

KLS's Gogte Institute of Technology

Department of Computer Science and Engineering

Lab Manual

INTRODUCTION TO WEB PROGRAMMING

(22PLC15A/25A)

Faculty Coordinator:

Prof.Veena V kangralkar

Prof. Seena M Kalghatgi

INTRODUCTION TO WEB PROGRAMMING

LABORATORY

Programming Assignments:

1. Create an XHTML page using tags to accomplish the following:
 - (i) A paragraph containing text “All that glitters is not gold”. Bold face and italicize this text
 - (ii) Create equation: $x = 1/3(y_1^2 + z_1^2)$
 - (iii) Put a background image to a page and demonstrate all attributes of background image.
 - (iv) Create unordered list of 5 fruits and ordered list of 3 flowers.

2. Create following table using XHTML tags. Properly align cells, give suitable cell padding and cell spacing, and apply background color, bold and emphasis necessary.

Department	Sem1	<i>SubjectA</i>
		<i>SubjectB</i>
		<i>SubjectC</i>
	Sem2	<i>SubjectE</i>
		<i>SubjectF</i>
		<i>SubjectG</i>
	Sem3	<i>SubjectH</i>
		<i>SubjectI</i>
		<i>SubjectJ</i>

3. Use HTML5 for performing following tasks:
 - (i) Draw a square using HTML5 SVG , fill the square with green color and make 6px brown stroke width.

(ii) Write the following mathematical expression by using HTML5 MathML $d = x^2 - y^2$.

(iii) Redirecting current page to another page after 5 seconds using HTML5 meta tag.

4. Demonstrate the following HTML5 Semantic tags- <article>, <aside>, <details>, <figcaption>, <figure>, <footer>, <header>, <main>, <mark>, <section> for a web page that gives information about travel experience.

5. Create a class called income, and make it a background color of #0ff.

Create a class called expenses, and make it a background color of #f0f.

Create a class called profit, and make it a background color of #f00.

Throughout the document, any text that mentions income, expenses, or profit, attach the appropriate class to that piece of text. Further create following line of text in the same document:

~~The current price is 50₹~~ and new price is 40₹.

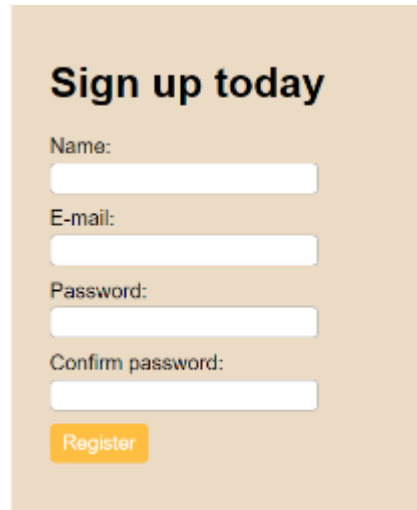
6. Change the tag li to have the following properties:

- ☐ A display status of inline
- ☐ A medium, double-lined, black border
- ☐ No list style type

Add the following properties to the style for li:

- ☐ Margin of 5px
- ☐ Padding of 10px to the top, 20px to the right, 10px to the bottom, and 20px to the left. Also demonstrate list style type with user defined image logos.

7. Create following web page using HTML and CSS with tabular layout.



Sign up today

Name:

E-mail:

Password:

Confirm password:

8. Create following calculator interface with HTML and CSS.



Note: In the examination each student picks one question from the lot of all 10 questions

Termwork 1:

1. Create an XHTML page using tags to accomplish the following:

- (i) A paragraph containing text “All that glitters is not gold”. Bold face and italicize this text
- (ii) Create equation: $x = 1/3(y_1^2 + z_1^2)$
- (iii) Put a background image to a page and demonstrate all attributes of background image.
- (iv) Create unordered list of 5 fruits and ordered list of 3 flowers.

