Diploma in

IT, Networking and Cloud

Module 3

Web Designing

Lab Manual

Table of contents

Learning outcomes

After completing this module a student will be able to:

1. create simple web pages using HTML5.
2. create styles of web pages using CSS.
3. create your own account in cloud and hosting.
4. configure embedded databases with different web pages using MongoDB.
5. design and develop dynamic websites using PHP.
6. make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

Learning outcome 1 - Able to create simple web pages using HTML5

After achieving this learning outcome, a student will be able to create simple web pages using HTML5. In order to achieve this learning outcome, a student has to complete the following:

Activities

1. Create HTML document using markup Tags in HTML editor (Notepad) (1 Hr)
2. Open/run the HTML file in a web browser to check the output (1Hr)
3. Modify above HTML document using heading – paragraphs (1Hr)
4. Modify the above HTML document using Line Breaks (1Hr)
5. Modify above HTML document using HTML Tags. (1Hr)
6. Create Text, Lists, Tables, and Frames (2Hrs)
7. Create Hyperlinks, Images and Multimedia Working with Forms and controls. (3 Hrs)

Activity 1

Aim: Create an HTML document using markup Tags in HTML editor (Notepad)

Learning outcome: Able to create simple web pages using HTML 5.

Duration: 1 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Editor (Default GEDIT) and a Browser

Program 1:

```
<!DOCTYPE html>
<html>
<head>                          <!-- Document Head Starts -->
    <title>Markup Tags</title>
</head>                 <!-- Document Head Ends -->
<body>                          <!-- Document Body Starts -->
    <p>My First Paragraph</p>                   <!--Paragraph element -->
</body>                 <! -- Document Body Ends -->
</html>
```

Output/Results snippet:

- A simple HTML webpage with the body.

Program 2:

```
<!DOCTYPE html>
<html>
<head>                          <!-- Document Head Starts -->
    <title>Markup Tags</title>
</head>                 <!-- Document Head Ends -->
<body>                          <!-- Document Body Starts -->
    <center>
            <font size="20">My Web Page</font> <!--Font element -->
    </center>
</body>                 <! -- Document Body Ends -->
```

</html>

Output/Results snippet:

Example -

- A simple HTML webpage with the body.

References:

- HTML Introduction - https://www.w3schools.com/html/html_intro.asp

Activity 2

Aim: Open/run the HTML file in a web browser to check the output.

Learning outcome: Able to create simple web pages using HTML 5.

Duration: 1 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. A Browser

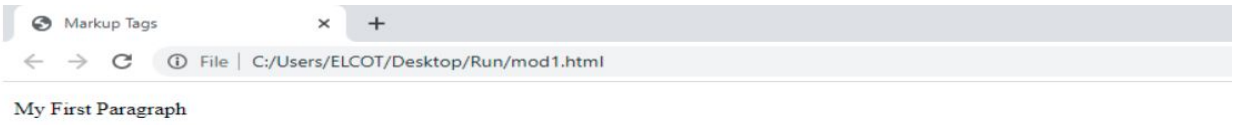
Steps to run the HTML file in a web browser:

1. Press "Windows-E" to launch Windows Explorer.
2. Navigate to the folder that contains your HTML file.

3. Double-click the file. Your default browser displays the HTML document. If the browser is not open, Windows launches it.

Output/Results snippet:

- A simple HTML webpage with the body



References:

- HTML Introduction - https://www.w3schools.com/html/html_intro.asp

Activity 3

Aim: Modify the above HTML document using heading – paragraphs.

Learning outcome: Able to create simple web pages using HTML 5.

Duration: 1 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Editor (Default GEDIT) and a Browser

Program:

```
<!DOCTYPE html>
<html>
<head>                          <!-- Document Head Starts -->
    <title> Heading &Paragraph</title>
</head>                 <!-- Document Head Ends -->
<body bgcolor="lightgreen" >          <!-- Document Body Starts -->
    <h1>My First Heading</h1>      <!--This is a heading-->
    <h2>My First Heading</h2>
    <h3>My First Heading</h3>
    <h4>My First Heading</h4>
    <h5>My First Heading</h5>
    <h6>My First Heading</h6>
    <p>                          <!--Paragraph element -->
            HTML is a Hypertext Mark-up Language file format used as the basis of a web page.
    </p>

</body>                 <! -- Document Body Ends -->
</html>
```

Output/Results snippet:

- A simple HTML webpage with the body.

References:

- HTML Introduction - https://www.w3schools.com/html/html_intro.asp

Activity 4

Aim: Modify the above HTML document using Line Break

Learning outcome: Able to create simple web pages using HTML 5.

Duration: 1 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Editor (Default GEDIT) and a Browser

Program:

```
<!DOCTYPE html>
<html>
<head>                          <!-- Document Head Starts -->
    <title> Heading Paragraph</title>
</head>                 <!-- Document Head Ends -->
<body >                 <!-- Document Body Starts -->
<p>                     <! – Paragraph Element with break -->
    To break lines</br>in a text,</br>use the br element.
</p>
</body>     <! -- Document Body Ends -->
</html>
```

Output/Results snippet:

- A simple HTML webpage with the body.

References:

- HTML Introduction - https://www.w3schools.com/html/html_intro.asp

Activity 5

Aim: Modify the above HTML document using HTML Tags.

Learning outcome: Able to create simple web pages using HTML 5.

Duration: 1 hour

List of Hardware/Software requirements:

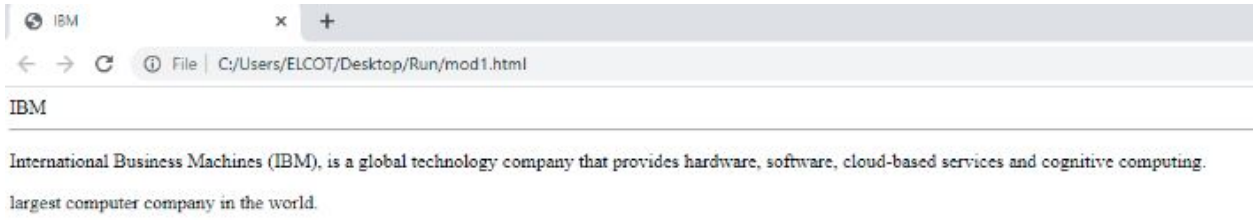
1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Editor (Default GEDIT) and a Browser

Program:

```
<!DOCTYPE html>
<html>
<head>                          <!-- Document Head Starts -->
    <title> IBM </title>
</head>             <!-- Document Head Ends -->
<body >             <!-- Document Body Starts -->
<h1>IBM</h1>                    <!--Heading Tag -->
<hr>                            <!--Horizontal Rule -->
<p>                 <!--Paragraph Element -->
International Business Machines (IBM), is a global technology company that provides hardware,
    software, cloud-based services and cognitive computing. </p>
<p>largest computer company in the world.</p>
</body>     <! -- Document Body Ends -->
</html>
```

Output/Results snippet:

- A simple HTML webpage with the body.



References:

- HTML Introduction - https://www.w3schools.com/html/html_intro.asp

Activity 6

Aim: Create Text, Lists, Tables, and Frames

Learning outcome: Able to create simple web pages using HTML 5.

Duration: 2 hours

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Editor (Default GEDIT) and a Browser

Program:

Texts

```
<!DOCTYPE html>
<html>
<head>                          <!-- Document Head Starts -->
    <title> Text,Lists,Tables and Frames </title>
</head>                 <! -- Document Head Ends -->
<body >                 <!-- Document Body Starts -->

<b>Bold text.</b><br>       <!-- This  text is Bold -->

<i>Italic text.</i><br>     <!-- This  text is Italic -->

<pre>Preformatted text     <!-- This  text is Preformatted text →

displays just

as you

WEB DESIGNING

5 | P a g e

type it...

...line breaks,

spaces...

...and all!

</pre>

<tt>Teletype text - sometimes referred to as typewriter text.</tt><br><!-- This  text is Teletype Text
    -->
```

```
<cite>Citation text.</cite><br>        <!-- This  text is Citation -->
<em>Emphasized text.</em><br>   <!-- This  text is Emphasize -->
<code>Code text.</code><br>        <!-- This  text is Codetext -->
<big>Big text.</big><br>              <!-- This  text is Big -->
<small>Small text.</small><br>      <!-- This  text is Small -->
<del>Delete this text.</del><br>     <!-- This  text is Delete Text -->
<ins>Insert this text.</ins><br>      <!-- This  text is Insertion -->
<kbd>Keyboard text - text to be entered by the user.</kbd><br>       <!-- This  text is Keyboard text
    -->
<q>Quotation text.</q><br>          <!-- This  text is Quotation text -->
<samp>Sample text (output from a computer program).</samp><br><!-- This  text is Sample Text
    -->
<var>Variable text.</var><br>       <!-- This  text is Variable Text -->
<dfn>Definition.</dfn>: To define the meaning of a word, phrase or term. <! -- This text is Document
    Type Definition -->
</body>      <! -- Document Body Ends -->
</html>
```

Tables:

```
<!DOCTYPE html>
<html>
<head>                          <!-- Document Head Starts -->
    <title> Tables </title>
</head>               <! -- Document Head Ends -->
<body >               <!-- Document Body Starts -->
<table class="example3" border="2">          <!--Table Tag Starts -->
<tr>                            <!--Table Row Starts -->
<th colspan="2">Table header</th> <!--This is Table Head -->
</tr>                           <!--Table Row Ends -->
<tr>                            <!--Table Row Starts -->
            <td width="20%">Table cell 1</td><td>Table cell 2</td>
```

```
</tr>                          <!--Table Row Ends -->
</table><br><!--Table Tag Ends -->
<table class="example4" border="2"> <!--Table Tag Starts -->
<tr>                           <!--Table Row Starts -->
<th rowspan="2" width="25%">Table header</th><!--This is Table Head -->

<td width="30%">Table cell 1</td><!--This is Table Data -->
</tr>                          <!--Table Row Ends -->
    <tr>                       <!--Table Row Starts -->
<td width="30%">Table cell 2</td><!--This is Table Data -->
</tr>                          <!--Table Row Ends -->
</table>     <!—Table Tag Ends -->
</body>      <! -- Document Body Ends -->
</html>
```

Lists

```
<!DOCTYPE html>
<html>
<head>              <!-- Document Head Starts -->
    <title> Lists </title>
</head>             <! -- Document Head Ends -->
<body>              <!-- Document Body Starts -->
<ol>                           <!-- OrderedList Starts -->
<li>List item 1</li> <!--List of Items -->
<li>List item 2</li>
<li>List item 3</li>
</ol>                          <! -- OrderedList Ends -->
<ul>                           <!-- UnOrderedList Starts -->
<li>List item 1</li> <!--List of Items -->
```

List item 2

List item 3

 <! -- UnOrderedList Ends --

<dl> <!—This is Definition List Starts-->

<dt>Term 1</dt><!—This is Definition Term -->

<dd>Definition of term 1</dd><!—This is Definition Data -->

<dt>Term 2</dt><!—This is Definition Term -->

<dd>Definition of term 2</dd> <!—This is Definition Data -->

</dl> <!--This is Definition List Ends-->

</body> <! -- Document Body Ends -->
</html>

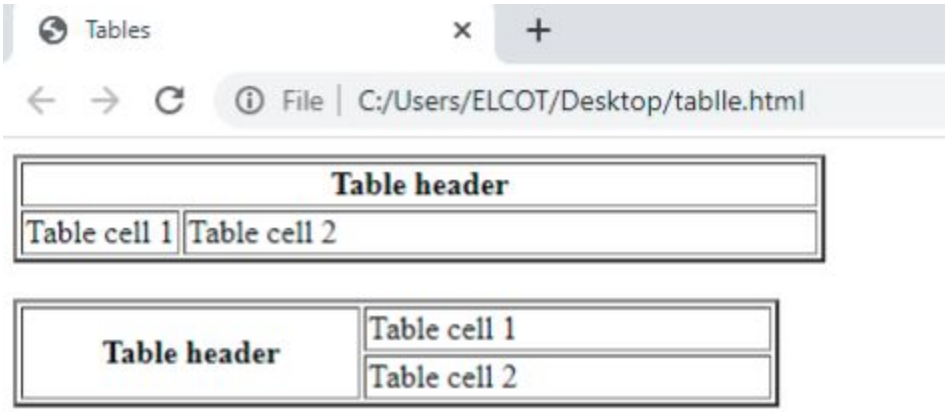
Frames

```
<!DOCTYPE html>
<html>
<head>                <! -- Document Head Starts -->
    <title>Lists</title>
</head>               <! -- Document Head Ends -->
<frameset cols="25%,50%,25%">   <! -- Frameset Starts -->
<frame src="frame_a.htm" >                  <!--Frame element -->
 <frame src="frame_b.htm">                  <! -- Frame element -->
 <frame src="frame_c.htm">                  <! -- Frame element -->
 </frameset>                  <! -- Frameset Ends -->
</html>
```

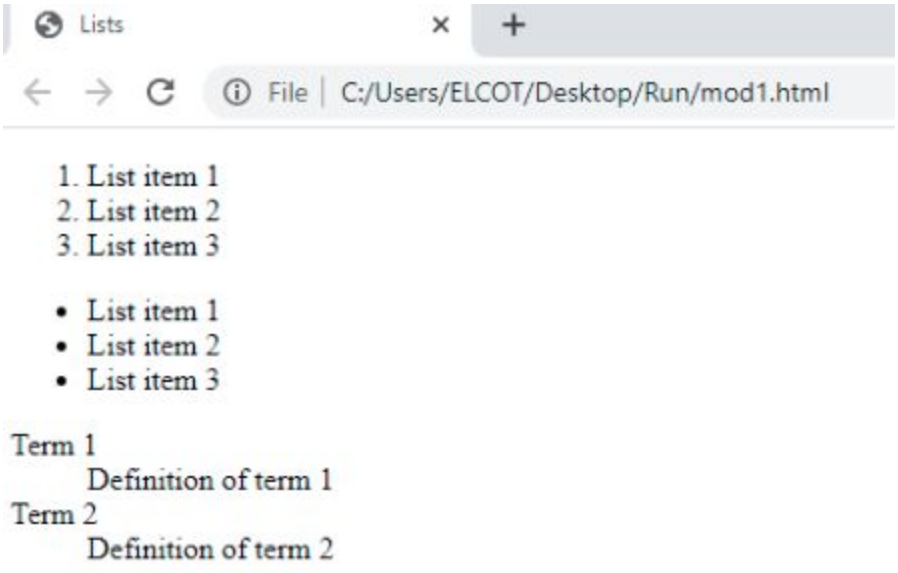
Output/Results snippet:

- A simple HTML webpage with the body Texts.

- A simple HTML webpage with the body Tables.



- A simple HTML webpage with the body Lists



- A simple HTML webpage with the body Frames

References:

- HTML Introduction - https://www.w3schools.com/html/html_intro.asp

Activity 7

Aim: Create Hyperlinks, Images and Multimedia Working with Forms and controls.

Learning outcome: Able to create simple web pages using HTML 5.

Duration: 3 hours

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Editor (Default GEDIT) and a Browser

Program:

Hyperlinks:

```
<!DOCTYPE html>
<html>
<head>                          <!-- Document Head Starts -->

    <title> Links with Paragraph</title>

</head>                 <! -- Document Head Ends -->
<body>                          <!-- Document Body Starts -->

    <a href="https://www.google.com">Welcome to HTML</a></br>

    <!--This is Hyperlink -->

    <p>Locked in a frame? <a href="https://edunetworld.com/" target="_top">Click here!</a></p>
    ><!--This is Paragraph with Hyperlink -->

</body>                 <! -- Document Body Ends -->
</html>
```

Image as a Link

```
<!DOCTYPE html>
<html>
<head>                          <!-- Document Head Starts -->

    <title> Links with Paragraph</title>

</head>                 <! -- Document Head Ends -->
<body>                          <!-- Document Body Starts -->

<a href="http://www.google.com">

 <img src="../smiley.png" alt="HTML tutorial"></a><!--This is image with Hyperlink -->
```

```
</body>                <! -- Document Body Ends -->
</html>
```

Multimedia

```
<!DOCTYPE html>
<html>
<head>                          <!-- Document Head Starts -->
    <title> Links with Paragraph</title>
</head>                <!-- Document Head Ends -->
<body>                          <!-- Document Body Starts -->
    <video  width = "300" height = "200" controls autoplay> <!-- video Element Starts-->
        <source src = "../video.mp4" type="video/mp4" />Your browser does not support the
    <video> element.
    </video>          <!-- video Element Ends-->


</body>                <! -- Document Body Ends -->
</html>
```

Forms and Controls

```
<!DOCTYPE html>
<html>
<head>                          <!-- Document Head Starts -->
    <title> Forms and Controls</title>
</head>                <!-- Document Head Ends -->
<body>                          <!-- Document Body Starts -->
    <form> <!-- Form Element Starts -->
<table cellpadding="2" width="20%" bgcolor="99FFFF" align="center" cellspacing="2"> <!--Table
    Tag Starts -->
    <tr>                <!--Table Row Starts -->
    <td colspan=2> <!--This is Table Data -->
    <center><font size=4><b>Student Registration Form</b></font></center><!--This is font-->
    </td>
    </tr>               <!--Table Row Ends -->
```

```
<tr>                <!--Table Row Starts -->
<td>Name</td><!--This is Table Data -->
<td><input type=”text” name=textnames id="textname" size="30"></td><!--This is Textbox-->
</tr><!--Table Row Ends -->
<tr><!--Table Row Starts -->
<td>Father Name</td><!--This is Table Data -->
<td><input type="text" name="fathername" id="fathername" size="30"></td><!--This is Textbox-->
</tr><!--Table Row Ends -->
<tr><!--Table Row Starts -->
<td>Postal Address</td><!--This is Table Data -->
<td><input type="text" name="paddress" id="paddress" size="30"></td><!--This is Textbox-->
</tr><!--Table Row Ends -->
<tr><!--Table Row Starts -->
<td>Personal Address</td><!--This is Table Data -->
<td><input type="text" name="personaladdress" id="personaladdress" size="30"></td>><!--This is Textbox-->
</tr><!--Table Row Ends -->
<tr><!--Table Row Starts -->
<td>Sex</td><!--This is Table Data -->
<td><input type="radio" name="sex" value="male" size="10">Male
<!--This is Table Data -->
<!--Input type is radio-->
<input type="radio" name="sex" value="Female" size="10">Female</td>
</tr><!--Table Row Ends -->
<tr><!--Table Row Starts -->
<td>City</td><!--This is Table Data --><! This is Dropdown List-->
<td><select name="City">
<option value="-1" selected>select..</option><!--This is option for Cities-->
```

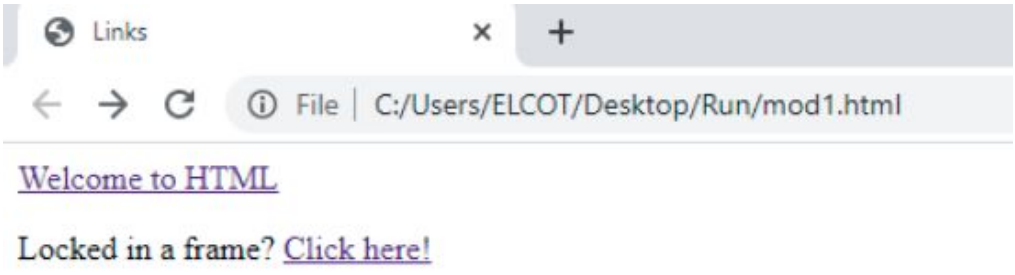
```
<option value="New Delhi">NEW DELHI</option>
<option value="Mumbai">MUMBAI</option>
<option value="Goa">GOA</option>
<option value="Patna">PATNA</option>
</select></td>
</tr><!--Table Row Ends -->
    <tr><!--Table Row Starts -->
<td>Course</td><!--This is Table Data -->
<td><select name="Course"><! This is Dropdown List-->
<option value="-1" selected>select..</option><!--This is option for Course-->
<option value="B.Tech">B.TECH</option>
<option value="MCA">MCA</option>
<option value="MBA">MBA</option>
<option value="BCA">BCA</option>
</select></td>
</tr><!--Table Row Ends -->
    <tr><!--Table Row Starts -->
<td>District</td><!--This is Table Data -->
<td><select name="District"><! This is Dropdown List-->
<option value="-1" selected>select..</option>><!--This is option for Districts-->
<option value="Nalanda">NALANDA</option>
<option value="UP">UP</option>
<option value="Goa">GOA</option>
<option value="Patna">PATNA</option>
</select></td>
</tr><!--Table Row Ends -->
<tr><!--Table Row Starts -->
<td>State</td><!--This is Table Data -->
```

```
<td><select Name="State"><! This is Dropdown List-->
<option value="-1" selected>select..</option><!--This is option for States-->
<option value="New Delhi">NEW DELHI</option>
<option value="Mumbai">MUMBAI</option>
<option value="Goa">GOA</option>
<option value="Bihar">BIHAR</option>
</select></td>
</tr><!--Table Row Ends -->
<tr><!--Table Row Starts -->
<td>PinCode</td><!--This is Table Data -->
    <td><input type="text" name="pincode" id="pincode" size="30"></td><!—This is Textbox-->
    </tr><!--Table Row Ends -->
    <tr><!--Table Row Starts -->
<td>EmailId</td><!--This is Table Data -->
    <td><input type="text" name="emailid" id="emailid" size="30"></td><!—This is Textbox-->
    </tr><!--Table Row Ends -->
    <tr><!--Table Row Starts -->
<td>DOB</td><!--This is Table Data -->
    <td><input type="text" name="dob" id="dob" size="30"></td><!—This is Textbox-->
</tr><!--Table Row Ends -->
    <tr><!--Table Row Starts -->
<td>MobileNo</td><!--This is Table Data -->
    <td><input type="text" name="mobileno" id="mobileno" size="30"></td><!—This is
    Textbox-->
</tr><!--Table Row Ends -->
<tr><!--Table Row Starts -->
<td><input type="reset"></td><!--This is Table Data  & Reset button-->
<td colspan="2"><input type="submit" value="Submit Form" /></td><!--submit button-->
```

```
</tr><!--Table Row Ends -->
</table> <!--Table Tag Ends-->
</form> <! -- Form Element Ends -->
</body>                <! -- Document Body Ends -->
</html>
```

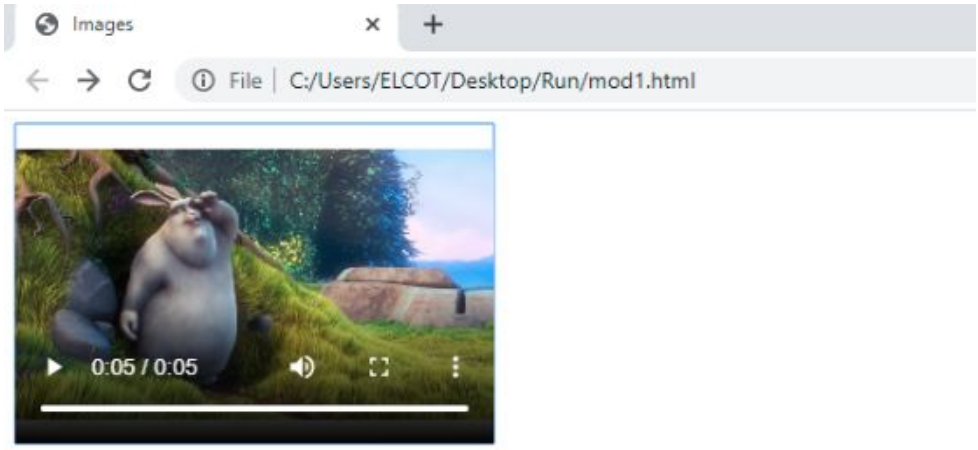
Output/Results snippet:

A simple HTML webpage with the body Links with Paragraph



A simple HTML webpage with the body Links with Image Link

A simple HTML webpage with the body video



A simple HTML webpage with the body Forms and Controls

References:

- HTML Introduction - https://www.w3schools.com/html/html_intro.asp

Learning outcome 2 - able to create styles of web pages using CSS

After achieving this learning outcome, a student will be able to create styles of web pages using CSS. In order to achieve this learning outcome, a student has to complete the following:

Activities

1. Create a CSS document by using an ID selector (1Hr)
2. Create CSS document by using Class selector, Universal selector and Grouping selector (1Hr)
3. Create CSS document with fonts : Bold, Italics, oblique (1 Hr)
4. Design Style sheet document with text transformation :Uppercase, Lowercase and capitalize (1Hr)
5. Create CSS document with font size in different pixels (1 Hr)
6. Create CSS document with font weight thinner, thicker, bold (1Hr)
7. Create CSS document with alignment centre, right and left (1Hr)
8. Create CSS document with background colours and font colours (1Hr)
9. Create CSS document with text hovering (1Hr)
10. Create CSS document with text decoration (1Hr)
11. Create CSS document with block elements and objects (1 Hr)
12. Create Lists and Tables (1 Hr)
13. Create Box Model by using borders, Padding and Margin (1Hr)
14. Create CSS document by Grouping, Dimension, Display, Positioning, Floating, Align, Pseudo class, NavigationBar, Image Sprites, Attribute sector (1Hr)
15. Creating page Layout and Site Designs. (1Hrs)

Activity 1

Aim: Create CSS document by using ID selector

Learning outcome: Able to create Styles of web pages using CSS.

