

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: Kevin Christopher Ruban

Department : AI-ML

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.

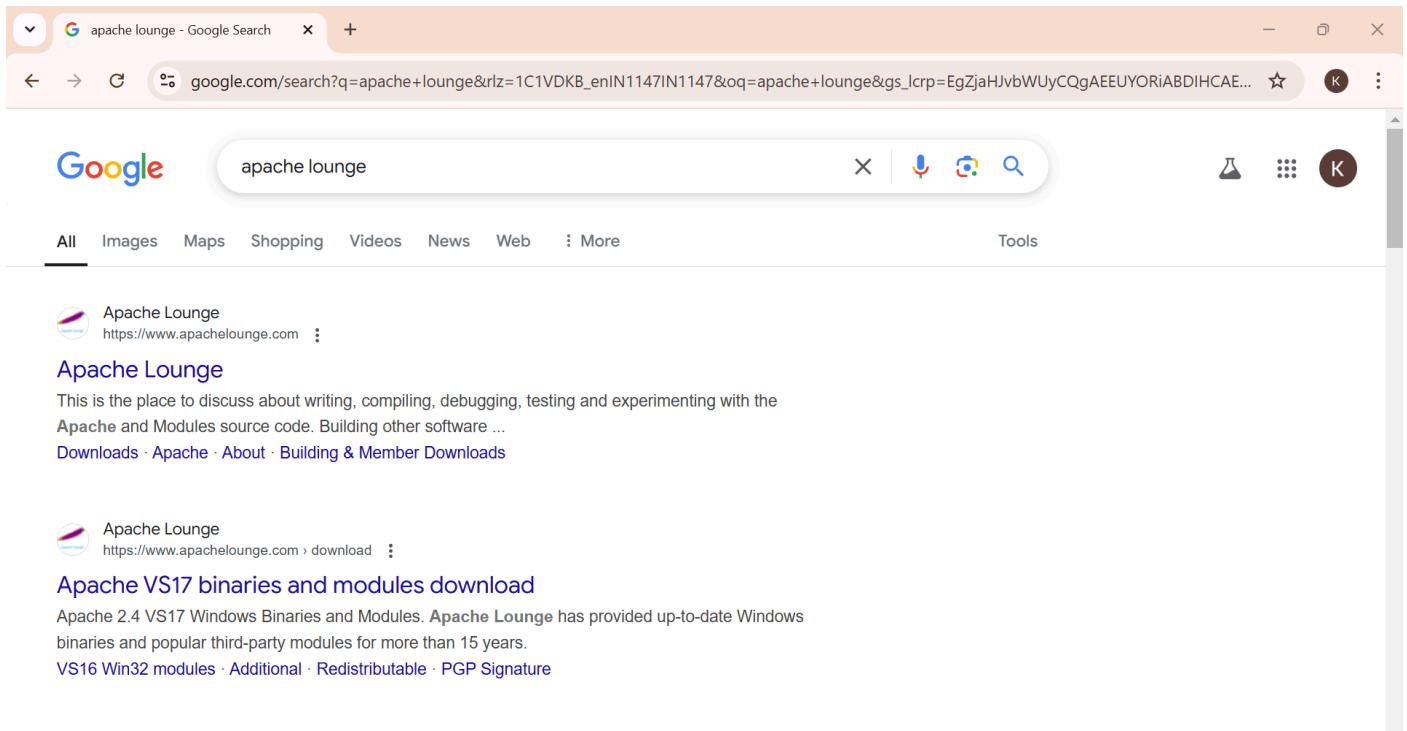
Testing Ground: Safely test and debug websites before deploying them to a live server.

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.

The screenshot shows the Apache Lounge website's download page for Apache 2.4 VS17 binaries and modules. The sidebar on the left lists recent updates, including a 'NEW' update for January 23, 2025. The main content area provides information about the binaries, their compatibility with VS17, and links to download the 64-bit and 32-bit versions. A table lists the download links, PGP signatures, and file sizes for both architectures.

