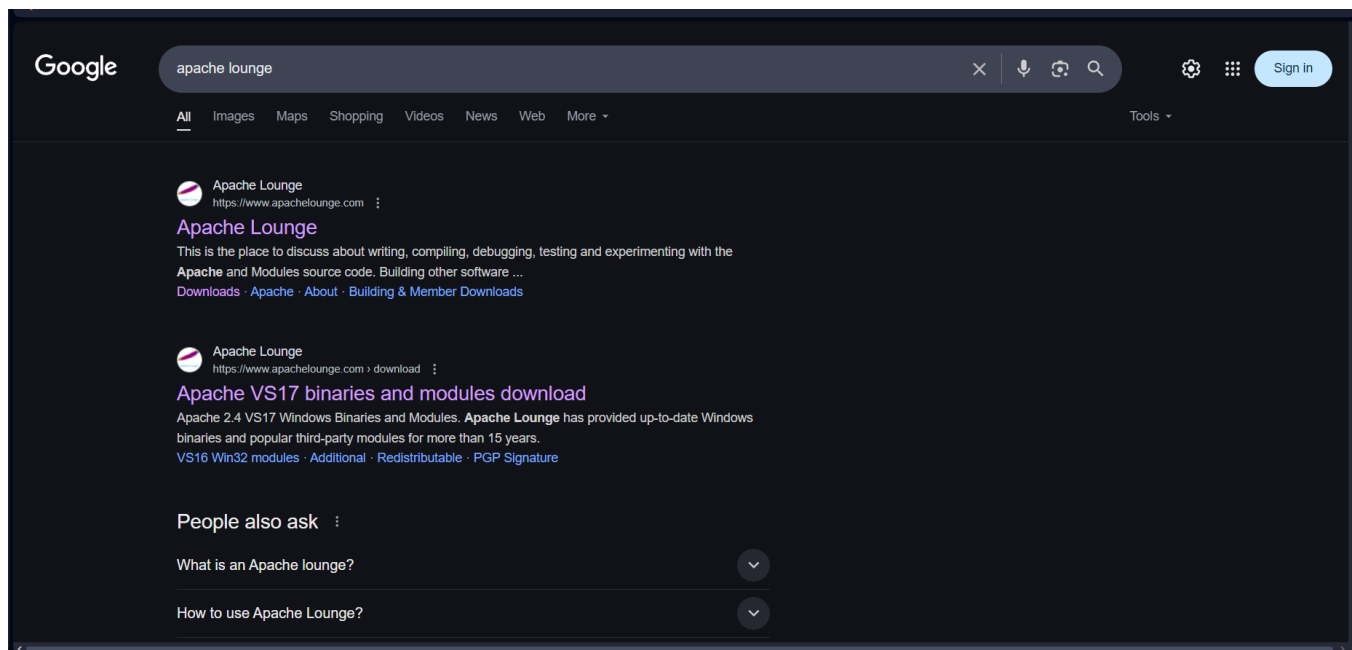
Testing Ground: Safely test and debug websites before deploying them to a live server.

Offline Development: Work on web projects without requiring an active internet connection.

Step-by-Step Procedure:

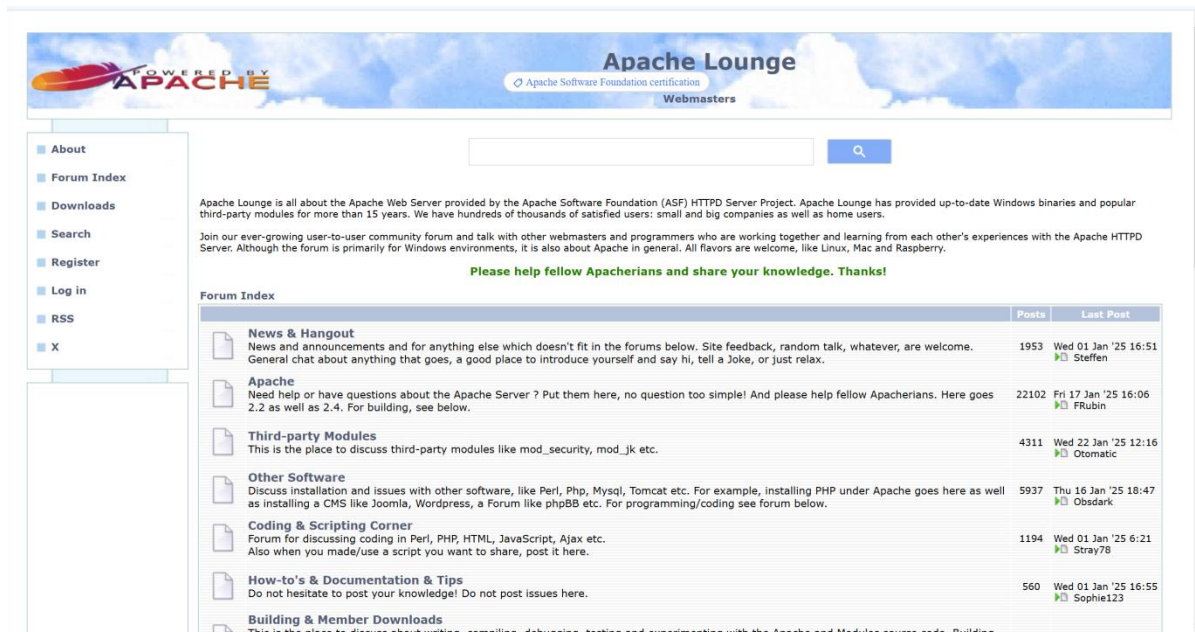
Step1:

Search for "Apache Lounge" on Google and open the webpage.



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 zip file.



Step 4 :

Open Command Prompt as Administrator (Windows + R, type cmd, right-click and select 'Run as Administrator') and use the command `cd C:\Apache\Apache24\bin` to set the path to the Apache bin folder.

```
C:\Windows\System32>cd C:\Apache\Apache24\bin
```

Step 5 :

Then Run the installation command :

