 

**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: MIRDULA C Department : ADS



**Introduction and Overview**

In this proof of concept (POC), we will explore how to locally host a static website using the Apache HTTP Server. This process includes setting up a local web server, configuring it properly, and serving a simple HTML page. By following these steps, you'll gain practical experience in configuring and running a local Apache server—an essential skill for web hosting and server management.

**Objective**

The objective of this project is to:

1. Set up a local web server using Apache.
2. Configure the server to host static files efficiently.
3. Create and host a simple HTML page that displays your name.

.

**Importance of Local Hosting**

Local hosting is a crucial skill for developers, enabling them to test and experiment with web applications in a controlled environment. It provides several key benefits, including:

* **Hands-On Learning:** Gain practical experience in setting up and configuring a web server.
* **Safe Testing Ground:** Test and debug websites securely before deploying them to a live server.
* **Offline Development:** Work on web projects without needing an active internet connection.

**Step-by-Step Overview**

**Steps to Set Up a Local Apache Server on Windows**

**Step 1: Download Apache**

* Open Google and search for "Apache Lounge".
* Click on the first link to access the official website.

**Step 2: Navigate to the Downloads Section**

* On the Apache Lounge website, click on the "Downloads" option located in the left-hand menu.

**Step 3: Download and Extract Apache**

* Click on the link "Apache 2.4.62-240904 Win64" (or the latest Windows version).
* Download the file and extract all its contents to a preferred location on your system.

**Step 4: Set Up the Apache Path in Command Prompt**

* Open Command Prompt as an Administrator:
* Press Windows + R, type cmd, then right-click and select 'Run as Administrator'.
* Navigate to the Apache bin directory using:

sh

CopyEdit

cd C:\path\to\apache\bin

**Step 5: Install Apache**

* Run the following command to install Apache:

sh

CopyEdit

httpd.exe -k install

**Step 6: Configure Apache**

* Navigate to the extracted Apache folder.
* Open the conf folder and locate the httpd.conf file.
* Right-click the file and select 'Edit with Notepad' (or any preferred text editor).

**Step 7: Modify the Configuration File**

* Update the SRVROOT directive with your Apache installation path.
* Ensure the configuration defines the server root directory, listening port, modules, document root, logging paths, and access permissions correctly.

**Step 8: Validate the Configuration**

* Open Command Prompt and run the following command to test the configuration file:

sh

CopyEdit

httpd.exe -t

* If the configuration is correct, you should see 'Syntax OK'.

**Step 9: Start the Apache Server**

* Run the following command to start the Apache server:

sh

CopyEdit

httpd.exe -k start

**Step 10: Edit the Default Web Page**

* Navigate to the htdocs folder inside the Apache directory.
* Locate the index.html file, right-click, and select 'Edit with Notepad'.

**Step 11: Create a Simple Web Page**

* Modify the index.html file to display your name.
* Optionally, add CSS for styling.

**Step 12: Access Your Hosted Website**

* Open Google Chrome (or any web browser).
* In the address bar, type:

bash

CopyEdit

localhost/index.html

* If configured correctly, your website should load successfully.

**Expected Outcome**

By completing this proof of concept (POC), you will:

1. Successfully set up and run an Apache server locally.
2. Host a static HTML website that displays your name.
3. Gain a fundamental understanding of web server configuration and file hosting.