**DEPARTMENT OF INFORMATION TECHNOLOGY**

**COMPUTER NETWORKING LAB**

**LAB1: 25/10/2022**

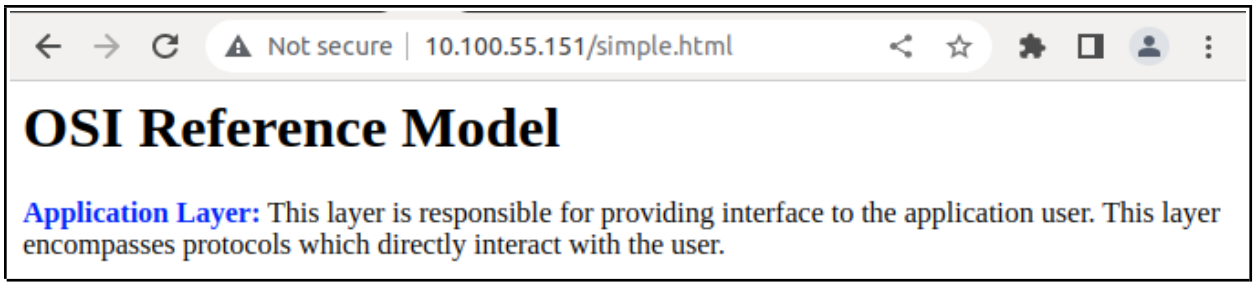
## Objective

To understand the concept web page development

Note: Add screenshots of the each step in the document and finally upload the same in Moodle.

**Evaluation: Q 3 to 10 for 1 Marks and Q11 is for 2 Marks : total 10 Marks.**

**The screen shots must be captured including IP address as follows**




If IP address is not mentioned in the screenshot then 0.5 marks will be deducted for that particular answer.

1. **Creating webpage in localhost in Ubuntu**

   First Install Apache with following command

   `sudo apt install` apache2

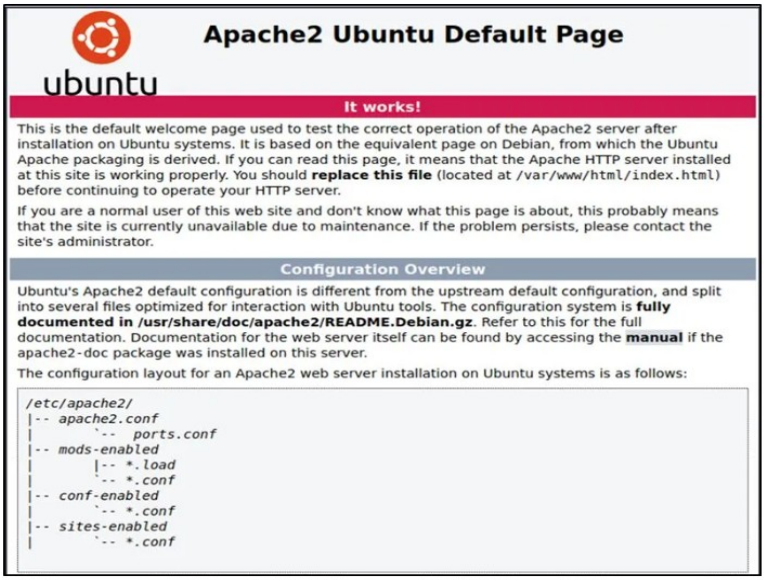
   Default index.html is stored in directory */var/www/html*

2. **Running First html program**

   Open the Browser and type as follows in address

   127.0.0.1

   You will be able to see the following html page




Now to create your own page either you can change the content in index.html or create new file filename.html. To access the page in browser type 127.0.0.1/filename.html

Note: If the *var*/www/html does not have permission to save files then in terminal change directory to *var*/www and type following command

**sudo chmod +777 html**

Then you will be able to save files as permission for same will be set through chmod.

3. **Write simple.html and browse the same [1 MARK]**

Type the following in html file, note down the content in rowser

```
<html>
 <head>
   <title>OSI Layers</title>
</head>
 <body>
Application Layer: This layer is responsible for providing interface to the application user. This layer encompasses protocols which directly interact with the user.
```

Presentation Layer: This layer defines how data in the native format of remote host should be presented in the native format of host.

Session Layer: This layer maintains sessions between remote hosts. For example, once user/password authentication is done, the remote host maintains this session for a while and does not ask for authentication again in that time span.

Transport Layer: This layer is responsible for end-to-end delivery between hosts.

Network Layer: This layer is responsible for address assignment and uniquely addressing hosts in a network.

Data Link Layer: This layer is responsible for reading and writing data from and onto the line. Link errors are detected at this layer.

Physical Layer: This layer defines the hardware, cabling wiring, power output, pulse rate etc.