httpd.exe -k install

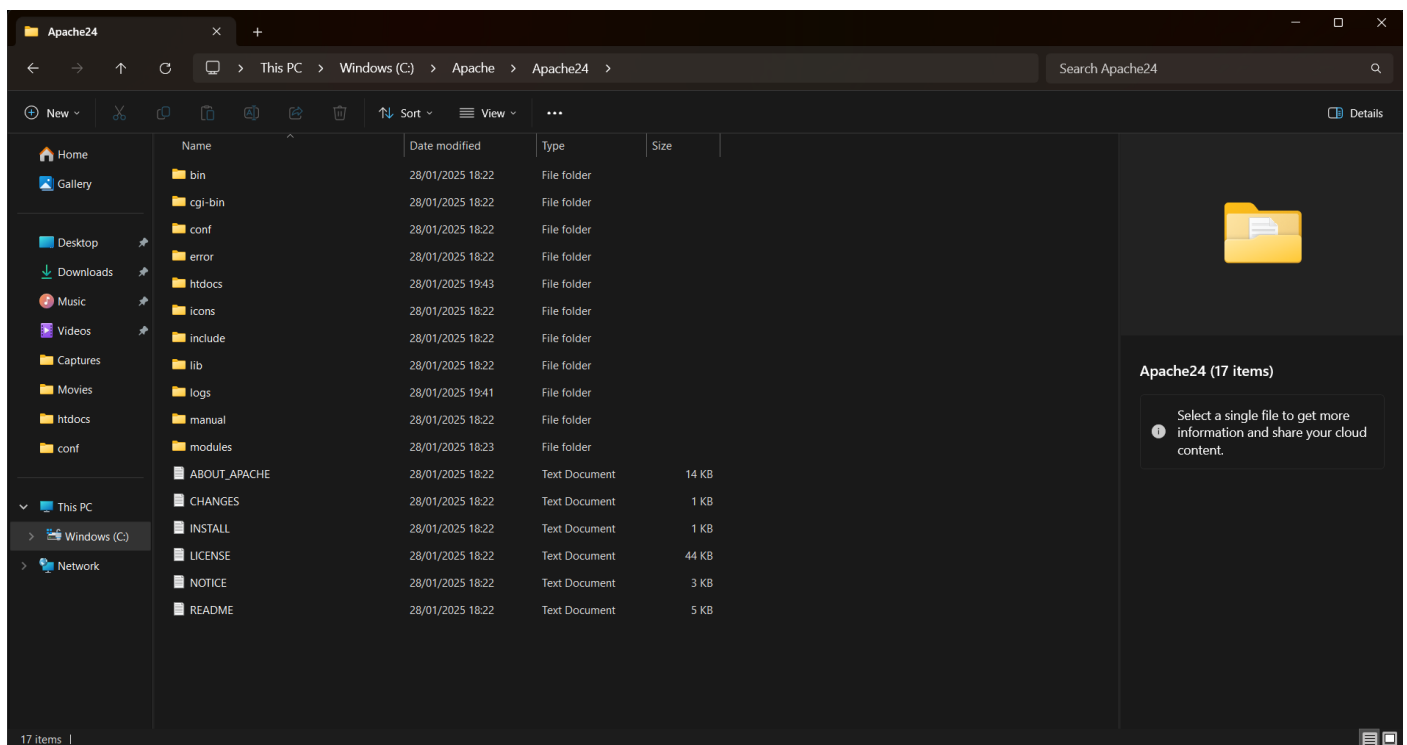
```
C:\Apache\Apache24\bin>httpd.exe -k
httpd.exe: illegal option -- k
Usage: httpd.exe [-D name] [-d directory] [-f file]
               [-C "directive"] [-c "directive"]
               [-w] [-k start|restart|stop|shutdown] [-n service_name]
               [-k install|config|uninstall] [-n service_name]