Duration: 1 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Editor (Default GEDIT) and a Browser

Program:

```
<!DOCTYPE html>
<html>
<head>                          <!-- Document Head Starts -->
    <title> ID Selector </title>
<style>
/* Style the element with the id "myHeader" */

#myHeader {

background-color: lightblue;

color: black;

padding: 40px;

text-align: center; }

</style>

</head>              <!-- Document Head Ends -->
<body>                          <!-- Document Body Starts -->

<h1 id="myHeader">My Header</h1>  <!--This is Heading Tag with id myHeader-->

</body>              <! -- Document Body Ends -->
</html>
```

Output/Results snippet:

- A simple HTML webpage with the body CSS id Selector.

References:

- HTML Introduction - https://www.w3schools.com/html/html_intro.asp

Activity 2

Aim: Create CSS document by using Class selector,Universal selector Grouping selector

Learning outcome: Able to create Styles of web pages using CSS.

Duration: 1 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Editor (Default GEDIT) and a Browser

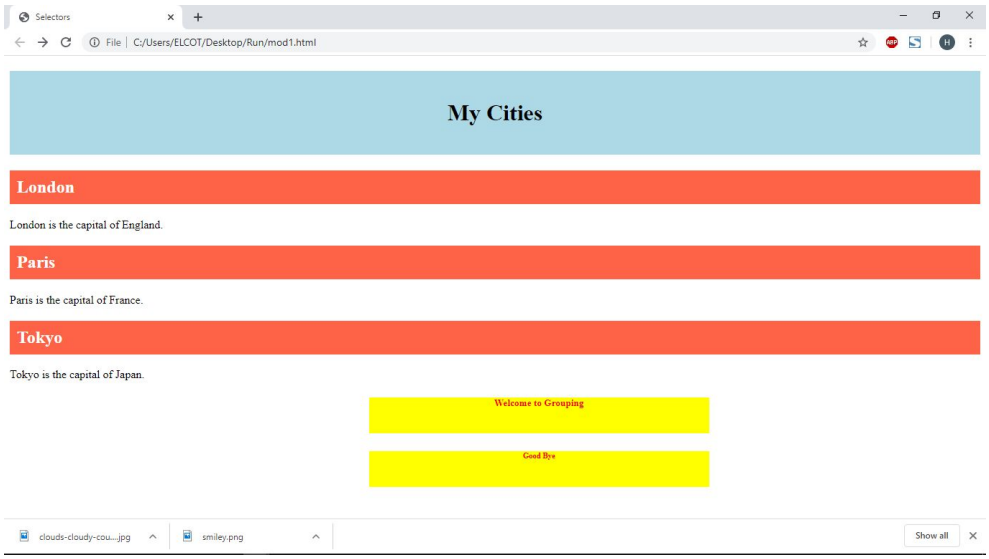
Program:

```
<!DOCTYPE html>
<html>
<head>                  <!-- Document Head Starts -->
    <title> Selectors </title>
<style>
/* Style the element with the id "myHeader" */
#myHeader {
background-color: lightblue;
color: black;
padding: 40px;
text-align: center;
}
/* Style all elements with the class name "city" */
.city {
background-color: tomato;
color: white;
padding: 10px;
}
/* Style all elements with the Group name "h5,h6" */
h5,h6
{
margin-left:500px;
color:red;
background-color:yellow;
width:35%;
height:50px;
text-align:center;
}
</style>
</head>                 <!-- Document Head Ends -->
<body>                          <!-- Document Body Starts -->
```

```
<!-- A unique element -->
<h1 id="myHeader">My Cities</h1>
<!-- Multiple similar elements -->
<h2 class="city">London</h2>
<p>London is the capital of England.</p>
<h2 class="city">Paris</h2>
<p>Paris is the capital of France.</p>
<h2 class="city">Tokyo</h2>
<p>Tokyo is the capital of Japan.</p>
<h5>Welcome to Grouping</h5>
<h6>Good Bye</h6>

</body>                <! -- Document Body Ends -->
</html>
```

Output/Results snippet:

- A simple HTML webpage with the body CSS Selectors.



References:

- HTML Introduction - https://www.w3schools.com/html/html_intro.asp

Activity 3

Aim: Create CSS document with fonts : Bold, Italics, oblique

Learning outcome: Able to create Styles of web pages using CSS.

Duration: 1 hour

List of Hardware/Software requirements:

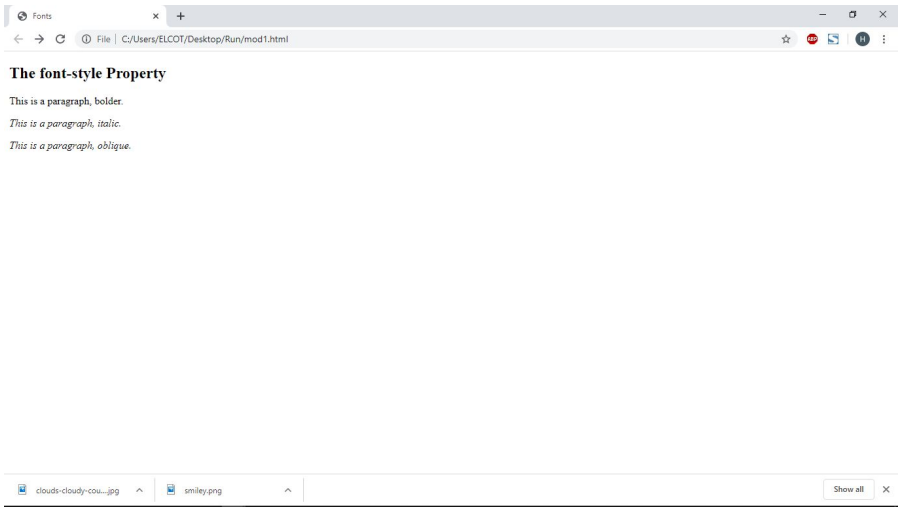
1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Editor (Default GEDIT) and a Browser

Program:

```
<!DOCTYPE html>

<html>

<head><!-- Document Head Starts -->

<title> Fonts</title>

<style>

/* Style all elements with the paragraph class name "p.a,p.b,p.c" */

p.a { font-style: bolder;}

p.b {font-style: italic;}

p.c {font-style: oblique;}

</style>

</head><!-- Document Head Ends -->

<body><!-- Document Body Starts -->
<h2>The font-style Property</h2> <!--This is Heading-->

<p class="a">This is a paragraph, bolder.</p> <!--Paragraph Element with bolder-->

<p class="b">This is a paragraph, italic.</p><!--Paragraph Element with Italic-->

<p class="c">This is a paragraph, oblique.</p><!--Paragraph Element with oblique -->

</body> <! -- Document Body Ends -->

</html>
```

Output/Results snippet:

- A simple HTML webpage with the body CSS Fonts.



References:

- HTML Introduction - https://www.w3schools.com/html/html_intro.asp

Activity 4

Aim: Design Style sheet document with text transformation :Uppercase, Lowercase and capitalize

Learning outcome: Able to create style of web pages using CSS.

Duration: 1 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Editor (Default Gedit or any other) and a Browser

Program:

```
<!DOCTYPE html>
    <html>
    <head>
    <style>
    /*  Style the element with the class "a"  */
    div.a {
    text-transform: uppercase;
    }
    /* Style the element with the class "b" */
    div.b {
    text-transform: lowercase;
    }
    /* Style the element with the class "c" */
    div.c {
    text-transform: capitalize;
    }
    </style>
    </head>
    <body>
    <h2>The text-transform Property</h2>
```

```
<h2>text-transform: uppercase:</h2>
<div class="a">Welcome to CSS</div>
<h2>text-transform: lowercase:</h2>
<div class="b">Welcome to CSS</div>
<h2>text-transform: capitalize:</h2>
<div class="c">Welcome to CSS</div>
</body>
</html>
```

Output/Results snippet:

- A Page with all the div has applied css

References:

- https://www.w3schools.com/html/html_intro.asp
- https://www.tutorialspoint.com/css/index.htm
- https://www.javatpoint.com/css-tutorial

Activity 5

Aim: Create CSS document with font size in different pixels

Learning outcome: Able to create style of web pages using CSS.

Duration: 1 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Editor (Default Gedit) and a Browser

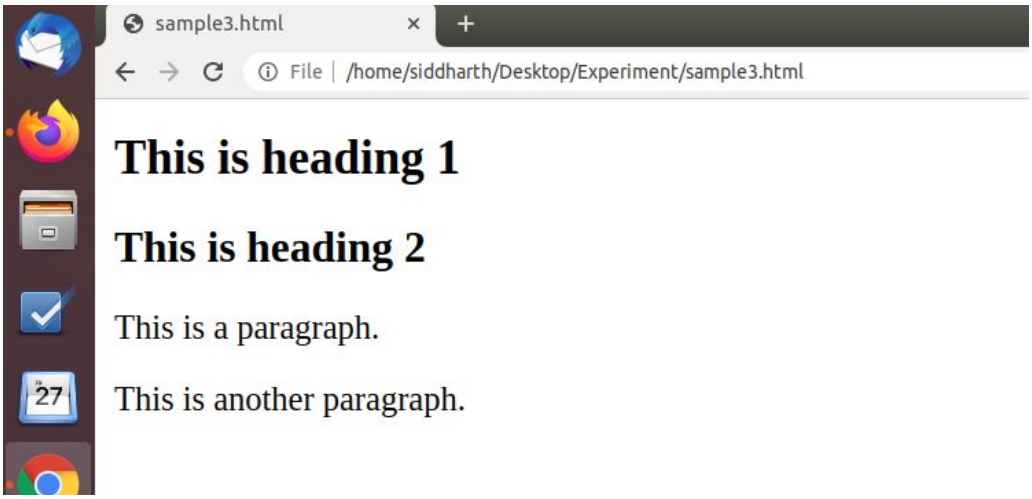
Program:

```
<!DOCTYPE html>
<html>
<head>
<style>
/*setting the font size of the heading tag h1*/
h1 {
font-size: 20px;
}
/*setting the font size of the heading tag h2*/
h2 {
font-size: 18px;
}
/*setting the font size of the paragraph*/
p {
font-size: 14px;
}
</style>
</head>
<body>
<h1>This is heading 1</h1>
```

```
<h2>This is heading 2</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
</body>
</html>
```

Output/Results snippet

- Applied css on heading and paragraph.



References:

- https://www.w3schools.com/html/html_intro.asp
- https://www.tutorialspoint.com/css/index.htm
- https://www.javatpoint.com/css-tutorial

Activity 6

Aim: Create CSS document with font weight thinner, thicker, bold

Learning outcome: Able to create style of web pages using CSS.

Duration: 1 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Editor (Default GEDIT) and a Browser

Program:

```
<!DOCTYPE html>
<html>
<head>
<style>
/*setting the font weight of the paragraph of class ".normal"*/
p.normal {
font-weight: normal;
}
/*setting the font weight of the paragraph of class ".light"*/
p.light {
   font-weight: lighter; }
/*setting the font weight of the paragraph of class ".thick"*/
p.thick {
font-weight: bold; }
/*setting the font weight of the paragraph of class ".thicker"*/
p.thicker {
font-weight: 900; }
</style>
</head>
<body>
```

```
<h2>The font-weight Property</h2>
<p class="normal">This is a paragraph.</p>
<p class="light">This is a paragraph.</p>
<p class="thick">This is a paragraph.</p>
<p class="thicker">This is a paragraph.</p>
</body>
</html>
```

Output/Results snippet

- Apply font-weight property to each paragraph.

References:

- https://www.w3schools.com/html/html_intro.asp
- https://www.tutorialspoint.com/css/index.htm
- https://www.javatpoint.com/css-tutorial

Activity 7

Aim: Create CSS document with alignment centre, right and left

Learning outcome: Able to create style of web pages using CSS.

Duration: 1 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Editor (default Gedit) and a Browser

Program:

```
<!DOCTYPE html>
<html>
<head>
<style>
/*setting the alignment of  the div of class ".a"*/
div.a {
text-align: center;
}
/*setting the alignment of  the div of class ".b"*/
div.b {
text-align: left;
}
/*setting the alignment of  the div of class ".c"*/
div.c {
text-align: right;
}
/*setting the alignment of  the div of class ".d"*/
div.d {
text-align: justify;
}
```

```
</style>
</head>
<body>
<h2>The text-align Property</h2>
<div class="a">
<h2>text-align: center:</h2>
<p>Welcome to alignment Welcome to alignment Welcome to alignment Welcome to alignment</p>
</div>
<div class="b">
<h2>text-align: left:</h2>
<p>Welcome to alignment Welcome to alignment Welcome to alignment Welcome to alignment</p>
</div>
<div class="c">
<h2>text-align: right:</h2>
<p>Welcome to alignment Welcome to alignment Welcome to alignment Welcome to alignment</p>
</div>
<div class="d">
<h2>text-align: justify:</h2>
<p>Welcome to alignment Welcome to alignment Welcome to alignment Welcome to alignment</p>
</div>
</body>
</html>
```

Output/Results snippet

- Alignment of the divisions with text-align property.

References:

- https://www.w3schools.com/html/html_intro.asp
- https://www.tutorialspoint.com/css/index.htm
- https://www.javatpoint.com/css-tutorial

Activity 8

Aim: Create CSS document with background colours and font colours

Learning outcome: Able to create style of web pages using CSS.

Duration: 1 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Editor (Default GEDIT) with a Browser

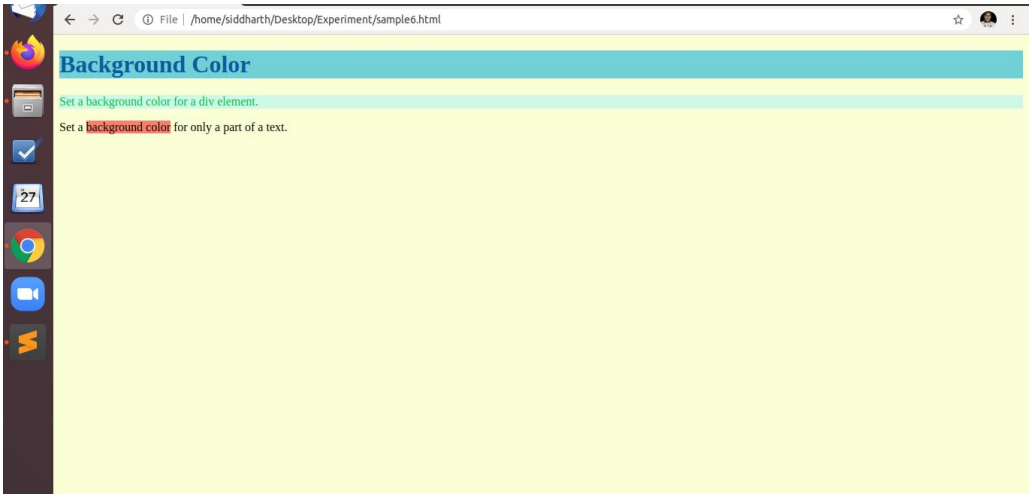
Program:

```
<!DOCTYPE html>
<html>
<head>
<style>
/*setting the background color of body */
body {
background-color: #fefbd8; }
/*setting the background color of  heading and also change its color */
h1 {
background-color: #80ced6;
color: #006699; }
/*changing the background color of  division with its color */
div {
background-color: #d5f4e6;
color: #4CAF50; }
/*setting the background color of  span tag */
span {
background-color: #f18973; }
</style>
</head>
```

```
<body>
<h1>Background Color</h1>
<div>Set a background color for a div element.</div>
<p>Set a <span>background color</span> for only a part of a text.</p>
</body>
</html>
```

Output/Results snippet

- Background color changes for heading , div and span.



References:

- https://www.w3schools.com/html/html_intro.asp
- https://www.tutorialspoint.com/css/index.htm
- https://www.javatpoint.com/css-tutorial

Activity 9

Aim: Create CSS document with text hovering

Learning outcome: Able to create style of web pages using CSS.

Duration: 1 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Editor (Default GEDIT) with a Browser

Program:

```
<!DOCTYPE html>
<html>
<head>
<style>
/*setting the background color on hover of the mouse */
a:hover {
background-color: yellow; }
</style>
</head>
<body>
<a href=”https://www.google.com”>google.com</a>
<a href=”https://www.wikipedia.org”>22ikipedia.org</a>
<p><b>Note:</b> The :hover selector style links on mouse-over.</p>
</body>
</html>
```

Output/Results snippet

- A mouse occurs when a mouse pointer is above link.

References:

- https://www.w3schools.com/html/html_intro.asp
- https://www.tutorialspoint.com/css/index.htm
- https://www.javatpoint.com/css-tutorial

Activity 10

Aim: Create CSS document with text decoration

Learning outcome: Able to create style of web pages using CSS.

Duration: 1 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Editor (default gedit) with Browser

Program:

```
<!DOCTYPE html>
<html>
<head>
<style>
/*setting the text decoration with overline property on h1 tag */
h1 {
text-decoration: overline; }
/*setting the text decoration with line-through property on h2 tag */
h2 {
text-decoration: line-through; }
/*setting the text decoration with underline property on h3 tag */
h3 {
text-decoration: underline; }
/*setting the text decoration with overline & underline property on h4 tag */
h4 {
text-decoration: underline overline; }
</style>
</head>
<body>
<h4>This is heading 1</h4>
```

```
<h3>This is heading 2</h3>
<h2>This is heading 3</h2>
<h1>This is heading 4</h1>
</body>
</html>
```

Output/Results snippet

Example -

- Decorate heading with line-through property.

References:

- https://www.w3schools.com/html/html_intro.asp
- https://www.tutorialspoint.com/css/index.htm
- https://www.javatpoint.com/css-tutorial

Activity 11

Aim: Create CSS document with block elements and objects

Learning outcome: Able to create style of web pages using CSS.

Duration: 1 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Editor (Default Gedit or any other) and a Browser

Program:

```
<!DOCTYPE html>
<html>
<body>
<div>Hello</div>
<div>World</div>
<p>The DIV element is a block element, and will start on a new line.</p>
<span>Hello</span>
<span>World</span>
<p>The SPAN element is an inline element, and will not start on a new line.</p>
<!-- setting the style i.e background color , color , padding  with inline css -->
<div style="background-color:black;color:white;padding:20px;">
<h2>London</h2>
<p>London is the capital city of England. It is the most populous city in the United
Kingdom, with a metropolitan area of over 13 million inhabitants.</p>
<p>Standing on the River Thames, London has been a major settlement for two millennia.</p>
</div>
<!-- setting the style i.e  color  with inline css -->
<h1>My <span style="color:red">Important</span> Heading</h1>
</body>
</html>
```

Output/Results snippet

- Design of a web page using div , span and CSS.

References:

- https://www.w3schools.com/html/html_intro.asp
- https://www.tutorialspoint.com/css/index.htm
- https://www.javatpoint.com/css-tutorial

Activity 12

Aim: Create Lists and Tables

Learning outcome: Able to create style of web pages using CSS.

Duration: 1 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Editor (Default Gedit or any other) and a Browser

Program:

```
<html>
<head>
<style>
/*setting the list style type of class a */
ul.a {
list-style-type: circle;
}
/*setting the list style type of class b */
ul.b {
list-style-type: square;
}
/*setting the list style type of class c */
ol.c {
list-style-type: upper-roman;
}
/*setting the list style type of class d */
ol.d {
list-style-type: lower-alpha;
}
/*setting the border collapse and width of table */
```

```
table {
border-collapse: collapse;
width: 100%;
}
/*setting alignment and padding of table heading and table data */
th, td {
text-align: left;
padding: 8px;
}
/*setting the background color property of table row  */
tr:nth-child(even) {background-color: #f2f2f2;}
</style>
</head>
<body>
<p>Example of unordered lists:</p>
<ul class="a">
<li>Coffee</li>
<li>Tea</li>
<li>Coca Cola</li>
</ul>
<ul class="b">
<li>Coffee</li>
<li>Tea</li>
<li>Coca Cola</li>
</ul>
<p>Example of ordered lists:</p>
<ol class="c">
<li>Coffee</li>
```

```
<li>Tea</li>
<li>Coca Cola</li>
</ol>
<ol class="d">
<li>Coffee</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>
<h2>Striped Table</h2>
<p>For zebra-striped tables, use the nth-child() selector and add a background-color to all
even (or odd) table rows:</p>
<table>
<tr>
<th>First Name</th>
<th>Last Name</th>
<th>Points</th>
</tr>
<tr>
<td>Peter</td>
<td>Griffin</td>
<td>$100</td>
</tr>
<tr>
<td>Lois</td>
<td>Griffin</td>
<td>$150</td>
</tr>
<tr>
```

<td>Joe</td>

<td>Swanson</td>

<td>$300</td>

</tr>

<tr>

<td>Cleveland</td>

<td>Brown</td>

<td>$250</td>

</tr>

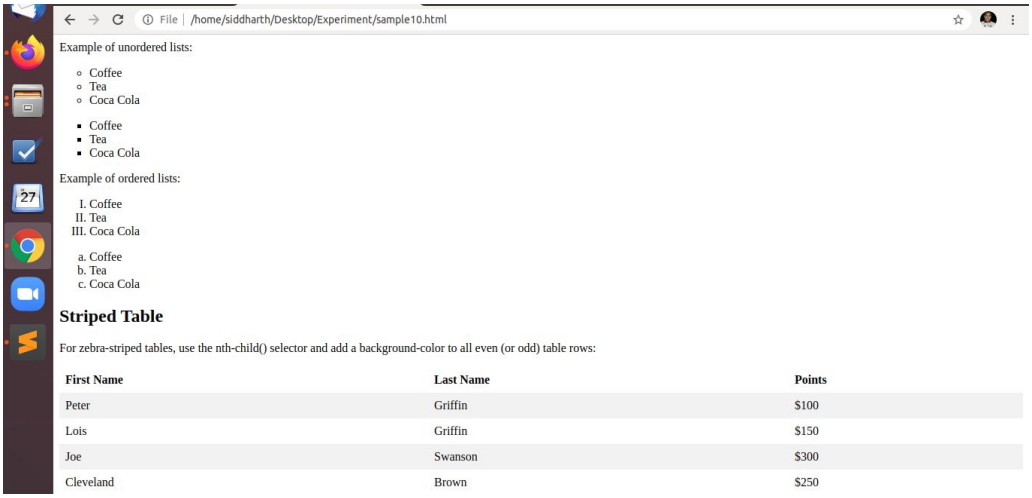
</table>

</body>

</html>

Output/Results snippet

- Designing of List and Tables using CSS.



