

## **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 : B.Harshana Perianayaki

Department : IT



## ***INTRODUCTION***

Hosting a static website locally is an essential first step for web development. It allows you to test and view your HTML, CSS,

and JavaScript code on your own machine before deploying it to the web. By setting up a local server with tools like **Apache** or **Nginx**, you can simulate how your website would behave in a real-world scenario.

## **OVERVIEW**

To host a static website locally, we follow these main steps:

**1. Install a Web Server:** A web server (like Apache or Nginx) is software that serves web pages to browsers. It reads and responds to requests for files like index.html, delivering them to users.

**a. Apache:** Commonly used and included in software bundles like XAMPP.

**b. Nginx:** Lightweight and highly performant, often used for modern web applications.

**2. Prepare the HTML Page:** Create a simple static web page using HTML. The content can include basic information, such as your name, displayed on the page.

**3. Configure the Web Server:** Place the HTML file in the server's root directory. The web server will fetch this file when a browser requests it.

**4. Access the Website Locally:** Once the server is running, the website can be accessed through <http://localhost> in your web browser.

## ***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***

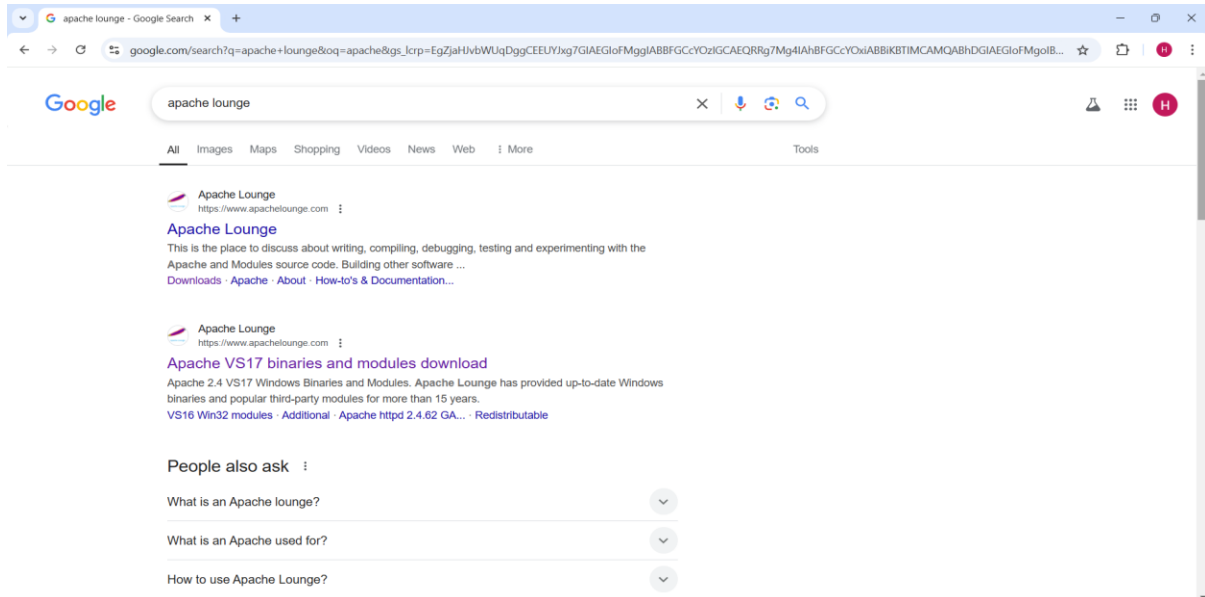
1. Development and Testing
2. No Internet Dependency
3. Cost-Effective Solution
4. Learning and Experimentation
5. Privacy and Security
6. Mirrors Real-World Hosting
7. Collaborative Development

## ***STEP-BY-STEP OVERVIEW***

Step1:

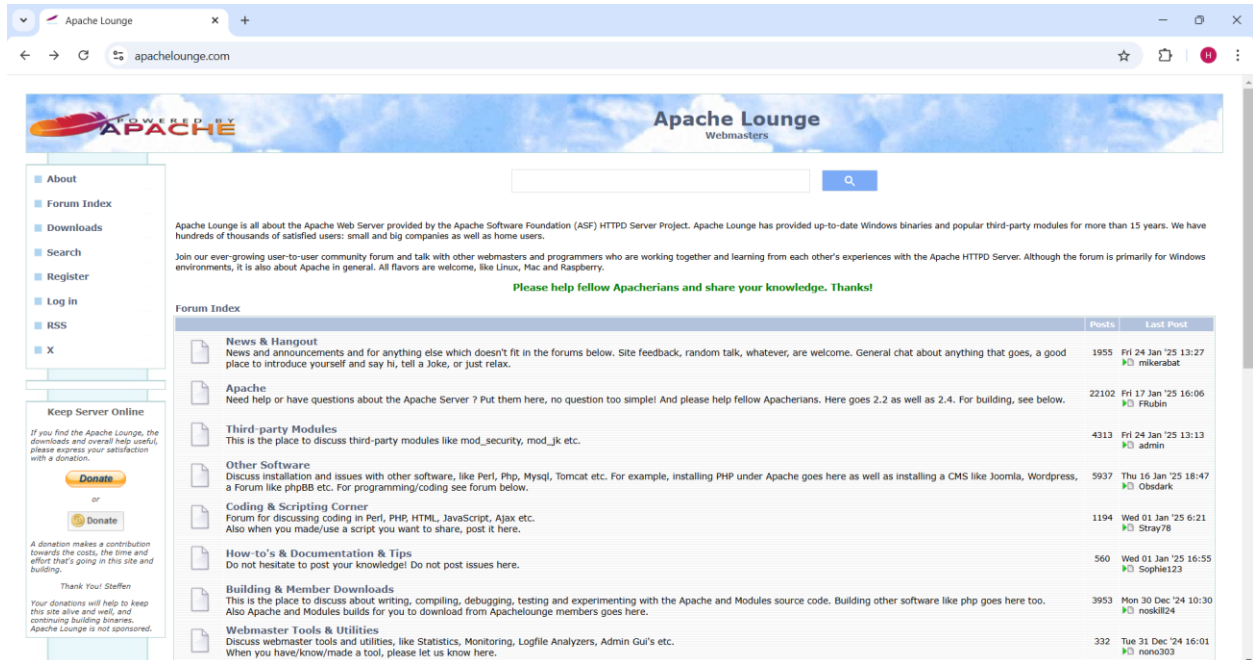
Search for "Apache Lounge" on Google

click the first link to access the official website.



Step 2 :

Click on the "Downloads" option.



## Step 3 :

Click on the link "**Apache 2.4.62-240904 Win64**" and download the file.

The screenshot shows the Apache Lounge website. The header includes the Apache logo and the text "Apache Lounge Webmasters". The main heading is "Apache 2.4 VS17 Windows Binaries and Modules". The page contains several paragraphs of text explaining the binaries, their compatibility, and how to install them. There are also links to download the binaries and modules. The sidebar on the left lists various updates and their dates.

**Apache 2.4 VS17 Windows Binaries and Modules**

Apache Lounge has provided up-to-date Windows binaries and popular third-party modules for more than 15 years. We have hundreds of thousands of satisfied users: small and big companies as well as home users. Always build with up to date dependencies and latest compilers, and tested thorough. The binaries are referenced by the ASF, Microsoft, PHP etc. and more and more software is packaged with our binaries and modules.

The binaries, are build with the sources from ASF at [httpd.apache.org](http://httpd.apache.org), contains the latest patches and latest dependencies like zlib, openssl etc. which makes the downloads here mostly more actual then downloads from other places. The binaries **do not run** on XP and 2003. Runs on: 7 SP1, Vista SP2, 8/8.1, 10, 11 Server 2008 SP2 / R2 SP1, Server 2012 / R2, Server 2016/2019/2022.

Build with the latest Windows Visual Studio C++ 2022 aka VS17. Has improvements, fixes and optimizations over VS16 in areas like Performance, MemoryManagement, New standard conformance features, Code generation and Stability. For example code quality tuning and improvements done across different code generation areas for "speed". And makes more use of latest processors and supported Windows editions (win7 and up) internal features.