               [-v] [-V] [-h] [-l] [-L] [-t] [-T] [-S] [-X]

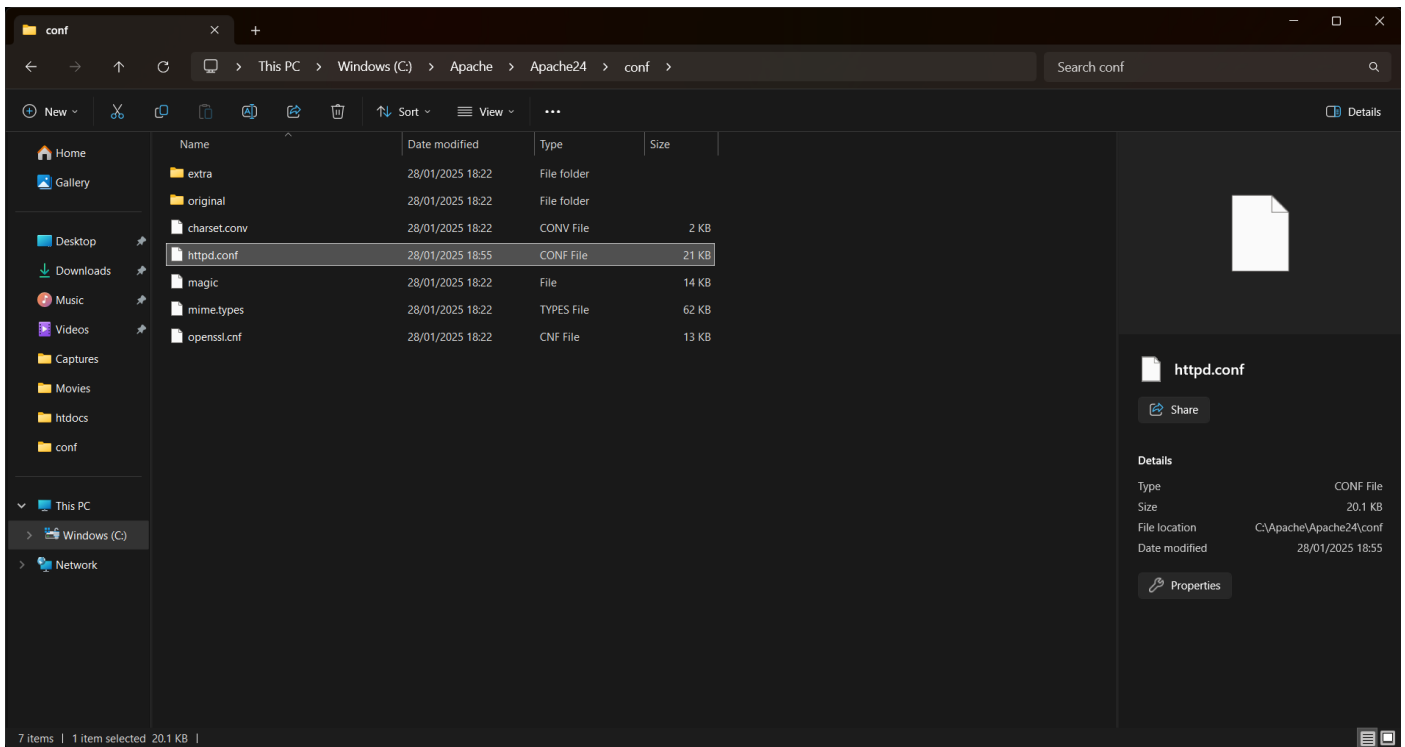
Options:
  -D name           : define a name for use in <IfDefine name> directives
  -d directory      : specify an alternate initial ServerRoot
  -f file           : specify an alternate ServerConfigFile
```