Theory:

XHTML stands for EXtensible HyperText Markup Language. The XHTML 1.0 is the first document type in the XHTML family. XHTML was developed by World Wide Web Consortium (W3C) to help web developers make the transition from HTML to XML.

Benefits of XHTML

Strict syntax: XHTML has a stricter syntax compared to HTML5, which means that it is more structured and easier to read.

Standardization: XHTML follows the syntax rules of XML, which is a standardized markup language. This makes it easier to create interoperable web pages that work well with different web browsers and devices.

Better for parsing: XHTML is easier to parse and process than HTML5, which makes it a better choice for developers who want to create web pages that can be easily processed by other software tools.

The Most Important Differences from HTML

- 1. `<!DOCTYPE>` is mandatory.
- 2. The `xmlns` attribute in `<html>` is mandatory.
- 3. `<html>`, `<head>`, `<title>`, and `<body>` are mandatory.
- 4. Elements must always be properly nested.

5. Elements must always be closed.
6. Elements must always be in lowercase.
7. Attribute names must always be in lowercase.
8. Attribute values must always be quoted.
9. Attribute minimization is forbidden.

XHTML Tags

	Defines bold text
	Defines emphasized text
<i>	Defines italic text
<p>	Defines a paragraph
<pre>	Defines preformatted text
<sub>	Defines subscripted text
<sup>	Defines superscripted text
	Defines a list item
<h1> to <h6>	Defines header 1 to header 6
<body>	Defines the body element
	Defines an image
	Defines an ordered list
	Defines an unordered list

Along with these, <a>,
, <hr>, <table>, <th>, <tr>, <td>, , <small>, , <meta>. and many more.

XHTML code:

```

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE      html      PUBLIC      "-//W3C//DTD      XHTML      1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">

<head>

<title>SAMPLE XHTML PAGE</title>

</head>

<body>

<style>

body {

    background-image: url("image.png");

    background-repeat: no-repeat;

    background-position: right bottom;

    background-attachment: fixed;

}

</style>

<h4>Paragraph</h4>

<p>

<b><i>All that glitters is not gold</i></b> is an aphorism stating that not everything that looks
precious or true turns out to be so.

```

While early expressions of the idea are known from at least the 12th-13th century, the current saying is derived from a 16th-century line by William Shakespeare,

<i>All that glisters is not gold</i>.

**

**

**<i>All that glisters is not gold</i>
**

Often have you heard that told.

Many a man his life hath sold

But my outside to behold.

Gilded tombs do worms enfold.

Had you been as wise as bold,

Young in limbs, in judgment old,

Your answer had not been inscrolled

Fare you well. Your suit is cold

-William Shakespeare, Merchant of Venice, Act II Scene 7

</p>

<h4>Equation</h4>

$$\langle i \rangle x \langle /i \rangle = 1/3(\langle i \rangle y \langle /i \rangle \langle \text{sub} \rangle 1 \langle / \text{sub} \rangle \langle \text{sup} \rangle 2 \langle / \text{sup} \rangle + \langle i \rangle z \langle /i \rangle \langle \text{sub} \rangle 1 \langle / \text{sub} \rangle \langle \text{sup} \rangle 2 \langle / \text{sup} \rangle)$$

<h4>Unordered Fruit List</h4>

Banana

Mango

Grapes

Apples

<h4>Ordered Flower List</h4>


```
<li>Jasmine</li>
```

```
<li>Rose</li>
```

```
<li>Lotus</li>
```

```
<li>Tulip</li>
```