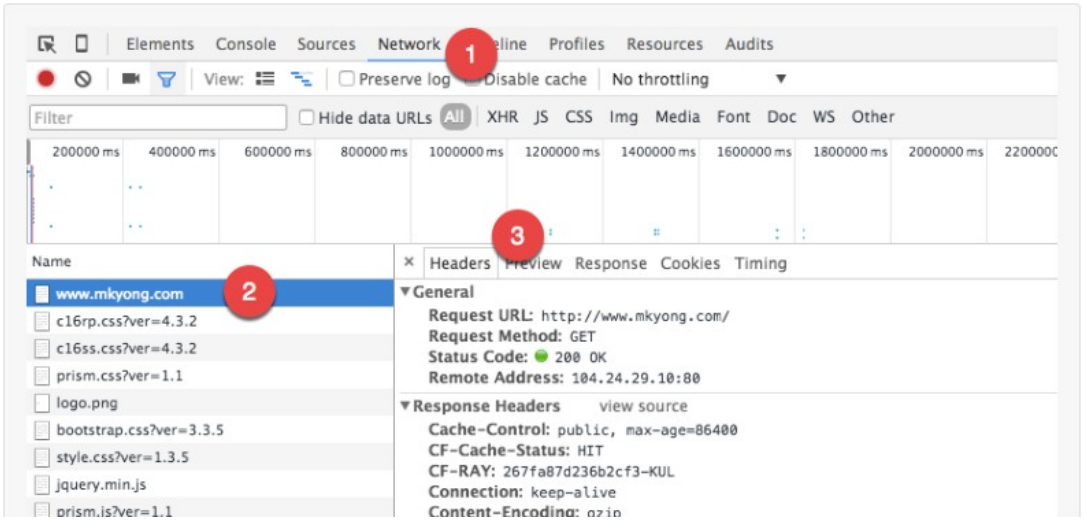
</body>

</html>

Record the result by accessing page as yoursystemipaddress/simple.html through web browser and write observation. (attach screenshot along with ip address )

To view the request or response HTTP headers in Google Chrome, take the following steps :

1. In Chrome, visit a URL, `right click`, select `Inspect` to open the developer tools.
2. Select `Network` tab.
3. Reload the page, select any HTTP request on the left panel, and the HTTP headers will be displayed on the right panel.



4. For the details of each layer attach the tag  and  at the beginning and end of the line.

   For example:

   ** **Application Layer: This layer is responsible for providing interface to the application user. This layer encompasses protocols which directly interact with the user.** **

   **Record the result through screenshot and explain the function of  and  [1 MARK]**

5. Use Bold  and  as follows and record the result and explain. **[1 MARK]**

   ** **Application Layer: **** This layer is responsible for providing interface to the application user. This layer encompasses protocols which directly interact with the user.** **

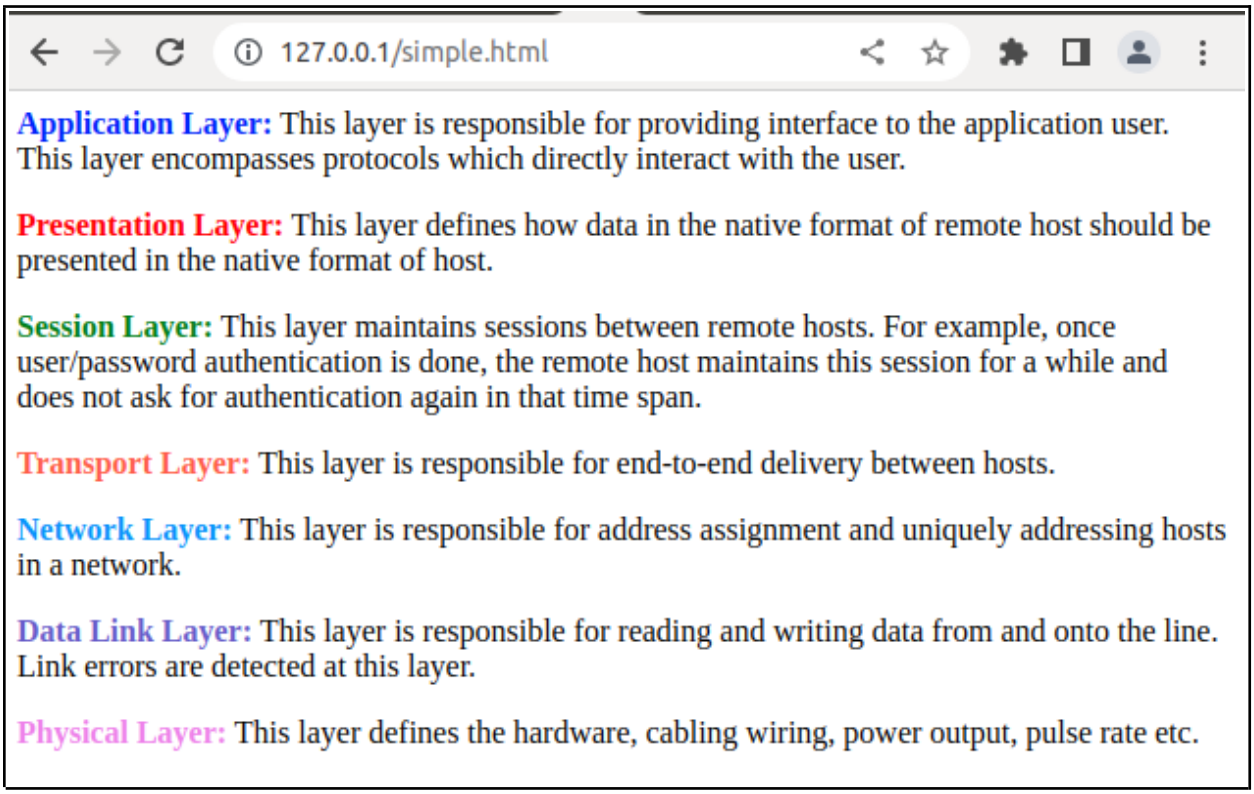
6. CSS stands for Cascading Style Sheets. It can control multiple web pages all at once. With CSS, you can control the color, font, the size of text, the spacing between elements, how elements are positioned and laid out, what background images or background colors are to be used, different displays for different devices and screen sizes, and much more!