Apache 2.4 binaries VS17		
Info & Changelog		
Apache 2.4.63-250122 Win64		
httpd-2.4.63-250122-win64-VS17.zip	22 Jan '25	11.787k
PGP Signature (Public PGP_key) , SHA1-SHA512 Checksums		
Apache 2.4.63-250122 Win32		
httpd-2.4.63-250122-win32-vs17.zip	22 Jan '25	10.589k
PGP Signature (Public PGP_key) , SHA1-SHA512 Checksums		

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.

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

Step 5 :

Then Run the installation command :

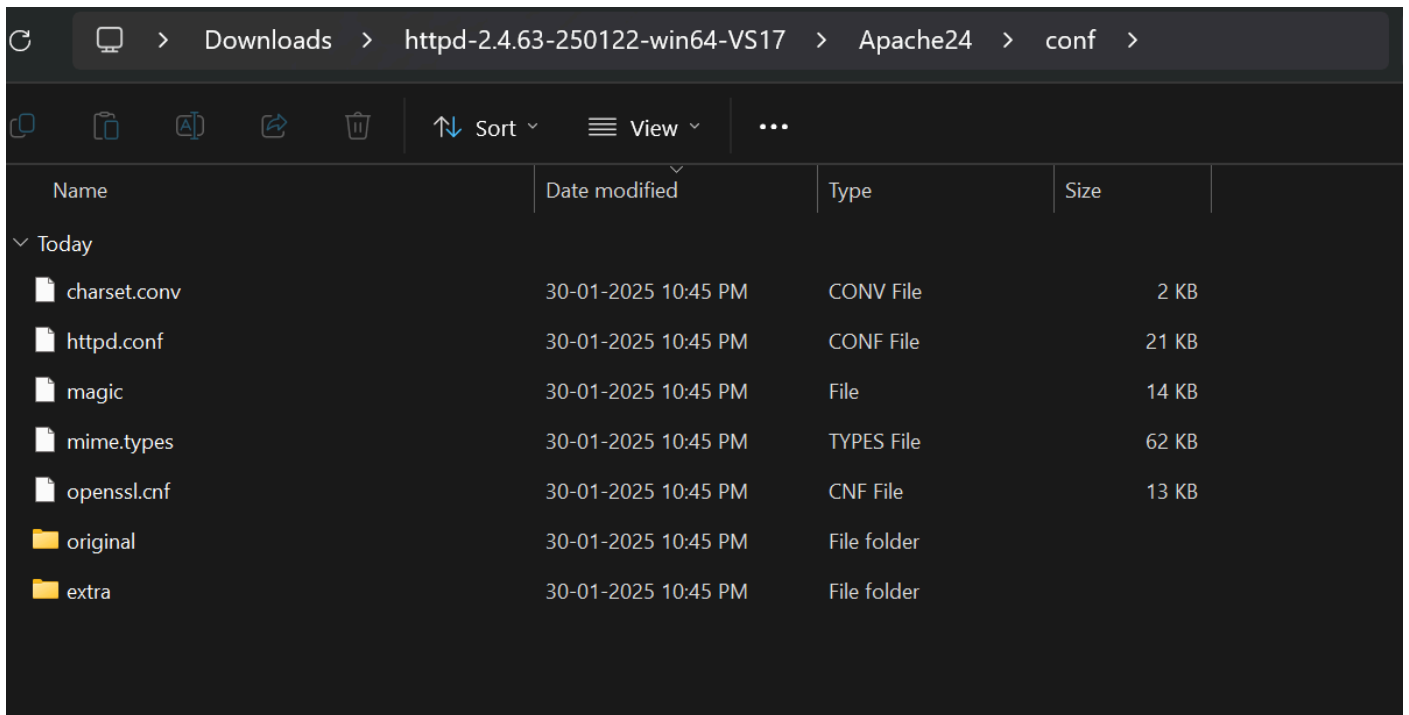
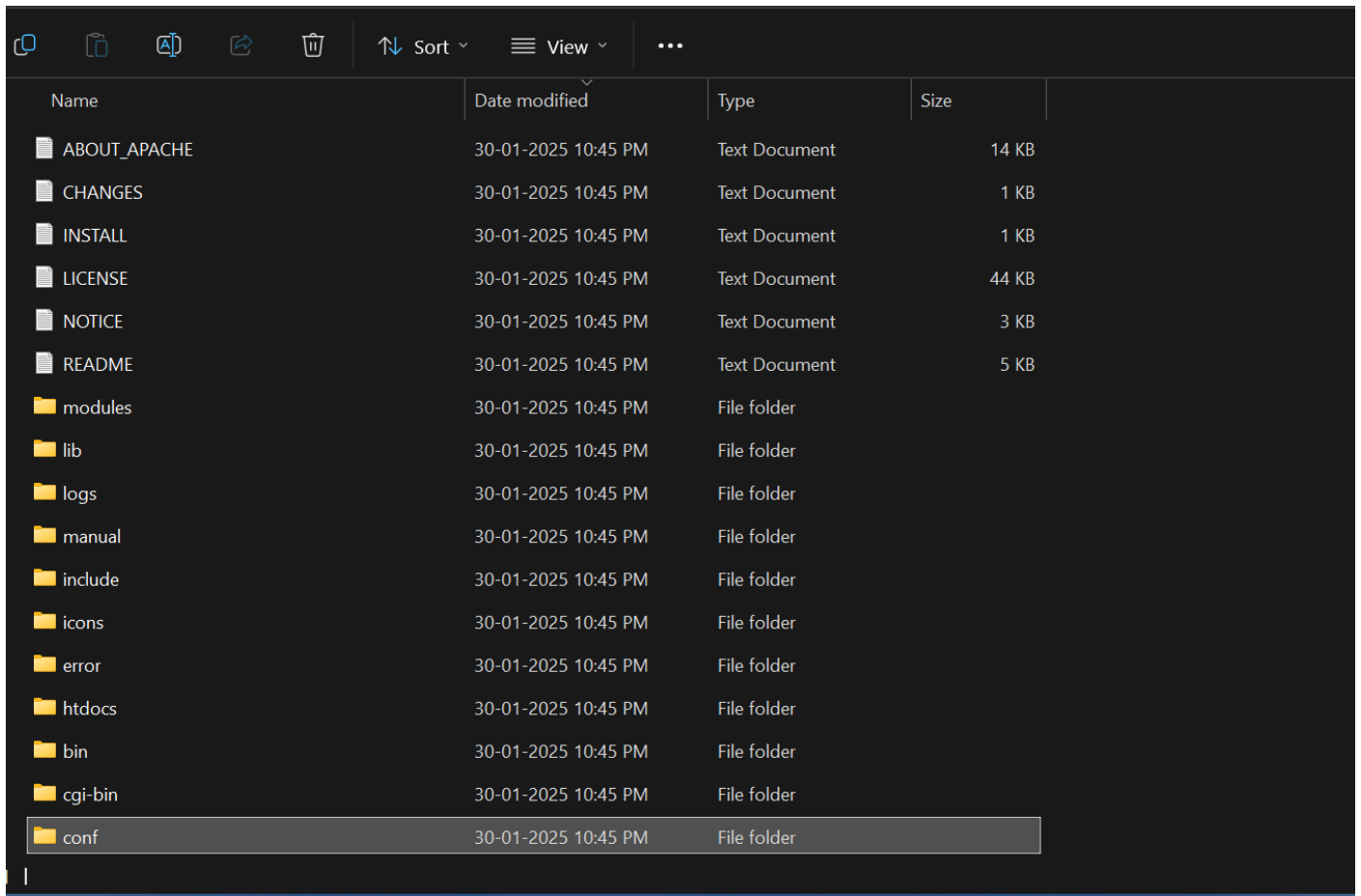
httpd.exe -k install

```
C:\Users\kevin\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.
```

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 :

