အခြေခံ ဝက်နည်းပညာ (ကြားဖြတ်သင်တန်း)

https://github.com/lawintun0110/project-oriented-curriculum/releases/tag/FWT-CC

Fundamental Web Technology

(Cross Course)

"This course is designed as a handbook specifically for first-year IT engineering students. The handbook is hosted via GitHub platform as the version-control-document , allowing both students and instructors to download it using the link provided on the cover. Created by Mg La Win Tun (VI-IT-15) as part of the HSS activities, this course aims to promote education, encourage community engagement, and support capacity building at TUM."

FOREWORD

THE EVER-EVOLVING FIELD OF WEB TECHNOLOGY PLAYS A VITAL ROLE IN SHAPING THE DIGITAL LANDSCAPE OF TODAY'S WORLD. THIS CROSS-COURSE HAS BEEN METICULOUSLY DESIGNED TO PROVIDE FIRST-YEAR IT ENGINEERING STUDENTS WITH A FOUNDATIONAL UNDERSTANDING OF WEB DEVELOPMENT CONCEPTS, BRIDGING THE GAP BETWEEN THEORETICAL KNOWLEDGE AND PRACTICAL APPLICATION. BY ALIGNING WITH THE OBJECTIVES OF HSS ACTIVITIES, THIS COURSE AIMS TO EMPOWER STUDENTS WITH ESSENTIAL SKILLS, FOSTER COMMUNITY ENGAGEMENT, AND ENHANCE THEIR CAPACITY FOR FUTURE CHALLENGES IN THE FIELD OF IT. I COMMEND THE EFFORTS OF MG LA WIN TUN (VI-IT-15) FOR CREATING SUCH A COMPREHENSIVE LEARNING RESOURCE TO BENEFIT THE ACADEMIC COMMUNITY AT TUM.

MG LA WIN TUN
IT ENGINEERING STUDENT
ACADEMIC FINAL YEAR (2024-2025)
TECHNOLOGICAL UNIVERSITY (MANDALAY)

Mg La Win Tun

Author

[illegible]

PREFACE

A project-oriented curriculum focuses on hands-on, practical learning through projects designed to help students apply theoretical knowledge in real-world scenarios. Courses in this curriculum emphasize critical thinking, problem-solving, and teamwork skills. Web development is a cornerstone of modern it education, and understanding its fundamental aspects is essential for aspiring engineers. This cross-course has been developed specifically for first-year it engineering students as part of the HSS initiative at TUM. The course content is carefully structured to introduce students to the core principles of frontend and backend development, types of websites, and the basic structure of a typical website. It also provides insights into widely used technologies and frameworks, ensuring students are equipped with the knowledge to navigate and build upon this exciting domain. I hope this course will not only serve as a stepping stone in the academic journey of students but also inspire them to explore the endless possibilities of web technology.

Curricular aid link :
https://github.com/lawintun0110/lawintun0110.github.io/releases/tag/Curricular_aid

ACKNOWLEDGEMENTS

This cross-course would not have been possible without the guidance, encouragement, and support of several individuals and organizations. I would like to express my heartfelt gratitude to:

- **My Faculty Mentor : Dr. Cho Me Me Maung (Professor at IT Engineering)** for her valuable teaching methods in HSS and constant motivation throughout the development of this course.
- **My Comrades : Mg Min Thant(VI-IT-19) and Mg Aung Naing Tun(VI-IT-18)** for their practical assistances on students who are following the course's instructions.

Finally, I extend my deepest appreciation to everyone who has supported and believed in my vision for this educational initiative.

Mg La Win Tun (VI-IT-15)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION TO WEB TECHNOLOGY

Introduction

Web technology refers to the techniques and tools used to develop services and products based on web browser platform. This chapter introduces foundational knowledge and the fundamental aspects of web development. Fundamental concept of a website is essentially a collection of files that can be hosted either locally or online. Hosting allows these files to be accessed via a URL on a browser platform. URL stands for "**Uniform Resource Locator"** in the browser address bar.

Things of Web Technology

1. Website
   Type of websites
   - Static website
   - Dynamic website
2. Web application
   Type of web applications
   - SPAs (Single-Page Application)
   - MPAs (Multiple-Page Application)

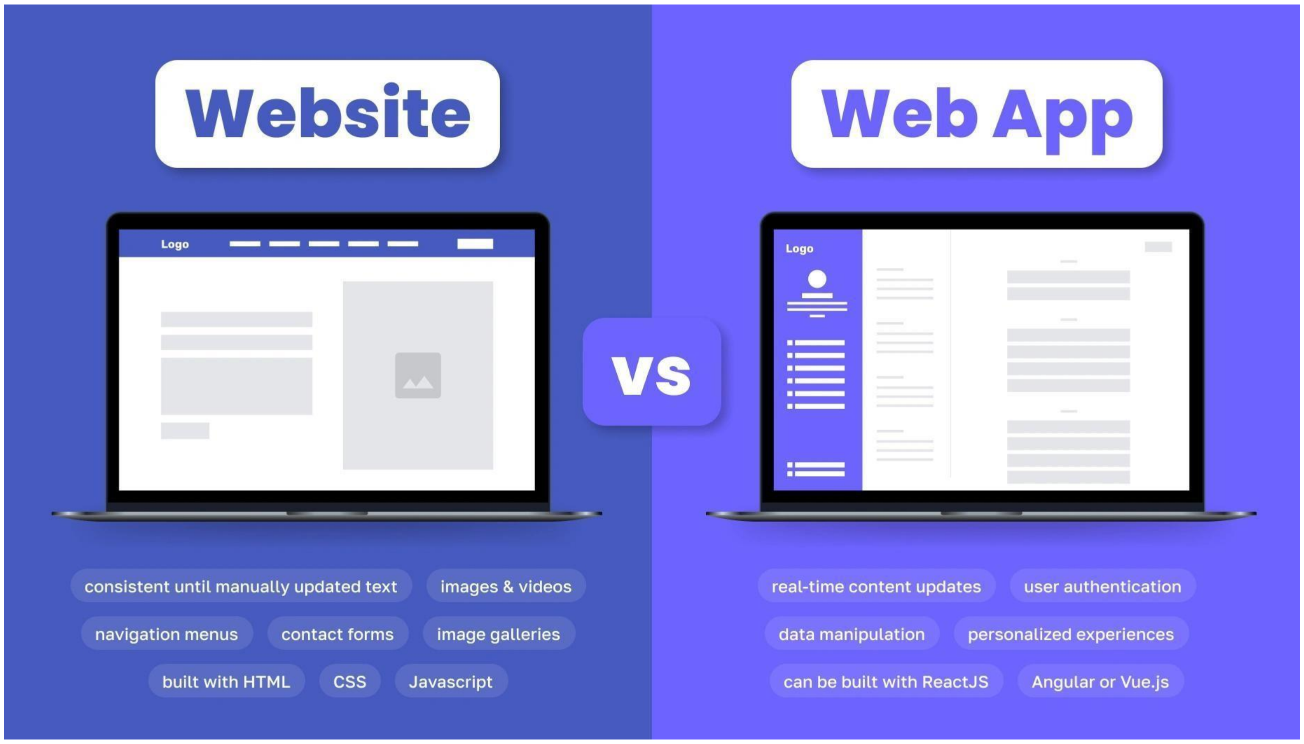


Figure 1.1. Website and Web Application (Web App)

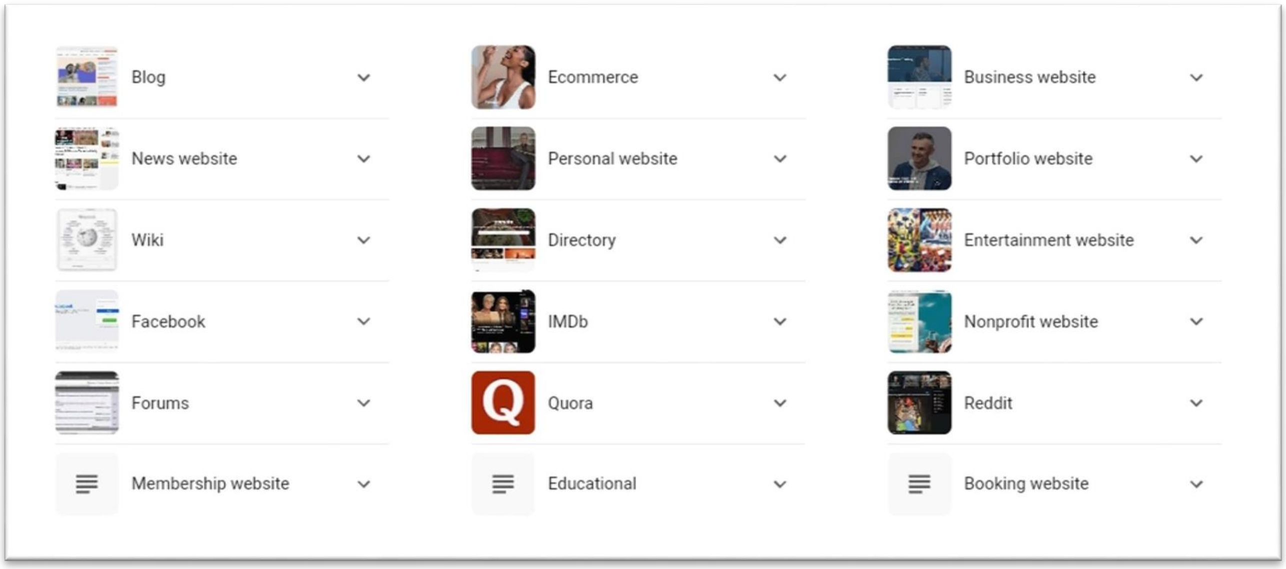
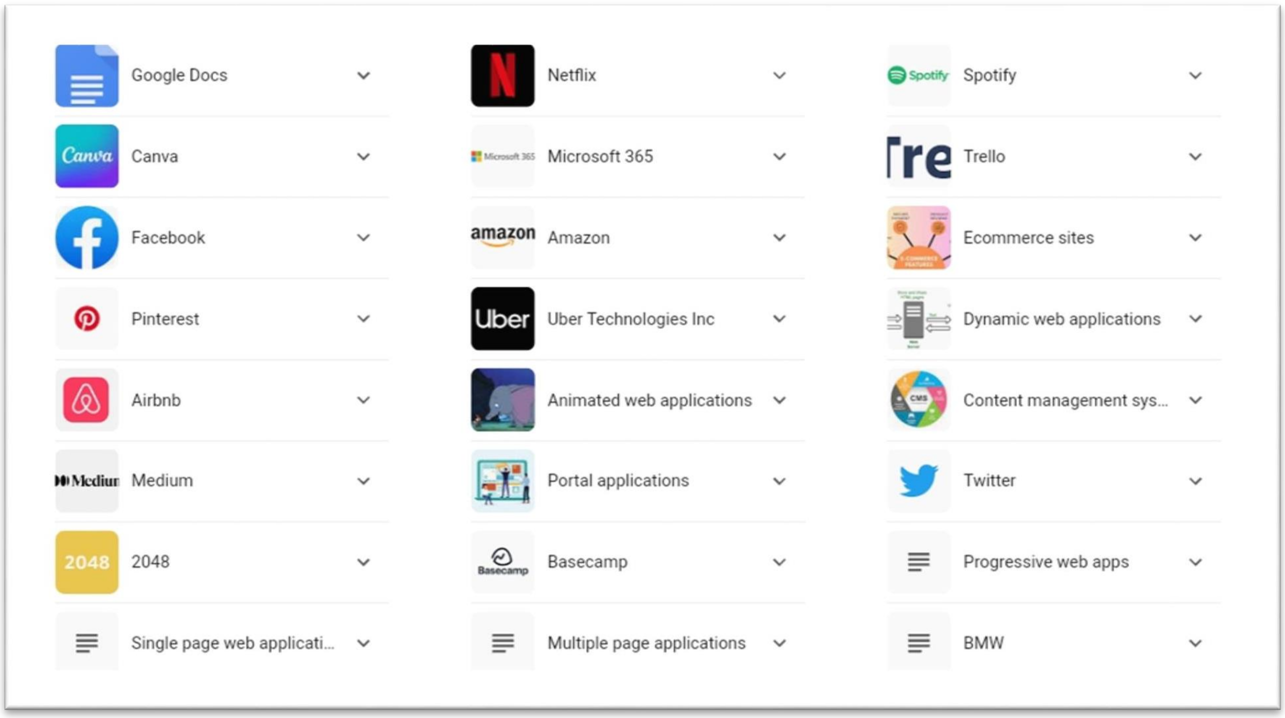


Figure 1.2. Examples of Websites



Google Docs	Netflix	Spotify
Canva	Microsoft 365	Trello
Facebook	Amazon	Ecommerce sites
Pinterest	Uber Technologies Inc	Dynamic web applications
Airbnb	Animated web applications	Content management sys...
Medium	Portal applications	Twitter
2048	Basecamp	Progressive web apps
Single page web applicati...	Multiple page applications	BMW