**VS17 is backward compatible**, That means, a VS16/15/14 module can be used inside the VS17 binary.

**Be sure** you installed latest 14.42.34433.0 Visual C++ Redistributable Visual Studio 2015-2022 : [vc\\_redist\\_x64](#) or [vc\\_redist\\_x86](#) see [Redistributable](#)

**Apache 2.4 binaries VS17**

[Info & Changelog](#)

**Apache 2.4.62-240904 Win64**

[httpd-2.4.62-240904-win64-VS17.zip](#) 04 Sep '24 11.432k  
PGP Signature (Public [PGP key](#)), SHA1-SHA512 [Checksums](#)

**Apache 2.4.62-240904 Win32**

[httpd-2.4.62-240904-win32-VS17.zip](#) 04 Sep '24 10.285k  
PGP Signature (Public [PGP key](#)), SHA1-SHA512 [Checksums](#)

To be sure that a download is intact and has not been tampered with, use PGP, see [PGP Signature](#)

**Apache 2.4 modules VS17**

Mail for the PGP signatures and/or SHA checksums to verify the contents of a file.

**Note: VS17 Win32 modules (like mod\_fcgid) use VS16 ones at VS16 Win32 modules**

**mod\_jk**  
Tomcat connector

[mod\\_jk-1.2.50-win64-VS17.zip](#) [Info](#) 13 Aug '24 169K

**17 November 2024**  
New C++ Redistributable

**06 November 2024**  
mod\_security Update

**22 October 2024**  
New C++ Redistributable

**21 October 2024**  
mod\_evasive Update

**18 October 2024**  
Libcurl update, see [here](#)

**04 September 2024**  
httpd 2.4.62 update

**13 August 2024**  
mod\_jk Update

**18 July 2024**  
httpd 2.4.62

**03 July 2024**  
httpd 2.4.61

**01 July 2024**  
httpd 2.4.60

**05 June 2024**  
httpd 2.4.59 Update

**22 May 2024**  
New C++ Redistributable

**04 April 2024**  
httpd 2.4.50

## Step 4 :

Open Command Prompt as Administrator and use the command `cd C:\path\to\apache\bin` to set the path to the Apache bin folder

```
C:\Users\HP\Downloads\httpd-2.4.62-240904-VS17\Apache24>cd C:\Users\HP\Downloads\httpd-2.4.62-240904-win64-VS17\Apache24\bin
```

## Step 5 :

Then Run the installation command :

## httpd.exe -k install

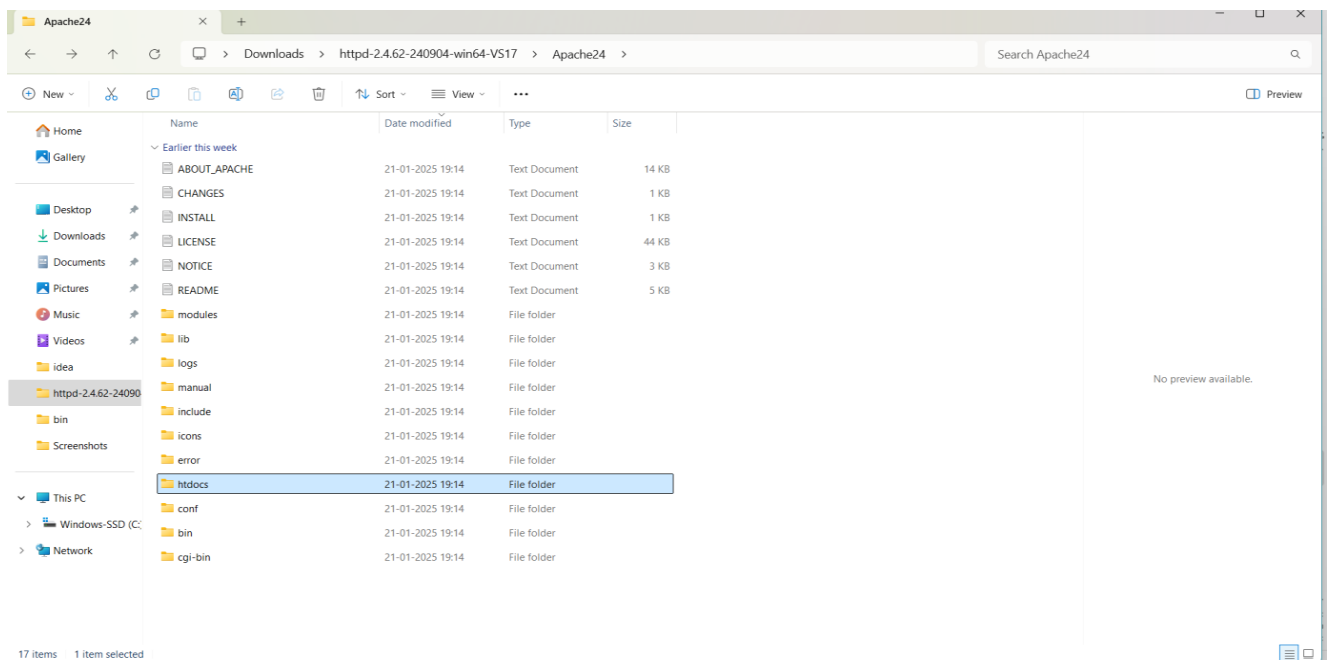
```
C:\Users\HP\Downloads\https-2.4.62-240904-win64-VS17\Apache24\bin>httpd.exe -k install
```