References:

- https://www.w3schools.com/html/html_intro.asp
- https://www.tutorialspoint.com/css/index.htm
- https://www.javatpoint.com/css-tutorial

Activity 13

Aim: Create Box Model by using borders, Padding and Margin

Learning outcome: Able to create style of web pages using CSS.

Duration: 1 hour

List of Hardware/Software requirements:

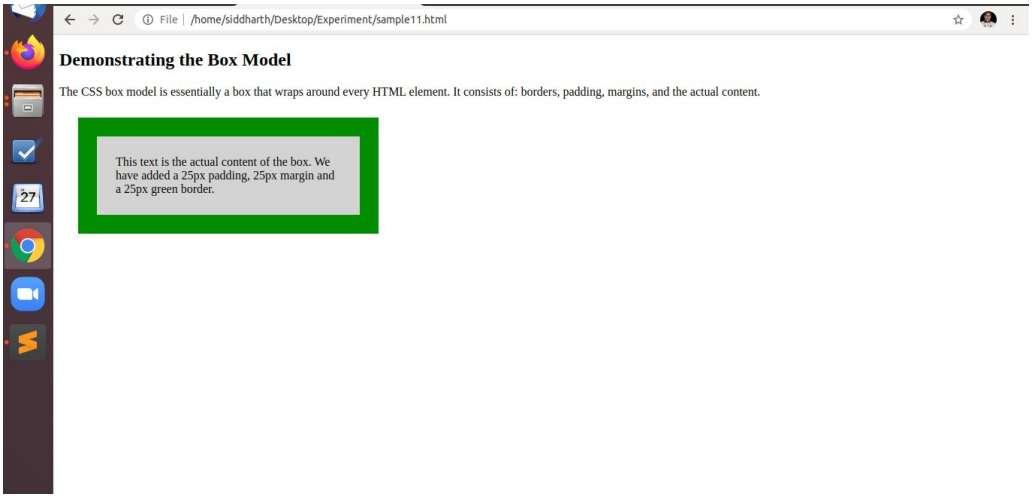
1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Editor (Default Gedit or any other) and a Browser

Program:

```
<!DOCTYPE html>
<html>
<head>
<style>
/*setting border padding and margin property */
div {
background-color: lightgrey;
width: 300px;
border: 25px solid green;
padding: 25px;
margin: 25px; }
</style>
</head>
<body>
<h2>Demonstrating the Box Model</h2>
<p>The CSS box model is essentially a box that wraps around every HTML element. It consists of:
    borders, padding, margins, and the actual content.</p>
<div>This text is the actual content of the box. We have added a 25px padding, 25px margin and a
    25px green border. </div>
</body>
</html>
```

Output/Results snippet

- Created a Box with border and color .



References:

- https://www.w3schools.com/html/html_intro.asp
- https://www.tutorialspoint.com/css/index.htm
- https://www.javatpoint.com/css-tutorial

Activity 14

Aim: Create CSS document by Grouping, Dimension, Display, Positioning, Floating, Align, Pseudo class, NavigationBar, Image Sprites, Attribute sector and Site Layout

Learning outcome: Able to create style of web pages using CSS.

Duration: 2 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Editor (Default Gedit or any other) and a Browser

Program:

```
<!DOCTYPE html>
<html>
<head>
<style>
* {
box-sizing: border-box;
}
body {
font-family: Arial;
padding: 10px;
background: #f1f1f1;
}
/* Header/Blog Title */
.header {
padding: 30px;
text-align: center;
background: white;
}
.header h1 {
```

```
font-size: 50px;
}
/* Style the top navigation bar */
.topnav {
overflow: hidden;
background-color: #333;
}
/* Style the topnav links */
.topnav a {
float: left;
display: block;
color: #f2f2f2;
text-align: center;
padding: 14px 16px;
text-decoration: none;
}
/* Change color on hover */
.topnav a:hover {
background-color: #ddd;
color: black;
}
/* Create two unequal columns that floats next to each other */
/* Left column */
.leftcolumn {
float: left;
width: 75%;
}
/* Right column */
```

```
.rightcolumn {
float: left;
width: 25%;
background-color: #f1f1f1;
padding-left: 20px;
}
/* Fake image */
.fakeimg {
background-color: #aaa;
width: 100%;
padding: 20px;
}
/* Add a card effect for articles */
.card {
background-color: white;
padding: 20px;
margin-top: 20px;
}
/* Clear floats after the columns */
.row:after {
content: "";
display: table;
clear: both;
}
/* Footer*/
.footer {
padding: 20px;
text-align: center;
```

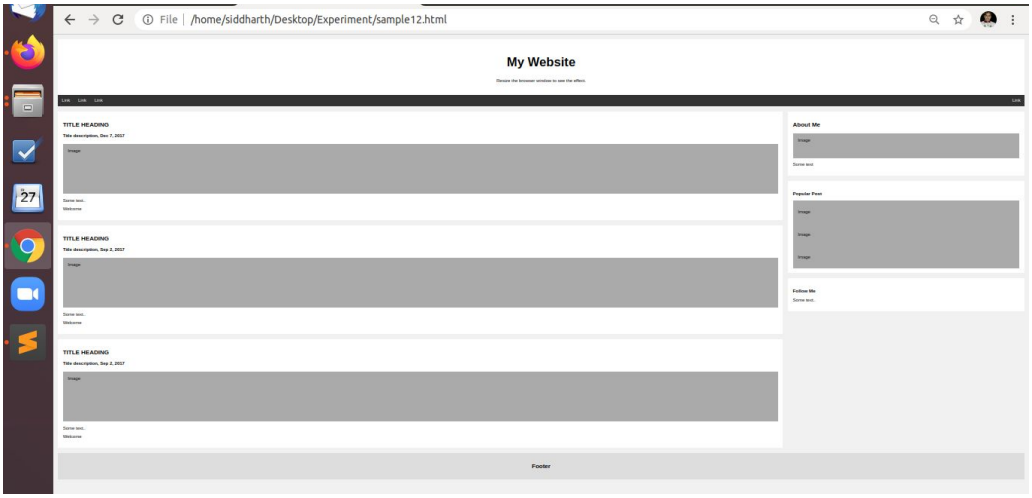
```
background: #ddd;
margin-top: 20px;
}
/* Responsive layout - when the screen is less than 800px wide, make the two columns
stack on top of each other instead of next to each other */
@media screen and (max-width: 800px) {
.leftcolumn, .rightcolumn {
width: 100%;
padding: 0;
}
}
/* Responsive layout - when the screen is less than 400px wide, make the navigation links
stack on top of each other instead of next to each other */
@media screen and (max-width: 400px) {
.topnav a {
float: none;
width: 100%;
}
}
</style>
</head>
<body>
<div class="header">
<h1>My Website</h1>
<p>Resize the browser window to see the effect.</p>
</div>
<div class="topnav">
<a href="#">Link</a>
```

```
<a href="#">Link</a>
<a href="#">Link</a>
<a href="#" style="float:right">Link</a>
</div>
<div class="row">
<div class="leftcolumn">
<div class="card">
<h2>TITLE HEADING</h2>
<h5>Title description, Dec 7, 2017</h5>
<div class="fakeimg" style="height:200px;">Image</div>
<p>Some text..</p>
<p>Welcome</p>
</div>
<div class="card">
<h2>TITLE HEADING</h2>
<h5>Title description, Sep 2, 2017</h5>
<div class="fakeimg" style="height:200px;">Image</div>
<p>Some text..</p>
<p>Welcome</p>
</div>
<div class="card">
<h2>TITLE HEADING</h2>
<h5>Title description, Sep 2, 2017</h5>
<div class="fakeimg" style="height:200px;">Image</div>
<p>Some text..</p>
<p>Welcome</p>
</div>
</div>
```

```
<div class="rightcolumn">
<div class="card">
<h2>About Me</h2>
<div class="fakeimg" style="height:100px;">Image</div>
<p>Some text </p>
</div>
<div class="card">
<h3>Popular Post</h3>
<div class="fakeimg"><p>Image</p></div>
<div class="fakeimg"><p>Image</p></div>
<div class="fakeimg"><p>Image</p></div>
</div>
<div class="card">
<h3>Follow Me</h3>
<p>Some text..</p>
</div>
</div>
</div>
<div class="footer">
<h2>Footer</h2>
</div>
</body>
</html>
```

Output/Results snippet

- Created Menu with responsive factor & layout of the web page .



References:

- https://www.w3schools.com/html/html_intro.asp
- https://www.tutorialspoint.com/css/index.htm
- https://www.javatpoint.com/css-tutorial

Learning outcome 3 - able to create their own account in cloud and hosting

After achieving this learning outcome, a student will be able to create their own account in cloud and hosting. In order to achieve this learning outcome, a student has to complete the following:

1. Create own account in Cloud (AWS) (3 Hr)
2. Create Virtual Server in Cloud (AWS) (7 Hr)
3. Install Web Server on Cloud (AWS) (5 Hr)
4. Hosting in Amazon Web Server (5 Hr)
5. Launch and count number of visitors (5 Hr)

Activity 1

Aim: Create own account in Cloud (AWS)

Learning outcome: Able to create own account in cloud and hosting.

Duration: 3 hour

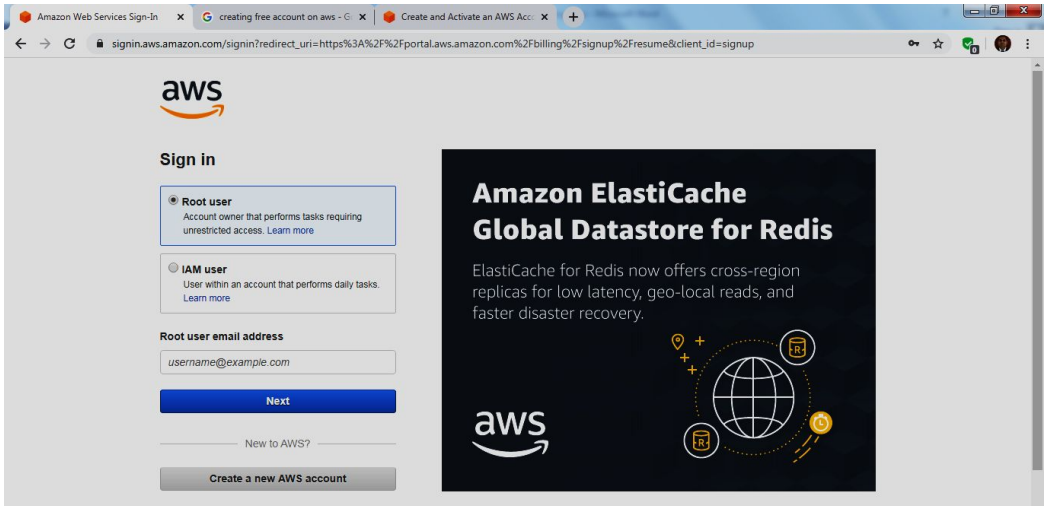
List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Web Browser (Chrome/Firefox)
3. Internet Connection

Steps to Create Own Account in Cloud (AWS):

1. Open web browser and open site http://www.aws.amazon.com/free
2. Click on 'Create Free Account' button

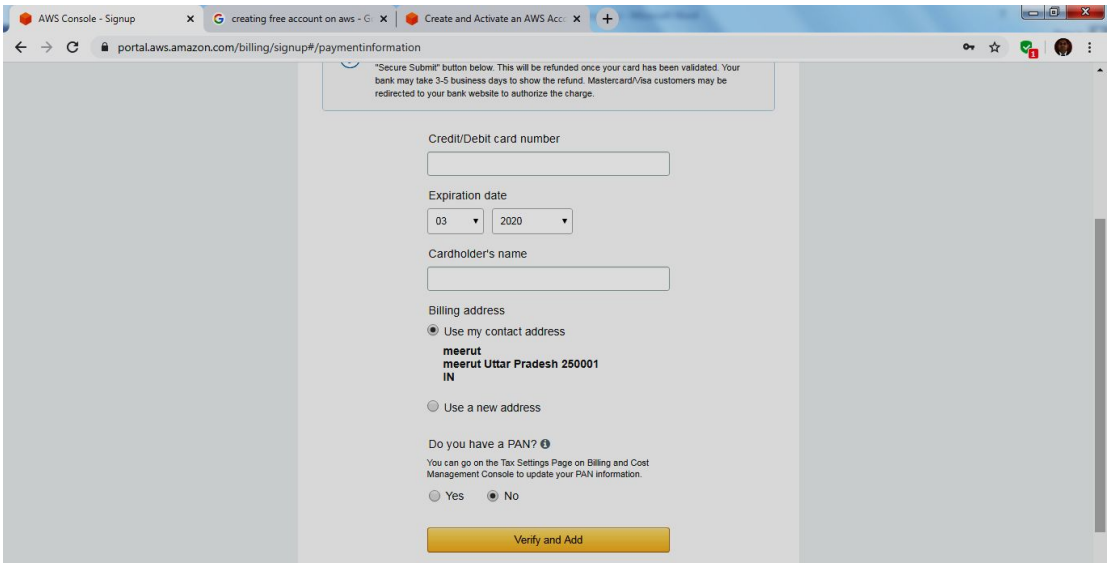
3. On next screen click on ‘Create a New AWS Account’ button below



4. Fill details in form and press ‘Continue’

5. On the next screen fill other details, remember to select ‘personal’ in account type. Then ‘create account and continue’

6. Enter credit/debit card details for payment method verification and and click ‘verify and add’



7. Verify your mobile number & email, your account will be activated after verification within a few minutes.
8. Login to your account using account credentials at http://www.aws.amazon.com/console

Output/Results Snippet:

- After login, you will see the dashboard as given below

References:

- Support URL from AWS Cloud- https://aws.amazon.com/premiumsupport/knowledge-center/create-and-activate-aws-account/

Activity 2

Aim: Create Virtual Server in Cloud (AWS)

Learning outcome: Able to create own account in cloud and hosting.

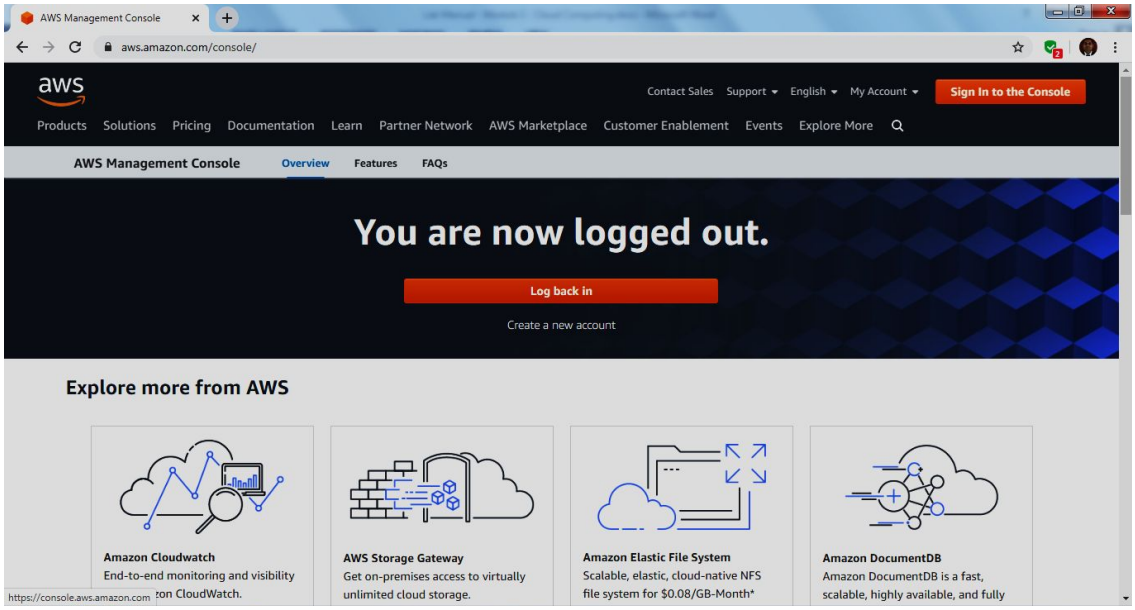
Duration: 7 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Web Browser (Chrome/Firefox)
3. Internet Connection

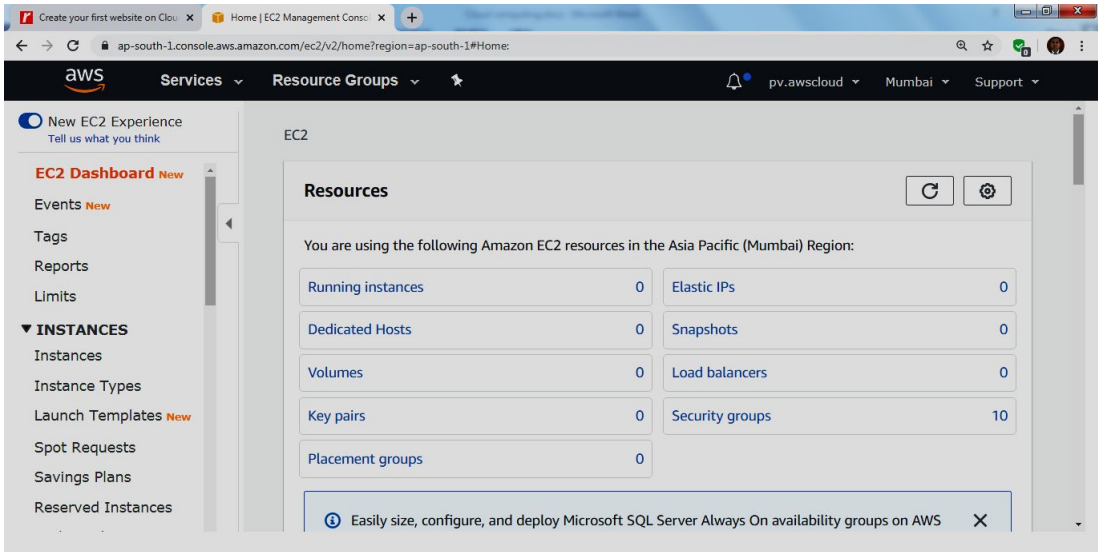
Steps to Create Virtual Server in Cloud (AWS):

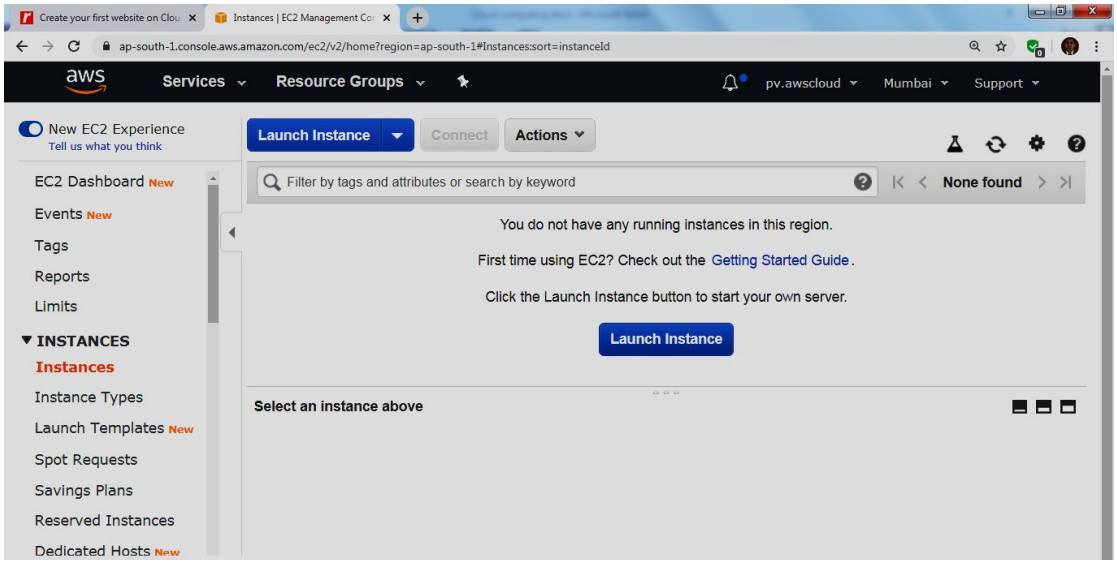
1. Log in to AWS account through http://www.aws.amazon.com/console/



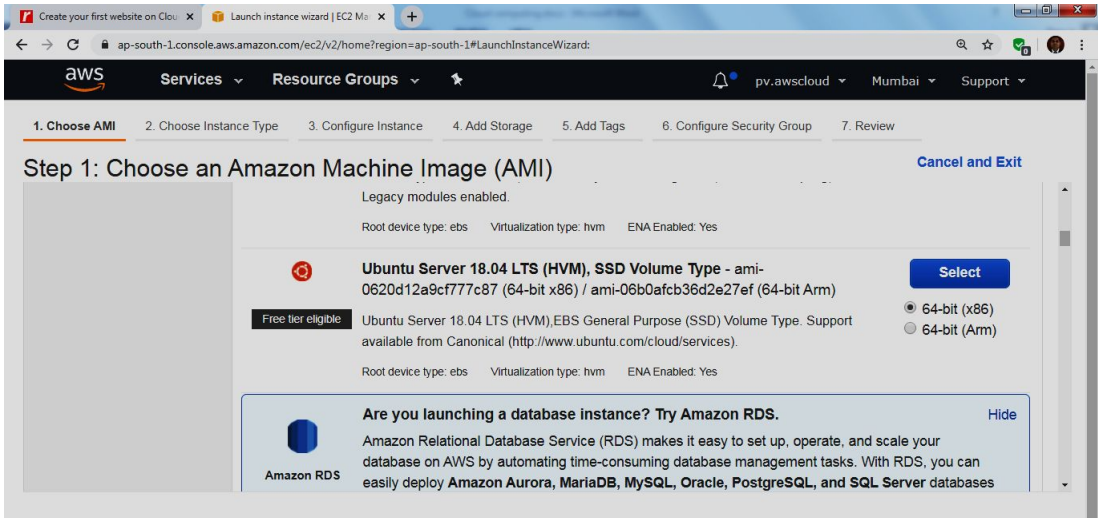
2. Search for EC2 service and select EC2 virtual server in cloud