Figure 1.3. Examples of Web Applications

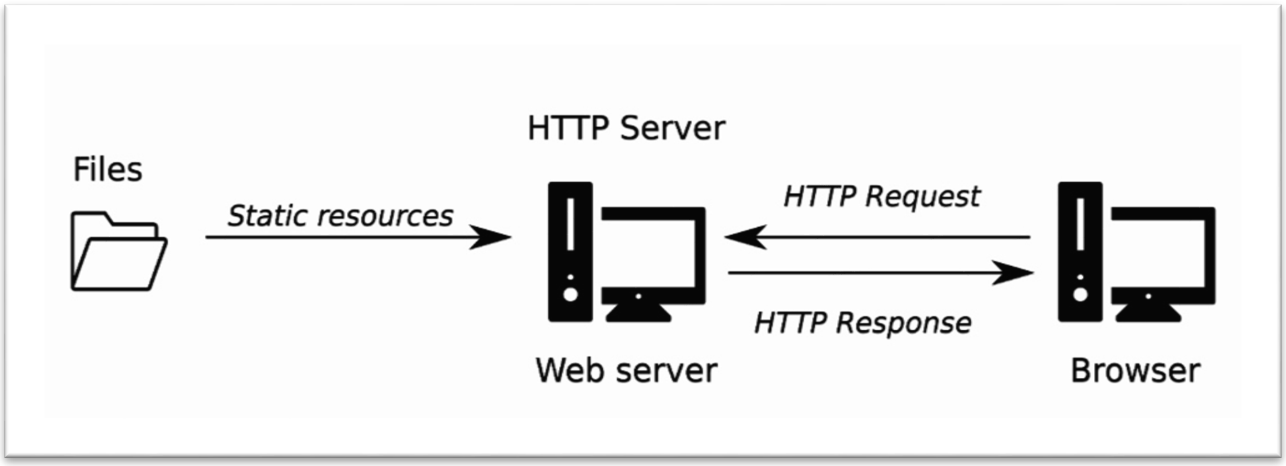


Figure 1.4. Static Website

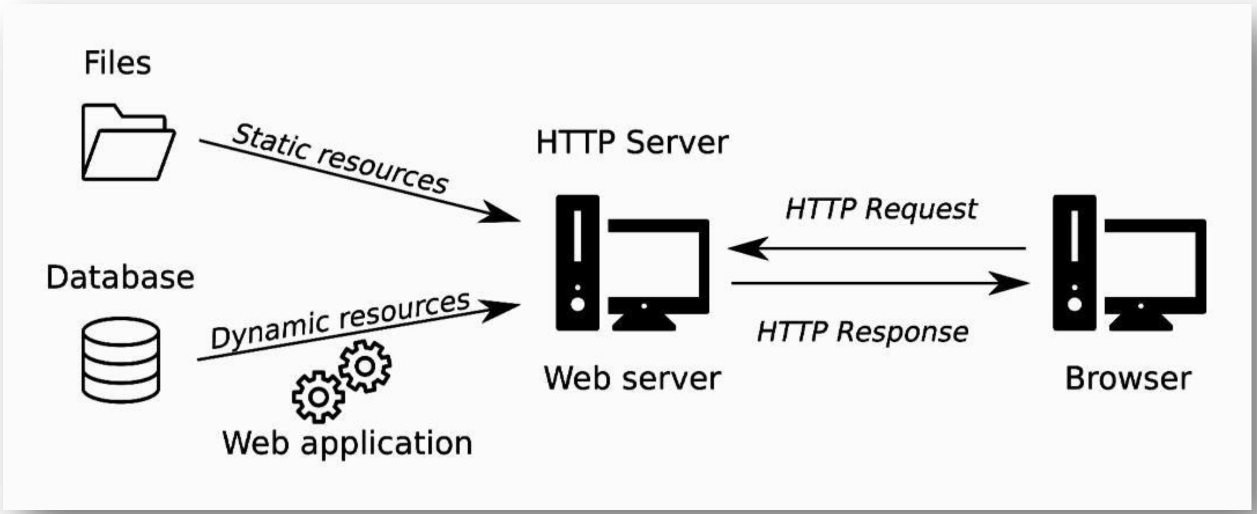


Figure 1.5. Dynamic Website

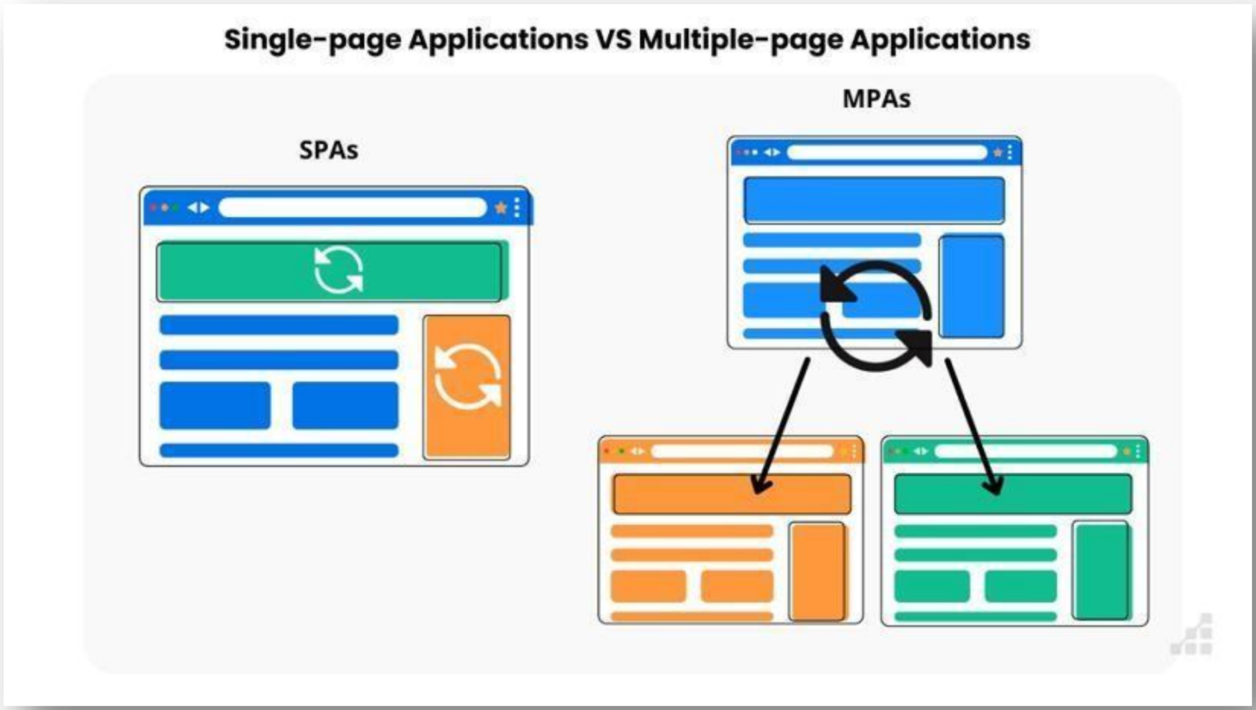


Figure 1.6. SPAs and MPAs

1.1. Web Development Concept

Types of Developments

- Frontend Development
- Backend Development

Frontend development focuses on designing and developing the visual and interactive aspects of a website, also known as the rendering process. It does not involve significant data processing or integration with database management systems. Backend development handles data manipulation, computations, and integration with database management systems to support the website's functionality.

1.1.1. Frontend Development

The core technologies for frontend development are:

- HTML: Stands for HyperText Markup Language and is used for structuring web content. (Note: It is not a programming language.)
- CSS: Stands for Cascading Style Sheets and is used to style and design web elements.
- JavaScript (JS): A programming language that enables interactivity and dynamic responses between users and the website.

1.1.2. Backend Development

Backend development involves server-side technologies to process data and manage the operations beyond the visible interface of a website. Common backend technologies include:

- PHP: Used for server-side scripting.
- SQL: Used for database management and queries.

1.2. Development of Static and Dynamic Websites

Static websites are built solely using frontend technologies without involving backend systems. Dynamic websites, on the other hand, use both frontend and backend technologies to deliver interactive and functional features.

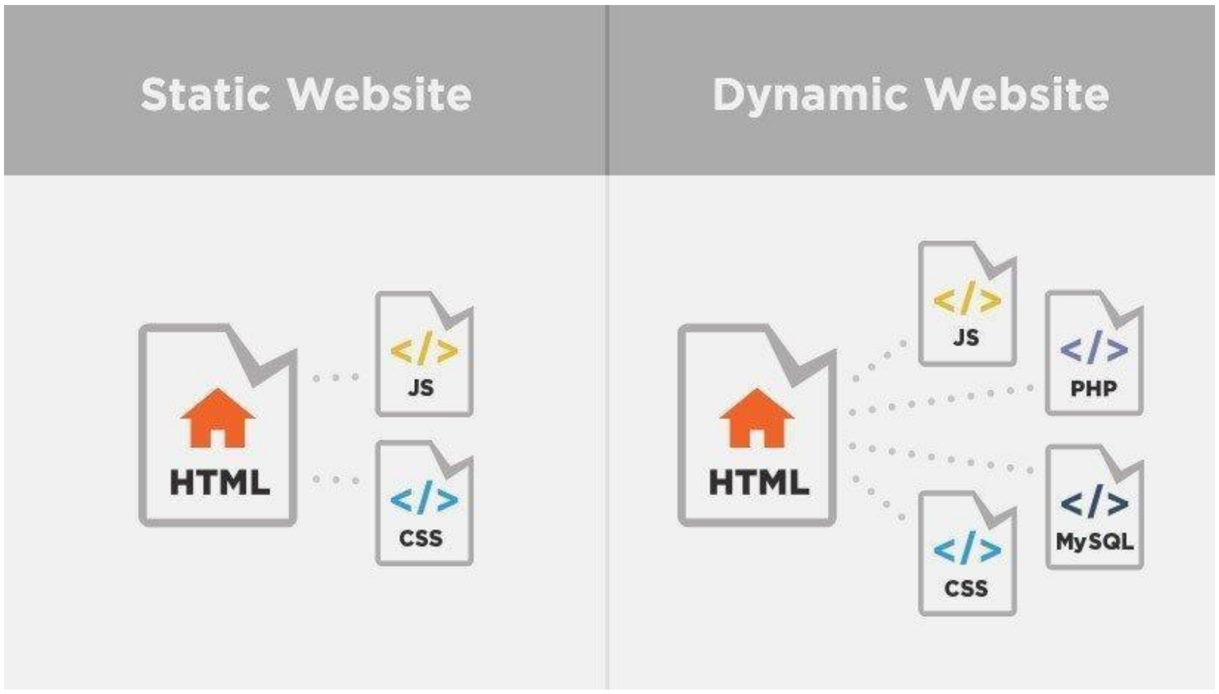


Figure 1.7. Development of Static and Dynamic Websites

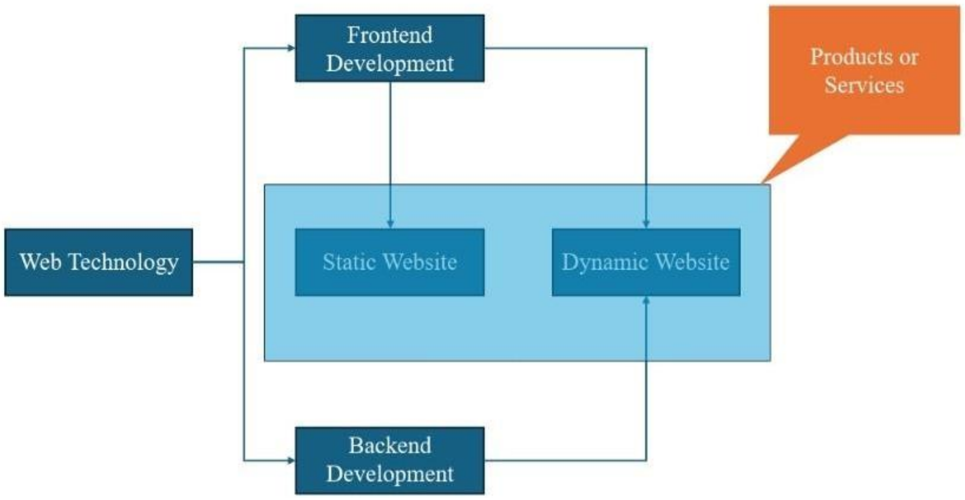


Figure 1.8. Tree Diagram of Development Approach

1.3. Typical Structure of a Webpage

A typical webpage consists of three main parts:

1. Head: Contains the header, logo, name, navigation bar, or highlights of the website.
2. Body: Displays the main content, including text, media, or services provided by the website.
3. Footer: Optional (but often used for additional navigation links, contact details, or other supplementary information. It sometimes serves as a secondary navigation bar, similar to the head.)

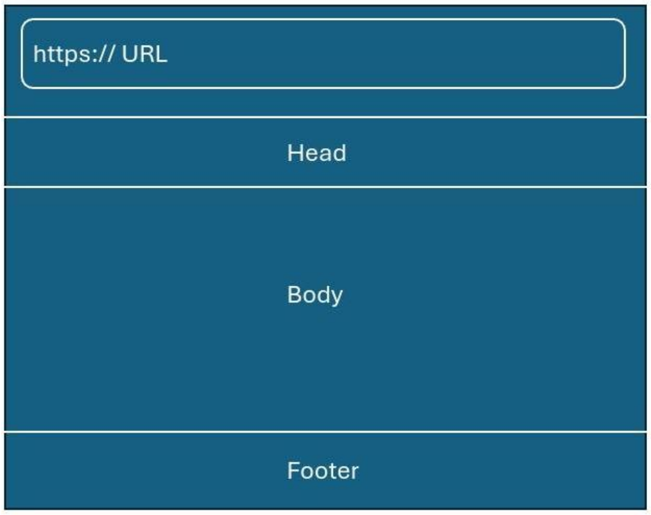


Figure 1.9. Typical Structure of a Webpage

1.4. Requirements for First Step

Hardware	Software
Computer , Laptop or Mobile	Text Editor, Sublime Text Editor (recommend),
	Web Browser , chrome or firefox
	Termux for mobile (Instead IDE) Pycharm Community For PC (IDE)