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.

```
httpd.conf
File Edit View
Define SRVROOT "C:\Users\kevin\Downloads\httpd-2.4.63-250122-win64-VS17\Apache24"
ServerRoot "${SRVROOT}"
Listen 80
ServerName localhost:80
LoadModule access_compat_module modules/mod_access_compat.so
LoadModule actions_module modules/mod_actions.so
LoadModule alias_module modules/mod_alias.so
LoadModule allowmethods_module modules/mod_allowmethods.so
LoadModule auth_basic_module modules/mod_auth_basic.so
LoadModule authn_core_module modules/mod_authn_core.so
LoadModule authz_core_module modules/mod_authz_core.so
LoadModule dir_module modules/mod_dir.so
LoadModule log_config_module modules/mod_log_config.so
LoadModule mime_module modules/mod_mime.so
LoadModule rewrite_module modules/mod_rewrite.so
LoadModule setenvif_module modules/mod_setenvif.so
DocumentRoot "${SRVROOT}/htdocs"
<Directory "${SRVROOT}/htdocs">
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
ErrorLog "${SRVROOT}/logs/error.log"
CustomLog "${SRVROOT}/logs/access.log" common
```

```
<IfModule mime_module>
    TypesConfig conf/mime.types
    AddType application/x-compress .Z
    AddType application/x-gzip .gz .tgz
</IfModule>
<IfModule dir_module>
    DirectoryIndex index.html
</IfModule>
```