Step 6 :

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 7 :

Setting up a local web server using Apache HTTP Server to host a static website. You will modify the `httpd.conf` file to configure the server's root directory, port, modules, and permissions for serving content. The `DocumentRoot` directive will point to the folder containing your HTML files, and the `Listen` directive will ensure Apache listens on port 80. After configuring Apache, you will create a simple HTML page and verify that the server correctly serves the content. This project provides a foundational understanding of web hosting and Apache server management.

```
#
Define SRVROOT "c:/Apache24"
Define SRVROOT C:\Apache\Apache24

ServerRoot "${SRVROOT}"

#
# Mutex: Allows you to set the mutex mechanism and mutex file directory
# for individual mutexes, or change the global defaults
#
# Uncomment and change the directory if mutexes are file-based and the default
# mutex file directory is not on a local disk or is not appropriate for some
# other reason.
#
# Mutex default:logs
```

Step 8 :

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:\Apache\Apache24\bin>httpd.exe -t
AH00558: httpd.exe: Could not reliably determine the server's fully qualified domain name, using fe80::ac8c:c822:6a04:d504. Set the 'ServerName' directive globally to suppress this message
Syntax OK
```

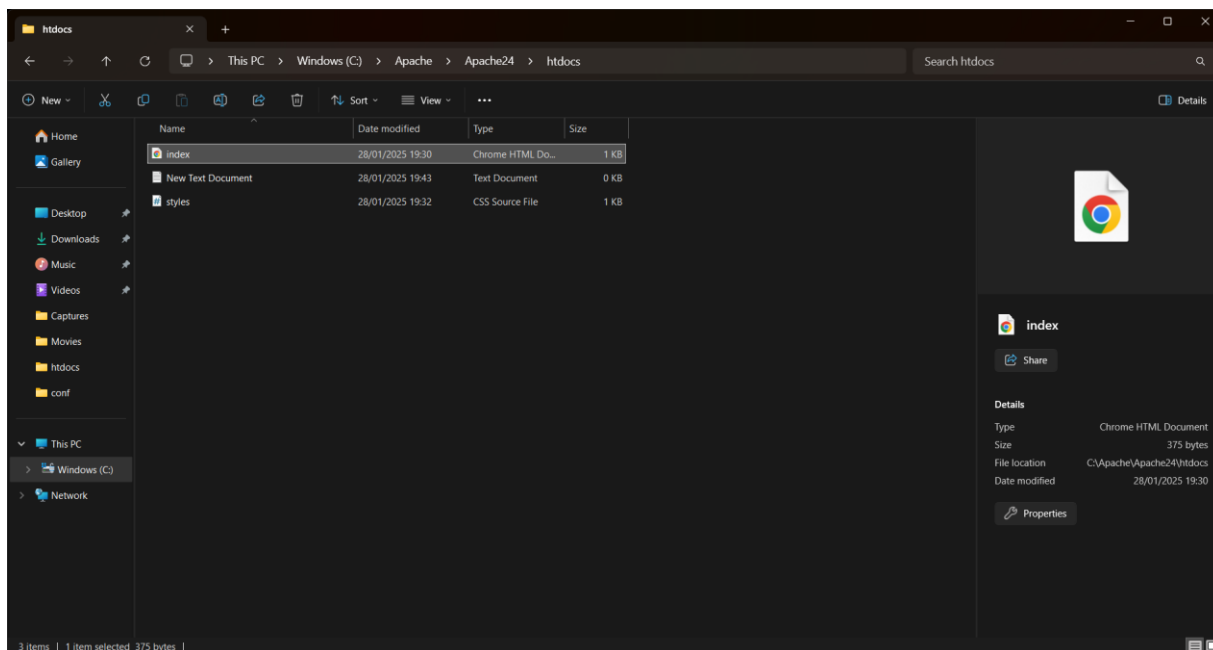
Step 9 :

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