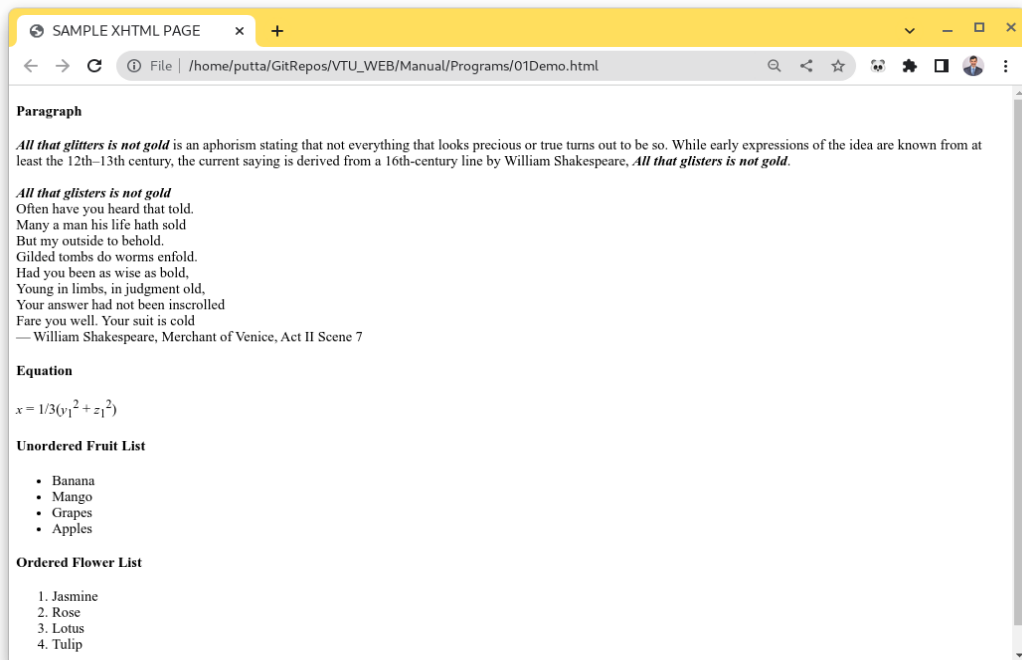
```
</ol>
```

```
<br />
```

```
</body>
```

```
</html>
```

OUTPUT:



Termwork 2

Create following table using XHTML tags. Properly align cells, give suitable cell padding and cell spacing, and apply background color, bold and emphasis necessary.

Department	Sem1	<i>SubjectA</i>
		<i>SubjectB</i>
		<i>SubjectC</i>
	Sem2	<i>SubjectE</i>
		<i>SubjectF</i>
		<i>SubjectG</i>
	Sem3	<i>SubjectH</i>
		<i>SubjectI</i>
		<i>SubjectJ</i>

Theory :

What is HTML Tables?

An HTML Table is an arrangement of data in rows and columns in tabular format. Tables are useful for various tasks, such as presenting text information and numerical data. A table is a useful tool for quickly and easily finding connections between different types of data. Tables are also used to create databases.

HTML Tags	Descriptions
<code><table></code>	Defines the structure for organizing data in rows and columns within a web page.
<code><tr></code>	Represents a row within an HTML table, containing individual cells.
<code><th></code>	Shows a table header cell that typically holds titles or headings.
<code><td></code>	Represents a standard data cell, holding content or data.
<code><caption></code>	Provides a title or description for the entire table.

● Adding Border to an HTML Table

By default, HTML tables do not have a border. We can set the border using the CSS border property.

● Adding Collapsed Borders to an HTML Table

The borders around the cells and the table are separated from each other, by default. We can use the CSS border-collapse property to add a collapsed border in HTML.

● Adding Padding to a Table

To add more space to the table cells, we can use the CSS padding property.

● Spanning Multiple Rows and Columns

With the rowspan and colspan attributes, we can make table rows and columns extend across multiple other rows and columns. In simple words, we can merge two or more rows into a single row; or two or more columns into a single column using the rowspan and colspan attributes respectively.

● Adding Captions to Tables

We can use the <caption> element to add a caption or title to our table. The <caption> element should be present after the opening <table> tag. By default, the caption is displayed at the top of the table.

● Adding a Background Color to a Table

We can add a background to an HTML table using the background-color option.