Table 1.1. Requirements

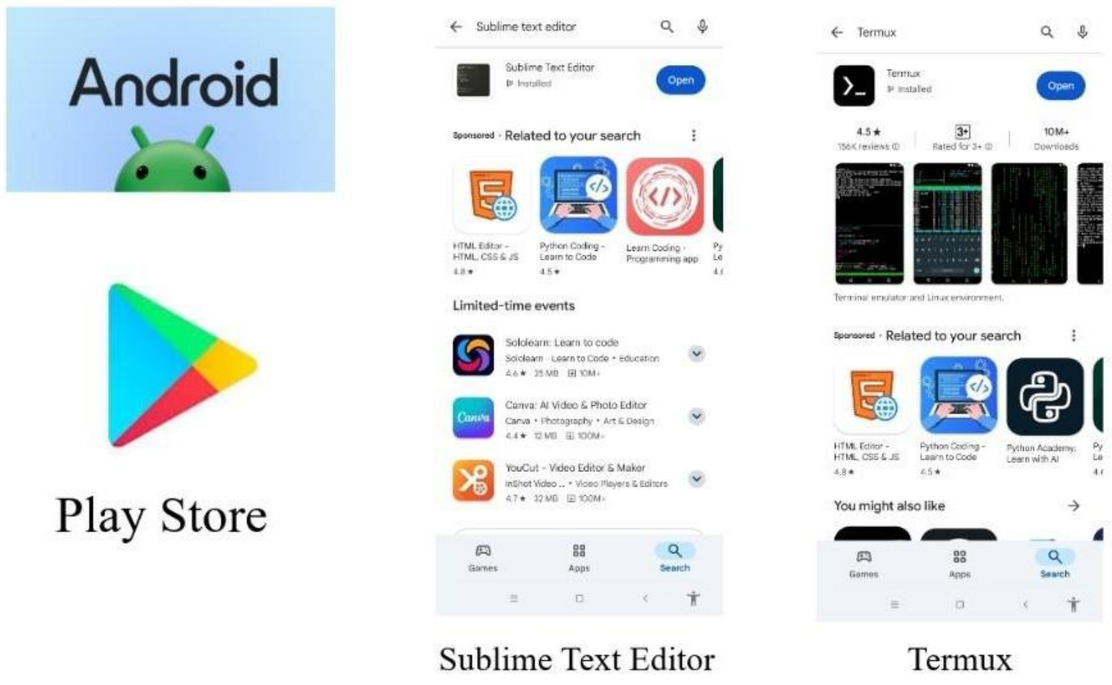


Figure 1.10. Requirements for Android Mobile

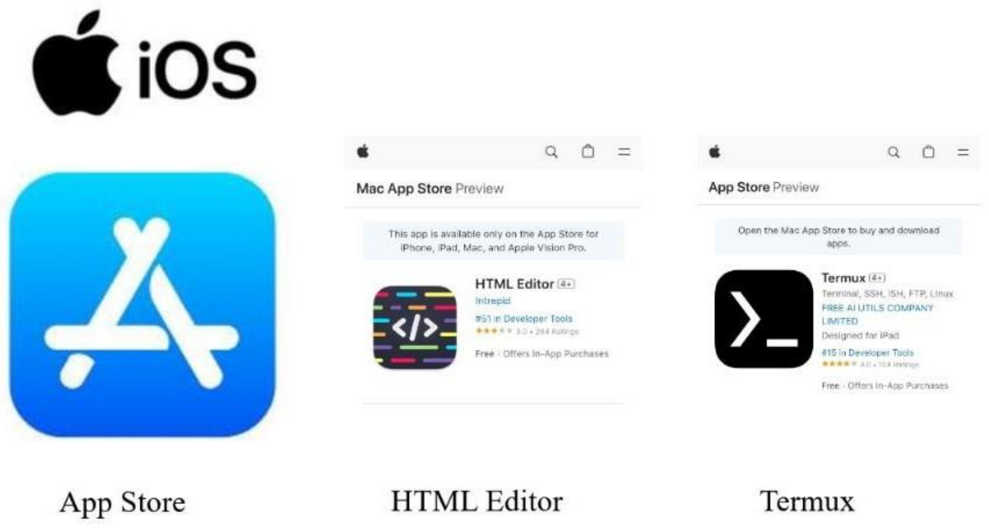
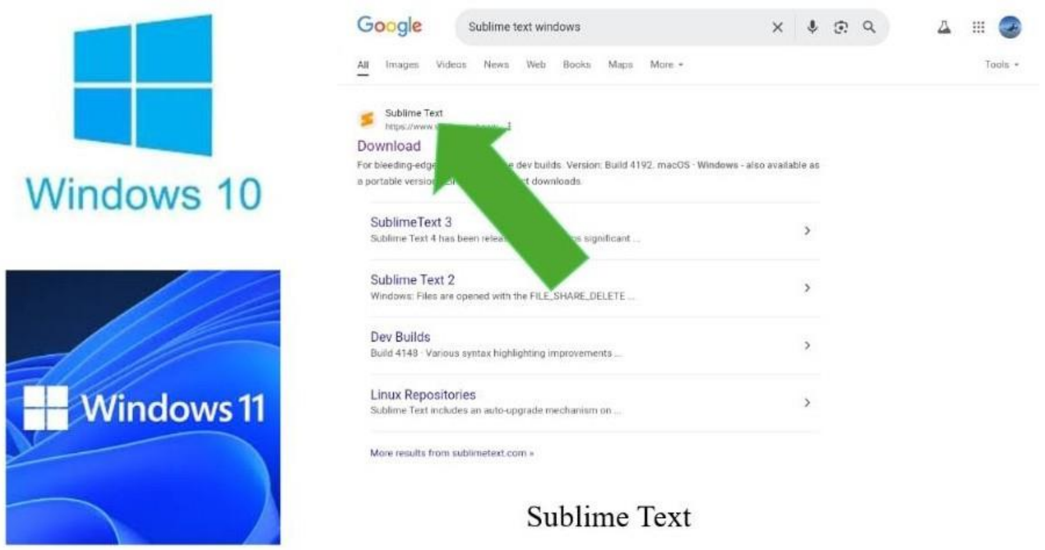
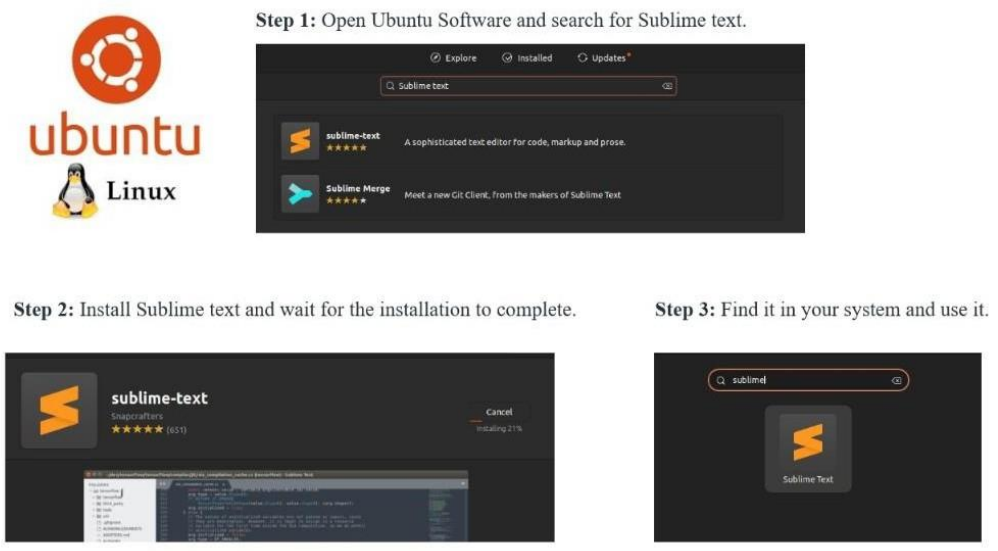


Figure 1.11.Requirements for IOS Mobile



Sublime Text
Download
For bleeding-edge … dev builds. Version: Build 4192. macOS · Windows - also available as a portable version … downloads.
SublimeText 3
Sublime Text 4 has been relea… significant …
Sublime Text 2
Windows: Files are opened with the FILE_SHARE_DELETE …
Dev Builds
Build 4148 · Various syntax highlighting improvements …
Linux Repositories
Sublime Text includes an auto-upgrade mechanism on …
More results from sublimetext.com »

Figure 1.12. Requirements for Window PC



Step 1: Open Ubuntu Software and search for Sublime text.

sublime-text — A sophisticated text editor for code, markup and prose.
Sublime Merge — Meet a new Git Client, from the makers of Sublime Text

Step 2: Install Sublime text and wait for the installation to complete.

Step 3: Find it in your system and use it.

Figure 1.13.Requirements for Ubuntu PC

Summary

This chapter provides a foundational understanding of web technology related development fields. It introduces key concepts such as frontend and backend development, differentiates between static and dynamic websites, and explains the typical structure of a webpage. Additionally, it outlines the basic requirements needed to begin web development, ensuring that beginners have a clear roadmap to follow. After chapter 1, what did you get, examine yourself honestly.

CHAPTER II

INTRODUCTION TO HTML WEBPAGE

Introduction

HTML (Hypertext Markup Language) is the standard markup language for making web pages. With only HTML you can create your own Website. However, HTML is not considered a "programming language"; it is a "markup language" which means it defines the structure and layout of content on a webpage. Although HTML does not have the capability to perform complex logic or calculations like a programming language does, it is a backbone of Web technology.

Key points in the history of the HTML in Web Technology:

1989: Tim Berners-Lee proposes the concept of the World Wide Web, outlining his vision for a system of interconnected documents accessible through hyperlinks.

1990: Berners-Lee develops the core technologies of the web including HTML, HTTP, and the first web browser/editor.

1991: The World Wide Web software is made publicly available, allowing other researchers and developers to start creating their own web pages.

1993: The Mosaic browser is released, considered a turning point in making the web more accessible to the general public due to its user-friendly interface.

April 1993: CERN releases the World Wide Web software into the public domain.

1994: Tim Berners-Lee establishes the World Wide Web Consortium (W3C) to oversee the development of web standards.

HTML is easy to learn - You will enjoy it.

- **Tag** : HTML tag starts with "<" and ends with ">" , example "**<p></p>**",
  In this case, "**<p>**" is called opening tag, and "**</p>**" is called closing tag.
  - **(i)** **Paired tag.**
    Require both opening tag and closing tag (<></>)
  - **(ii)** **Unpaired tag.**
    Require only opening tag (<>)

- **Attribute** : HTML attribute lays in first sub-tag ,
  example "**<p align = "center"></p>**".
  In this case, "align" is attribute, assigned the word "center" is attribute value.

- **Element** : HTML element is the whole both tags with content ,
  example "**<p> + content +</p>**".
  In this case, the whole structure is element.
  - (iii) Structure without attribute.
    "**<opening tag> + content + </closing tag>**".
    "**<p>Hello World</p>**". (is called Element)
  - (iv) Structure with attribute.
    "**<opening tag + attribute> + content + </closing tag>**".
    "**<p align= "left">Hello World</p>**". (is called Element too)

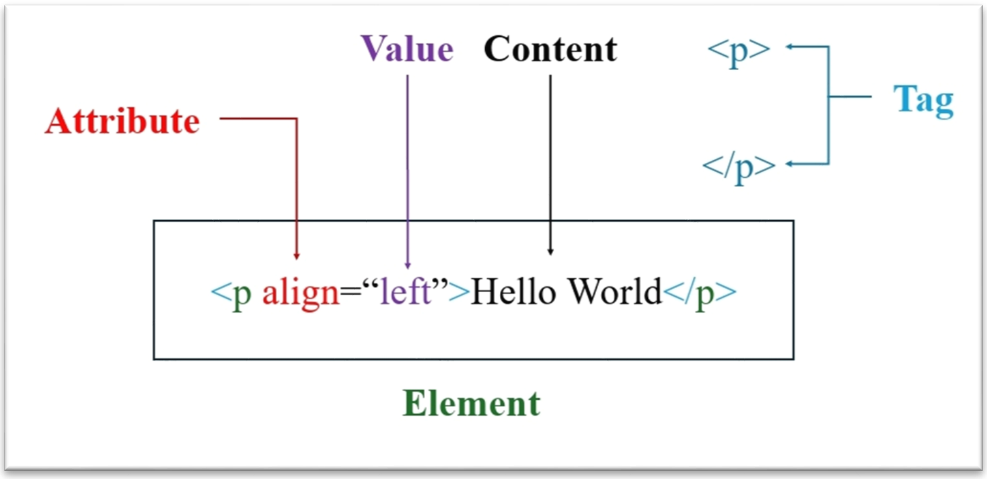


Figure 2.1.The Components of HTML Element

2.1. Startup HTML

Step 1 : Create new "website" folder.

Step 2 : Use "**Sublime Text**" software , open new file in name "index" with ".html" extension. (it is meant that "index.html"). Save this HTML file under "website" folder.

Step 3 : Code in this file, as shown below, and then save it.

Step 4 : Open HTML file with browser app.

The HTML code of typical structure of a Webpage.

```
<html>
        <head>
                <title></title>
        </head>
        <body>
        </body>
</html>
```

Step 4 : Open HTML file with browser app.

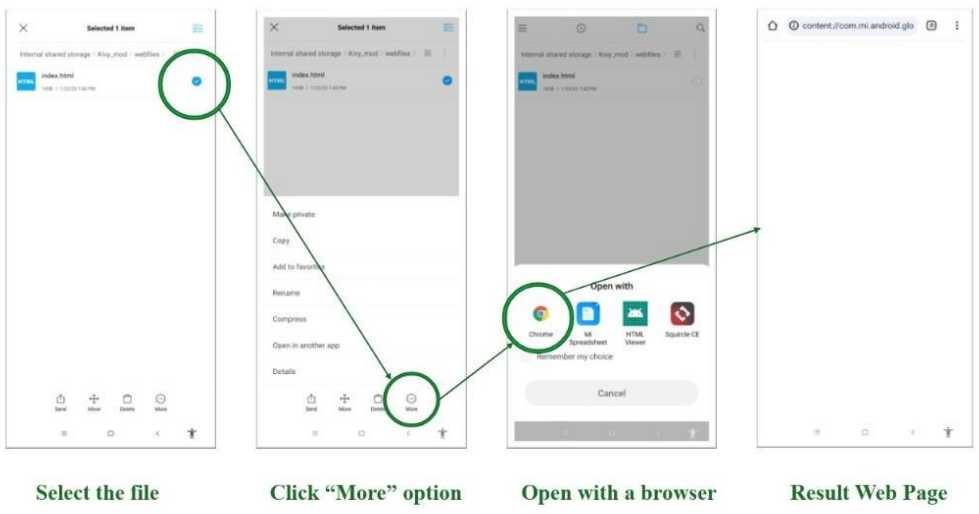


Figure 2.2. Startup HTML with Mobile

Note : The result webpage must be white plane. It should not returned any texts or messages. It is to understand that only HTML tags would not be rendered.

2.2. HTML Elements Testing

The HTML code of a Webpage with some information. Edit the source code of previous HTML file and refresh browser.

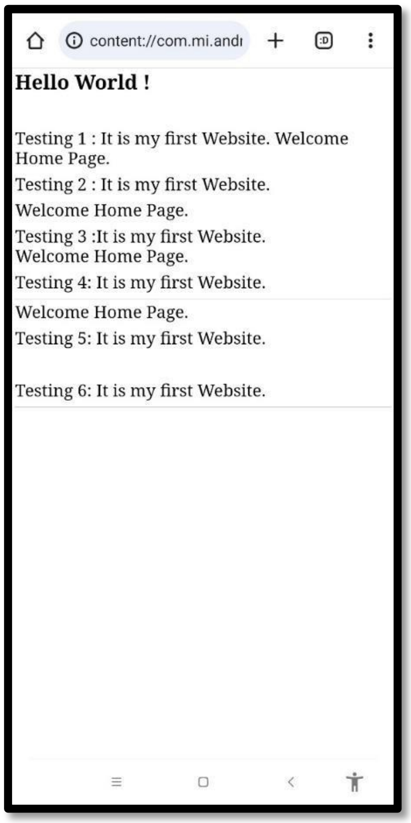
```
<html>
        <head>
                <title>#Replace your name</title>
        </head>
        <body>
                <h1>Hello World !</h1> <br>
                <p>Testing 1 :  It is my first Website.
                    Welcome Home Page.</p>
                <p>Testing 2 : It is my first Website.</p><p> Welcome Home Page.</p>
```

```
            <p>Testing 3 :It is my first Website.<br>Welcome Home Page.</p>
            <p>Testing 4: It is my first Website.<hr>Welcome Home Page.</p>
            <p>Testing 5: It is my first Website.</p><br>
            <p>Testing 6: It is my first Website.</p><hr>
        </body>
</html>
```

Note : Unnecessary to code entirely or create the new file. Just edit the source code as following.