3. Select Instances from left panel, and then click Launch Instance



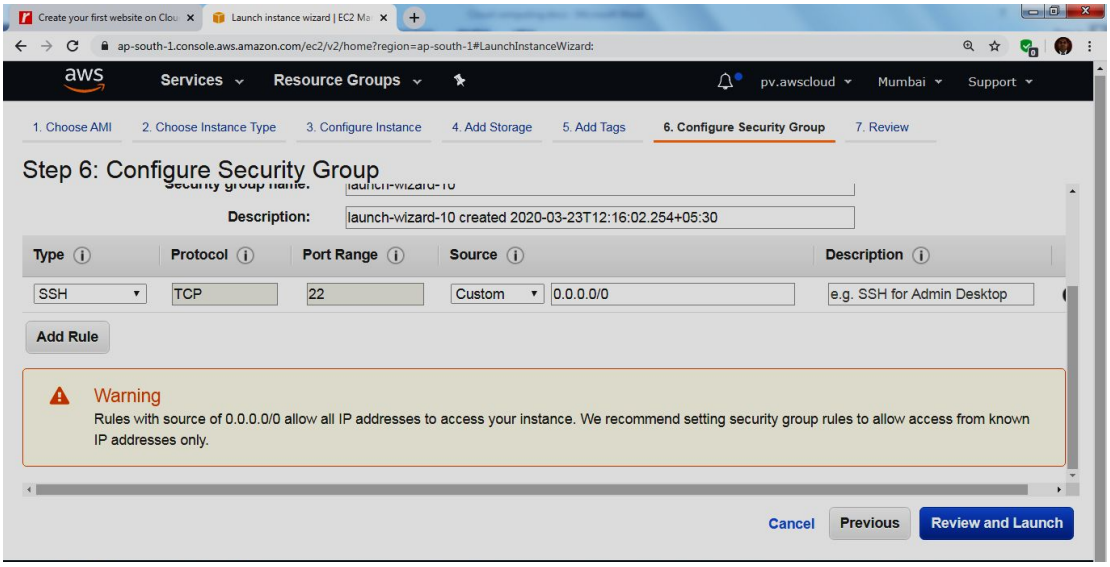


4. Select AMI (Amazon Machine Image- OS Platform) type (Ubuntu 18.04 free tier eligible only)



5. Select type of Server (t2-micro for free tier)

6. Click on Review and Launch



7. Select create new key pair & download key pair to local system

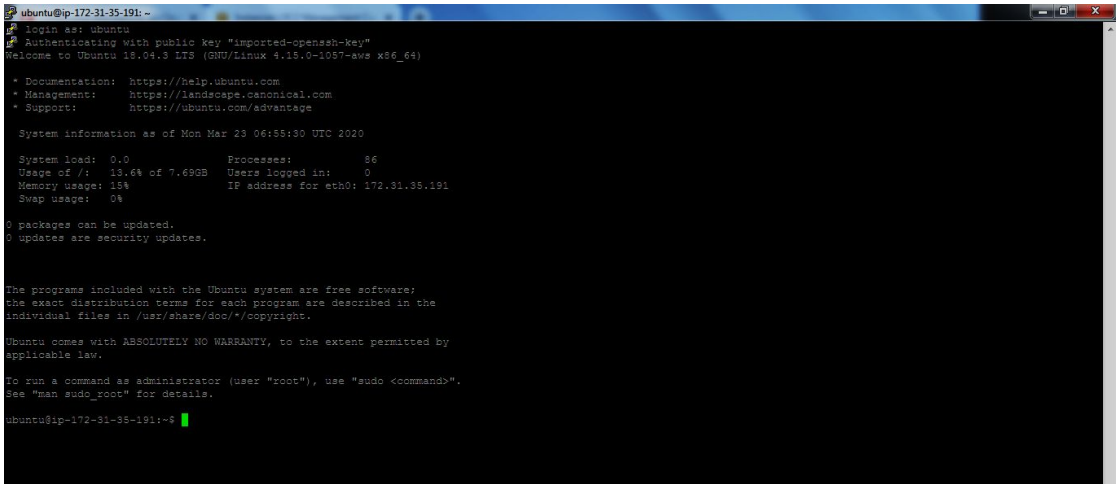
8. Click Launch Instances & view instances from left panel

9. Once instance is launched successfully and shows status checks done as 2/2, click on Connect button

10. Copy SSH command code from dialog and paste in local system terminal & press Enter

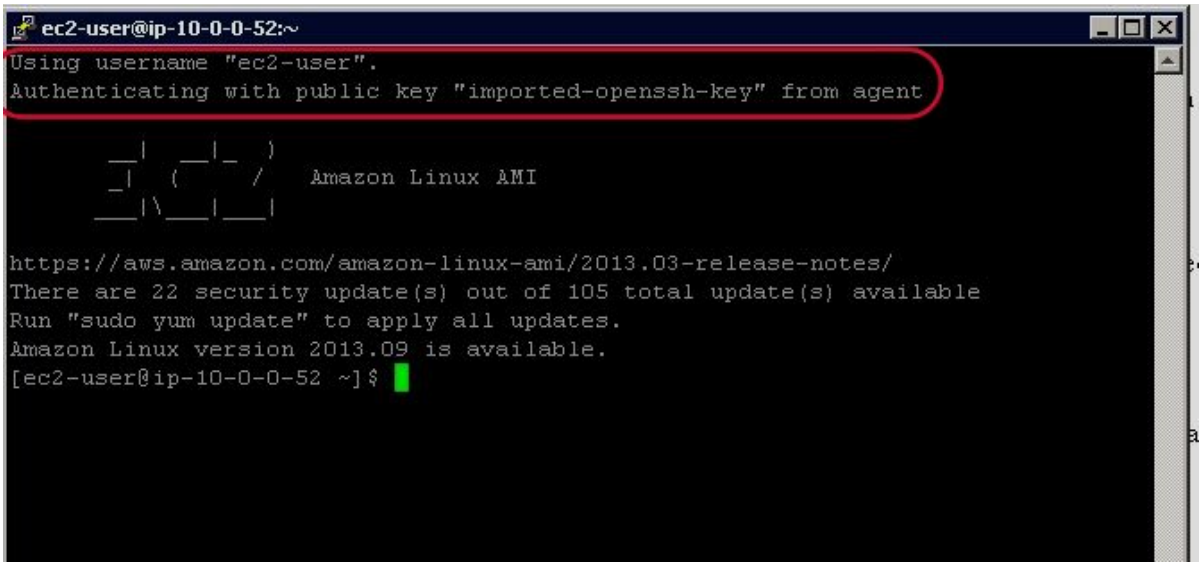
Output/Results Snippet:

You will be logged into remote server using SSH command shell



Example –

- Example image if you log into Amazon Linux AMI Server



References:

- Youtube Video on Amazon Web Service Free Ubuntu Server Setup
  https://www.youtube.com/watch?v=fPZuN_fibjM

Activity 3

Aim: Install Web Server on Cloud (AWS)

Learning outcome: Able to create own account in cloud and hosting.

Duration: 5 hour

List of Hardware/Software requirements:

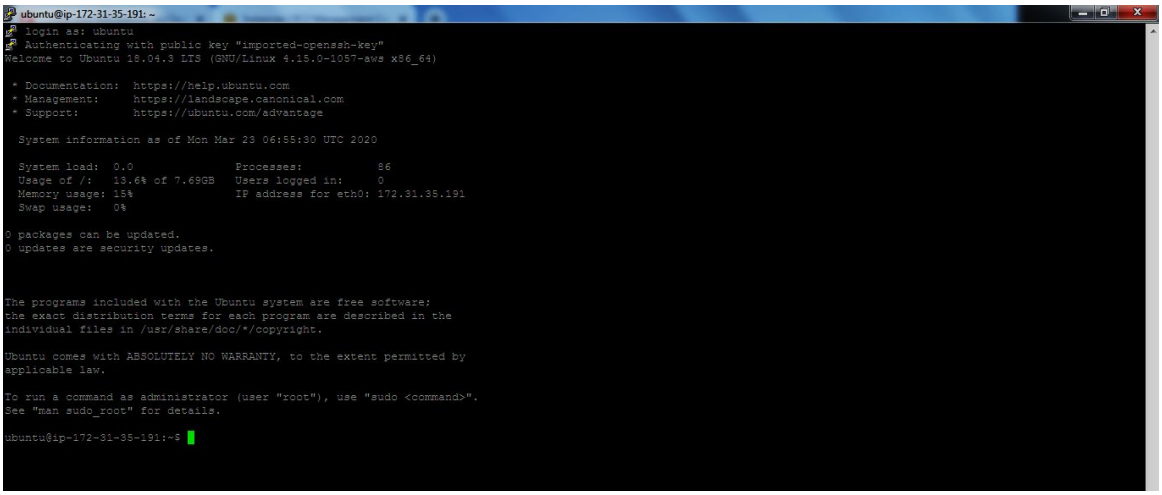
1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Web Browser (Chrome/Firefox)
3. High speed Internet Connection

Steps to Install Web Server in Cloud:

1. Log in to AWS account using your credentials from http://aws.amazon.com/console

2. Select EC2 from the dashboard screen. This will navigate you to EC2 instances dashboard where you can view your created virtual server

3. Click on the ‘Connect’ button and copy the SSH command, paste command into the local system terminal and press Enter. You will get screen as below after you connect to remote virtual machine



4. To install LAMP stack, having Apache Server, PHP & MySQL; follow the usage of commands into terminal window
5. To download LAMPP stack from internet resource portal:

   sudo wget
   https://downloadsapachefriends.global.ssl.fastly.net/7.3.0/xampp-linux-x64-7.3.0-0-installer.run?from_af=true

6. To change permission of execution for downloaded file

sudo chmod 777 xampp-linux-x64-7.3.0-0-installer.run?from_af=true

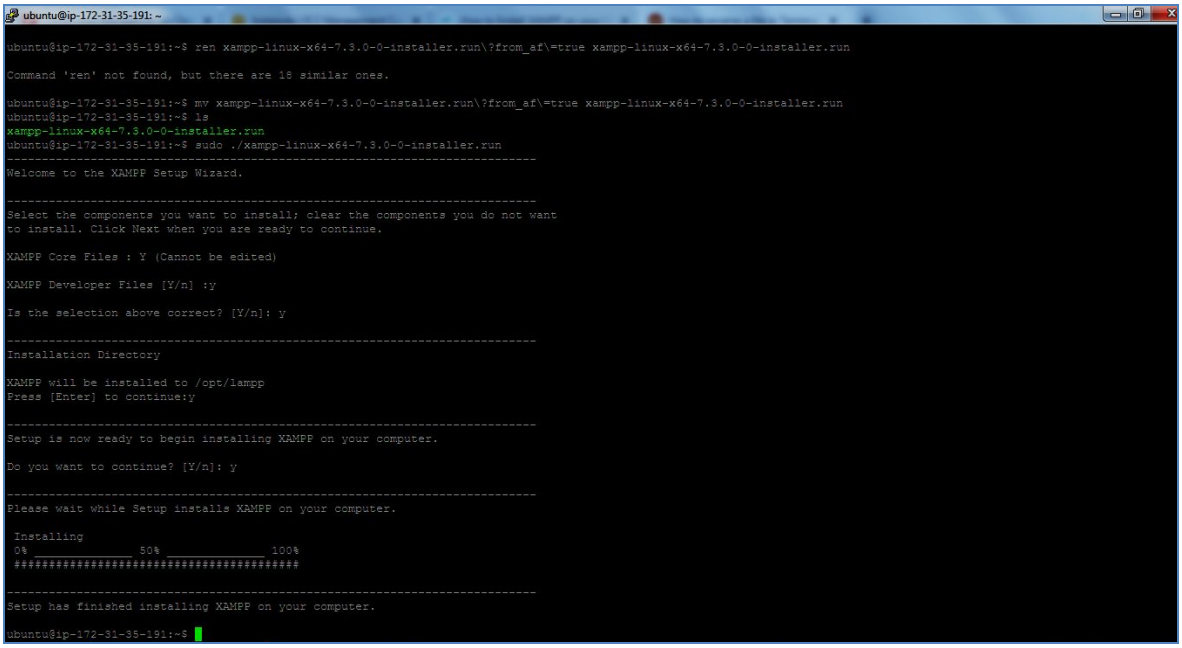
7. To rename the file as shell script file extension

sudo mv xampp-linux-x64-7.3.0-0-installer.run?from_af=true xampp-linux-x64-7.3.0-0-installer.run

8. Run the installer on virtual server

sudo ./xampp-linux-x64-7.3.0-0-installer.run

9. Press ‘y’ for all prompts



10. Change directory to view installation

cd /opt/lampp/

ls -l

11. To start server type following command

sudo /opt/lampp/lampp start

Output/Results Snippet:

References:

- For LAMPP on Ubuntu server- https://dmsbilas.wordpress.com/2019/03/12/how-to-install-xampp-in-amazon-aws-ubuntu-linux/
- To install LAMP stack on Amazon Linux AMI- https://gist.github.com/aronwoost/1105007

Activity 4

Aim: Hosting in Amazon Web Server

Learning outcome: Able to create own account in cloud and hosting.

Duration: 5 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Web Browser (Chrome/Firefox)
3. Internet Connection

Code for hosting along with steps to host:

1. Log in to AWS account and connect EC2 instance using terminal SSH command (Refer Activity-2 steps 1 to 3 on how to connect to EC2 using SSH command)
2. In virtual machine terminal type commands given below

   cd /opt/lampp/htdocs/

   sudo nano index.php

3. Replace the code in file *index.php* with sample PHP code given below for demonstration

   <?php

   echo("<h1>Welcome to AWS Virtual Server Hosted Web Page</h1>");

   echo("
<h3>This is sample page to demonstrate web hosting on AWS EC2 Server using LAMP stack</h3>");

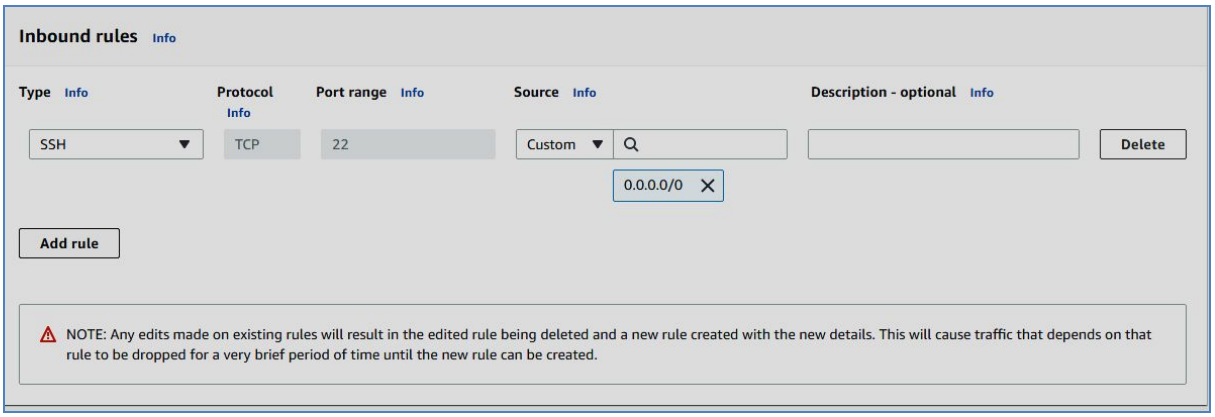
   ?>

4. Press ctrl+o to save file in nano editor and then press Enter
5. Press ctrl+x to exit nano editor and return to terminal prompt
6. Now return to web browser in local system and select go to EC2 instance dashboard
7. Scroll horizontally to extreme end and check name of security group
8. Click on security group name to visit security group dashboard in AWS account

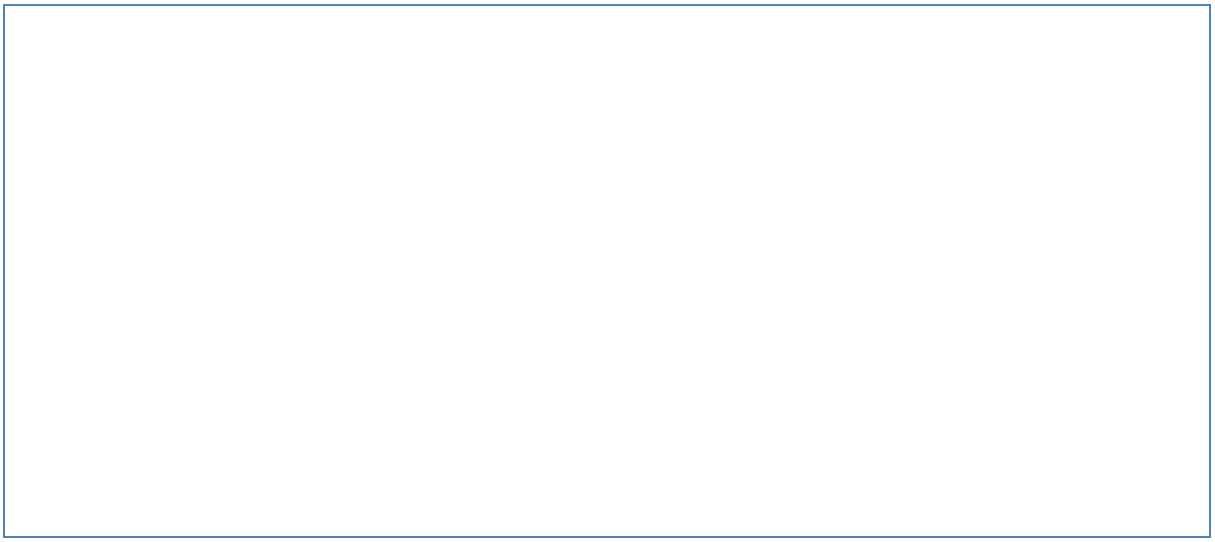
9. Select security group in security group dashboard and click on ‘Inbound Rules’ tab below

10. Click on ‘Edit Inbound Rules’ button on right side

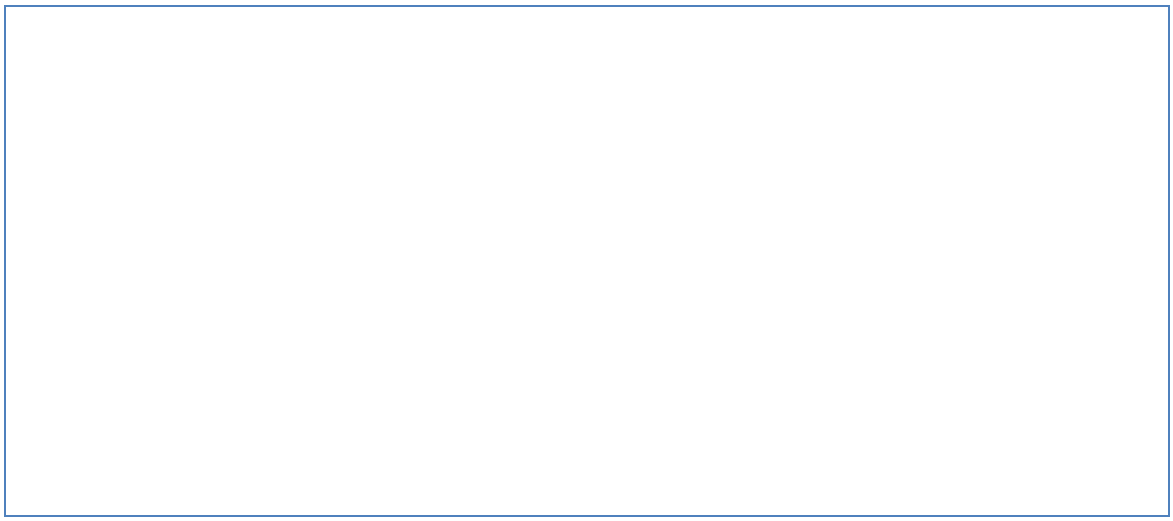
11. On next screen click ‘Add rule’ button



12. Select HTTP in Type list



13. Select ‘Anywhere’ under Source list



14. Click on ‘Save rules’ button below
15. This will enable HTTP access to web server hosting
16. To view your web page output got to EC2 instances dashboard by selecting instances in left panel
17. Select your EC2 virtual machine instance
18. In description tab below find Public DNS(IPv4)
19. Copy the URL value from Public DNS(IPv4) and paste in new tab in web browser, then press Enter

Output/Results Snippet:

1. Output page would look similar to the page below

Activity 5

Aim: Launch and count number of visitors

Learning outcome: Able to create own account in cloud and hosting.

Duration: 5 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Web Browser (Chrome/Firefox)
3. Internet Connection

Program:

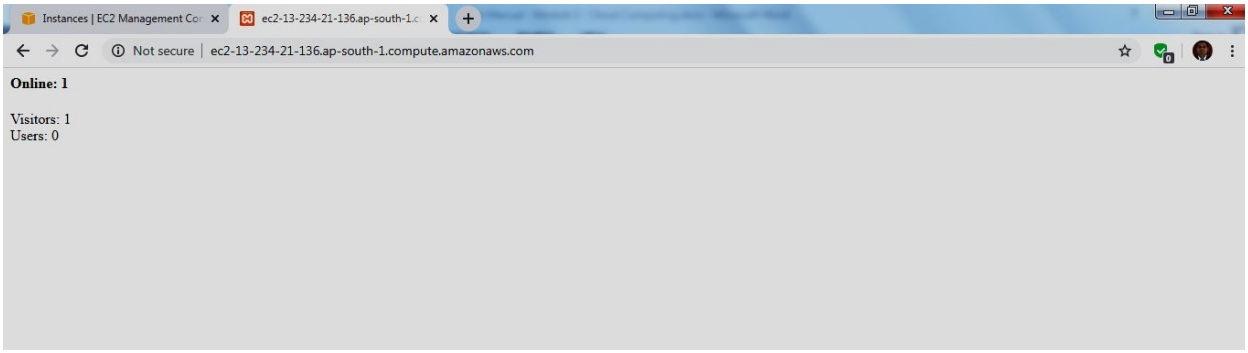
1) In place of sample code in Activity-3, replace code in *index.php* file with code given below using nano editor with command- sudo nano index.php
2) Copy and paste the code given below in nano editor using right click & run using Public DNS of EC2 virtual server instance as in Activity-3

```
<?php

// Script Online Users and Visitors -http://coursesweb.net/php-mysql/

if(!isset($_SESSION)) session_start(); // start Session, if not already started

$filetxt = 'userson.txt'; // the file in which the online operators /visitors are stored

$timeon = 120; // number of seconds to keep a operator online

$sep = '^^'; // characters used to separate the operator name and date-time

$vst_id = '-vst-'; // an identifier to know that it is a visitor, not logged user

// get the user name if it is logged, or the visitors IP (and add the identifier)

$uvon = isset($_SESSION['nume']) ? $_SESSION['nume'] : $_SERVER['SERVER_ADDR'].
    $vst_id;

$rgxvst = '/^([0-9\.]*)'. $vst_id. '/i'; // regex to recognize the line with visitors

$nrvst = 0; // to store the number of visitors

// sets the row with the current user /visitor that must be added in $filetxt (and current timestamp)

$addrow[] = $uvon. $sep. time();

// check if the file from $filetxt exists and is writable

if(is_writable($filetxt)) {

// get into an array the lines added in $filetxt
```

```
$ar_rows = file($filetxt, FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
$nrrows = count($ar_rows);
// number of rows
// if there is at least one line, parse the $ar_rows array
    if($nrrows>0) {
for($i=0; $i<$nrrows; $i++) {
// get each line and separate the user /visitor and the timestamp
$ar_line = explode($sep, $ar_rows[$i]);
// add in $add row array the records in last $timeon seconds
if($ar_line[0]!=$uvon && (intval($ar_line[1])+$timeon)>=time()) {
$addrow[] = $ar_rows[$i];
                                    }
                            }
                    }
    }
$nruvon = count($addrow); // total online
$usron = ''; // to store the name of logged users
// traverse $add row to get the number of visitors and users
for($i=0; $i<$nruvon; $i++) {
if(preg_match($rgxvst, $addrow[$i])) $nrvst++; // increment the visitors
else {
// gets and stores the user's name
$ar_usron = explode($sep, $addrow[$i]);
$usron .= '<br/> - <i>'. $ar_usron[0]. '</i>';
                    }
    }
$nrusr = $nruvon - $nrvst; // gets the users (total - visitors)
// the HTML code with data to be displayed
```

```
$reout = '<div id="uvon"><h4>Online: '. $nruvon. '</h4>Visitors: '. $nrvst. '<br/>Users: '. $nrusr.
    $usron. '</div>';
// write data in $filetxt
if(!file_put_contents($filetxt, implode("\n", $addrow))) $reout = 'Error: Recording file not exists, or is
    not writable';
// if access from <script>, with GET 'uvon=showon', adds the string to return into a JS statement
// in this way the script can also be included in .html files
if(isset($_GET['uvon']) && $_GET['uvon']=='showon') $reout = "document.write('$reout');";
echo $reout; // output /display the result
?>
```

Output/Results Snippet:



References:

- To get registration script http://coursesweb.net/php-mysql/register-login-script-users-online_s2

Learning outcome 4 - able to configure an embedded database with different web pages using MongoDB

After achieving this learning outcome, a student will be able to configure an embedded database with different web pages using MongoDB. In order to achieve this learning outcome, a student has to complete the following:

Activities

1. Install of MongoDB in the system (2Hrs)
2. Create data with the following Data types – String, Integer, Boolean, double, min/max keys, arrays, timestamp, object, Null, symbol, date, object ID, Binary data, Code, Regular Expression (3Hrs)
3. Insert Document in database (1Hr)
4. Update document in database (1Hr)
5. Delete document in database (1Hr)
6. Project document in document (2Hr)
7. Create a MongoDB query to display all the documents in the collection data (Trainees data)(5Hrs)
8. Create a MongoDB query to display the fields id, trainee name, lab name, Certificate No., course title, course starting date,course ending date for all the documents in the collection trainees data. (5 Hrs)
9. Create a MongoDB query to display the fields id, trainee name, lab name, Certificate No., course name, course starting date, course ending date for all the documents in the collection trainees data, but excluding lab name (5 Hrs)
10. Create a MongoDB query to display all the trainees who attended course on PHP (2Hrs)
11. Create a MongoDB query to display the 1stbatch trainees of PHP (3Hrs)
12. Create a MongoDB query to display the 2ndbatch trainees of PHP (2Hrs)
13. Create a MongoDB query to find the course where maximum trainees attended (3Hrs)
14. Create a MongoDB query to find lab wise details of trainees (5Hrs)
15. Create a MongoDB query with course wise details of trainees (5Hrs)
16. Print the queries (5Hrs)

Activity 1

Aim: To install MongoDB in the computer system running with Ubuntu 18.04 LTS

Learning outcome: Able to configure embedded databases with different web pages using MongoDB.