XHTML Code:

```

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/ xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">

<head>

<title>Table Demo XHTML PAGE</title>

<style>
table, th, td {

border: 1px solid black;
border-collapse: collapse;
}
th, td{
padding-left: 10px;
padding-bottom: 20px
}
table {
border-spacing: 30px;
}
</style>
</head>

<body>

<h3>Tables in XHTML</h3>

<table align="center" width="70%" style="height:450px">

<tr >

```

```

<td rowspan="9" align="center" bgcolor=DEEEEEE>
<b>Department</b>
</td>
<td rowspan="3" align="center" bgcolor=9E7BA0>
<b>Sem1</b>
</td>
<td padding:15px>
<em>SubjectA</em>
</td>
</tr>
<tr>
<td ><em>SubjectB</em></td>
</tr>
<tr>
<td ><em>SubjectC</em></td>
</tr>
<tr>
<td rowspan="3" align="center" bgcolor=9E7BA0>
<b>Sem2</b>
</td>
<td ><em>SubjectE</em></td>
</tr>
<tr>
<td ><em>SubjectF</em></td>
</tr>
<tr>

```

```

<td ><em>SubjectG</em></td>

</tr>

<tr>

65

<td rowspan="3" align="center" bgcolor=9E7BA0>

<b>Sem3</b>

</td>

<td ><em>SubjectH</em></td>

</tr>

<tr>

<td ><em>SubjectI</em></td>

</tr>

<tr>

<td ><em>SubjectJ</em></td>

</tr>

</table>

</body>

</html>

```

OUTPUT

Table Demo XHTML PAGE x +

File | /home/putta/GitRepos/VTU_WEB/Manual/Programs/02Table.html

Tables in XHTML

Department	Sem1	SubjectA
		SubjectB
		SubjectC
	Sem2	SubjectE
		SubjectF
		SubjectG
	Sem3	SubjectH
		SubjectI
		SubjectJ

Termwork 3:

Use HTML5 for performing following tasks:

- (i) Draw a square using HTML5 SVG , fill the square with green color and make 6px brown stroke width.
- (ii) Write the following mathematical expression by using HTML5 MathML $d = x^2 - y^2$.
- (iii) Redirecting current page to another page after 5 seconds using HTML5 meta tag.

Theory:

SVG stands for Scalable Vector Graphics. It defines vector-based graphics in XML format. They are graphics that are scalable, and do not lose any quality if they are zoomed or resized. SVG is supported by all major browsers.

How does it Work?

SVG has elements and attributes for rectangles, circles, ellipses, lines, polygons, curves, and more. SVG also supports filter and blur effects, gradients, rotations, animations, interactivity with JavaScript, and more. A simple SVG document consists of the `<svg>` root element and several basic shape elements that will build a graphic together.

SVG Rectangle - `<rect>`

The `<rect>` element is used to create a rectangle and variations of a rectangle shape. The `<rect>` element has six basic attributes to position and shape the rectangle:

Attribute	Description
width	Required. The width of the rectangle
height	Required. The height of the rectangle
x	The x-position for the top-left corner of the rectangle
y	The y-position for the top-left corner of the rectangle
rx	The x radius of the corners of the rectangle (used to round the corners). Default is 0
ry	The y radius of the corners of the rectangle (used to round the corners). Default is 0

- The `fill` attribute defines the fill color of the rectangle
- The `stroke` property defines the color of the border of the rectangle.

- The CSS `stroke-width` property defines the width of the border of the rectangle.
- The CSS `fill-opacity` property defines the opacity of the fill color (legal range: 0 to 1)
- The CSS `stroke-opacity` property defines the opacity of the stroke color (legal range: 0 to 1).

Mathematical Markup Language (MathML) is an XML-based language for describing mathematical notation.