- Fill your name between opening and closing " title" tags. It will decorate the URL with your name.
- Code new "h1" , "hr" , "p" and "br" tags within "body" tag. And then fill the contents as the instructions.
- You will see "h1" and "p" elements act heading and paragraph behaviors (They are **Paired tags**) but <hr> = horizontal ruler and <br> = line break behaviors. (They are **Unpaired tags**

It is to understand that only HTML elements are rendered in the webpage.



Hello World !

Testing 1 : It is my first Website. Welcome Home Page.

Testing 2 : It is my first Website.

Welcome Home Page.

Testing 3 :It is my first Website.
Welcome Home Page.

Testing 4: It is my first Website.

Welcome Home Page.

Testing 5: It is my first Website.

Testing 6: It is my first Website.

Figure 2.3.Result of Rendering HTML Elements

2.3. Concepts and Tags

Webpages mostly behave document screens so that documental concepts such as thematic and semantic, are essential. Therefore, there are thematic and semantic tags in HTML.

- What is **Thematic Tag**: define distinct sections within a webpage to help structure the page logically and improve visual presentation for reader. Thematic tags may be paired or unpaired tags.
  Example tags: <div></div>, <center></center> , <hr> , <br> , etc.
- What is **Sematic Tag :** define particular meaning of the contents within webpage to help both browser and developer but not visual presentation improvement for reader.
  Example tags: <table></table>,<form></form>,<h></h>,<p></p>, etc.

Web contents include symbols and text decorations they are not the particular part of webpage so that tags become differed in HTML as inline element and block element.

- What is **Inline Element** : this element doesn't take new line or new block space within webpage.
- What is **Block Element** : this element takes new line or new block space within webpage.

Here are the block-level elements in HTML:

<address>,<article>,<aside>,<blockquote>,<canvas>,<dd>,<div>,<dl><dt>,<fieldset>,<figcaption>,<figure>,<footer>,<form>,<h1>,<h2>,<h3>,<h4>,<h5>,<h6>,<header>,<hr>,<li>,<main>,<nav>,<noscript>,<ol>,<p>,<pre>,<section>,<table>,<tfoot>,<ul>,<video>,etc.

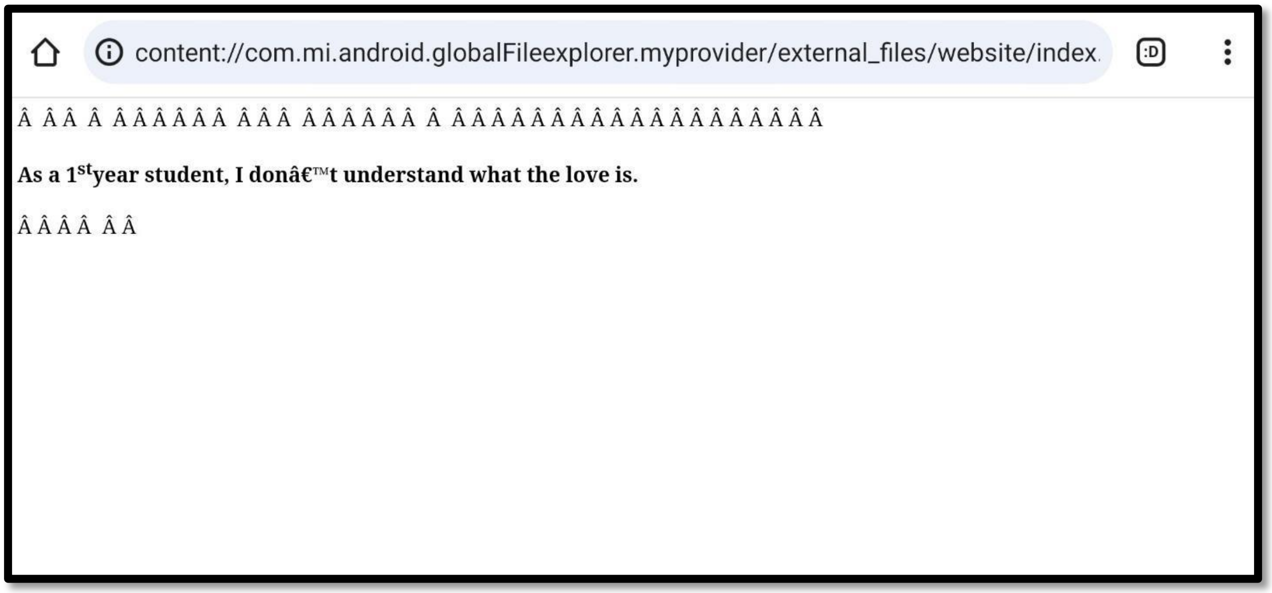
Here are the inline elements in HTML:

<a>,<abbr>,<acronym>,<b>,<bdo>,<big>,<br>,<button>,<cite>,<code>,<dfn>,<em>,<i>,<img>,<input>,<kbd>,<label>,<map>,<object>,<output>,<q>,<samp>,<script>,<select>,<small>,<span>,<strong>,<sub>,<sup>,<textarea>,<time>,<tt>,<var>,etc.

Note: An inline element cannot contain a block-level element, but a block-level element can contain the inline element.

Example : <h4>As a 1<sup>st</sup>year student, I don't understand what the love is.</h4>

```
<html>
   <head>
      <title></title>
   </head>
      <body>
            <h4> " As a 1<sup>st</sup> year student, I don't understand what the love is." </h4>
     </body>
</html>
```


content://com.mi.android.globalFileexplorer.myprovider/external_files/website/index.

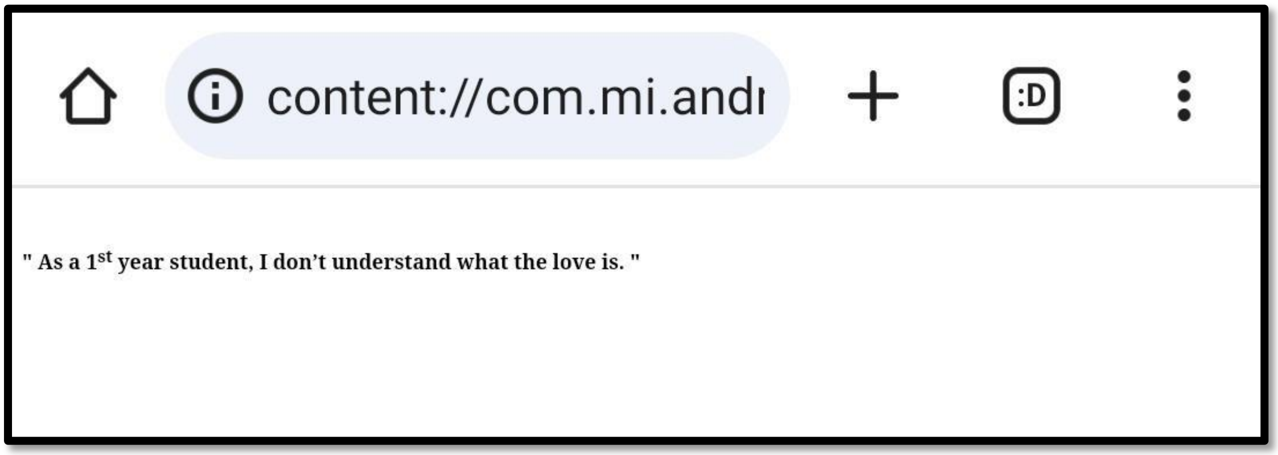
```
Â ÂÂ Â ÂÂÂÂÂÂ ÂÂÂ ÂÂÂÂÂÂ Â ÂÂÂÂÂÂÂÂÂÂÂÂÂÂÂÂÂÂ

As a 1styear student, I donâ€™t understand what the love is.

ÂÂÂÂ ÂÂ
```

Figure 2.4.Uncorrected Text Character Encoding

Solution : <meta> tag is needed to use, for encoding text character correctly.

Insert <meta charset = “UTF-8”> within head tag. The meta tag has only opening tag but not closing tag so that it is called **Unpaired tag,**

```
<html>
        <head>
           <title></title>
           <meta charset="UTF-8">
        </head>
        <body>
            <h4> “ As a 1<sup>st</sup> year student, I don’t understand what the love is.” </h4>
        </body>
</html>
```



" As a 1st year student, I don't understand what the love is. "

Figure 2.5.The Effect of Meta Tag

In that case, <h4> is block level element and <sup> is inline level element, so that block element can contain inline element.

2.4. HTML Comment

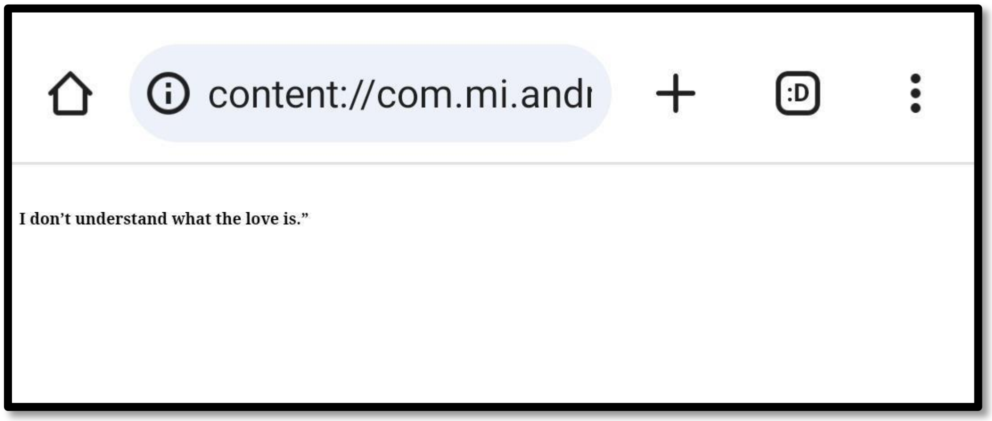
Comments do not appear on the front end or browser. Symbol is < ! - - (starting comment) and - - > (ending comment) .Types of HTML Comments : there are three types of comments .

- Single-line comment
- Inline comment
- Multi-lines comment

Now, let us learn about them.

```
<html>
    <head>
        <title></title>
        <meta charset="UTF-8">
    </head>
        <body>
< ! --  <h4> " As a 1<sup>st</sup> year student,</h4> - - >
       <h4> I don't understand what the love is." </h4>
```

```
        </body>
</html>
```



I don't understand what the love is."

Figure 2.6. Single-line Comment Result

```
<html>
    <head>
          <title></title>
          <meta charset="UTF-8">
    </head>
         <body>
< ! --  <h4> " As a 1<sup>st</sup> year student,</h4>
        <h4> I don't understand what the love is." </h4> - - >
         </body>
</html>
```

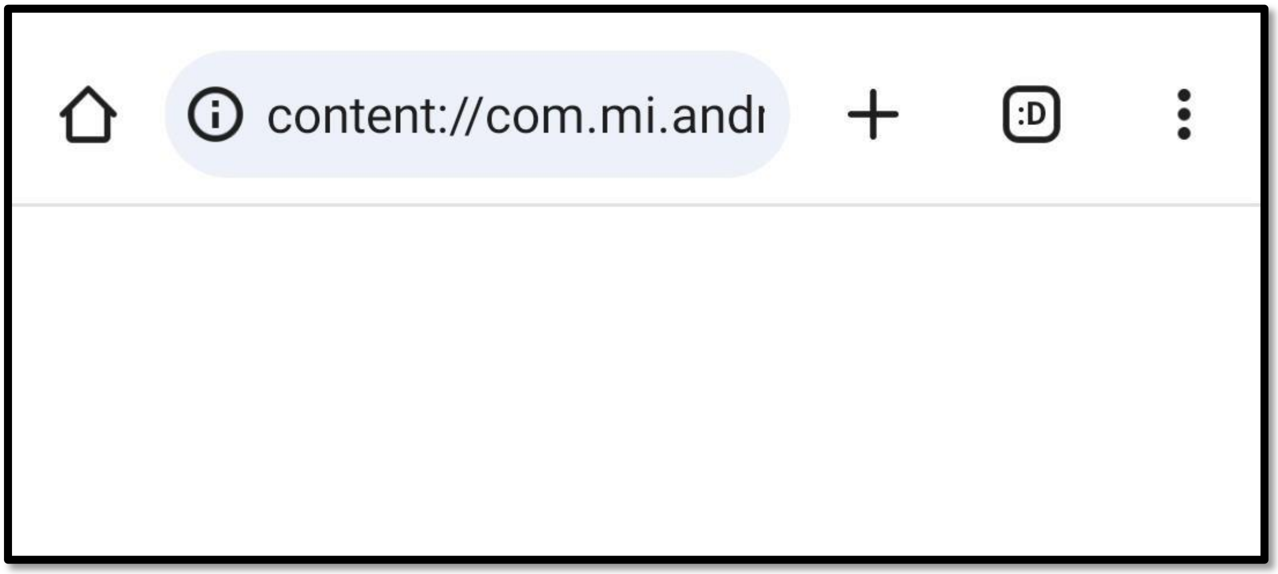
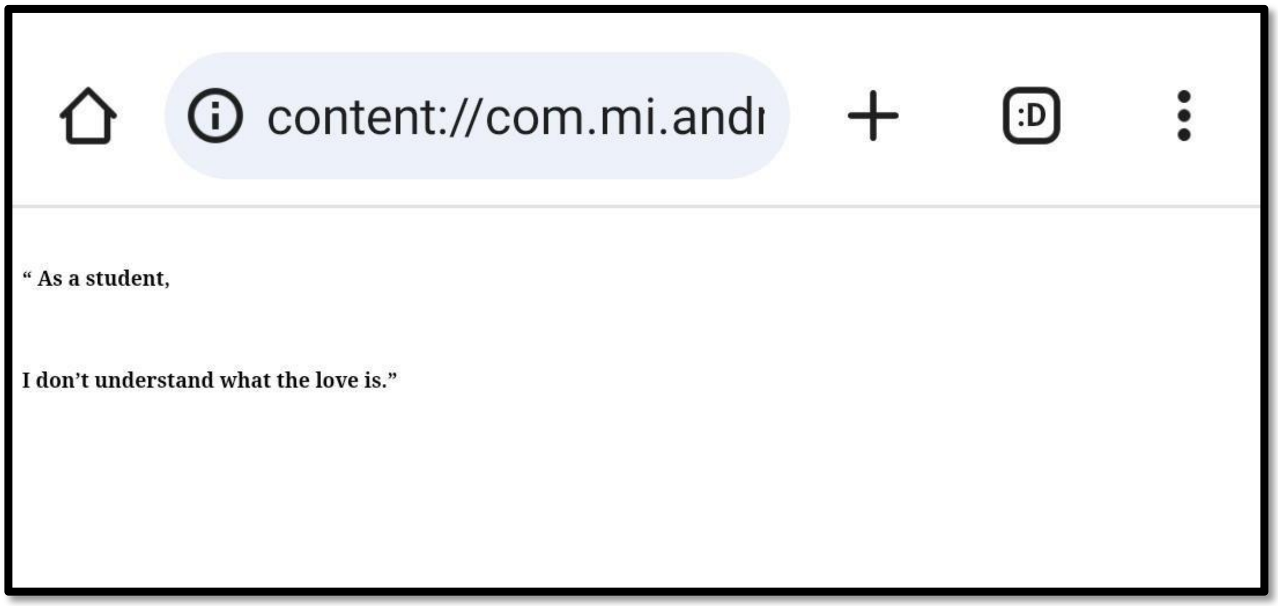


Figure 2.7.Multi-line Comment Result

```
<html>
    <head>
         <title></title>
         <meta charset="UTF-8">
    </head>
        <body>
       <h4> " As a < ! -- 1<sup>st</sup> year - - > student,</h4>
       <h4> I don't understand what the love is." </h4>
        </body>
</html>
```



```
“ As a student,

I don’t understand what the love is.”
```

Figure 2.8.Inline Comment Result

Summary

This chapter focuses on HTML, the backbone of web pages. It guides readers through creating their first HTML file, testing elements, and understanding essential tags. The chapter also explains HTML comments and their role in structuring code. By the end, readers will have a solid grasp of how to structure and render a simple webpage.