Duration: 2 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. MongoDB 4.2 Community Edition

Installation Steps of MongoDB Community Edition:

1) Import the public key used by the package management system.

From a terminal, issue the following command to import the MongoDB public GPG Key from https://www.mongodb.org/static/pgp/server-4.2.asc

```
wget -qO - https://www.mongodb.org/static/pgp/server-4.2.asc | sudo apt-key add –
```

The operation should respond with an OK.

However, if you receive an error indicating that gnupg is not installed, you can:

a) Install gnupg and its required libraries using the following command:

```
sudo apt-get install gnupg
```

b) Once installed, retry importing the key:

```
wget -qO - https://www.mongodb.org/static/pgp/server-4.2.asc | sudo apt-key add -
```

2) Create a /etc/apt/sources.list.d/mongodb-org-4.2.list file for MongoDB.

Create the list file using the command appropriate for your version of Debian:

```
echo "deb http://repo.mongodb.org/apt/debian buster/mongodb-org/4.2 main" | sudo tee
/etc/apt/sources.list.d/mongodb-org-4.2.list
```

3) **Reload local package database**.

Issue the following command to reload the local package database:

```
sudo apt-get update
```

4) Install the MongoDB packages

To install the latest stable version, run command as following

```
sudo apt-get install -y mongodb-org
```

5) Start MongoDB

You can start the mongod process by issuing the following command:

```
sudo systemctl start mongod
```

6) Begin using MongoDB.

Run the following command

```
mongo
```

References:

- https://docs.mongodb.com/manual/tutorial/install-mongodb-on-debian/

Activity 2

Aim: Create MongoDB database with the following Data types – String, Integer, Boolean, double, min/max keys, arrays, timestamp, object, Null, symbol, date, object ID, Binary data, Code, Regular Expression

Learning outcome: Able to configure embedded databases with different web pages using MongoDB.

Duration: 3 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. MongoDB 4.2 Community Edition
3. XAMPP- PHP 7
4. PHP-MongoDB driver 1.7.4

Create a Data:

a) Insert a Document without Specifying an _id Field
In the following example, the document passed to the insert() method does not contain the _id field:

```
db.products.insert( { item: "card", qty: 15 } )
```

Output:

```
{ "_id" : ObjectId("5063114bd386d8fadbd6b004"), "item" : "card", "qty" : 15 }
```

b) Insert Multiple Documents
The following example performs a bulk insert of three documents by passing an array of documents to the insert() method.

```
db.products.insert(
  [
    { _id: 11, item: "pencil", qty: 50, type: "no.2" },
    { item: "pen", qty: 20 },
    { item: "eraser", qty: 25 }
  ]
)
```

Output:

```
{ "_id" : 11, "item" : "pencil", "qty" : 50, "type" : "no.2" }
{ "_id" : ObjectId("51e0373c6f35bd826f47e9a0"), "item" : "pen", "qty" : 20 }
{ "_id" : ObjectId("51e0373c6f35bd826f47e9a1"), "item" : "eraser", "qty" : 25 }
```

We can either set the value of *_id* for each document or else *_id* value is assigned automatically. Let's take a look at the example below

```
db.student.insertMany(
[
{name : "Alex", age : 19}, {name : "Albert" , age : 20}, {name : "Bob" , age : 19}
]
)
```

Output:

```
{
    "acknowledged" : true,
    "insertedIds" : [
        ObjectId("5b680cbc80847beb3aa3e837"),
        ObjectId("5b680cbc80847beb3aa3e838"),
        ObjectId("5b680cbc80847beb3aa3e839")
    ]
}
```

Reference:

- https://docs.mongodb.com/manual/reference/method/db.collection.insert/

Activity 3

Aim: Insert Document in database

Learning outcome: Able to configure embedded databases with different web pages using MongoDB.

Duration: 1 hour

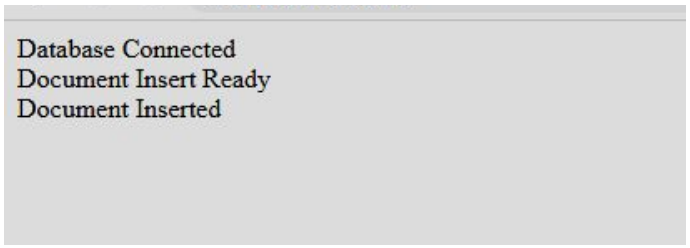
List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. MongoDB 4.2 Community Edition
3. XAMPP- PHP 7
4. PHP-MongoDB driver 1.7.4

Program:

```
<?php
//connecting to database
if($con=new MongoDB\Driver\Manager("mongodb://localhost:27017"))
    echo "Database Connected";
//creating document as array
$doc=array("_id"=>"124", "Name"=>"John", "Age"=>"23", "Location"=>"India");
//creating write database object
$single_insert=new MongoDB\Driver\BulkWrite();

if($single_insert->insert($doc))        //preparing query statement for insert
    echo "<br>Document Insert Ready";

if($con->executeBulkWrite("mydb.mycol", $single_insert))  //executing write query
    echo "<br>Document Inserted";
?>
```

Output:



Activity 4

Aim: Update Document in database

Learning outcome: Able to configure embedded databases with different web pages using MongoDB.

Duration: 1 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. MongoDB 4.2 Community Edition
3. XAMPP- PHP 7
4. PHP-MongoDB driver 1.7.4

Program:

```
<?php
//connecting database
if($con=new MongoDB\Driver\Manager("mongodb://localhost:27017"))
    echo "Database Connected";
//creating write object
$single_update=new MongoDB\Driver\BulkWrite();
//creating query for update
$single_update->update(["_id"=>"124"], ["Name"=>"Raj", "Age"=>"26"], ["multi"=>false,
"upsert"=>false]);
//executing query for update
if($con->executeBulkWrite("mydb.mycol", $single_update))
    echo "<br>Document Updated";
?>
```

Output:

Activity 5

Aim: Delete Document in database

Learning outcome: Able to configure embedded databases with different web pages using MongoDB.

Duration: 1 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. MongoDB 4.2 Community Edition
3. XAMPP- PHP 7
4. PHP-MongoDB driver 1.7.4

Program:

```
<?php
//connecting database
if($con=new MongoDB\Driver\Manager("mongodb://localhost:27017"))
    echo "Database Connected";
//creating object for write
$del=new MongoDB\Driver\BulkWrite();
//creating delete query
$del->delete(["Name"=>"John"], ["limit"=>0]);
//executing delete query
if($con->executeBulkWrite("mydb.mycol", $del))
    echo "<br>Document Deleted";
?>
```

Output:

Activity 6

Aim: Project Document in database

Learning outcome: Able to configure embedded databases with different web pages using MongoDB.

Duration: 2 hours

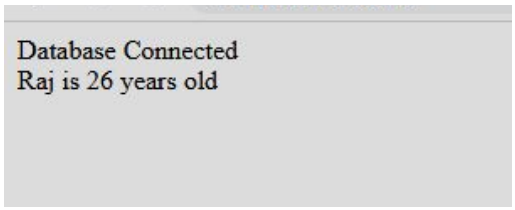
List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. MongoDB 4.2 Community Edition
3. XAMPP- PHP 7
4. PHP-MongoDB driver 1.7.4

Program:

```
<?php
//connecting to database
if($con=new MongoDB\Driver\Manager("mongodb://localhost:27017"))
    echo "Database Connected";
$filter=["Name"=>"Raj"];    //defining projection filter
$option=[];                 //leaving options blank

$read=new MongoDB\Driver\Query($filter, $option);          //creating query

$single_user=$con->executeQuery("mydb.mycol", $read);    //executing query

foreach($single_user as $user){                 //foreach loop for traversing result displaying it
    echo "<br>".$user->Name." is ".$user->Age." years old";
}

?>
```

Output:



Activity 7

Aim: Create a MongoDB query to display all the documents in the collection data (Trainees data)

Learning outcome: Able to configure embedded databases with different web pages using MongoDB.

Duration: 5 hours

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. MongoDB 4.2 Community Edition
3. XAMPP- PHP 7
4. PHP-MongoDB driver 1.7.4

Program:

```
<?php
//connecting to database
if($con=new MongoDB\Driver\Manager("mongodb://localhost:27017"))
    echo "Database Connected<br><br>";

$filter=[];             //no filter
$option=[];             //no options

$read=new MongoDB\Driver\Query($filter, $option);          //creating query

$single_user=$con->executeQuery("mydb3.mycol", $read);  //executing query

//creating HTML table for displaying data
echo "<table border=3 cellspacing=5
    cellpadding=7><thead><th>ID<th>Name<th>Lab<th>Certificate No.<th>Course<th>Start
    Date<th>End Date</thead>";

foreach($single_user as $user){                 //parsing through results in loop
    echo "<tr>";
    echo
    "<td>".$user->_id."<td>".$user->name."<td>".$user->lab."<td>".$user->certNo."<td>".$user->c
    ourse."<td>".$user->start."<td>".$user->end;
}           //end of loop for parsing

echo "</table>";                //end of table

?>
```

Output:

Activity 8

Aim: Create a MongoDB query to display the fields id, trainee name, lab name, Certificate No., course title, course starting date,course ending date for all the documents in the collection trainees data.

Learning outcome: Able to configure embedded databases with different web pages using MongoDB.

Duration: 5 hours

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. MongoDB 4.2 Community Edition
3. XAMPP- PHP 7
4. PHP-MongoDB driver 1.7.4

Program:

```
<?php
//connecting to database
if($con=new MongoDB\Driver\Manager("mongodb://localhost:27017"))
    echo "Database Connected<br><br>";
//no filters & no options applied
$filter=[];
$option=[];
//creating query object
$read=new MongoDB\Driver\Query($filter, $option);
//executing query and receive results
$single_user=$con->executeQuery("mydb3.mycol", $read);
//create table view
echo "<table border=3 cellspacing=5 cellpadding=7><thead><th>ID<th>Name<th>Certificate
    No.<th>Course<th>Start Date<th>End Date</thead>";

foreach($single_user as $user){                    //parse through results using loop
    echo "<tr>";
    echo
    "<td>".$user->_id."<td>".$user->name."<td>".$user->certNo."<td>".$user->course."<td>".$user
    ->start."<td>".$user->end;
}              //end of loop

echo "</table>";                  //end of table

?>
```

Output:

Activity 9

Aim: Create a MongoDB query to display the fields id, trainee name, lab name, Certificate No., course name, course starting date, the course ending date for all the documents in the collection trainee's data, but excluding lab name

Learning outcome: Able to configure embedded databases with different web pages using MongoDB.

Duration: 5 hours

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. MongoDB 4.2 Community Edition
3. XAMPP- PHP 7
4. PHP-MongoDB driver 1.7.4

Program:

```
<?php
//connecting to database
if($con=new MongoDB\Driver\Manager("mongodb://localhost:27017"))
    echo "Database Connected<br><br>";

$filter=[];    //no filters
$option=[ 'projection'=>['lab'=>0] ];          //options to eliminate lab column from result

$read=new MongoDB\Driver\Query($filter, $option);          //creating query

$single_user=$con->executeQuery("mydb3.mycol", $read);  //executing query
//creating table view without column for lab
echo "<table border=3 cellspacing=5 cellpadding=7><thead><th>ID<th>Name<th>Certificate
    No.<th>Course<th>Start Date<th>End Date</thead>";

foreach($single_user as $user){                  //parsing results in loop
    echo "<tr>";
    echo
    "<td>".$user->_id."<td>".$user->name."<td>".$user->certNo."<td>".$user->course."<td>".$user
    ->start."<td>".$user->end;
}              end loop

echo "</table>";                //end table
?>
```

Output:

Database Connected

ID	Name	Certificate No.	Course	Start Date	End Date
1	simmi	11	IBM	2018	2020
2	suman	22	IBM	2018	2020
3	anju	33	IBM	2018	2020
4	uma	44	IBM	2018	2020
5	meenakshi	55	IBM	2018	2020
6	deepika	66	IBM	2018	2020
7	muskan	77	IBM	2018	2020
8	kiran	88	IBM	2018	2020
9	nishika	99	IBM	2018	2020
10	nutan	1	IBM	2018	2020
11	jyoti	2	IBM	2018	2020
12	seema	21	IBM	2018	2020
13	seema B	22	IBM	2018	2020
14	preeti	23	EF	2019	2021

Activity 10

Aim: Create a MongoDB query to display all the trainees who attended course on PHP

Learning outcome: Able to configure embedded databases with different web pages using MongoDB.

Duration: 2 hours

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. MongoDB 4.2 Community Edition
3. XAMPP- PHP 7
4. PHP-MongoDB driver 1.7.4

Program:

```
<?php
//connecting to database
if($con=new MongoDB\Driver\Manager("mongodb://localhost:27017"))
    echo "Database Connected<br><br>";

$filter=[];  //no filters
$option=[ 'projection'=>['lab'=>0] ];                //options to eliminate lab column from result

$read=new MongoDB\Driver\Query($filter, $option);   //creating query

$single_user=$con->executeQuery("mydb3.mycol", $read);      //executing query
//creating table view without column for lab
echo "<table border=3 cellspacing=5
    cellpadding=7><thead><th>ID<th>Name<th>Certificate No.<th>Course<th>Start
    Date<th>End Date</thead>";

foreach($single_user as $user){              //parsing results in loop
    echo "<tr>";
    echo
    "<td>".$user->_id."<td>".$user->name."<td>".$user->certNo."<td>".$user->course."<td
    >".$user->start."<td>".$user->end;
}             end loop

echo "</table>";              //end table
?>
```

Output:

Database Connected

ID	Name	Certificate No.	Course	Start Date	End Date
1	simmi	11	IBM	2018	2020
2	suman	22	IBM	2018	2020
3	anju	33	IBM	2018	2020
4	uma	44	IBM	2018	2020
5	meenakshi	55	IBM	2018	2020
6	deepika	66	IBM	2018	2020
7	muskan	77	IBM	2018	2020
8	kiran	88	IBM	2018	2020
9	nishika	99	IBM	2018	2020
10	nutan	1	IBM	2018	2020
11	jyoti	2	IBM	2018	2020
12	seema	21	IBM	2018	2020
13	seema B	22	IBM	2018	2020
14	preeti	23	EF	2019	2021

Activity 11

Aim: Create a MongoDB query to display the 1st batch trainees of PHP

Learning outcome: Able to configure embedded databases with different web pages using MongoDB.

Duration: 2 hours

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. MongoDB 4.2 Community Edition
3. XAMPP- PHP 7
4. PHP-MongoDB driver 1.7.4

Program:

```
<?php
//connecting to database
if($con=new MongoDB\Driver\Manager("mongodb://localhost:27017"))
    echo "Database Connected<br><br>";
//filetrs to select lab PHP only for batch 1st i.e. admitted in 2018
$filter=[ 'lab'=>'PHP', 'start'=>2018 ];
$option=[  ];                    //no options required

$read=new MongoDB\Driver\Query($filter, $option);          //create query

$single_user=$con->executeQuery("mydb3.mycol", $read);  //execute query
//create table view
echo "<table border=3 cellspacing=5
    cellpadding=7><thead><th>ID<th>Name<th>Lab<th>Certificate No.<th>Course<th>Start
    Date<th>End Date</thead>";

foreach($single_user as $user){       //parsing through loop on result data
    echo "<tr>";
    echo
    "<td>".$user->_id."<td>".$user->name."<td>".$user->lab."<td>".$user->certNo."<td>".$user->c
    ourse."<td>".$user->start."<td>".$user->end;
}              //end loop

echo "</table>";                //end table

?>
```

Output:

Activity 12

Aim: Create a MongoDB query to display the 2nd batch trainees of PHP (Notepad)

Learning outcome: Able to configure embedded database with different web pages using MongoDB

Duration: 2 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. MongoDB 4.2 Community Edition
3. XAMPP- PHP 7
4. PHP-MongoDB driver 1.7.4

Program:

```
<?php
//connecting to database
if($con=new MongoDB\Driver\Manager("mongodb://localhost:27017"))
    echo "Database Connected<br><br>";
//filters to select PHP lab data only for 2nd batch i.e. admitted in 2019
$filter=[ 'lab'=>'PHP', 'start'=>2019 ];
$option=[  ];          //no options req

$read=new MongoDB\Driver\Query($filter, $option);          //create query

$single_user=$con->executeQuery("mydb3.mycol", $read);  //execute query
/create table view
echo "<table border=3 cellspacing=5
    cellpadding=7><thead><th>ID<th>Name<th>Lab<th>Certificate No.<th>Course<th>Start
    Date<th>End Date</thead>";

foreach($single_user as $user){                  //parsing results in loop
    echo "<tr>";
    echo
    "<td>".$user->_id."<td>".$user->name."<td>".$user->lab."<td>".$user->certNo."<td>".$user->c
    ourse."<td>".$user->start."<td>".$user->end;
}              //end loop

echo "</table>";      //end table

?>
```

Output:

References:

- HTML Introduction - https://www.w3schools.com/
- https://www.tutorialspoint.com/mongodb/mongodb_php.htm

Activity 13

Aim: Create a MongoDB query to find the course where maximum trainees attended (Notepad)

Learning outcome: Able to configure embedded databases with different web pages using MongoDB.

Duration: 3 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. MongoDB 4.2 Community Edition
3. XAMPP- PHP 7
4. PHP-MongoDB driver 1.7.4

Program:

```
<?php
//connecting to database
if($con=new MongoDB\Driver\Manager("mongodb://localhost:27017"))
    echo "Database Connected<br><br>";
//no filter or options
$filter=[ ];
$option=[ ];

$ibm=0;     //counetr for IBM course
$ef=0;              //counter for EF course

$read=new MongoDB\Driver\Query($filter, $option);          //create query

$single_user=$con->executeQuery("mydb3.mycol", $read);  //execute query

foreach($single_user as $user){                 //loop for parsing all data
    if($user->course=='IBM')
            $ibm++;                 //counter update if course is IBM
    else $ef++;                     //counter update if course is EF
}

if($ibm>$ef) echo('Maximum Trainees attended course IBM');          //display if IBM is max
else echo('Maximum Trainees attended course EF');              //display if EF is max

?>
```

Output:

References:

- HTML Introduction - https://www.w3schools.com/
- https://www.tutorialspoint.com/mongodb/mongodb_php.htm

Activity 14

Aim: Create a MongoDB query to find lab wise details of trainees (Notepad)

Learning outcome: Able to configure embedded databases with different web pages using MongoDB.

Duration: 5 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. MongoDB 4.2 Community Edition
3. XAMPP- PHP 7
4. PHP-MongoDB driver 1.7.4

Program:

```
<?php
//connecting to database
if($con=new MongoDB\Driver\Manager("mongodb://localhost:27017"))
    echo "Database Connected<br><br>";
//no filter or options
$filter=[ ];
$option=[ ];

$ibm=0;     //counetr for IBM course
$ef=0;              //counter for EF course

$read=new MongoDB\Driver\Query($filter, $option);          //create query

$single_user=$con->executeQuery("mydb3.mycol", $read);  //execute query

foreach($single_user as $user){                 //loop for parsing all data
    if($user->course=='IBM')
            $ibm++;                 //counter update if course is IBM
    else $ef++;                     //counter update if course is EF
}

if($ibm>$ef) echo('Maximum Trainees attended course IBM');          //display if IBM is max
else echo('Maximum Trainees attended course EF');              //display if EF is max

?>
```

Output:

References:

- HTML Introduction - https://www.w3schools.com/
- https://www.tutorialspoint.com/mongodb/mongodb_php.htm

Activity 15

Aim: Create a MongoDB query with course wise details of trainees (Notepad)

Learning outcome: Able to configure embedded databases with different web pages using MongoDB.

Duration: 5 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. MongoDB 4.2 Community Edition
3. XAMPP- PHP 7
4. PHP-MongoDB driver 1.7.4

Program:

```
<?php
//connecting to database
if($con=new MongoDB\Driver\Manager("mongodb://localhost:27017"))
    echo "Database Connected<br><br>";

$filter=[ ];            //no filters
$option=[ 'sort'=>['course'=>1] ];              //sorting options on course name

$read=new MongoDB\Driver\Query($filter, $option);          //create query

$single_user=$con->executeQuery("mydb3.mycol", $read);  //execute query
//create table view
echo "<table border=3 cellspacing=5
    cellpadding=7><thead><th>ID<th>Name<th>Lab<th>Certificate No.<th>Course<th>Start
    Date<th>End Date</thead>";

foreach($single_user as $user){                 //parsing results though loop
    echo "<tr>";
    echo
    "<td>".$user->_id."<td>".$user->name."<td>".$user->lab."<td>".$user->certNo."<td>".$user->c
    ourse."<td>".$user->start."<td>".$user->end;
}   //end loop

echo "</table>";     //end table

?>
```

Output:

References:

- HTML Introduction - https://www.w3schools.com/
- https://docs.mongodb.com/manual/reference/method/cursor.limit/

Activity 16

Aim: Print the queries (Notepad)

Learning outcome: Able to configure embedded databases with different web pages using MongoDB.