```
C:\Apache\Apache24\bin>httpd.exe -k
```

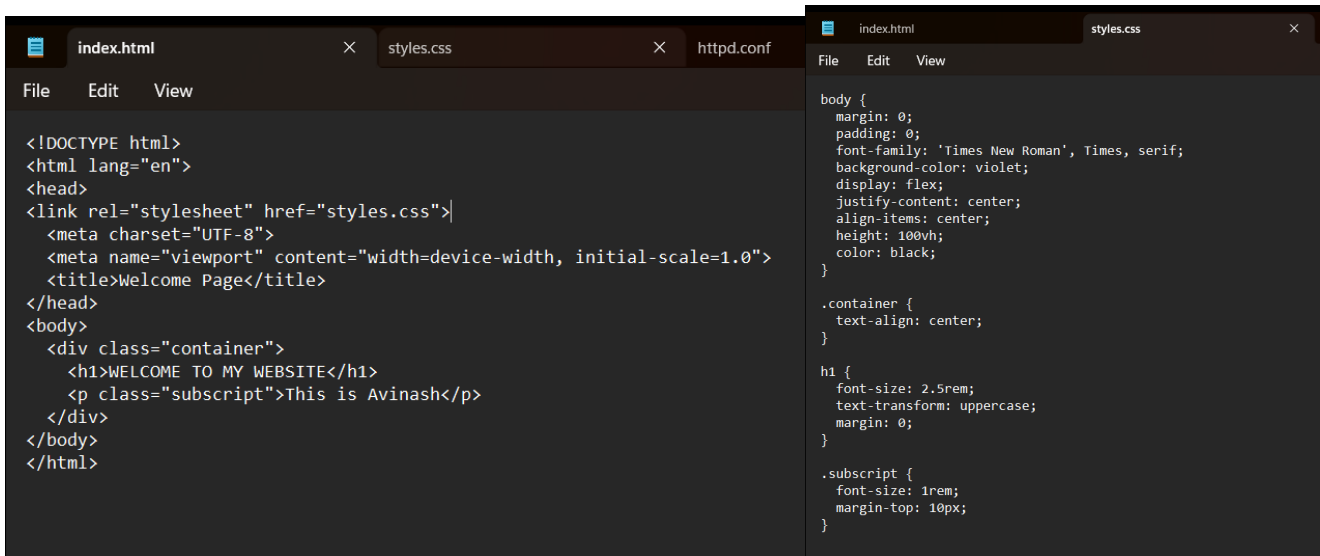
Step10:

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'.



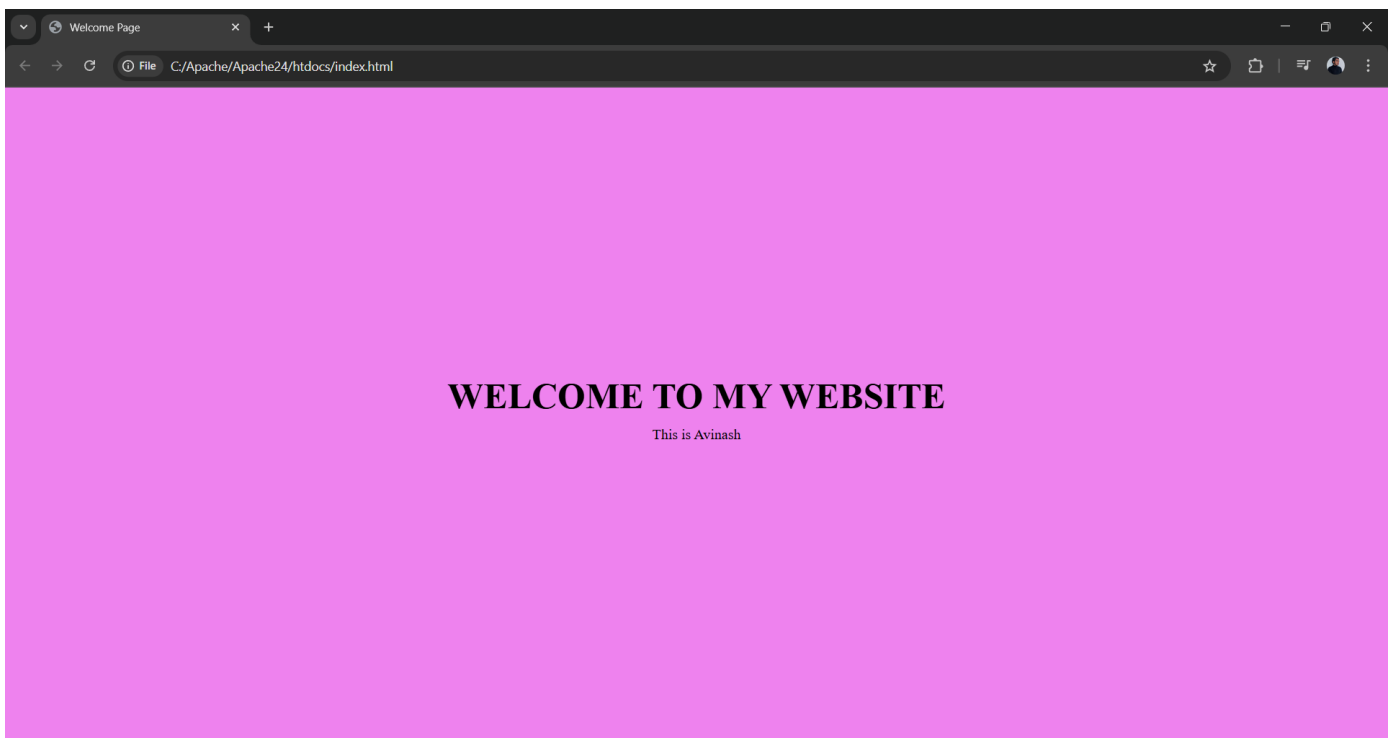
Step 11 :

Create a simple website having the context “Welcome to my Website” with your name using HTML and CSS.



Step 12 :

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



Expected Outcome:

By completing this POC, you will:

1. Successfully configure and run an Apache server locally, ensuring it serves content properly.

2. Host a static HTML website that displays your name, demonstrating the ability to serve simple content.
3. Gain an understanding of the basics of web server configuration, including how to manage document roots, set permissions, and configure the server to serve static files.