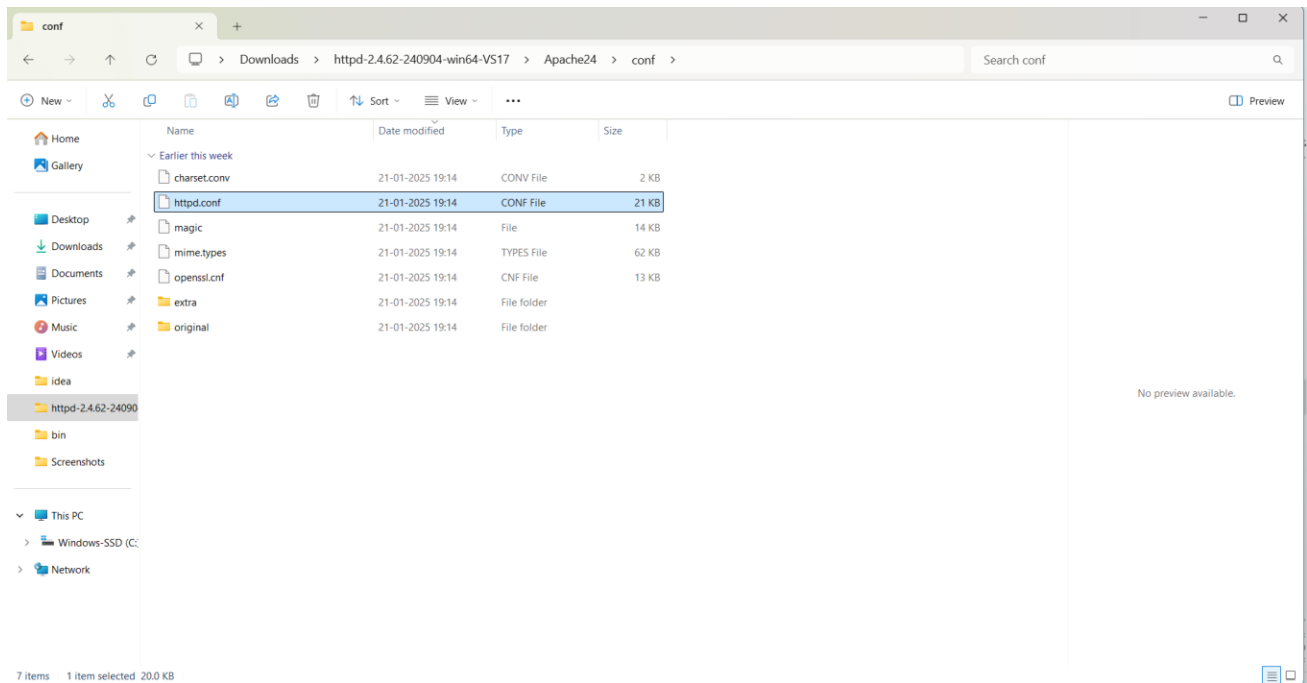
Installing the 'Apache2.4' service

The 'Apache2.4' service is successfully installed.

## 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' .





## Step 7 :

Inside the **httpd.conf** file, replace the content with the provided configuration.