Duration: 5 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Apache 7.4
3. MongoDB

Program:

```
//Insert Record
<?php
require 'vendor/autoload.php';
// Creating Connection
$con = new MongoDB\Client("mongodb://localhost:27017");
// Creating Database
$db = $con->javatpoint;
// Creating Document
$collection = $db->employee;
// Inserting Record
$collection->insertOne( [ 'name' =>'Peter', 'email' =>'peter@abc.com' ] );
// Fetching Record
$record = $collection->find( [ 'name' =>'Peter'] );
foreach ($record as $employe) {
echo $employe['name'], ': ', $employe['email']."<br>";
}
?>
//Update
<?php
```

```
require 'vendor/autoload.php';
// Creating Connection
$con = new MongoDB\Client("mongodb://localhost:27017");
// Creating Database
$db = $con->javatpoint;
// Creating Document
$collection = $db->employee;
// Inserting Record
$collection->insertOne( [ 'name' =>'Peter', 'email' =>'peter@abc.com' ] );
// Fetching Record
$record = $collection->find( [ 'name' =>'Peter'] );
foreach ($record as $employe) {
echo $employe['name'], ': ', $employe['email']."<br>";
}
?>
//Update Program
<?php
// connect to mongodb
require 'vendor/autoload.php';
// Creating Connection
$con = new MongoDB\Client("mongodb://localhost:27017");
echo "Connection to database successfully";
// select a database
$db = $con->javapoint;
echo "Database javapoint selected";
$collection = $db->employee;
echo "Collection selected succsessfully";
// Fetching Record
```

```
$record = $collection->find( [ 'name' =>'Satish'] );
foreach ($record as $employe) {
echo $employe['name'], ': ', $employe['email']."<br>";
}
// Using PHP Library
$updateResult = $collection->updateOne(
   ['name' => 'Peter'],
   ['$set' => ['email' => 'mong@ab.com']]
     );
?>
```

Output/Results snippet:

References:

- HTML Introduction - https://www.w3schools.com/
- https://docs.mongodb.com/manual/reference/method/cursor.limit/

Learning outcome 5 - able to design and develop dynamic websites with PHP

After achieving this learning outcome, a student will be able to design and develop dynamic websites with PHP. In order to achieve this learning outcome, a student has to complete the following:

Activities:

1. Capturing Form Data Dealing with Multi-value field Generating File uploaded form (3Hrs)
2. Redirecting a form after submission (2Hrs)
3. Write a PHP script to get the PHP version and configuration information (2 Hrs)
4. Write a PHP script to display the strings (3Hr)
5. Create a simple HTML form and accept the user name and display the name through PHP echo statement (2 Hrs)
6. Write a e PHP script to display string, values within a table (3Hrs)
7. Write a PHP script to count lines in a file (2Hr)
8. Write a PHP function to test whether a number is greater than 30, 20 or 10usingternary operator (3H
9. Write a script which will display the string (2Hrs)
10. Write a PHP script which will display the colors (3Hrs)
11. Write a PHP script to sorting (3Hrs)
12. Write a PHP script to calculate and display average temperature, five lowest and highest temperatures in given data (2 Hr)
13. Write a program to calculate and print the factorial of a number using a for loop (2 Hr)
14. Write a PHP script using nested for loop (3Hrs)
15. Write a PHP program to generate and display the first n lines of a Floyd (2 Hrs)
16. Write a function to calculate the factorial of a number (2 Hr)
17. Write a function to check a number is prime or not(1Hr)
18. Write a function to reverse a string (1Hr)
19. Write a PHP function that checks whether a passed string is a palindrome or not? (2Hr)
20. Write a simple PHP class which displays the given string (2 Hr)
21. Write a PHP Calculator class which will accept two values as arguments, then add them, subtract them, multiply them together, or divide them on request (5 Hrs)
22. Write a PHP script to : - a) transform a string all uppercase letters.b)transform a string all lowercase letters. c) make a string's first character uppercase. d) make a string's first character of all the words uppercase (5 Hrs)

Activity 1

Aim: Capturing Form Data Dealing with Multi-value field Generating File uploaded form (Notepad)

Learning outcome: Able to design and develop dynamic websites with PHP.

Duration: 3 hour

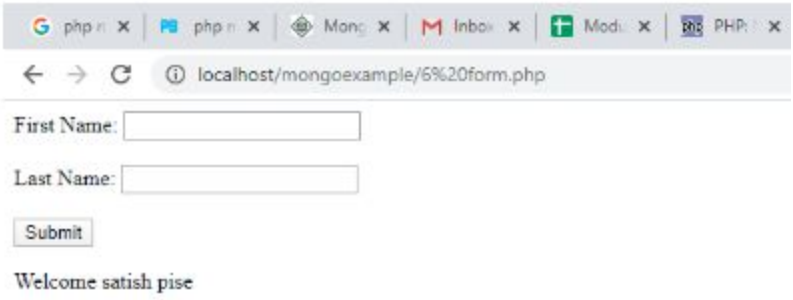
List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Apache 7.4

Program:

```
//6 form.php example
<html>
<body>
<form action="6 form.php" method="post">
First Name: <input type="text" name="name"><br><br>
Last Name: <input type="text" name="lname"><br><br>
<input type="submit">
</form>
</body>
</html>
<html>
<body>
Welcome <?php echo $_POST["name"]." "; ?>
<?php echo $_POST["lname"]; ?>
</body>
</html>
```

Output/Results snippet:



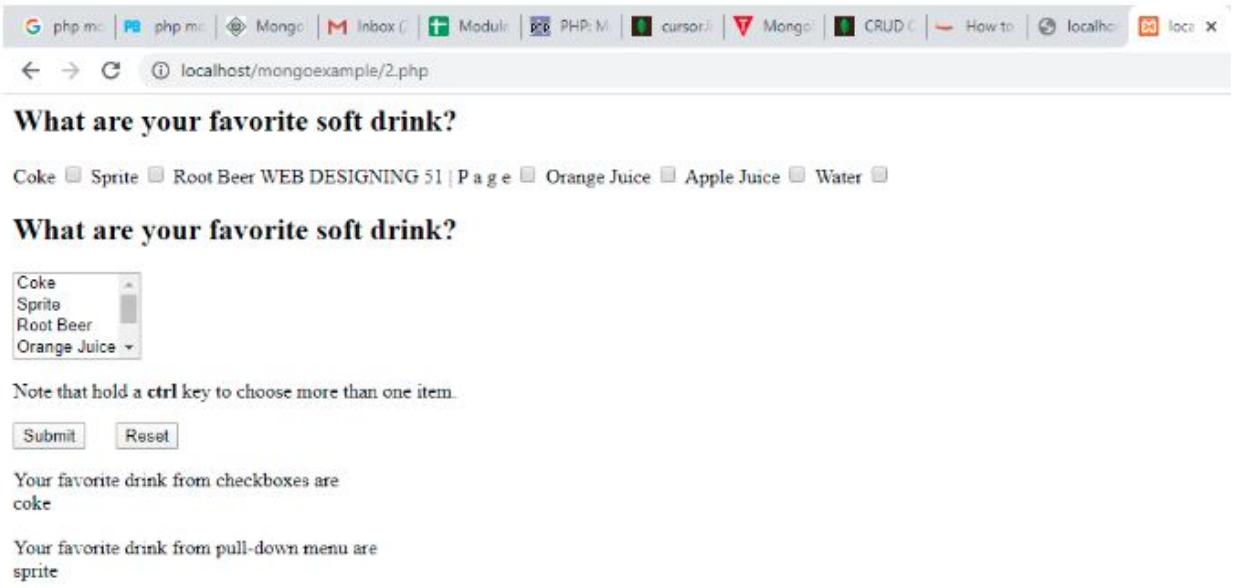
References:

- HTML Introduction - https://www.w3schools.com/
- PHP - https://www.w3schools.com/

```
// Example 2  multi field
<form action="2.php" method="post">
<h2>What are your favorite soft drink?</h2>
<label>Coke</label>
<input type="checkbox" name="drink[]" value="coke" />
<label>Sprite</label>
<input type="checkbox" name="drink[]" value="sprite" />
<label>Root Beer</label>
<input type="checkbox" name="drink[]" value="root beer" />
<label>Orange Juice</label>
<input type="checkbox" name="drink[]" value="orange juice" />
<label>Apple Juice</label>
<input type="checkbox" name="drink[]" value="apple juice" />
<label>Water</label>
<input type="checkbox" name="drink[]" value="water" />
```

```
<h2>What are your favorite soft drink?</h2>
<select name="favor[]" size = "4" multiple = "multiple">
<option value="coke">Coke</option>
<option value="sprite">Sprite</option>
<option value="root beer">Root Beer</option>
<option value="orange juice">Orange Juice</option>
<option value="apple juice">Apple Juice</option>
<option value="water">Water</option>
</select>
<p>Note that hold a <b>ctrl</b> key to choose more than one item.</p>
<input type="submit" name="submit" id="moveRight" value="Submit" />
<input type="reset" name="reset" value="Reset" style="margin-left: 20px;display:inline;" />
</form>
<?php
$drinklist=$_POST["drink"]; //assign an array to a local array
$favorlist=$_POST["favor"];
echo "Your favorite drink from checkboxes are <br />";
foreach($drinklist as $drink)
echo $drink . "<br /> ";
echo "<br />";
echo "Your favorite drink from pull-down menu are <br /> ";
foreach($favorlist as $favor)
echo $favor. "<br />";
?>
```

Output/Results snippet:



References:

- HTML Introduction - https://www.w3schools.com/
- PHP - https://www.w3schools.com/

```
// File Upload example html form
<!DOCTYPE html>
<html>
<body>
<form action="6-3upload.php" method="post" enctype="multipart/form-data">
    Select image to upload:
    <input type="file" name="fileToUpload" id="fileToUpload">
    <input type="submit" value="Upload Image" name="submit">
</form>
</body>
</html>
```

```
//File upload php file
<?php
$target_dir = "uploads/";
$target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);
$uploadOk = 1;
$imageFileType = strtolower(pathinfo($target_file,PATHINFO_EXTENSION));
// Check if image file is a actual image or fake image
if(isset($_POST["submit"])) {
    $check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
    if($check !== false) {
        echo "File is an image - " . $check["mime"] . ".";
        $uploadOk = 1;
    } else {
        echo "File is not an image.";
        $uploadOk = 0;
    }
}?>
```

Output/Results snippet:

References:

- HTML Introduction - https://www.w3schools.com/
- PHP - https://www.w3schools.com/

Activity 2

Aim: Redirecting a form after submission (Notepad)

Learning outcome: Able to design and develop dynamic websites with PHP.

Duration: 2 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Apache 7.4

Program:

```
<html>
 <body>
  <form action="https://google.com" method="POST">
   <input type="hidden" name="fullname" value="john" />
   <input type="hidden" name="address" value="street 2, 32 ave" />
   <input type="submit" value="Submit request" />
  </form>
 </body>
</html>
```

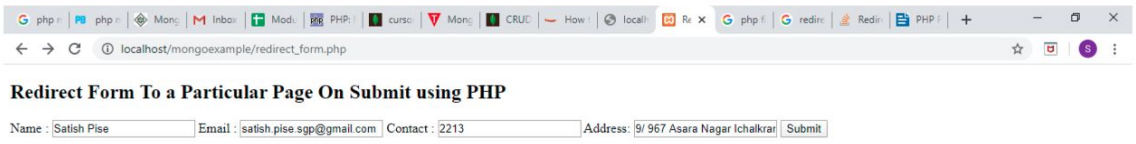
Output/Results snippet:

```
//PHP redirection – redirect_form.php
<!DOCTYPE html>
<html>
```

```
<head>
<title>Redirect Form To a Particular Page On Submit - Demo Preview</title>
<meta content="noindex, nofollow" name="robots">
<link href='css/redirect_form.css' rel='stylesheet' type='text/css'> <!--== Include CSS File Here ==-->
</head>
<body>
<div class="main">
<div class="first">
<h2>Redirect Form To a Particular Page On Submit using PHP</h2>
<form action="redirect_form.php" id="#form" method="post" name="#form">
<label>Name :</label>
<input id="name" name="name" placeholder='Your Name' type='text'>
<label>Email :</label>
<input id="email" name="email" placeholder='Valid Email Address' type='text'>
<label>Contact :</label>
<input id="contact" name="contact" placeholder='Contact' type='text'>
<label>Address:</label>
<input id="address" name="address" placeholder='Address' type='text' value="">
<input id='btn' name="submit" type='submit' value='Submit'>
<!---- Including PHP File Here ---->
<?php
include "redirect.php";
?>
</form>
</div>
</div>
</body>
</html>
```

```
//Redirect.php
<?php
if(isset($_POST['submit'])){
// Fetching variables of the form which travels in URL
$name = $_POST['name'];
$email = $_POST['email'];
$contact = $_POST['contact'];
$address = $_POST['address'];
if($name !="&& $email !="&& $contact !="&& $address !=")
{
//  To redirect form on a particular page
header("Location:https://www.formget.com/app/");
}
else{
?><span><?php echo "Please fill all fields.....!!!!!!!!!!!!";?></span> <?php
}
}
?>
```

Output/Results snippet:



References:

- HTML Introduction - https://www.w3schools.com/
- PHP - https://www.w3schools.com/

Activity 3

Aim: Write a PHP script to get the PHP version and configuration information **(**Notepad)

Learning outcome: Able to design and develop dynamic websites with PHP.

Duration: 2 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Apache 7.4

Program:

```
<?php

phpinfo();

?>
```

Output/Results snippet:

References:

- HTML Introduction - https://www.w3schools.com/
- PHP - https://www.w3schools.com/

Activity 4

Aim: Write a PHP script to display the strings (Notepad)

Learning outcome: Able to design and develop dynamic websites with PHP.

Duration: 2 hour

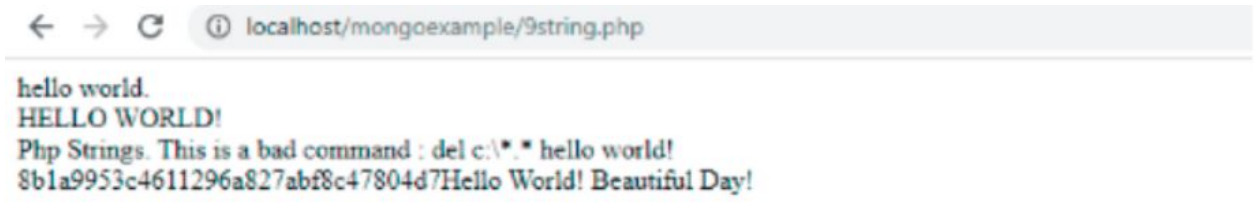
List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Apache 7.4

Program:

```
<?php

echo strtolower("Hello WORLD.<br>");

echo strtoupper("Hello WORLD!<br>");

echo "Php Strings."."\n";

echo "This is a bad command : del c:\\*.*"."\n";

echo lcfirst("Hello world!<br>");

$str = "Hello";

echo md5($str);

$arr = array('Hello','World!','Beautiful','Day!');

echo join(" ",$arr);

?>
```

Output/Results snippet:



References:

- PHP - https://www.w3schools.com/
- https://www.tutorialrepublic.com/faq/how-to-create-a-new-line-in-php.php

Activity 5

Aim: Create a simple HTML form and accept the user name and display the name through PHP echo statement (Notepad)

Learning outcome: Able to design and develop dynamic websites with PHP.

Duration: 2 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Apache 7.4

Program:

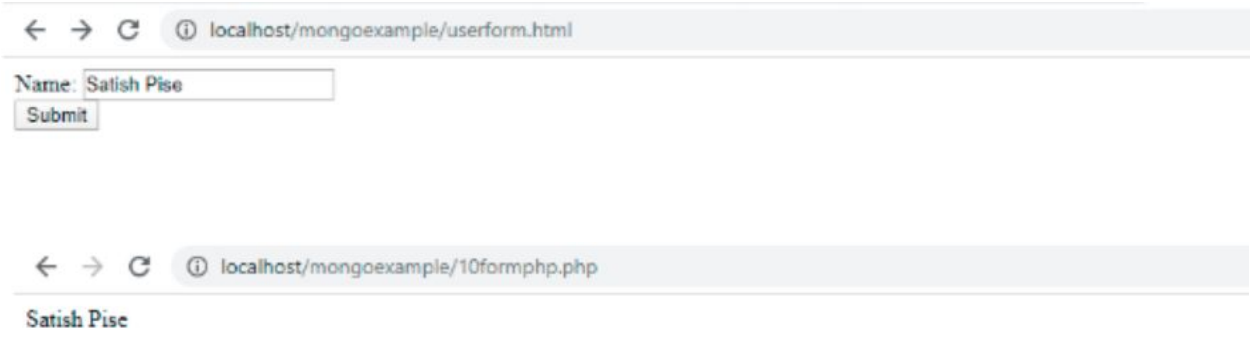
Userform.html

```
<html>
<body>
<form action="10formphp.php" method="post">
Name: <input type="text" name="name"><br>
<input type="submit">
</form> </body>
</html>
```

Formphp.php

```
<?php
echo $_POST["name"];
?>
```

Output/Results snippet:



References:

- PHP - https://www.w3schools.com/
- https://www.tutorialrepublic.com/faq/how-to-create-a-new-line-in-php.php

Activity 6

Aim: Write a PHP script to display string, values within a table

Learning outcome: Able to design and develop dynamic websites with PHP

Duration: 3Hrs

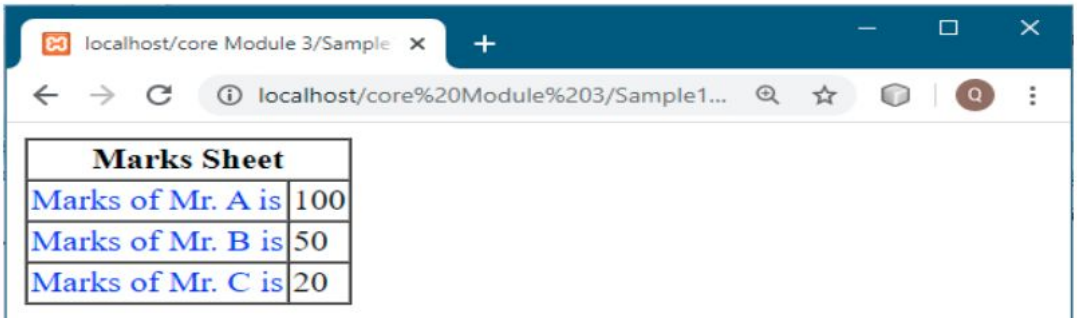
List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Notepad++
3. Xampp Server.

Program 1:

```
<?php
$a=100;
$b=50;
$c=20;
echo "<table border=1 cellspacing=0 cellpading=0>
<!— column  which spans 2 -->
<tr> <th colspan= 2> Marks Sheet </th></tr>
<tr> <td><font color=blue>Marks of Mr. A is</td>
<td>$a</font></td></tr>
<tr> <td><font color=blue>Marks of Mr. B is</td>
<td>$b</font></td></tr>
<tr> <td><font color=blue>Marks of Mr. C is</td>
<td>$c</font></td></tr>
</table>";
?>
```

Output/Results snippet:



Program 2:

```
<?php
echo "<table border=1 cellspacing=0 cellpading=0>
 <tr>
   <th>Name</th>
   <td>Diploma Course</td>
  </tr>
  <tr>
<!— row  which spans 2 -->
   <th rowspan=2>Offered by</th>
   <td>IBM</td>
  </tr>
  <tr>
   <td>Edunet</td>
  </tr>";
?>
```

Output/Results snippet:

Reference:

- https://www.w3resource.com/php-exercises/php-basic-exercise-13.php

Activity 7

Aim: Write a PHP script to count lines in a file

Learning outcome: Able to design and develop dynamic websites with PHP

Duration: 2 hour

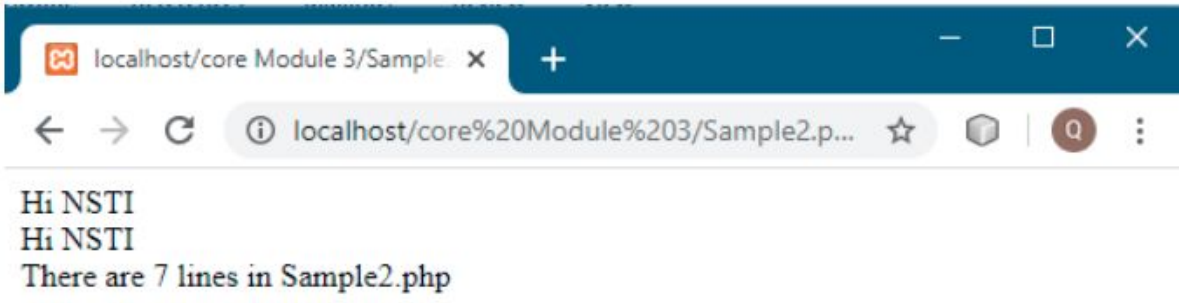
List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Notepad++
3. Xampp Server

Program:

```
<?php
// nl2br is used for newlines.
echo nl2br("Hi NSTI \n");
echo nl2br("Hi NSTI \n");
$file = basename($_SERVER['PHP_SELF']);
$no_of_lines = count(file($file));
echo "There are $no_of_lines lines in $file"."\n";
?>
```

Output/Results snippet:



Reference: https://www.w3resource.com/php-exercises/php-basic-exercise-16.php

Activity 8

Aim: Write a PHP function to test whether a number is greater than 30, 20 or 10 using ternary operator

Learning outcome: Able to design and develop dynamic websites with PHP

Duration: 3Hrs

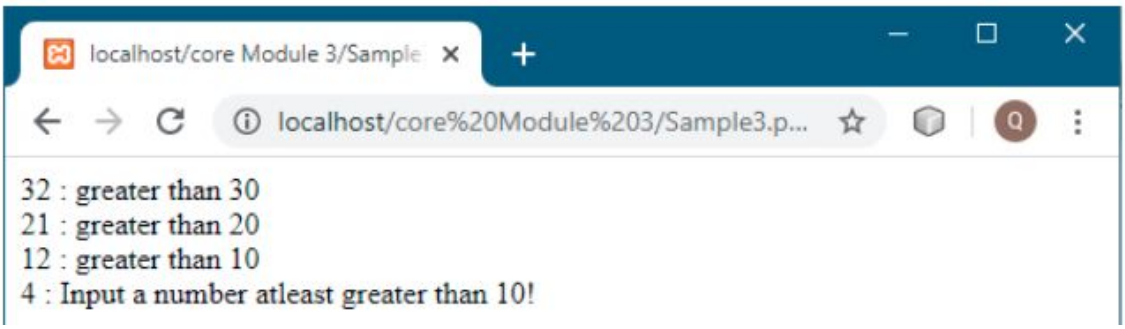
List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Notepad++
3. Xampp Server

Program 1:

```
<?php
//  Declaring a function called Test
function trinary_Test($n){
$r = $n > 30
? nl2br("greater than 30 \n")
: ($n > 20
? nl2br("greater than 20 \n")
: ($n >10
? nl2br("greater than 10 \n")
: "Input a number atleast greater than 10!"));
echo $n." : ".$r."\n";
}
// Calling the function Test
trinary_Test(32);
trinary_Test(21);
trinary_Test(12);
trinary_Test(4);
?>
```

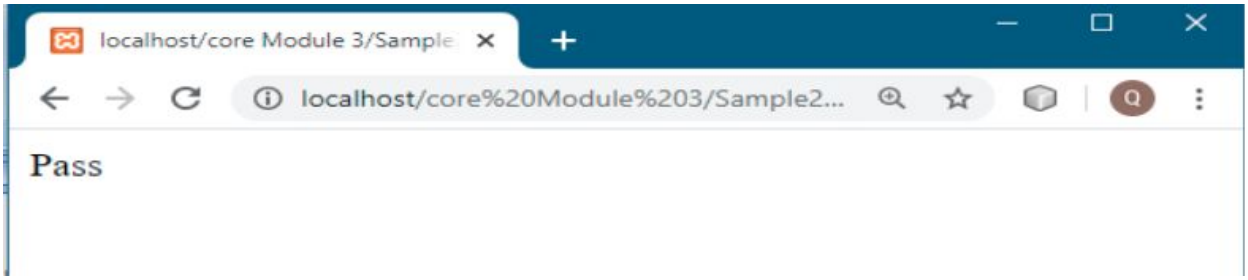
Output/Results snippet:



Program 2: Example of Ternary operators

```
<?php
$marks=50;
// using the print in php to display the output
print ($marks>=40) ? "Pass" : "Fail";
?>
```

Output/Results snippet:



Reference:

- https://www.w3resource.com/php-exercises/php-basic-exercise-21.php

Activity 9

Aim: Write a script which will display the string.