CHAPTER III
LOCAL WEB HOSTING TECHNOLOGY

Introduction

In this chapter would cover hosting in static website , not dynamic. At first , students need to know about web server. And then local hosting scenario would be engaged by a tiny web development project, implied micro project. After all, students get server knowledge and local web hosting experience practically.

3.1. About Web Server

Website is a set of files so that web server is a program that is responsible to host this resources. Hosting is the handling HTTP, client site (browser) send HTTP request (user searching in URL bar , by using targeted server IP address) and server site (program) reply with HTTP response (user getting webpage).

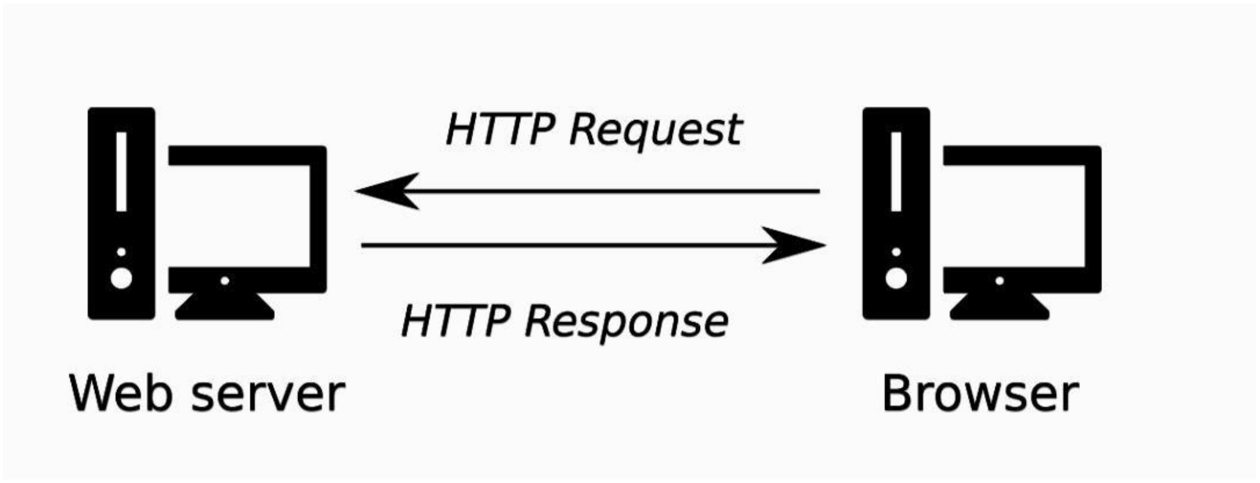


Figure 3.1.How Web Server Work

3.2. Startup Micro Web Project

Project target is to host static webpage in locally, in this project , student will have to practice how to host static website in local network . Website is including media thing (image,audio and video) in static webpage. For project man power organizing , grouping 8 or 10 members per group. Each group much report a result to the instructor. No all students need to report the result. One result for one group. In this chapter , only hosting technology will be engaged, HTML coding will not be emphasized. Next chapter , web development in live hosting (real world online).

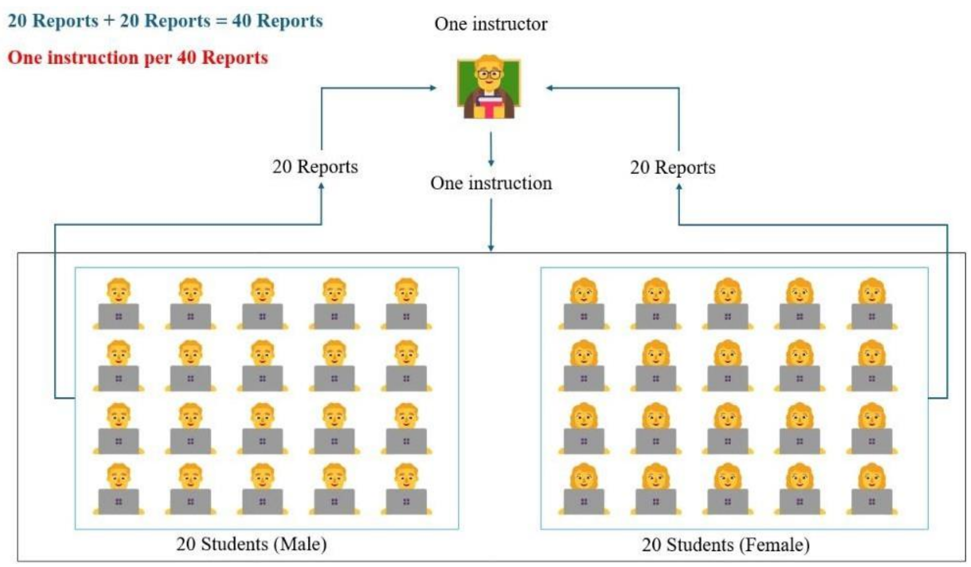


Figure 3.2.Ordinary Class Time

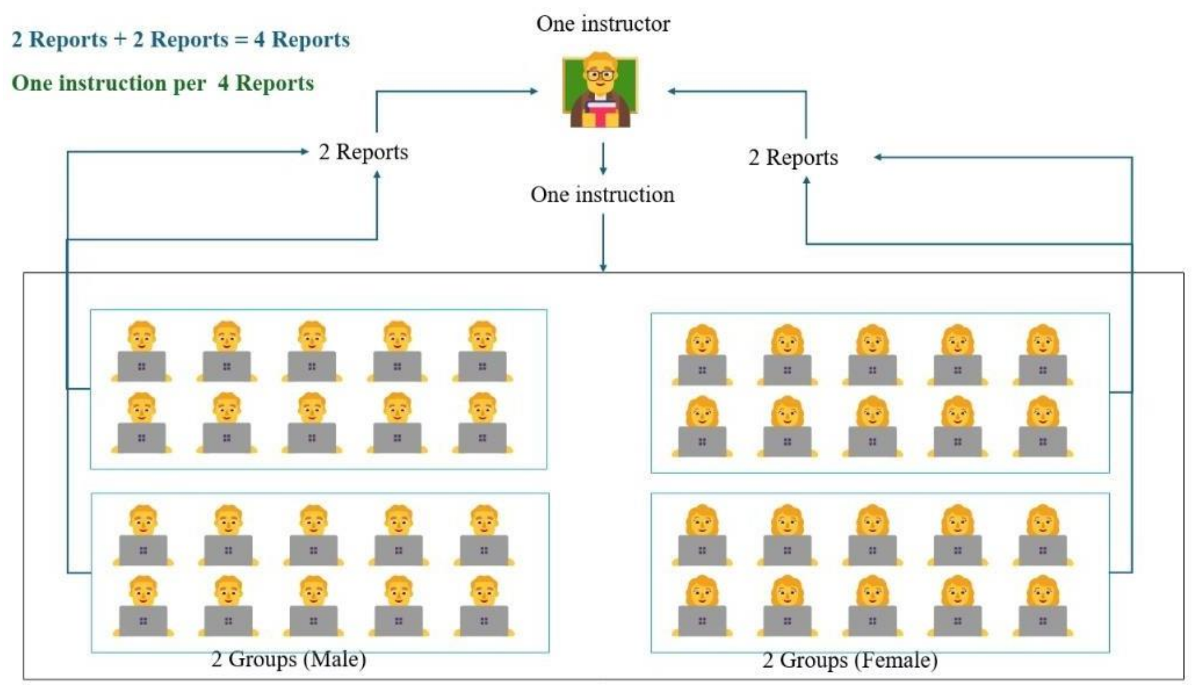


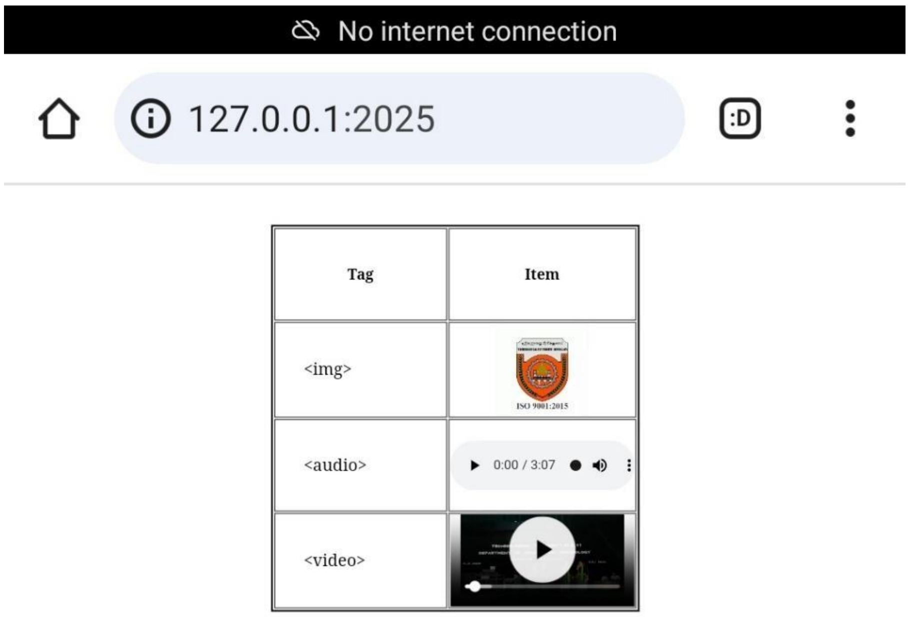
Figure 3.3.Project Class Time

Students will have to participate an informal class system, as the nature of project-oriented-curriculum, it is diversed from formal system.

- Ordinery class : had applied ordinary curriculum (settlement full course to qualify in certain academic goal such as 1st year academic level, 2nd year academic level, etc). Ordinary class purpose the studends could generate new knowledge , theories or, methods in finale. It is to build a proper professtionale such as an engineer, a medical doctor, an army officer, a computer scientist, an architect, etc.

- Project class : had applied some informal class system, it is espescially an essential part in technical education, it is only for not only candidating course but also deploying some knowledge and enchancing proper skill sets. It had no intention to generate new knowledge. Difference between ordinary class and project class is that distinct differ in teaching rate, in ordinary class, one instructor has to handle all feedbacks each individual. In project class, one instructor has to handle collective results of fellow groupping.

3.3. Introduction to Data Collection



Tag	Item
<img>	ISO 9001:2015
<audio>	0:00 / 3:07
<video>	

Figure 3.4. Expected Resulant (http://127.0.0.1:2025)

Step 1 : Create new source folder, name in “website” . (No need to create if it had been)

Step 2 : Open source folder , then create “index.html” file and “asset” folder. (No need to create “index.html” file if it had been).

Step 3 : Downlod image file from https://github.com/lawintun0110/lawintun0110.github.io/blob/main/asset/logo.jpg

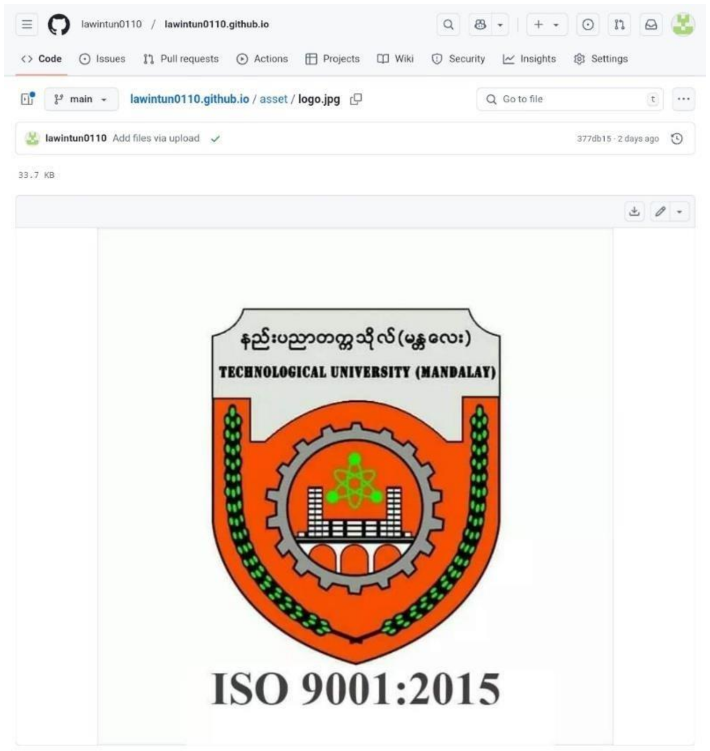

lawintun0110 / lawintun0110.github.io
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
main
lawintun0110.github.io / asset / logo.jpg
Go to file
lawintun0110 Add files via upload
377db15 · 2 days ago
33.7 KB
TECHNOLOGICAL UNIVERSITY (MANDALAY)
ISO 9001:2015

Figure 3.5.Image File Location

Step 4 : After downloading image file, move the logo (logo.jpg) under asset folder.