   To change color of the text use inline style attribute inside HTML element as follows and record results.

   Example:

    Application Layer: This layer is responsible for providing interface to the application user. This layer encompasses protocols which directly interact with the user.

   Use different colors for each layers: eg. tomato, DodgerBlue, SlateBlue, violet, red, blue, green **[1 MARK]**




### 7. Adding images [1 MARK]

<img> is used for adding images as follows. Download the OSI_model.jpg from moodle and save it in var/www/html

```
<img src="OSI_model.jpg" alt="OSI communication model" width="250" height="350">
```

Observe and record the result

**Application Layer:** This layer is responsible for providing interface to the application user. This layer encompasses protocols which directly interact with the user.

**Presentation Layer:** This layer defines how data in the native format of remote host should be presented in the native format of host.

**Session Layer:** This layer maintains sessions between remote hosts. For example, once user/password authentication is done, the remote host maintains this session for a while and does not ask for authentication again in that time span.

**Transport Layer:** This layer is responsible for end-to-end delivery between hosts.

**Network Layer:** This layer is responsible for address assignment and uniquely addressing hosts in a network.

**Data Link Layer:** This layer is responsible for reading and writing data from and onto the line. Link errors are detected at this layer.

**Physical Layer:** This layer defines the hardware, cabling wiring, power output, pulse rate etc.


**8. Insert Heading as follows and record the result [1 MARK]**

**<H1>OSI Reference Model </H1>**

 Application Layer: This layer is responsible for providing interface to the application user. This layer encompasses protocols which directly interact with the user.

**9. Use of Simple Button [1 MARK]**

Type following at the end of simple.html. Record and observe the result

```
<H3> Did you like the Information </H3>

<button type="button" onclick="alert('Thank You')">CLICK</button>
```

Following method can be used to open new page on button click

```
<H3> Did you like the Information </H3>

<button type="button" onclick="newpage()">CLICK</button>

<script>

function newpage(){

window.open("index.html");

}

</script>
```

**********

Open new page called index.html and write following and check what happen on button click

```
<!DOCTYPE html >

<html>

 <head>

   <title>index page</title>

</head>

 <body>

   <h1 style="color:tomato;">Thank you for visiting</h1>

 </body>

 </html>
```

**10. Using PHP for script writing: Write following in index.html [1 MARK]**

```
<!DOCTYPE html >
<html>
 <head>
    <title>index page</title>
</head>
 <body>
    <h1 style="color:tomato;">Thank you for visiting</h1>
<form action="welcome_get.php" method="get">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>
 </body>
 </html>
```

**Write following code in welcome_get.php**

```
<!DOCTYPE html >
<html>
 <head>
    <title>index page</title>
</head>
 <body>
    <h1 style="color:tomato;">Thank you for visiting</h1>

<form action="welcome_get.php" method="get">
Name: <input type="text" name="name"><br>
```

```
E-mail: <input type="text" name="email"><br>

<input type="submit">

</form>

 </body>

 </html>
```

**Browse simple.html and observe & record the results.**

**11. Exercise: [2 MARKS]**

Write a simple html code to read the details of a student and display it in new page when a button is clicked.

Fields of information to be collected : Register number, name, address, email and contact number.

Use colors, button, heading, etc for each field.