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\kevin\Downloads\httpd-2.4.63-250122-win64-VS17\Apache24\bin>httpd.exe -t
Syntax OK
```

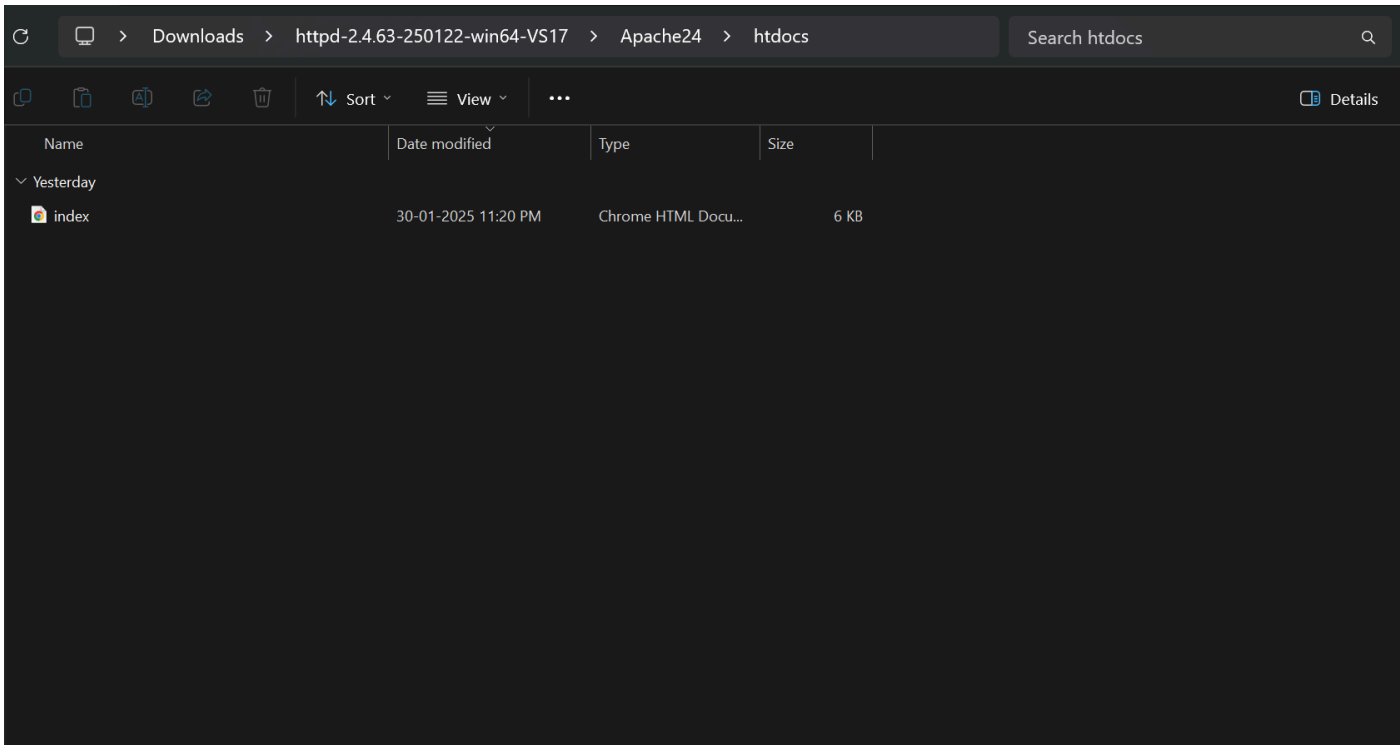
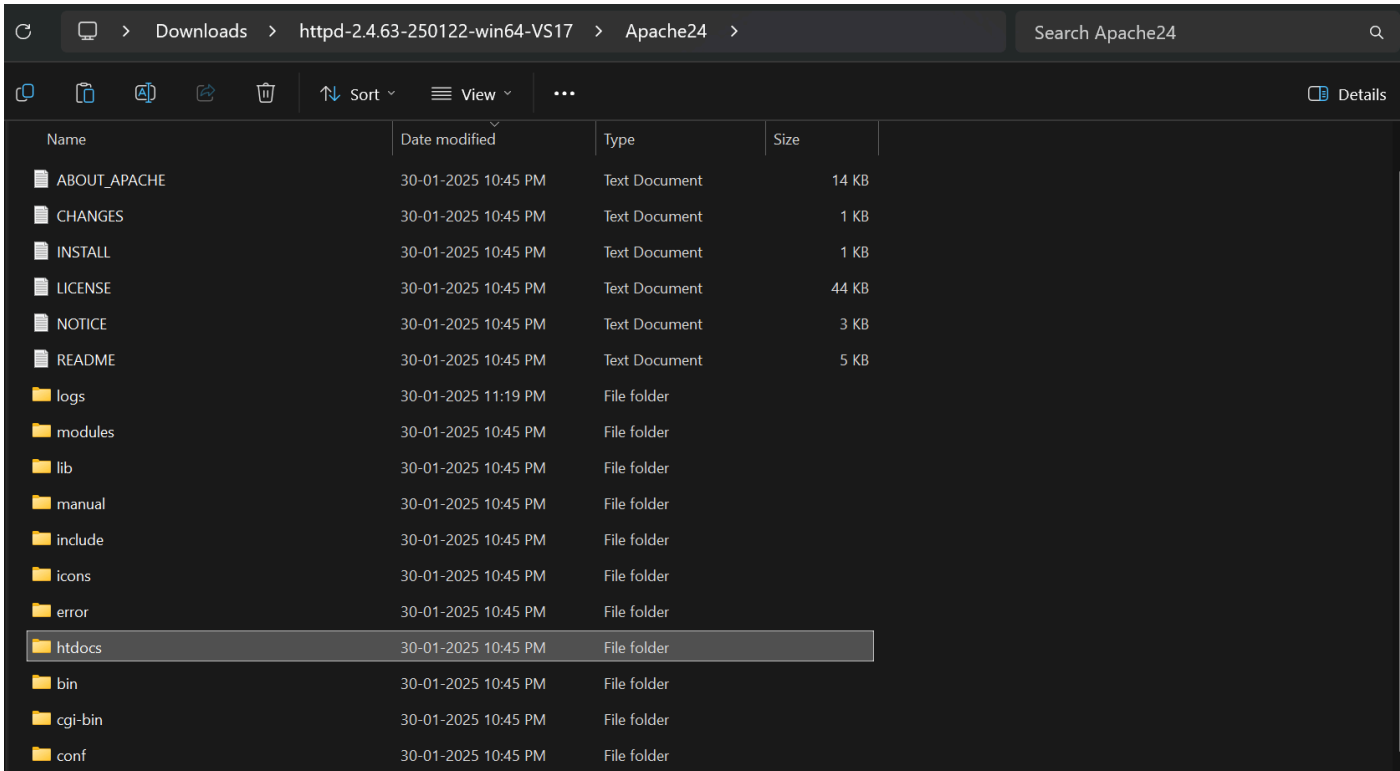
Step 9 :

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

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

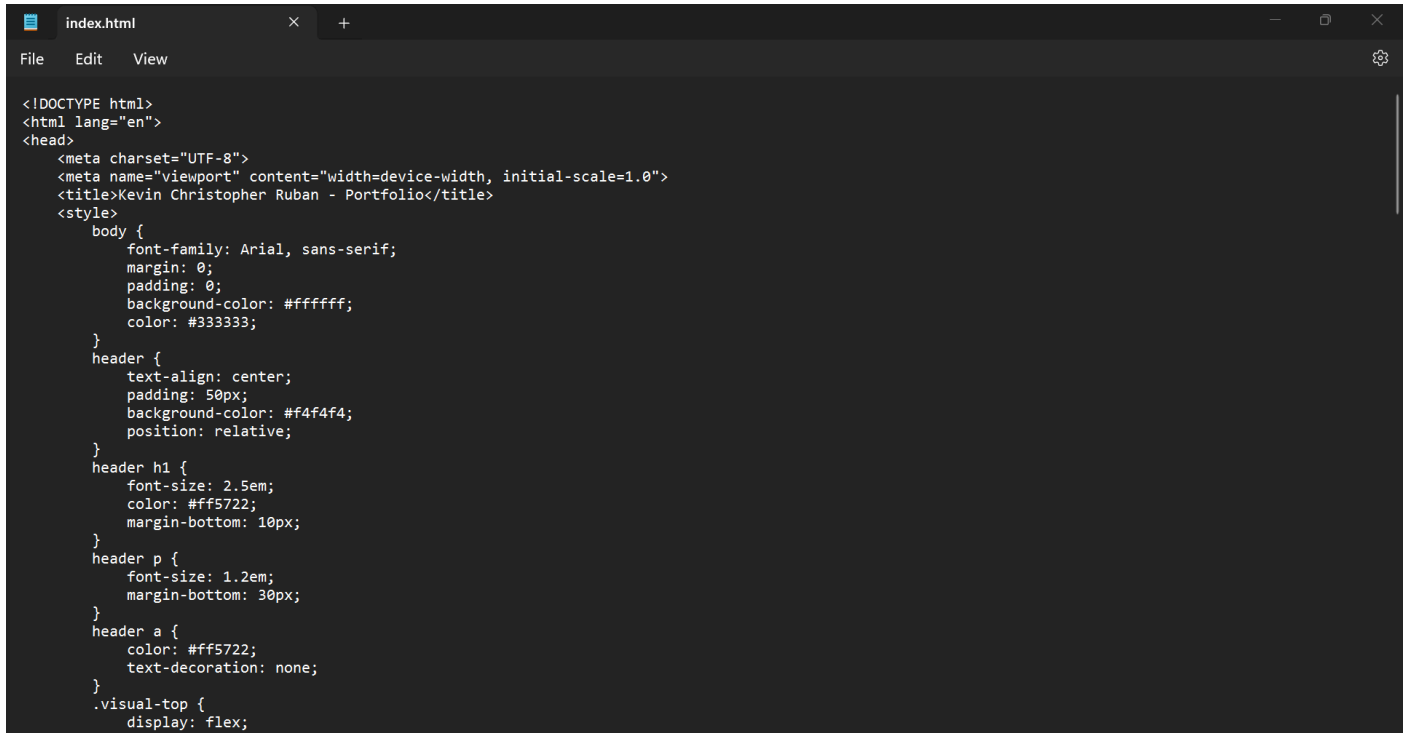
Step10:

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 :

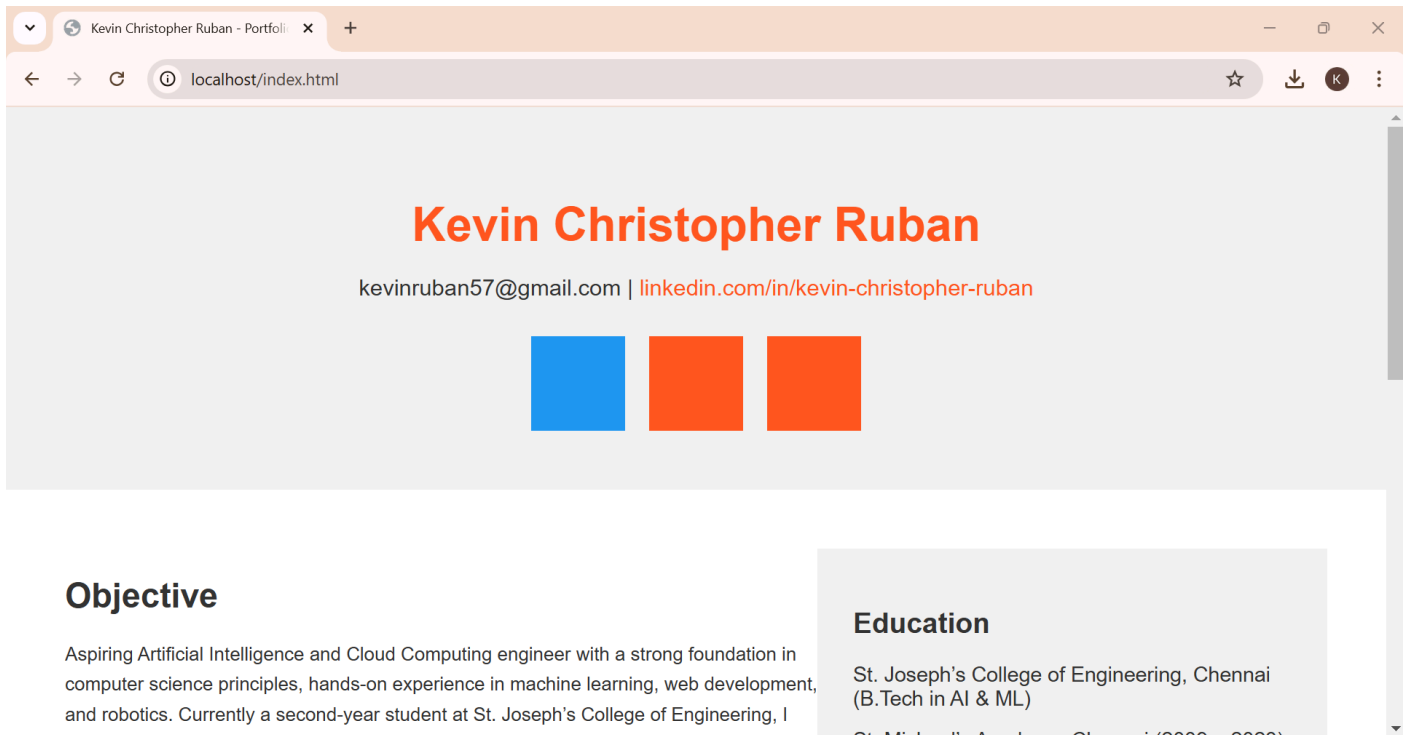
Create a simple model to display your name in HTML (you may optionally add CSS for styling).

A screenshot of a code editor window titled 'index.html'. The editor has a dark theme and shows the following code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Kevin Christopher Ruban - Portfolio</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      background-color: #ffffff;
      color: #333333;
    }
    header {
      text-align: center;
      padding: 50px;
      background-color: #f4f4f4;
      position: relative;
    }
    header h1 {
      font-size: 2.5em;
      color: #ff5722;
      margin-bottom: 10px;
    }
    header p {
      font-size: 1.2em;
      margin-bottom: 30px;
    }
    header a {
      color: #ff5722;
      text-decoration: none;
    }
    .visual-top {
      display: flex;
      justify-content: space-between;
      align-items: center;
    }
  </style>
</head>
<body>
  <div class="visual-top">
    <div class="header">
      <h1>Kevin Christopher Ruban</h1>
      <p>Full Stack Developer</p>
      <a href="#">Portfolio</a>
    </div>
  </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.



Expected 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.