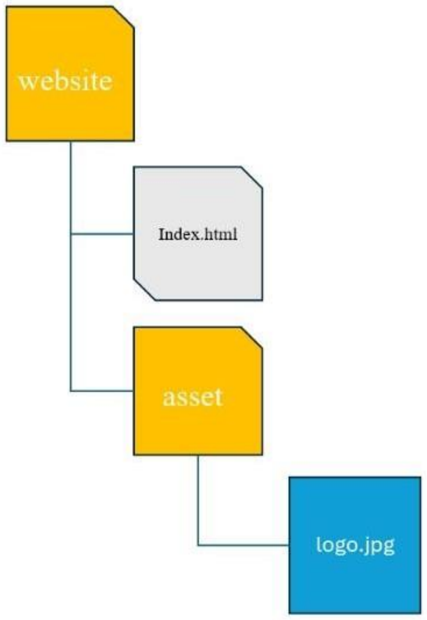
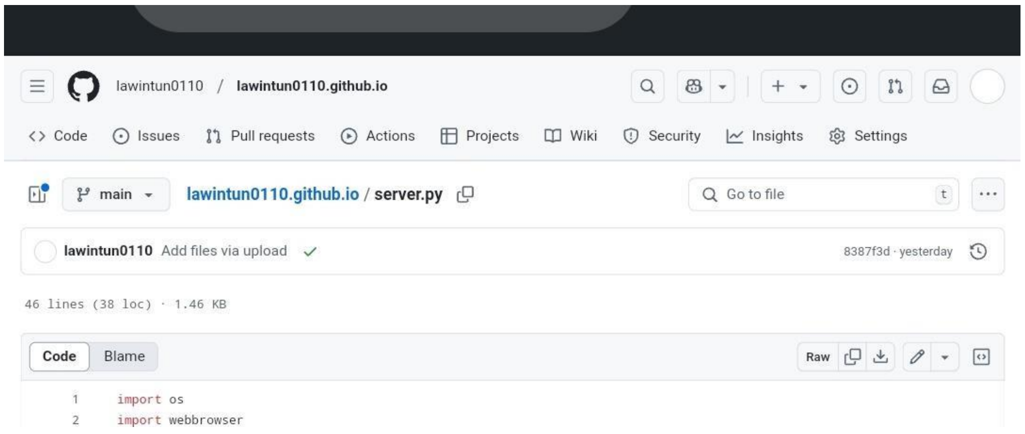


Figure 3.6.Common File Structure

Step 5 : Download server program file from https://github.com/lawintun0110/lawintun0110.github.io/blob/main/server.py


lawintun0110 / lawintun0110.github.io
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
main
lawintun0110.github.io / server.py
Go to file
lawintun0110 Add files via upload
8387f3d · yesterday
46 lines (38 loc) · 1.46 KB
Code
Blame
Raw

```
import os
import webbrowser
```

Figure 3.7. Server File Location

Step 6 : After downloading server.py file, move to under website folder.

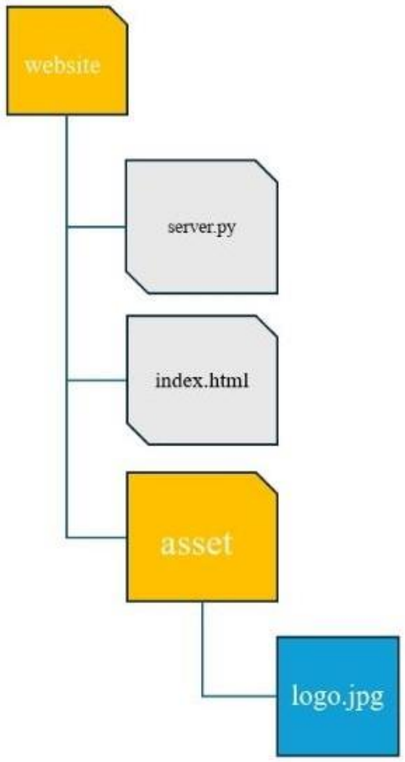


Figure 3.8.Updated File Structure

Step 7 : Download audio file from https://github.com/lawintun0110/lawintun0110.github.io/blob/main/asset/LWT%26PP.mp3

Step 8 : After downloading, move LWT&PP.mp3 to under asset folder.

Step 9 : Download video file from

https://github.com/lawintun0110/lawintun0110.github.io/blob/main/asset/IT-Welcome-2008.mp4

Step 10 : After downloading, move IT-Welcome-2008.mp4 to under asset folder.

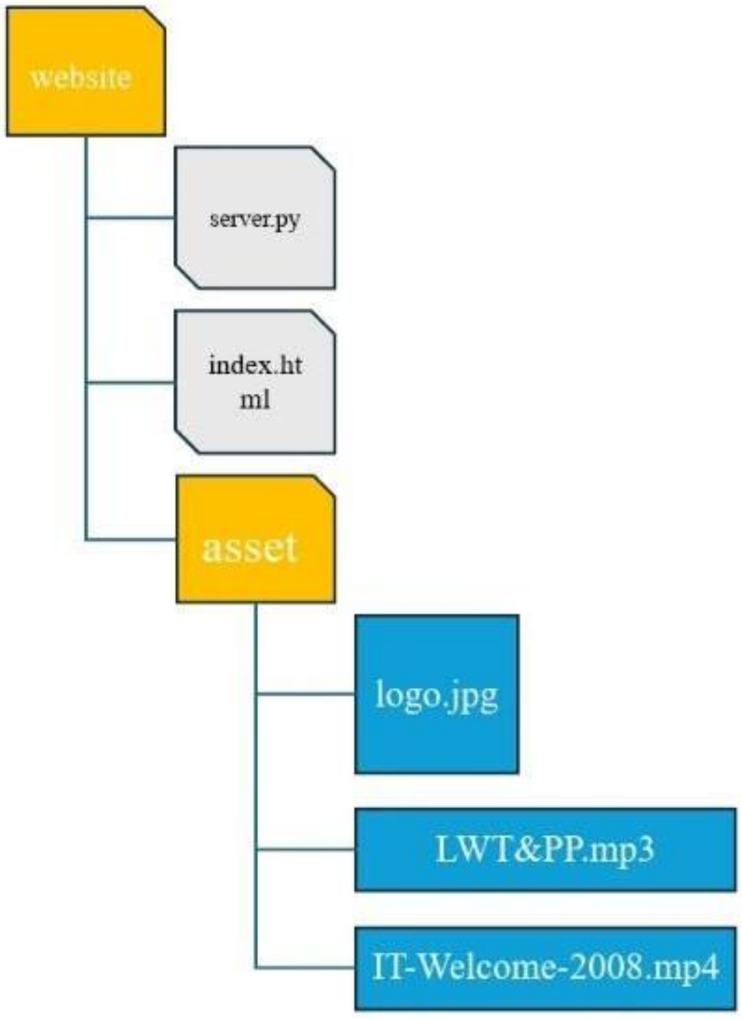


Figure 3.9.Resultant File Structure

3.4. Introduction to Implementation

Step 1: Open Pycharm if you are Window user and Open Sublime if you are Ubuntu or Android or ios user. In this case, Pycharn and Sublime are assumed as Development Environment.

Step 2 : Open project folder (website folder) from proper environment. (Sublime or Pycharm) .

Step 3 : Open server.py file in text editor (Sublime) or IDE (Pycharm) , and then customize default path with your actual project folder (called website) location.

server.py

```
import os

import webbrowser

from threading import Thread

from http.server import SimpleHTTPRequestHandler

from socketserver import TCPServer


class CustomTCPServer(TCPServer):

    allow_reuse_address = True  # Allow reuse of the same port


class WebServer:

    def start(self):

        # Start the local server and open the web page

        if self.start_server():

            self.open_web_page()


    def start_server(self):

        directory = "/storage/emulated/0/website/" #for Android  (Default)

        # “C:\Users\website\” for Window

       # “/Name/Home/website/” for Ubuntu

        if not os.path.exists(directory):
            print(f"Directory does not exist: {directory}")
            return False

        os.chdir(directory)
        print(f"Serving files from: {directory}")

        def server_thread():
            handler = SimpleHTTPRequestHandler
            with CustomTCPServer(("127.0.0.1", 2025), handler) as httpd:
                try:
                    print("Server started at http://127.0.0.1:2025")
                    httpd.serve_forever()
                except KeyboardInterrupt:
                    print("Shutting down the server...")
                    httpd.shutdown()

        thread = Thread(target=server_thread)
        thread.daemon = True
        thread.start()
        return True

    def open_web_page(self):
        webbrowser.open("http://127.0.0.1:2025")  # Open in the default browser

if _name___== "_main_":
    server = WebServer()
    server.start()
```

```
input("Press Enter to stop the server...")
```

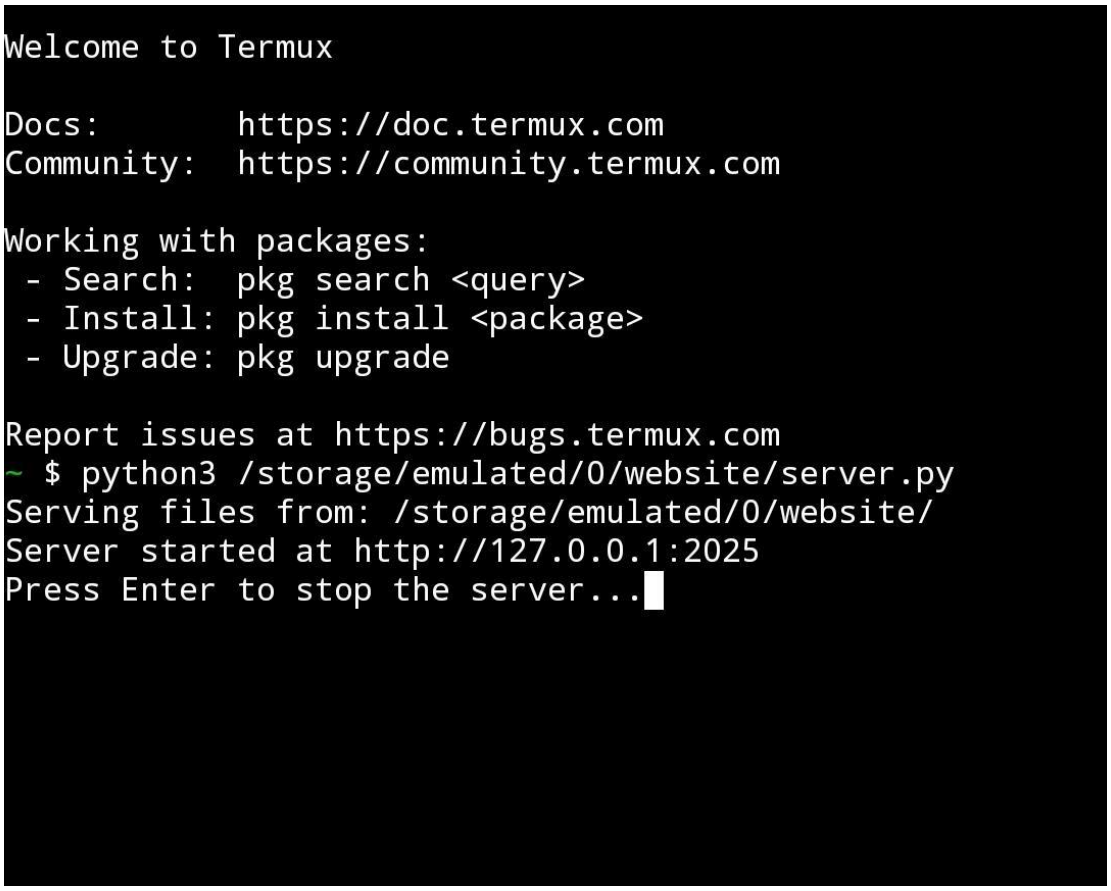
Step 4 : Run **server.py** (Server program),

(A) In android, apply **Termux ,** open internet or wifi , open Termux app.

- Allow storage access, use command "termux-setup-storage"
- Update & upgrade , use command "pkg update -y && pkg upgrade -y" (if any error encountered , skip this step)
- Installation Python3, use command "pkg install python3"

And then, you can turn off internet connection or wifi,

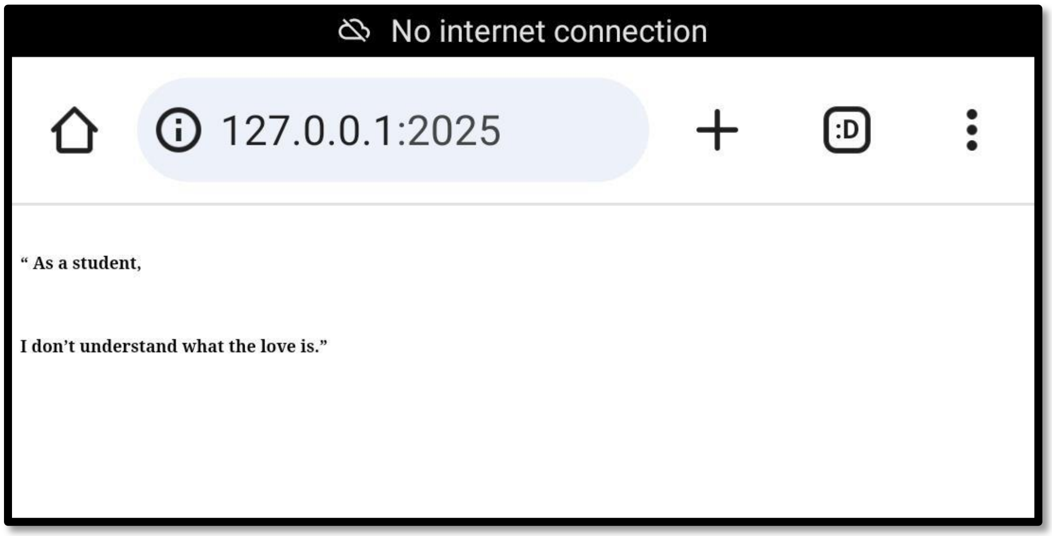
- Use command "python3 /storage/emulated/0/website/server.py". (Offline running)



```
Welcome to Termux

Docs:       https://doc.termux.com
Community:  https://community.termux.com

Working with packages:
 - Search:  pkg search <query>
 - Install: pkg install <package>
 - Upgrade: pkg upgrade

Report issues at https://bugs.termux.com
~ $ python3 /storage/emulated/0/website/server.py
Serving files from: /storage/emulated/0/website/
Server started at http://127.0.0.1:2025
Press Enter to stop the server...
```

Figure 3.10.HTTP Server Hosting in Termux

- Exit with home key, not back key, ensure termux session is running, it is hosting server, Open browser (Chrome or FireFox), type address http://127.0.0.1:2025 in the search bar, it is called making HTTP request.