## 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:\Users\HP\Downloads\httpd-2.4.62-240904-win64-VS17\Apache24\bin>httpd.exe -t  
  
Syntax ok
```

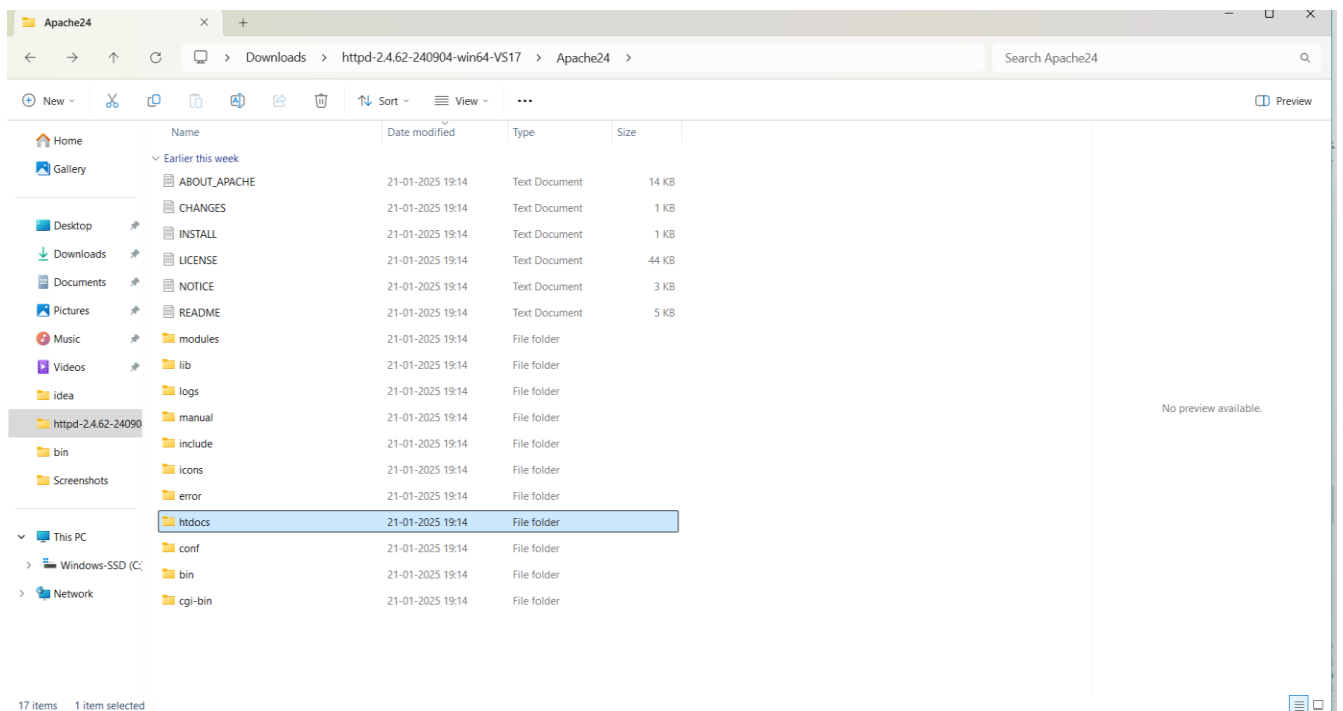
## Step 9 :

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

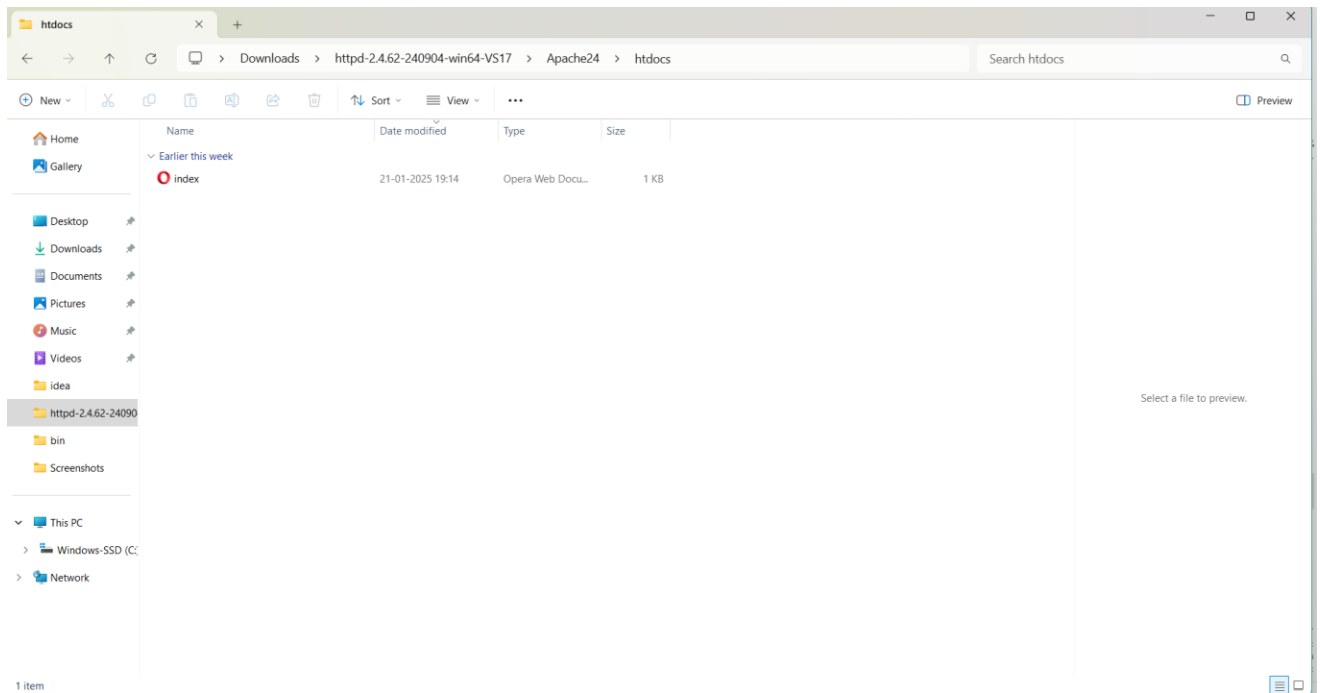
```
C:\Users\HP\Downloads\httpd-2.4.62-240904-win64-VS17\Apache24\bin>httpd.exe -k start
```

## 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 model to display your name in HTML

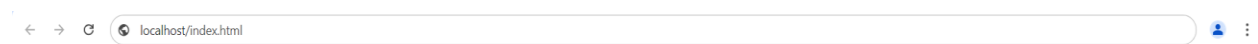
```