Learning outcome: Able to design and develop dynamic websites with PHP

Duration: 2 hours

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Notepad++
3. Xampp Server

Program:

```
<?php
    // using the echo in php to display the output
echo "<h2>PHP is Fun!</h2>";
echo "Hello world!<br>";
echo "I'm about to learn PHP!<br>";
echo "This ", "string ", "was ", "made ", "with multiple parameters.";
    // using the echo in php to display the output
print "<h2>PHP is Fun!</h2>";
print "Hello world!<br>";
print "I'm about to learn PHP!";
?>
```

Output/Results snippet:

References:

- https://www.w3schools.com/
- https://www.w3resource.com/

Activity 10

Aim: Write a PHP script which will display the array.

Learning outcome: Able to design and develop dynamic websites with PHP

Duration: 3 hours

List of Hardware/Software requirements:

1. Notepad++
2. Xampp Server

Program:

```
<?php
// Declaring a array
$color = array('white', 'green', 'red');
// using the foreach loop to loop through in array
foreach ($color as $c)
{
echo "$c, "; }
sort($color);                    // Sorting the array
// Displaying the content of array in unordered list
echo "<ul>";
foreach ($color as $y)
{ echo "<li>$y</li>";
} echo "</ul>";
?>
```

Output/Results snippet:

Reference:

- https://www.w3resource.com/php-exercises/php-array-exercise-2.php

Activity 11

Aim: Write a PHP script to sorting

Learning outcome: Able to design and develop dynamic websites with PHP

Duration: 3Hrs

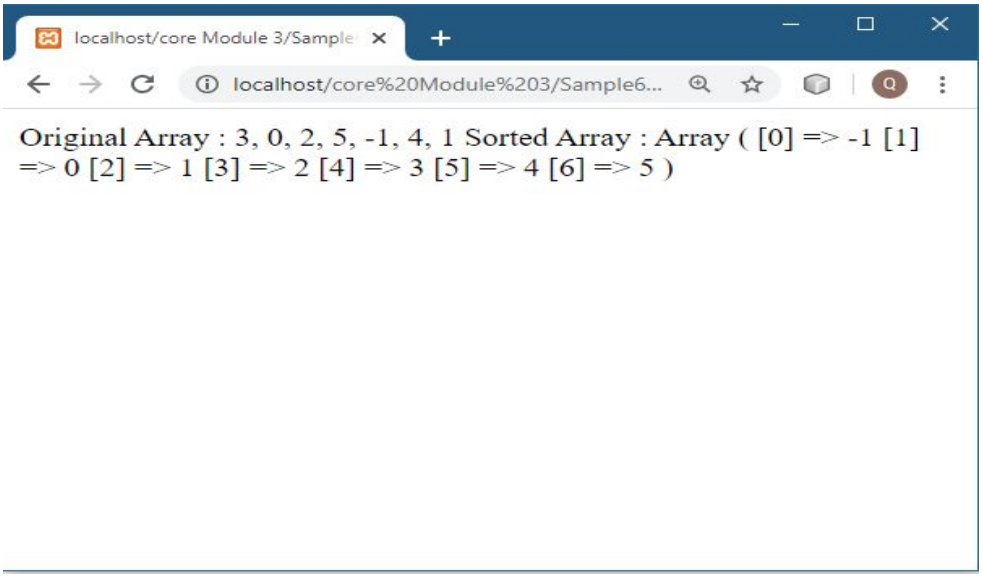
List of Hardware/Software requirements:

1. Notepad++
2. Xampp Server

Program:

```
<?php
// Writing the function to sort an array
// This function requires the parameter
function insertion_Sort($my_array)
{
for($i=0;$i<count($my_array);$i++){
$val = $my_array[$i];
$j = $i-1;
while($j>=0 && $my_array[$j] > $val){
$my_array[$j+1] = $my_array[$j];
$j--;
}
$my_array[$j+1] = $val;
}
return $my_array;
}
//Writing the array which need to be sort
$test_array = array(3, 0, 2, 5, -1, 4, 1);
echo "Original Array :\n";
echo implode(', ',$test_array );
echo "\nSorted Array :\n";
//Calling the function with parameter
//Displays the content of array in sorted manner
print_r(insertion_Sort($test_array));
?>
```

Output/Results snippet:



Reference:

- https://www.w3resource.com/php-exercises/searching-and-sorting-algorithm/searching-and-sorting-algorithm-exercise-3.php

Activity 12

Aim: Write a PHP script to calculate and display average temperature, five lowest and highest temperatures in given data

Learning outcome: Able to design and develop dynamic websites with PHP

Duration: 2Hrs

List of Hardware/Software requirements:

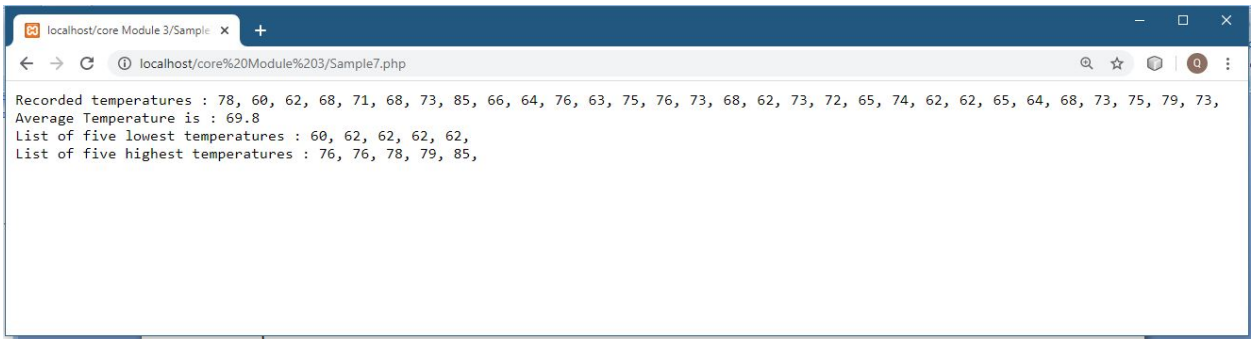
1. Notepad++
2. Xampp Server

Program:

```
<?php
echo "<pre>";
// Hard coded values for temperature
$temperatures = array(78, 60, 62, 68, 71, 68, 73, 85, 66, 64, 76, 63, 75, 76, 73, 68, 62, 73, 72, 65, 74,
    62, 62, 65, 64, 68, 73, 75, 79, 73);
// Declaring a parameterized function
// Function name is listvalues
function listvalues($value)
{
echo "$value, ";
}
// Function name is printAverage
function printAverage($array)
{
$total = 0;
foreach($array as $element)
{
$total += $element;
}
echo number_format($total / count($array), 1);
}
echo "Recorded temperatures : ";
array_walk($temperatures, "listvalues");
echo "<br>";
// Displaying the average of given temperatures
echo "Average Temperature is : ";
printAverage($temperatures);
echo "<br>";
```

```
//sort the temperatures in ascending order for both of the following lists.
sort($temperatures);
//print the first 5 values
echo "List of five lowest temperatures : ";
for($i = 0; $i < 5; $i++)
{
echo "$temperatures[$i], ";
}
echo "<br>";
//print the last 5 values
echo "List of five highest temperatures : ";
for($i = count($temperatures) - 5; $i <= count($temperatures) - 1; $i++)
{
echo "$temperatures[$i], ";
}
echo "<br>";
echo "</pre>";
?>
```

Output/Results snippet:



Reference:

- https://www.w3resource.com/php-exercises/php-array-exercise-9.php

Activity 13

Aim: Write a program to calculate and print the factorial of a number using a for loop.

Learning outcome: Able to design and develop dynamic websites with PHP

Duration: 2Hrs

List of Hardware/Software requirements:

1. Notepad++
2. Xampp Server

Program:

```
<?php
// Declaring a variable whose factorial is to be find
$n = 6;
$x = 1;
//Calculating the factorial of a given number
for($i=1;$i<=$n-1;$i++)
{
$x*=($i+1);
}
// Displaying the factorial of a given number
echo "The factorial of $n = $x"."\n";
?>
```

Output/Results snippet:

Activity 14

Aim: Write a PHP script using nested for loop

Learning outcome: Able to design and develop dynamic websites with PHP

Duration: 3Hrs

List of Hardware/Software requirements:

1. Notepad++
2. Xampp Server

Program:

```
<?php
$n=5;
// A for loop
for($i=1; $i<=$n; $i++)
{
//Declaring a for loop inside in the another for loop
// A nester for loop
for($j=1; $j<=$i; $j++)
{
echo ' * ';
}
echo nl2br("\n");
}
// A for loop
for($i=$n; $i>=1; $i--)
{
//Declaring a for loop inside in the another for loop
// A nester for loop
for($j=1; $j<=$i; $j++)
{
echo ' * ';
}
echo nl2br("\n ");
}
?>
```

Output/Results snippet:

Activity 15

Aim: Write a PHP program to generate and display the first n lines of a Floyd

Learning outcome: Able to design and develop dynamic websites with PHP

Duration: 2Hrs

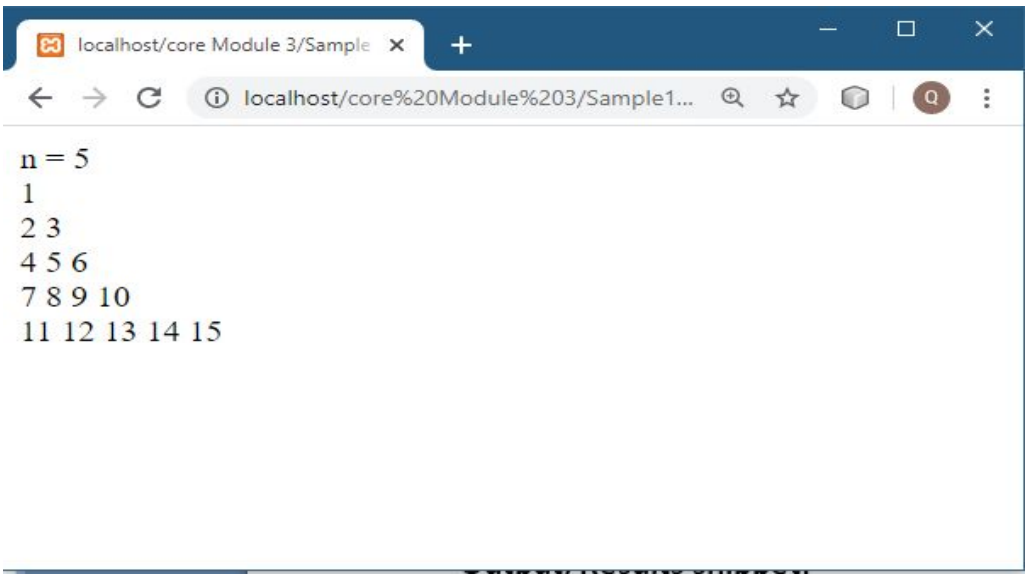
List of Hardware/Software requirements:

1. Notepad++
2. Xampp Server

Program:

```
<?php
// Declaring number of lines we need in a triangle
$n = 5;
echo "n = " . $n . "<br>";
//Specifying from where we need to print
$count = 1;
for ($i = $n; $i > 0; $i--)
{
for ($j = $i; $j < $n + 1; $j++)
{
printf("%4s", $count);
$count++;
}
echo "<br>";
}
?>
```

Output/Results snippet:



References:

- https://www.w3resource.com/php-exercises/php-for-loop-exercise-12.php
- https://www.w3resource.com/php-exercises/
- https://www.w3schools.com/php/default.asp
- https://www.javatpoint.com/php-tutoria

Activity 16

Aim: Write a function to calculate the factorial of a number

Learning outcome: Able to design and develop dynamic websites with PHP.

Duration: 2 hour

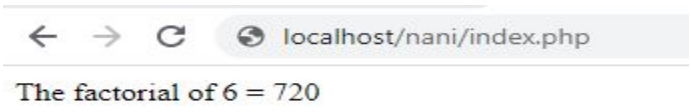
List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Lamp Server with PHP

Program:

```
<?php
// PHP code to get the factorial of a number
// function to get factorial in iterative way
function Factorial($number){
   if($number <= 1){
      return 1;
   }
   else{
      return $number * Factorial($number - 1);
   }
}
// Driver Code
$number = 10;
$fact = Factorial($number);
echo "Factorial = $fact";
?>
```

Output/Results snippet:



References:

1. https://www.javatpoint.com/php-factorial-Program

Activity 17

Aim: Write a function to check a number is prime or not

Learning outcome: Able to design and develop dynamic websites with PHP.

Duration: 1 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. LAMP Server with PHP

Program:

```
<?php
function primeCheck($number){ //creating a function primeCheck
 if ($number == 1)
 return 0;
 for ($i = 2; $i <= $number/2; $i++){
 if ($number % $i == 0)
 return 0; }
 return 1;
}
$number = 31;                                    //input number 31
$flag = primeCheck($number);        //checking prime or not
if ($flag == 1)
echo "$number is Prime";
else
echo "$number is Not Prime"
?>
```

Output/Results snippet:

References:

1. https://www.w3resource.com/php-exercises/php-function-exercise-2.php

Activity 18

Aim: Write a function to reverse a string

Learning outcome: Able to design and develop dynamic websites with PHP.

Duration: 1 hour

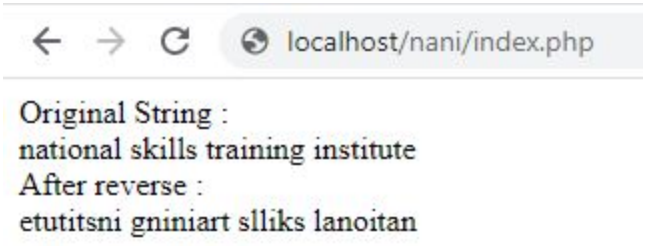
List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. LAMP Server with PHP

Program:

```
<?php
function Reverse($str){       //creating a function Reverse and passing $str as parameter
 return strrev($str);         //used strrev built-in function to get reverse of string
}
$str = "national skills training institute";      //input for $str as a string
echo "Original String :<br>";                          // Display original string
echo $str."<br>";
echo "After reverse : <br>";                      //Display reverse of string
echo Reverse($str);
?>
```

Output/Results snippet:



References:

1. https://www.geeksforgeeks.org/php-reverse-string/

Activity 19

Aim: Write a PHP function that checks whether a passed string is a palindrome or not?

Learning outcome: Able to design and develop dynamic websites with PHP.

Duration: 2 hours

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. LAMP Server with PHP

Program:

```
<?php
function check_palindrome($string) //creating a function check_palindrome
{
 if ($string == strrev($string))  //comparing string with reverse of same string
 return 1;                        //return 1 if reverse and original string is same
 else
 return 0;                        //return 0 if reverse and original strings are not same
}
$string="madam";   //taking example string as madam
if(check_palindrome($string)==1)    //checking palindrome or not
echo "$string is Palindrome";          //display the output
else
echo "$string is Not a palindrome";
?>
```

Output/Results snippet:

References:

1. https://www.geeksforgeeks.org/php-palindrome-check/

Activity 20

Aim: Write a simple PHP class which displays the given string

Learning outcome: Able to design and develop dynamic websites with PHP.

Duration: 2 hours

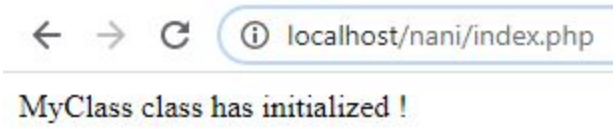
List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. LAMP Server with PHP

Program:

```
<?php
class MyClass { //creating a class  'MyClass' class name
 public function __construct() //creating a constructor for class
 {
 echo 'MyClass class has initialized !'."\n";
 }
}
$userclass = new MyClass; //creating object for class
?>
```

Output/Results snippet:



References:

1. https://www.w3resource.com/php-exercises/php-class-exercise-1.php

Activity 21

Aim: Write a PHP Calculator class which will accept two values as arguments, then add them, subtract them, multiply them together, or divide them on request

Learning outcome: Able to design and develop dynamic websites with PHP.

Duration: 5 hours

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. LAMP Server with PHP

Program:

```
<?php
class MyCalculator { //class name MyCalculator
private $_fval, $_sval; //Data members declared as private
public function __construct( $fval, $sval ) { //passing parameters to the function
$this->_fval = $fval; //initializing the data members
$this->_sval = $sval; //initializing the data members
}
public function add() {   // creating a add function
return $this->_fval + $this->_sval; //adding the values
}
public function subtract() { //creating the subtract function
return $this->_fval - $this->_sval; //subtraction is done here
}
public function multiply() { //creating the multiply function
return $this->_fval * $this->_sval; //multiplication is done here
}
public function divide() { //creating the divide function
return $this->_fval / $this->_sval; //division is done here
}}
$mycalc = new MyCalculator(12, 6); //passing the parameter values to the function 12,6
echo "addition ",$mycalc-> add()."<br>"; // Displays addition 12+6=18
echo "multiplication ",$mycalc-> multiply()."<br>"; // Displays  multiplication 12*6=72
echo "substraction ",$mycalc-> subtract()."<br>"; // Displays subtraction 12-6=6
echo "division ",$mycalc-> divide(); // Displays division 12/6=2
?>
```

Output/Results snippet:

References:

1. https://www.w3resource.com/php-exercises/php-class-exercise-6.php

Activity 22

Aim: Write a PHP script to: - a) transform a string all uppercase letters. b) transform a string all lowercase letters. c) make a string's first character uppercase. d) make a string's first character of all the words

Learning outcome: Able to design and develop dynamic websites with PHP.

Duration: 5 hours

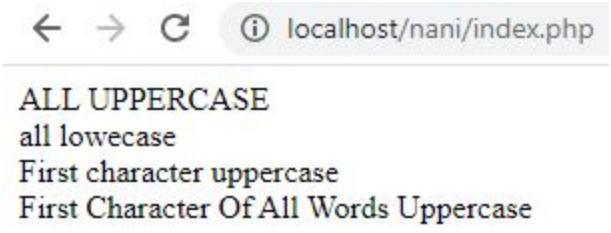
List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Lamp Server with PHP

Program:

```
<?php
//strtoupper is in-built function used for to make uppercase letters
print(strtoupper("all uppercase<br>"))."\n";
//strtolower is in-built function used for to make all into lower letters
print(strtolower("all lowecase<br>"))."\n";
// ucfirst is in-built function used for to make first character in a statement as Uppercase
print(ucfirst("first character uppercase<br>"))."\n";
// ucwords is in-built function used for to make first character of every word in a sentence as
    Uppercase
print(ucwords("first character of all words uppercase"))."\n";
?>
```

Output/Results snippet:



References:

1. https://www.w3resource.com/php-exercises/php-string-exercise-1.php

Learning outcome 6 - able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python

After achieving this learning outcome, a student will be able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python. In order to achieve this learning outcome, a student has to complete the following:

Activities:

1. Print a string using print statement (2Hrs)
2. Print given string using indentation (space between characters) (1Hrs)
3. Define Integer Variables, floating variables and string variables (1Hr)
4. Write a program to add numbers and strings to the correct list using the append list method (2Hrs)
5. Write a python program to add, subtract, multiply and divide given two numbers by using arithmetic operators(2Hrs)
6. Write a python program multiplying strings to form string with repeating sequence (2Hrs)
7. Write a Python program to get the largest number from a list by using max and mini commands (1Hr)
8. Write a Python program to find whether a given number (accept from the user) is even or odd by using if else command (2Hrs)
9. Write a Python program to create a histogram from a given list of integers by using for while loop (1Hrs)
10. Write a Python program to compute the greatest common divisor (GCD) of two positive integers by using loops (2Hrs)
11. Write a Python program to get the least common multiple (LCM) of two positive integers using if else and while commands(2Hrs)
12. Write a Python program to sort (ascending and descending) a dictionary by value (2Hrs)
13. Write a Python program to create a tuple. (2Hrs)
14. Write a Python program to create a tuple with different data types (1Hrs)
15. Write a Python program to create a set (2Hrs)
16. Write a Python program to add member(s) in a set (1 Hrs)
17. Write a Python program to find maximum and the minimum value in a set. (1 Hrs)
18. Write a Python program to find the length of a set(1 Hrs)
19. Write a Python program to convert temperatures to and from CentigradetoFahrenheit (2Hrs)
20. Write a python program to find Fibonacci series (2Hrs)
21. Write a python program to find factorial using function (2 Hrs)
22. Write a python program to find whether the given string is palindrome or not by using function (2Hrs)
23. Write a python class to reverse a string word by word (2Hrs)
24. Write a python class named as circle by a radius and two methods of computer area and perimeter of a circle (3 Hrs)

25. Write a python program to sort a list of elements using bubble sort algorithm (2Hrs)
26. Write a python program to copy content of a file to another file (3Hrs)
27. Write a python program to find the frequency of words in a file (2Hrs)

Activity 1

Aim: Write a Python program to print a string using the print statement

Learning outcome: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

Duration: 4 hours

List of Hardware/Software requirements:

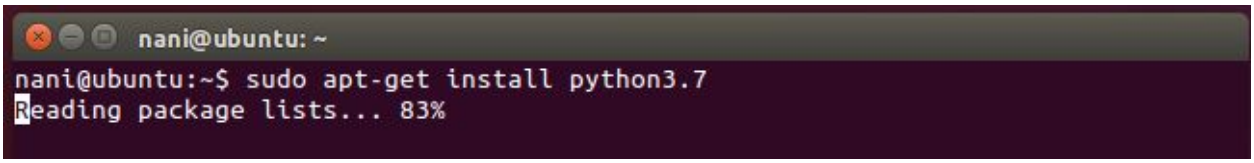
1. Laptop/Computer with Windows/Linux OS - Ubuntu 18.04 LTS
2. Python Software

Program:

Installation steps for Linux:

- Ctrl+Alt+T Open Terminal
- Type Command : sudo apt-get update

- Type command : sudo apt-get install python3.7



- After installation check python version

- Start programming

```
nani@ubuntu: ~
nani@ubuntu:~$ python3
Python 3.4.3 (default, Nov 12 2018, 22:25:49)
[GCC 4.8.4] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print ("Welcome to Python Practical") #print() is a library function to disp
lay the text on screen
Welcome to Python Practical
>>>
```

Installation Steps for Windows:

- Install Python software in the system.
- Open the browser and type the python.org/downloads.
- Click on download .
- Choice either Windows x86-64 executable installer for 64-bit or Windows x86 executable installer for 32-bit.
- After downloading a file the below page will appear.

Figure 18: Installing

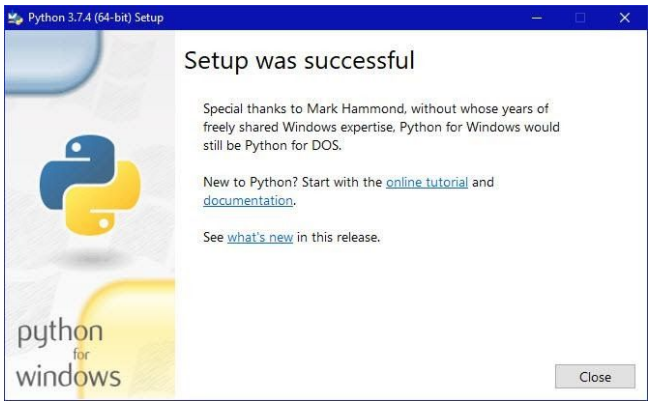


Figure 19: Installed successfully

print ("Welcome to Python Practical") #print() is a library function to display the text on screen

Output/Results snippet:

References:

1. https://www.ics.uci.edu/~pattis/common/handouts/pythoneclipsejava/python.html

Activity 2

Aim: Write a python program to print given string using indentation (space between characters)

Learning outcome: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

Duration: 1 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Python3 Software

Program:

```
print("\n\n\th e l l o \n \t \t \t w o r l d \n") # tab \t, new line \n and space is used
```

Output/Results snippet:

References:

1. https://stackoverflow.com/questions/18756510/printing-with-indentation-in-python

Activity 3

Aim: Define Integer Variables, floating variables and string variables.