- The **<math>** MathML element is the top-level MathML element, used to write a single mathematical formula.
- The **<mi>** MathML element indicates that the content should be rendered as an identifier, such as a function name, variable or symbolic constant.
- By default **<mi>** elements that contain multiple characters are rendered as normal text, while single character characters are rendered as italic
- The **<mn>** MathML element represents a numeric literal which is normally a sequence of digits with a possible separator (a dot or a comma).
- The **<mo>** MathML element represents an operator in a broad sense. Besides operators in strict mathematical meaning, this element also includes "operators" like parentheses, separators like comma and semicolon, or "absolute value" bars.
- The **<mrow>** MathML element is used to group sub-expressions, which usually contain one or more operators with their respective operands (such as **<mi>** and **<mn>**). This element renders as a horizontal row containing its arguments.
- The **<msub>** MathML element is used to attach a subscript to an expression.

The <meta> tag : Defines metadata about an HTML document. Metadata is data (information) about data. **<meta>** tags always go inside the **<head>** element, and are typically used to specify character set, page description, keywords, author of the document, and viewport settings.

Metadata will not be displayed on the page, but is machine parsable. Metadata is used by browsers (how to display content or reload page), search engines (keywords), and other web services.

The *charset* attribute specifies the character encoding for the HTML document. The HTML5 specification encourages web developers to use the UTF-8 character set, which covers almost all of the characters and symbols in the world!

```
<meta charset="utf-8" />
```

Refresh document after 5 seconds and redirect it to VTU website.:

```
<meta http-equiv="refresh" content="5; URL=http://www.vtu.ac.in">
```

HTML Code:

```

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>HTML5 Demo</title>

<meta http-equiv="refresh" content="5; URL=http://www.vtu.ac.in">

</head>

<body>

<h3>HTML5 SVG</h3>

<svg width="200" height="200" align="centre">

<rect x="50" y="50" width="100" height="100" fill="green" stroke="brown"
stroke-width="6px"/>

</svg>

<h3>HTML5 MathML</h3>

<math xmlns = "http://www.w3.org/1998/Math/MathML">

<mrow>

<msup><mi>d</mi></msup>

<mo> = </mo>

<msup><mi>x</mi><mn>2</mn></msup>

<mo>-</mo>

```

```

<msup><mi>y</mi><mn>2</mn></msup>

</mrow>

</math>

<h3>This page redirects in 5 seconds</h3>

</body>

</html>

```

OUTPUT:

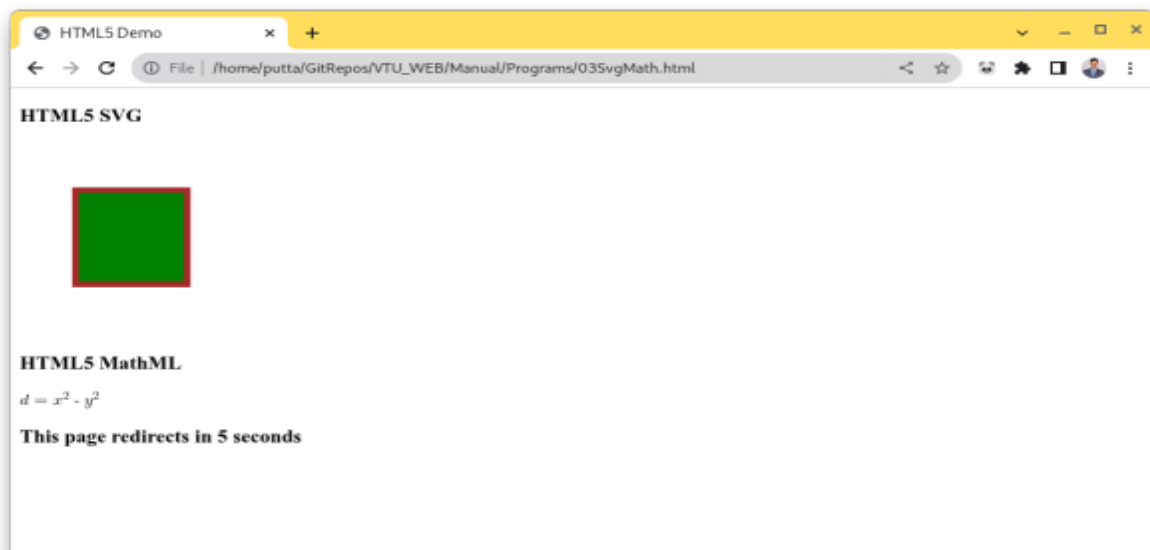


Figure 3.1: Initial Page



Figure 3.2: Redirected Page

Termwork 4:

Demonstrate the following HTML5 Semantic tags- <article>, <aside>, <details>, <figcaption>, <figure>, <footer>, <header>, <main>, <mark>, <section> for a web page that gives information about travel experience.