" As a student,

I don't understand what the love is."

Figure 3.11. HTTP Response with Webpage

- After you type http://127.0.01:2025 in browser search bar, you will get webpage as Figure 3.8, it is called HTTP response,

(B) In PC (computer laptop or desktop) ,apply pycharm , open server.py file,

- Run the program in pycharm (offline)
- Open Browser , type http://127.0.01:2025 in the search bar
- You will get webpage as HTTP response.

Step 5 : Download source code file (index.html) from https://github.com/lawintun0110/lawintun0110.github.io/blob/main/index.html

Step 6 : Move this index.html to website folder and replace old index.html with this, and save it.

Step 7 : Clear browser tag, and retype http://127.0.0.1:2025 , finally we will get expected resulant like Figure 3.1.

Summary

Here, the focus shifts to hosting web pages locally. The chapter introduces the concept of web servers and walks readers through setting up a small-scale project using a local server. It also provides an introduction to data collection and basic implementation methods, preparing learners for more advanced hosting techniques.

CHAPTER IV
LIVE WEB HOSTING TECHNOLOGY

Introduction

Welcome from finale chapter. In previous chapter, target defining and project grouping had been , this chapter will cover how to live hosting real world. Deployment of Github page is that everyone would get a webpage with subdomain that include “github.io” extension .

Example : lawintun0110.github.io

In this case

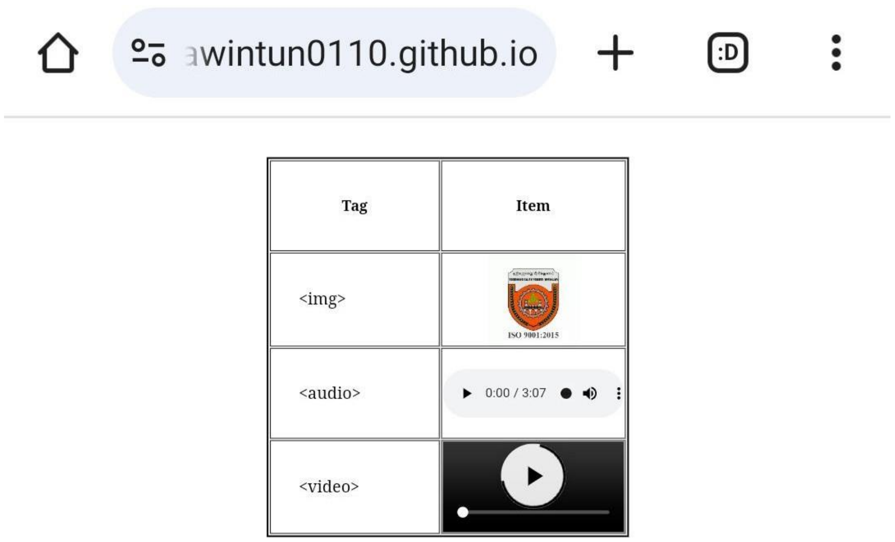
“.io” is TLD (Top Level Domain)

“github.” is SLD (Secondary Level Domain)

“lawintun0110.” is subdomain.

This subdomain provided by github could not be managed and controlled, you would not get panal session of DNS server. It could not be integrated with other tools or software or database to evaluate dynamic site, maintaining only static website but everyone taste live hosting experiences.

Sample resultant of live hosting : https://lawintun0110.github.io/



Tag	Item
<img>	
<audio>	0:00 / 3:07
<video>	

Figure 4.1.Expected Resultant (https://lawintun0110.github.io/)

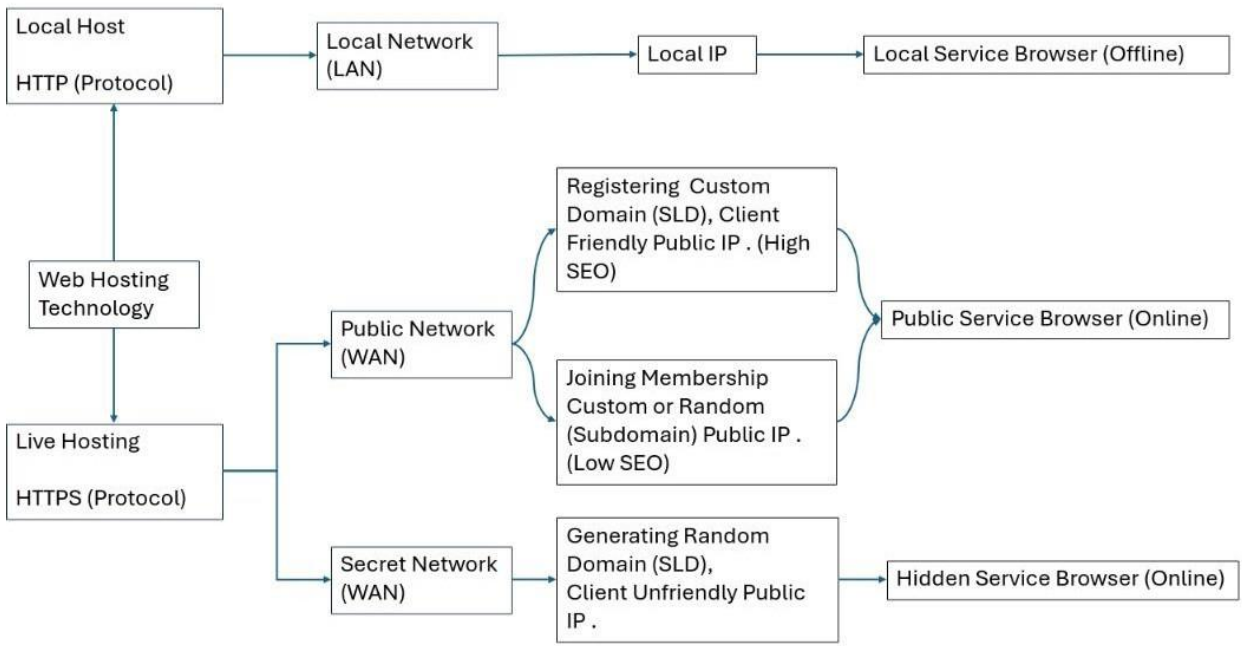

Local Host
HTTP (Protocol)
Local Network (LAN)
Local IP
Local Service Browser (Offline)
Registering Custom Domain (SLD), Client Friendly Public IP . (High SEO)
Web Hosting Technology
Public Network (WAN)
Public Service Browser (Online)
Joining Membership Custom or Random (Subdomain) Public IP . (Low SEO)
Live Hosting
HTTPS (Protocol)
Generating Random Domain (SLD), Client Unfriendly Public IP .
Secret Network (WAN)
Hidden Service Browser (Online)

Figure 4.2. Web Hositing Technology Map

4.1. ICANN And Webdomain

ICANN, which stands for "Internet Corporation for Assigned Names and Numbers," is a non-profit organization established in 1998 to manage and coordinate the internet's naming system, including domain names and IP addresses, ensuring a stable and unified global internet by providing a central registry for these unique identifiers across the world; essentially acting as the "address book" for the internet, allowing users to access websites using easy-to-remember domain names instead of complex IP addresses.

Figure 4.3.Logo of ICANN

TLD Management: ICANN is responsible for approving new top-level domains (like .com, .org, .net) and managing existing ones.

Function: ICANN does not directly register domain names, but oversees the process by accrediting domain registrars who handle user registrations.

Registering a Domain: When you want to register a domain name, you go through a domain registrar who is accredited by ICANN.

Database Update: The registrar submits your domain name and corresponding IP address to the ICANN database, which is essentially a central registry for all domain names.

4.1.1. Historical Context of ICANN

Early Internet: Before ICANN, the management of domain names was primarily handled by the US government through the Internet Assigned Numbers Authority (IANA).

Formation of ICANN: In 1998, the US government transitioned the responsibility of managing domain names to a non-profit organization, ICANN, with a more global stakeholder involvement.

Figure 4.4. Headquarter Building of ICANN

Abbreviation : ICANN Founded September 30, 1998; 26 years ago .

Tax ID no : 95-4712218

Focus : Manage Internet Protocol numbers and Domain Name System root.

Headquarters : Los Angeles, California, United States.

Key people : Kurt Erik Lindqvist (CEO and president), Tripti Sinha (Chair of the Board) and Jon Postel (founder).

Employees : 428

Website : icann.org

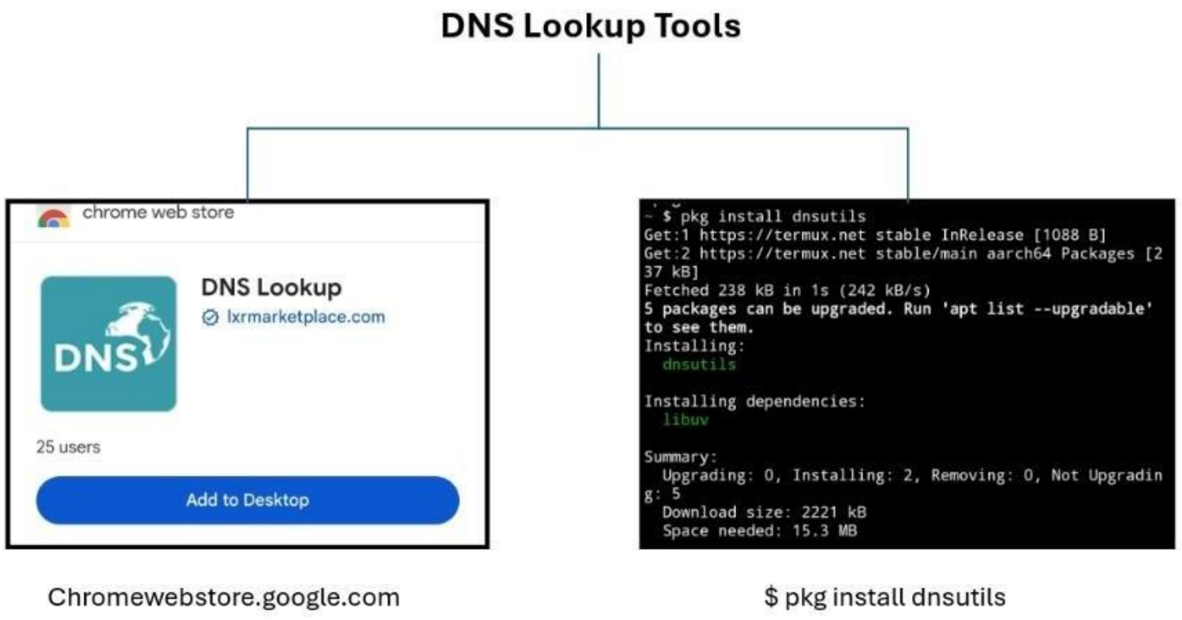
4.1.2. Brief About Webdomain

Ordinary web domains, like .com, .org, or .biz, are issued by the Internet Corporation for Assigned Names and Numbers (ICANN). There are thousands of different domains out there, but not all of them can be used by everyone (like .apple, for example). Users have to submit proposals to ICANN to register a domain and sub-domain (the part before the period). There are usually costs associated with registering and maintaining the domain of your choice.

Domain Structure: A domain name consists of two parts: a second-level domain (like "google") and a top-level domain (like ".com").

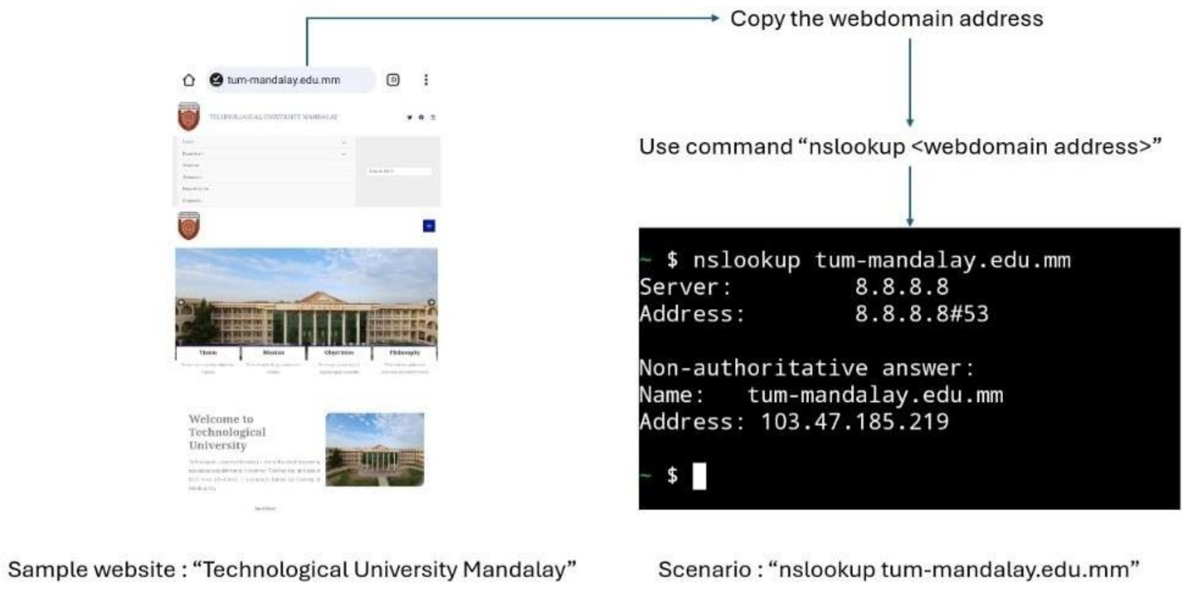
DNS (Domain Name System): When you type a domain name into your browser, the DNS system translates it into an IP address, allowing your computer to locate the website.

DNS Lookup: When you type a domain name in your browser, your computer queries a DNS server, which looks up the associated IP address in the ICANN database and directs your browser to the correct website.