Learning outcome: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

Duration: 1 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Python3 Software

Program:

```
myint = 7                       # declare integer variable
print(myint)                    # print the value of the variable
myfloat = 7.0                           # declare floating variable
print(myfloat)
myfloat = float(7)
print(myfloat)
mystring = 'hello'              # declare string variable
print(mystring)
mystring = "hello"
print(mystring)
```

Output:

7

7.0

7.0

hello

hello

References:

- https://www.tutorialspoint.com/python/python_overview.htm
- https://www.w3schools.com/python/
- https://www.javatpoint.com/python-tutorial

Activity 4

Aim: Write a program to add numbers and strings to the correct list using the append list method.

Learning outcome: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

Duration: 2 hours

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Python3 Software

Program:

```
numbers = [1,2,3]
strings = ["Hello","World"]                    # declare and initialize list
names = ["John", "Eric", "Jessica"]
second_name = names[1]
print(numbers)                                 # print the values of list
print(strings)
print("The second name on the names list is %s" % second_name)
```

Output:

[1, 2, 3]

['Hello', 'World']

The second name on the names list is ['Eric']

References:

- https://www.tutorialspoint.com/python/python_overview.htm
- https://www.w3schools.com/python/
- https://www.javatpoint.com/python-tutorial

Activity 5

Aim: Write a python program to add, subtract, multiply and divide given two numbers by using arithmetic operators.

Learning outcome: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

Duration: 2 hours

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Python3 Software

Program:

```
print("1. Addition");
print("2. Subtraction");
print("3. Multiplication");
print("4. Division");
print("5. Exit");
choice = int(input("Enter your choice: "));          # take input from console
if (choice>=1 and choice<=4):                        # taking input from 1 to 4
    print("Enter two numbers: ");
    num1 = int(input());
    num2 = int(input());
if choice == 1:                                          # perform addition if 1
    res = num1 + num2;
    print("Result = ", res);
elif choice == 2:                                    # perform subtraction if 2
    res = num1 - num2;
    print("Result = ", res);
elif choice == 3:                                    # perform multiplication if 3
    res = num1 * num2;
    print("Result = ", res);
elif choice == 4:                                    # perform division if 4
    res = num1 / num2;
    print("Result = ", res);
elif choice == 5:
    exit();                                          # exit if 5
else:
    print("Wrong input..!!");                        # print message for any other input
```

Output:

1. Addition

2. Subtraction

3. Multiplication

4. Division

5. Exit

Enter your choice: 1

Enter two numbers:

2

4

('Result = ', 6)

References:

- https://www.tutorialspoint.com/python/python_overview.htm
- https://www.w3schools.com/python/
- https://www.javatpoint.com/python-tutorial

Activity 6

Aim: Write a python program multiplying strings to form a string with repeating sequence.

Learning outcome: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

Duration: 2 hours

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Python3 Software

Program:

```
x = 'Welcome '
y = 'python '
print ((x+y)*5)                    # printing the concatenated strings 5 times
```

Output:

Welcome python Welcome python Welcome python Welcome python Welcome python

References:

- https://www.tutorialspoint.com/python/python_overview.htm
- https://www.w3schools.com/python/
- https://www.javatpoint.com/python-tutorial

Activity 7

Aim: Write a Python program to get the largest number from a list by using max and min commands.

Learning outcome: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

Duration: 1 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Python3 Software

Program:

```
list = [1, 2, 3]
print (max(list))          # max() finds the maximum among values
print (min(list))          # min() finds the minimum among values
```

Output:

3

1

References:

- https://www.tutorialspoint.com/python/python_overview.htm
- https://www.w3schools.com/python/
- https://www.javatpoint.com/python-tutorial

Activity 8

Aim: Write a Python program to find whether a given number (accept from the user) is even or odd by using if else command.

Learning outcome: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

Duration: 2 hours

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Python3 Software

Program:

```
num = int(input("Enter a number: "))          # taking input from console
mod = num % 2                                 # extracting the remainder value of input
if mod > 0:
    print("This is an odd number.")
else:
    print("This is an even number.")
```

Output:

Enter a number: 5

This is an odd number.

Enter a number: 2

This is an even number.

References:

- https://www.tutorialspoint.com/python/python_overview.htm
- https://www.w3schools.com/python/
- https://www.javatpoint.com/python-tutorial

Activity 9

Aim: Write a Python program to create a histogram from a given list of integers by using for while loop.

Learning outcome: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

Duration: 1 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Python3 Software

Program:

```
def histogram( items ):                         # create function
    for n in items:                             # for loop through values of n
            output = ''
            times = n
            while( times > 0 ):                 # check condition of counter
                    output += '*'
                    times = times–1         # decrement the counter
            print(output)                       # print the output
histogram([2, 3, 6, 5])
```

Output:

**

***

******

*****

References:

- https://www.tutorialspoint.com/python/python_overview.htm
- https://www.w3schools.com/python/
- https://www.javatpoint.com/python-tutorial

Activity 10

Aim: Write a Python program to compute the greatest common divisor (GCD) of two positive integers by using loop

Learning outcome: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

Duration: 2 hours

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Python3 Software

Program:

```
def gcd(x, y):                                          # create function gcd()
    gcd = 1
    if x % y == 0:
            return y
    for k in range(int(y / 2), 0, -1):          # for loop from y/2 to 0 increment -1
            if x % k == 0 and y % k == 0:
                    gcd = k
                    break
    return gcd
print(gcd(12, 17))                                      # print the gcd value
print(gcd(4, 6))
```

Output:

```
1
2
```

References:

- https://www.tutorialspoint.com/python/python_overview.htm
- https://www.w3schools.com/python/
- https://www.javatpoint.com/python-tutorial

Activity 11

Aim: Write a Python program to get the least common multiple (LCM) of two positive integers using if else and while commands.

Learning outcome: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

Duration: 2 hours

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Python3 Software

Program:

```
def lcm(x, y):                                  # define function
    if x > y:
            z = x
    else:
            z = y
    while(True):
            if((z % x == 0) and (z % y == 0)):  # check if both conditions are true
                    lcm = z
                    break
            z += 1
    return lcm
print(lcm(4, 6))                                # print the lcm value
print(lcm(15, 17))
```

Output:

```
12
255
```

References:

- https://www.tutorialspoint.com/python/python_overview.htm
- https://www.w3schools.com/python/
- https://www.javatpoint.com/python-tutorial

Activity 12

Aim: Write a Python program to sort (ascending and descending) a dictionary by value

Learning outcome: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

Duration: 2 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Python 3 software

Program:

```
import operator
d = {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}
print('Original dictionary : ',d)
sorted_d = sorted(d.items(), key=operator.itemgetter(0))
print('Dictionary in ascending order by value : ',sorted_d)
sorted_d = sorted(d.items(), key=operator.itemgetter(0),reverse=True)
print('Dictionary in descending order by value : ',sorted_d)
```

Output/Results snippet:

References:

- https://www.w3resource.com/python-exercises/dictionary/python-data-type-dictionary-exercise-1.php
- https://stackoverflow.com/questions/20577840/python-dictionary-sorting-in-descending-order-based-on-values/41866830
- https://www.youtube.com/watch?v=trWU2GDqXS4
- https://www.youtube.com/watch?v=-UGJvJNZ7i4

Activity 13

Aim: Write a Python program to create a tuple.

Learning outcome: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

Duration: 2 hour

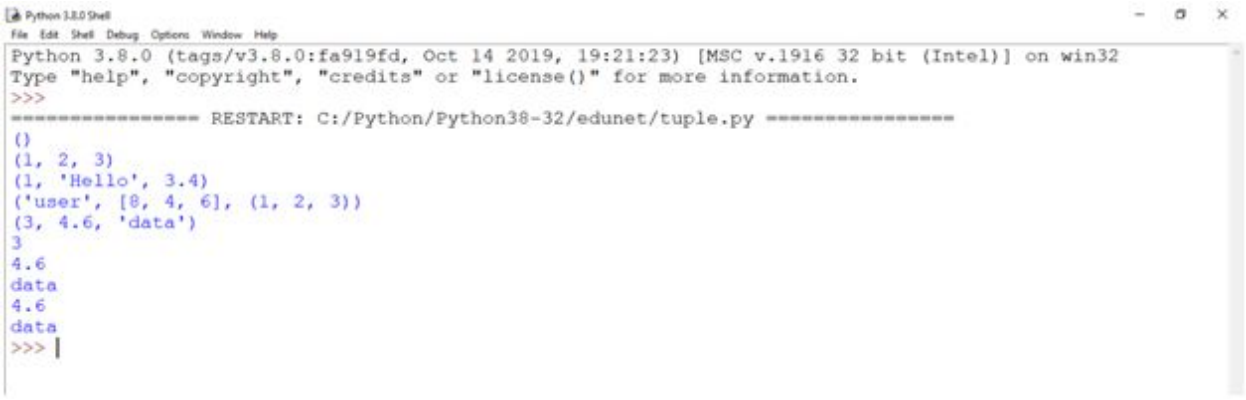
List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Python 3 software

Program:

```
# empty tuple
# Output: ()
my_tuple = ()
print(my_tuple)
# tuple having integers
# Output: (1, 2, 3)
my_tuple = (1, 2, 3)
print(my_tuple)
# tuple with mixed data types
# Output: (1, "Hello", 3.4)
my_tuple = (1, "Hello", 3.4)
print(my_tuple)
# nested tuple
# Output: ("user", [8, 4, 6], (1, 2, 3))
my_tuple = ("user", [8, 4, 6], (1, 2, 3))
print(my_tuple)
# tuple can be created without parentheses
# also called tuple packing
# Output: 3, 4.6, "data"
my_tuple = 3, 4.6, "data"
print(my_tuple)
# tuple unpacking is also possible
```

Output/Results snippet:



References:

- https://www.w3schools.com/python/python_tuples.asp
- https://www.w3resource.com/python-exercises/tuple/python-tuple-exercise-3.php
- https://www.programiz.com/python-programming/tuple
- https://www.geeksforgeeks.org/tuples-in-python/

Activity 14

Aim: Write a Python program to create a tuple with different data types

Learning outcome: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

Duration: 1 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Python 3 software

Program:

```
#Create a tuple with different data types
 tuplex = ("tuple", False, 3.2, 1)
print(tuplex)
```

Output/Results snippet:

References:

- https://www.w3resource.com/python-exercises/tuple/python-tuple-exercise-2.php
- https://www.youtube.com/watch?v=eXnZfHwzSiI
- http://python.mykvs.in/Programs/class%20xi/cs/tuple%20p.pdf

Activity 15

Aim: Write a Python program to create a set

Learning outcome: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

Duration: 2 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Python 3 software

Program:

```
#A new empty set
setx = set()
print(setx)
set()
#A non empty set
n = set([0, 1, 2, 3, 4, 5])
print(n)
```

Output/Results snippet:

References:

- https://www.w3schools.com/python/python_sets.asp
- https://www.geeksforgeeks.org/python-sets/
- https://www.youtube.com/watch?v=MEPlLAjPvXY

Activity 16

Aim: Write a Python program to add member(s) in a set

Learning outcome: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

Duration: 1 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Python 3 software

Program:

```
#A new empty set
color_set = set()
#Add a single member
color_set.add("Red")
print(color_set)
#Add multiple items
color_set.update(["Blue", "Green"])
print(color_set)
```

Output/Results snippet:

References:

- https://www.w3resource.com/python-exercises/sets/python-sets-exercise-3.php
- https://www.geeksforgeeks.org/set-add-python/

Activity 17

Aim: Write a Python program to find maximum and the minimum value in a set.

Learning outcome: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

Duration: 1 hour

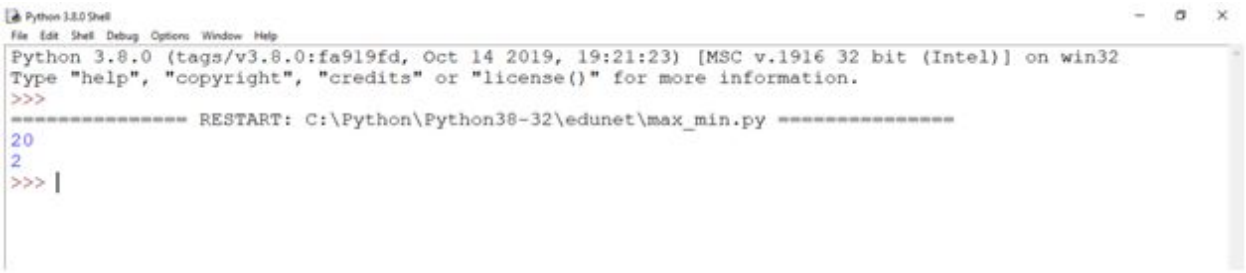
List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Python 3 software

Program:

```
#Create a set
seta = set([5, 10, 3, 15, 2, 20])
#Find maximum value
print(max(seta))
#Find minimum value
 print(min(seta))
```

Output/Results snippet:



References:

- https://www.w3resource.com/python-exercises/sets/python-sets-exercise-14.php
- https://www.geeksforgeeks.org/python-maximum-minimum-set/

Activity 18

Aim: Write a Python program to find the length of a set (1 Hrs)

Learning outcome: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

Duration: 1 hour

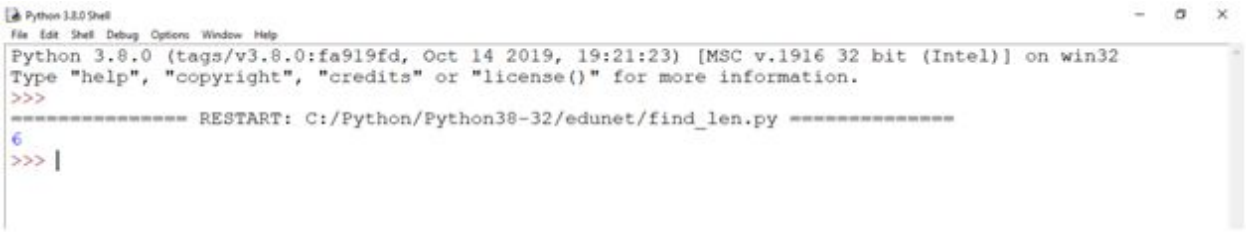
List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Python 3 software

Program:

```
#Create a set
seta = set([5, 10, 3, 15, 2, 20])
#Find the length use len()
print(len(seta))
```

Output/Results snippet:



References:

- https://www.w3resource.com/python-exercises/sets/python-sets-exercise-15.php
- https://www.edureka.co/blog/python-list-length/
- https://www.youtube.com/watch?v=hbVekSSSzVM

Activity 19

Aim: Write a Python program to convert temperatures to and from Centigrade to Fahrenheit

Learning outcome: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

Duration: 2 hour

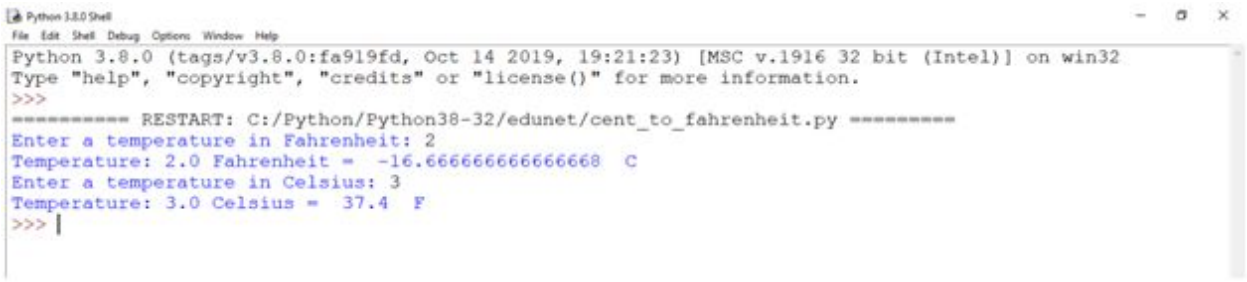
List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Python 3 software

Program:

```
Fahrenheit =  float(input("Enter a temperature in Fahrenheit: "))
Celsius = (Fahrenheit - 32) * 5.0/9.0
print ("Temperature:", Fahrenheit, "Fahrenheit = ", Celsius, " C")
Celsius =  float(input("Enter a temperature in Celsius: "))
Fahrenheit = 9.0/5.0 * Celsius + 32
print ("Temperature:", Celsius, "Celsius = ", Fahrenheit, " F")
```

Output/Results snippet:



References:

- https://beginnersbook.com/2019/05/python-Program-to-convert-celsius-to-fahrenheit-and-vice-versa/
- https://www.youtube.com/watch?v=_fxLlOO0Pts
- https://www.Programming-techniques.com/2019/03/python-Program-to-convert-celsius-to-fahrenheit-and-vice-versa.html

Activity 20

Aim: Write a python program to find Fibonacci series

Learning outcome: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

Duration: 2 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Python 3 software

Program:

```
pterms = int(input("How many terms? "))
n1 = 0
n2 = 1
count = 0
if pterms <= 0:
 print("Please enter a positive integer")
elif pterms == 1:
 print("Fibonacci sequence upto",pterms,":")
 print(n1)
else:
 print("Fibonacci sequence upto",pterms,":")
 while count < pterms:
    print(n1,end='\n')
     nth = n1 + n2
     n1 = n2
     n2 = nth
     count += 1
```

Output/Results snippet:

Activity 21

Aim: Write a python program to find factorial using function in Python IDLE.

Learning outcome: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

Duration: 2 hour

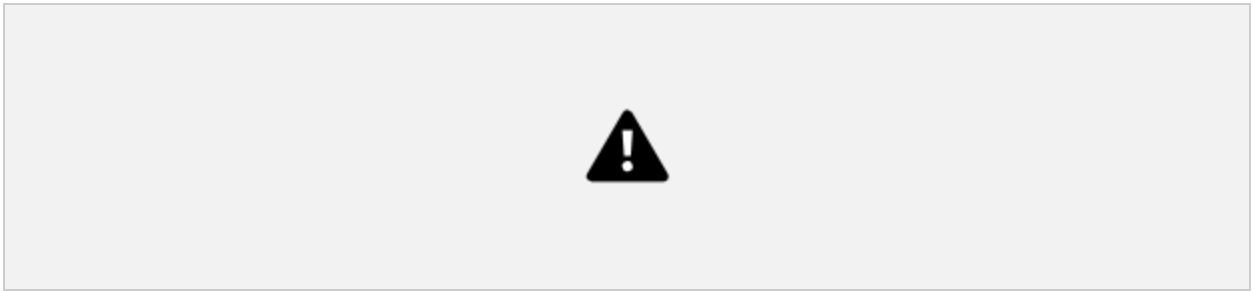
List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Python 3 software

Program:

```
# Function name using recursion_function(n)
def recursion_factorial(n):
if n == 1:
return n
else:
return n*recursion_factorial(n-1)
# read the value of input for factorial number
num = int(input('\n Enter your factorial number '))
# check if the number is negative
if num < 0:
print("Sorry, factorial does not exist its negative numbers")
elif num == 0:
print("The factorial of 0 is 1")
else:
print("The factorial of", num, "is", recursion_factorial(num))
```

Output/Results snippet



Activity 22

Aim: Write a python program to find whether the given string is palindrome or not by using function.

Learning outcome: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

Duration: 2 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Python 3 software

Program:

```
#enter input of the word
Word = str(input("Enter a word :"))
#check the word whether palindrome or not
if(Word==Word[::-1]):
print("Your Word is palindrome")
else:
print("Your Word isn't palindrome")
```

Output/Results Snippet

References

- https://docs.python.org/3/tutorial/
- https://www.tutorialsteacher.com/python
- https://realpython.com/

Activity 23

Aim: Write a python class to reverse a string word by word.

Learning outcome: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

Duration: 2 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Python 3 software

Program:

```
# Python program to Reverse each word of a string
# function definition
def reverseword(input_word):
w = input_word.split(" ")
# Splitting the string into a list of words
# reversing each word and creating a new list of words
nw = [i[::-1] for i in w]
# Joining the new list of words to form a new string
ns = " ".join(nw)
return ns
# main() method
input_word = input("Enter the string: ")
print(reverseword(input_word))
```

Output/Result Snippet

```
Python 3.7.7 (tags/v3.7.7:d7c567b08f, Mar 10 2020, 10:41:24) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:\Users\Ragavan\AppData\Local\Programs\Python\Python37\reverse.py =
Enter the string: PYTHON PROGRAM
NOHTYP MARGORP
>>>
```

References

- https://docs.python.org/3/tutorial/
- https://www.tutorialsteacher.com/python
- https://realpython.com/

Activity 24

Aim: Write a python class named as circle by a radius and two methods of computer area and perimeter of a circle.

Learning outcome: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

Duration: 2 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Python 3 software

Program:

```
#using math package
import math
#using math function calculate the value
class circle():
def __init__(self,radius):
self.radius=radius
def area(self):
return math.pi*(self.radius**2)
def perimeter(self):
return 2*math.pi*self.radius
#read value of area input from user
value_of_circle=int(input("Enter radius of circle: "))
#Object for Class
obj=circle(value_of_circle)
print("Area of circle:",round(obj.area(),2))
print("Perimeter of circle:",round(obj.perimeter(),2))
```

Output/Result Snippet

Activity 25

Aim: Write a python program to sort a list of elements using bubble sort algorithm.

Learning outcome: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

Duration: 2 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Python 3 software

Program:

```
# Python Program for Bubble Sort
def bubblesort(a, number):
for i in range(number -1):
for j in range(number - i - 1):
if(a[j] > a[j + 1]):
temp = a[j]
a[j] = a[j + 1]
a[j + 1] = temp
a = []
number = int(input("Please Enter the Total Number of Elements : "))
for i in range(number):
value = int(input("Please enter the %d Element of List1 : " %i))
a.append(value)
bubblesort(a, number)
print("The Sorted List in Ascending Order : ", a)
```

Note: Here, we are using Nested for Loop to iterate each element in a given List. Inside the loop, we are using the If statement to sort items in an ascending order using Bubble Sort

Output Snippet

Activity 26

Aim: Write a python program to copy content of a file to another file (3Hrs)

Learning outcome: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

Duration: 2 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Python 3 software

Program:

```
#Input files for Read data
#Output file to copy content from input file
with open("input.txt" ,"r") as f:
with open("output.txt", "w") as f1:
#writes content from existing to new file
for line in f:
f1.write(line)
#display content of copying text
print("Copy content text is : ",line)
#close files
f.close()
f1.close()
```

Output Snippet

References

- https://docs.python.org/3/tutorial/
- https://www.tutorialsteacher.com/python
- https://realpython.com/

Activity 27

Aim: Write a python program to find the frequency of words in a file.

Learning outcome: Able to make websites, web servers, game frameworks, desktop and CLI applications, and IDE using Python.

Duration: 2 hour

List of Hardware/Software requirements:

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. Python 3 software

Program:

```
#import counter for read data
from collections import Counter
def word_count(fname):
#Open a file and read text
with open(fname) as f:
#Separating a text word by word using split()
return Counter(f.read().split())
print("Number of words in the file :",word_count("output.txt"))
```

Output Snippet

References

- https://docs.python.org/3/tutorial/
- https://www.tutorialsteacher.com/python
- https://realpython.com/

Annexure

1. Installing Notepad++

a. Download Notepad++. You can download Notepad++ by clicking here. Click on the "Download" button, and the program will begin downloading to your computer.
   Click here Download 32-bit x86
   Click here Download 64-bit x64
b. Double-click on the downloaded installer to start the installation. It is a wizard-driven installation.
c. Let's start…….
   Select language.
d. Click next and follow instructions