</DOCTYPE html>
<html>
<head>
  <title>Display My Name</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      text-align: center;
      margin-top: 100px;
    }
    .container {
      padding: 20px;
      display: inline-block;
    }
    .name {
      font-size: 24px;
      font-weight: bold;
      color: #4CAF50;
    }
  </style>
</head>
<body>
  <div class="container">
    <p class="name">Hello, Harshana Perianayaki B !
    </p>
  </div>
</body>
</html>
|

```

## 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.



Hello, Harshana Perianayaki B !

# OUTCOMES

By completing the task of hosting a static website locally using Apache or Nginx and creating a simple HTML page with your name, you achieve the following outcomes:

## 1. Practical Knowledge of Web Hosting

- **Understanding Server Basics:** You gain hands-on experience with how web servers like Apache and Nginx work, including how they serve static content.
- **File Structure Familiarity:** Learn where to place files (e.g., server root directory) and how servers locate and serve them.

## 2. Development Environment Setup

- **Local Hosting Environment:** Successfully set up a personal development environment to test and preview websites without requiring internet access or external hosting.
- **Efficient Testing:** Test changes to your code in real time without the need for deployment to a remote server.

## 3. Basic HTML Page Hosted

- **Hosted Static Website:** You create and serve a simple `index.html` file that displays your name, demonstrating your ability to develop and host basic web pages.
- **Browser Accessibility:** Access your hosted website at <http://localhost>, simulating a live hosting experience.

## 4. Troubleshooting and Problem-Solving

- **Server Configuration:** Learn to troubleshoot common issues like incorrect file paths, permissions errors, or port conflicts.
- **Error Handling:** Gain insight into identifying and resolving errors during server setup or while hosting files.

## 5. Skill Building for Real-World Hosting

- **Deployment Preparation:** The experience mirrors the process of deploying a website to a live server, making you more comfortable with deployment in the future.
- **Foundation for Dynamic Websites:** Understand the role of web servers as a stepping stone for hosting more complex websites with server-side functionality.

## 6. Privacy and Security Awareness

- **Safe Workspace:** Hosting locally ensures privacy, as files remain on your machine and are not exposed to the internet.

## 7. Confidence Boost

- **Hands-On Achievement:** Successfully hosting your first website locally builds confidence and motivates you to explore more advanced web technologies.