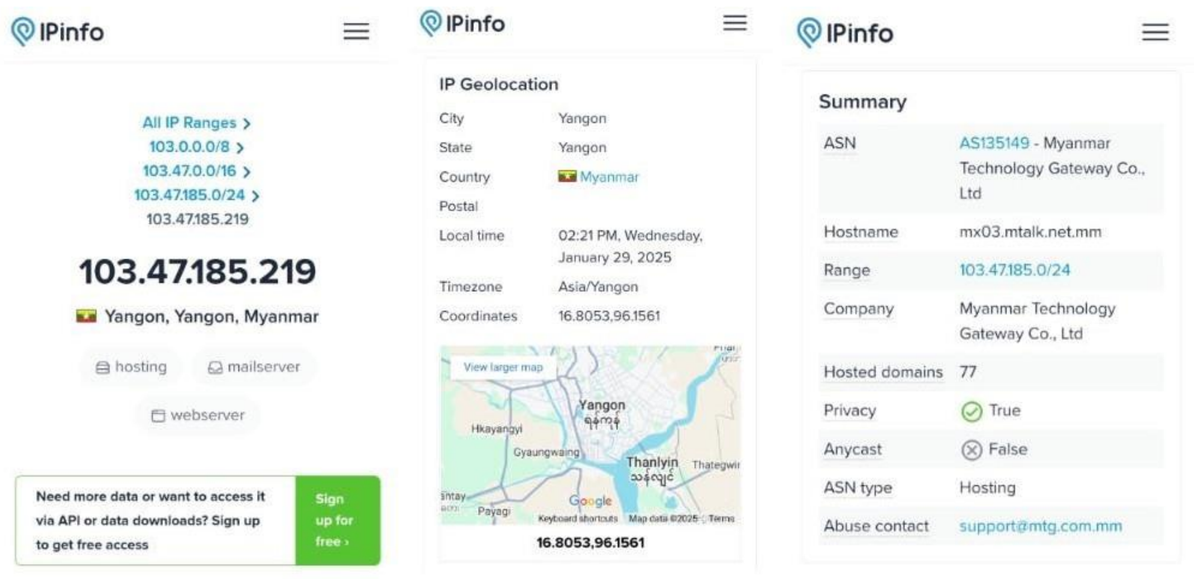
```
~ $ pkg install dnsutils
Get:1 https://termux.net stable InRelease [1088 B]
Get:2 https://termux.net stable/main aarch64 Packages [2
37 kB]
Fetched 238 kB in 1s (242 kB/s)
5 packages can be upgraded. Run 'apt list --upgradable'
to see them.
Installing:
  dnsutils

Installing dependencies:
  libuv

Summary:
  Upgrading: 0, Installing: 2, Removing: 0, Not Upgradin
g: 5
  Download size: 2221 kB
  Space needed: 15.3 MB
```

Figure 4.5.DNS Lookup Tools


Copy the webdomain address
Use command “nslookup <webdomain address>”
tum-mandalay.edu.mm
Welcome to Technological University
Sample website : “Technological University Mandalay”
Scenario : “nslookup tum-mandalay.edu.mm”

```
~ $ nslookup tum-mandalay.edu.mm
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
Name:   tum-mandalay.edu.mm
Address: 103.47.185.219

~ $ █
```

Figure 4.6. Example of DNS Lookup Usage



103.47.185.219

Yangon, Yangon, Myanmar

Need more data or want to access it via API or data downloads? Sign up to get free access

IP Geolocation

City	Yangon
State	Yangon
Country	Myanmar
Postal	
Local time	02:21 PM, Wednesday, January 29, 2025
Timezone	Asia/Yangon
Coordinates	16.8053,96.1561

16.8053,96.1561

Summary

ASN	AS135149 - Myanmar Technology Gateway Co., Ltd
Hostname	mx03.mtalk.net.mm
Range	103.47.185.0/24
Company	Myanmar Technology Gateway Co., Ltd
Hosted domains	77
Privacy	✓ True
Anycast	⊗ False
ASN type	Hosting
Abuse contact	support@mtg.com.mm

Figure 4.7. IPinfo of 103.47.185.219

4.2. Preliminary Studies Before Practice

There are two types of options

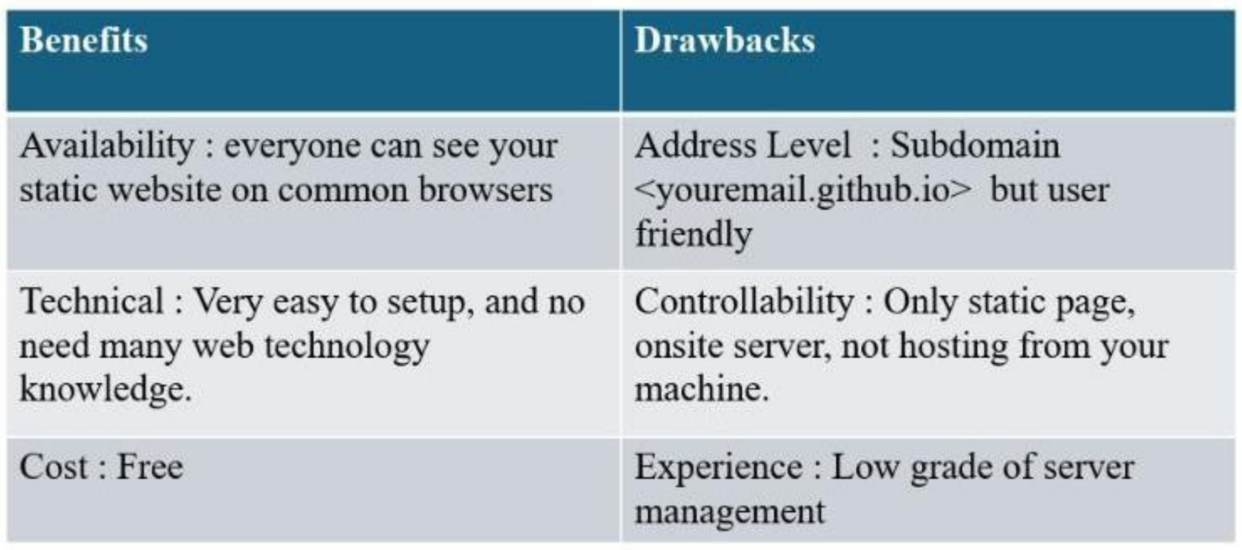
A. Public Network : Public Service Browser

B. Secret Network : Hidden Service Browser

Option (A) info

Platform : GitHub.com

Requirement : Email , GitHub account, a browser app, an authenticator app.



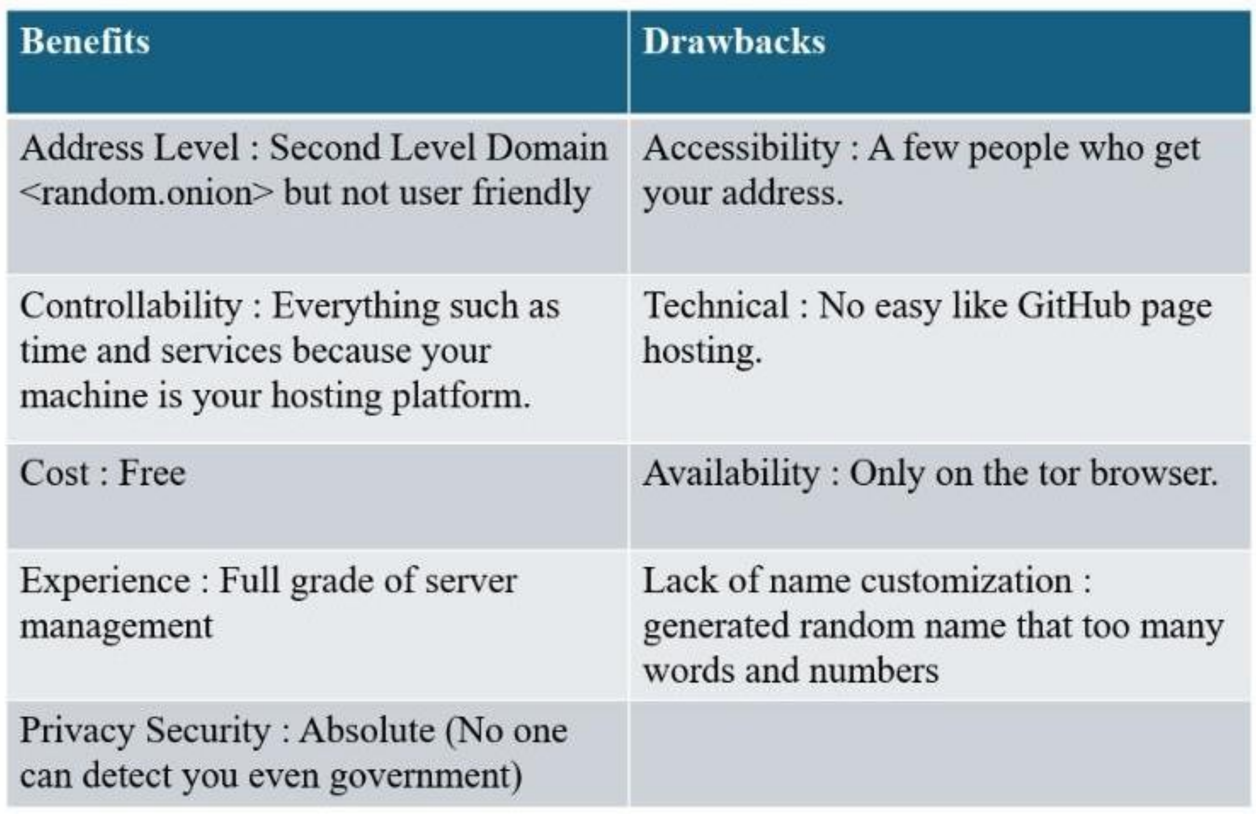
Benefits	Drawbacks
Availability : everyone can see your static website on common browsers	Address Level : Subdomain <youremail.github.io> but user friendly
Technical : Very easy to setup, and no need many web technology knowledge.	Controllability : Only static page, onsite server, not hosting from your machine.
Cost : Free	Experience : Low grade of server management

Table 4.1. Option (A) : Benefits and Drawbacks

Option (B) info

Platform : Your machine, performs as dedicated server. (recommended mobile phone)

Requirement : a device (mobile), terminal app (termux), tor libraries, Tor browser app, server program.



Benefits	Drawbacks
Address Level : Second Level Domain <random.onion> but not user friendly	Accessibility : A few people who get your address.
Controllability : Everything such as time and services because your machine is your hosting platform.	Technical : No easy like GitHub page hosting.
Cost : Free	Availability : Only on the tor browser.
Experience : Full grade of server management	Lack of name customization : generated random name that too many words and numbers
Privacy Security : Absolute (No one can detect you even government)	

Table 4.2. Option (B) : Benefits and Drawbacks

4.3. Instructions for Option (A)

Step 1: Create a New GitHub Repository

- Go to www.github.com and log in. If account not had been, sign up with your email (example : ponpon2000@gmail.com)
- Click on the "+" button in the top-right corner and select "New repository".
- Enter a repository name with your email name and “.guthub.io” extention (e.g., “ponpon2000.github.io”). Structure :| <youremailname>.github.io | . Note that you can give the name as you like any name but there will be not webpage name with example pattern.

- Set the repository to Public (GitHub Pages does not support free private repositories).
- Check the box "Add a README file" (optional, but helps to initialize the repository).

- Click "Create repository".

Step 2: Upload Your Website Files

- Open the newly created repository.
- Click on "Add file" > "Upload files".
- Drag and drop your website files (e.g., index.html, asset/, etc.). To create folder on repository "folder_name/" . There will auto create folder and let this empty txt file temporarily. And then you can add your files under this folder.
- Click "Commit changes" to save them.

Important: Ensure that your homepage file is named index.html because GitHub Pages automatically looks for this file.

Step 3: Enable GitHub Pages

- In your repository at first, and then go to "Settings".
- In the left sidebar, look for "Pages", click on "Pages".
- Under "Branch", select "main" (or master if that is your default branch). Now, you must select main.
- Click "Save".

Your website will be live at: https://your-github-username.github.io/

Step 4: Verify Your Website is Live

- Visit your GitHub Pages URL: https://your-github-username.github.io/

Final Notes

⬛ No Git or terminal commands needed.

⬛ Fully managed via GitHub website.

🟩 Custom subdomain requires DNS setup (if using an external domain).

4.4. Instructions for Option (B)

Step 1: Install Required Packages in Termux

- First, update Termux and install Python and Tor.

- Type “pkg update && pkg upgrade -y pkg install tor -y” (If you face any error, skip this step) and then type “pkg install tor -y”.

Step 2: Create a Web Server and Website folder

- In this case, you had already server.py since chapter 3,
- Create a directory for your project . Before creating, check file path at first , type “pwd” . The answer must be “/data/data/com.termux/files/home”
- Type “mkdir website”.
- Type “whoami” , you will get your <username> . Copy that and path on note.

Step 3: Set File Owner and Permission

- Type “chown -R <username> :<username> /data /data /com.termux /files /home /website”.
- Type “chmod -R 700 /data/data/com.termux/files/home/website”.

Step 4: Open server.py file and Edit directory source

- Replace directory with “/data/data/com.termux/files/home/website/”. After that, save it.

Step 5: Move Resources

- “mv   /storage/emulated/0/website   /data/data/com.termux/files/home/website”.
- Pattern : “mv” + space + “initial_path” + space + “destination_path”.

Step 6: Start the Local Server

- Run the Python web server: (previous chapter had covered this step but location is changed so use below command)
- Type “python3 /data/data/com.termux/files/home/website/server.py”.

Your local server should now be running at:

http://127.0.0.1:2025

Step 7: Configure Tor Hidden Service

- At first , open new session (drag from screen left margin to right and click “+” from right buttom) in termux, go to “cd /data/data/com.termux/files/usr/etc/tor”.
- Open “nano torrc”.

- Add the following lines at the end of the file:

“HiddenServiceDir /data/data/com.termux/files/home/website/”

“HiddenServicePort 80 127.0.0.1:2025”

- Save and exit. (Ctrl + s, Ctrl + x)

Step 8: Start Tor Service

- Type “tor”

Once Tor starts, find your .onion domain:

Step 9: Look and Copy the hostname

- At first open new session in termux, and then type “cd”.
- Go to “cd website”.
- Look “cat hostname”. And copy that text end with “.onion”. It is your onion domain.

You will see an address like:

“Gdryhvji643dfhhhii8875fchjji9abc123xyz.onion” .

Step 10: Access the Server on Tor Browser

- Open Tor Browser.
- Enter your .onion address.

Your local server should now be accessible via the Tor network.

Step 7: Keep the Server Running (Optional , don’t apply, just for knowledge)

- To keep the server running even after closing Termux, following command is helpful.
- Use “nohup python3 /data/data/com.termux/files/home/website/erver.py & nohup tor”

Step 8 : To check running processes:

- Use “ps aux | grep python3 ps aux | grep tor”

Final Notes

🟩 Your .onion site is now live on Tor

✅ Works without port forwarding

⬛ Only accessible via Tor Browser

Summary

This chapter explores the process of deploying websites online. It introduces ICANN and the concept of domain names, providing historical context and explaining their importance. The chapter also covers preliminary considerations before selecting a hosting option and offers step-by-step instructions for two hosting methods—one deployment of remote (file hosting platform, version-control-system) and one dedicated inside own device as server IOT —allowing readers to choose the best option based on their needs.