Theory: Semantic elements = elements with a meaning. A semantic element clearly describes its meaning to both the browser and the developer.

Examples of non-semantic elements: <div> and - Tells nothing about its content.

Examples of semantic elements: <form>, <table>, and <article> - Clearly defines its content.

A Guide to Semantic Tags in HTML

<header>	Defines the header of a page or section.	Used for introductory content or navigational links.
<footer>	Specifies the footer of a document or section.	Contains authorship, copyright, contact info, etc.
<nav>	Defines a set of navigation links.	Used for major navigation blocks.
<section>	Represents a standalone section in a document.	Groups thematically related content.
<article>	Encapsulates a self-contained composition.	For blog posts, news articles, etc.,
<aside>	Marks content aside from its surrounding content.	For sidebars, related links, etc.
<main>	Specifies the main content of a document.	Contains the unique content of a document.

HTML5 Semantic Tags for Text

Tag Name	Description	Use
<mark>	Highlights parts of the text.	Used to spotlight important or highlighted text.
<time>	Represents a date/time.	Marks dates, times, or durations.
	Emphasizes text.	For stressing emphasis.
	Indicates strong importance.	Used for strong emphasis.
<small>	Defines smaller text.	For fine print or side comments.
<cite>	Denotes the title of a creative work.	Used for citing titles like books, songs, etc.
<q>	Defines a short inline quotation.	For embedding short quotes in text.

All these tags are part of the HTML5 specification and are supported by modern web browsers.

HTML CODE:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>HTML5 Semantic Tags Demo</title>

<style>

  body{

background-color: #FFDDDD;

}

  aside {

float: right;

width: 25%;

background-color: cyan;

font-style: italic;

padding

}

  main {

float: left;

width: 70%;

}

  footer {

position: fixed;
```

left: 0;

25

bottom: 0;

26

width: 100%;

27

text-align: center;

28

}

29

mark {

30

background-color: yellow;

31

color: black;

32 *}*

33

figure {

34

display: inline-block;

35

margin: auto;

36 *}*

37

38 *figcaption {*

9 CHAPTER 4. HTML5 SEMANTIC TAGS

Department of computer Science & Engineering,
KLS's GIT, Belagavi

39

font-style: italic;

40 }

41

</style>

42

</head>

<body>

<article>

46

<header>

47

<h1>My Travelogue</h1>

48

<p>Random Escapades</p>

49

</header>

<main>

<figure>

**

<figcaption>The road never ends</figcaption>

</figure>

<section>

<h2>Ooty</h2>

<p>Ooty is a popular hill station located in the Nilgiri Hills. It is popularly called the **"Queen of Hill Stations"**.**</p>**

</section>

62

63 **<section>**

64 **<h2>Mysore</h2>**

65 **<p>** The city is also known as the City of Palaces, Mysuru has always enchanted its visitors with its quaint charm.**</p>**

66 **</section>**

67

68 **</main>**

69

70 **<aside>**

71 **<section>**

72 **<p>**Upcoming Trek planned to **<mark>Kumara Parvata</mark>** will be sharing detils soon**</p>**

73 **<details>**

74

<summary>Tentative Dates</summary>

75

<p>24th January 2023**</p>**

76 **</details>**

77 **</section>**

```

78 </aside>
79
80
<footer>
81
<p>© 2023 Prabodh C P</p>
82
</footer>
83
84 </article>
85 </body>
86 </html>

```

OUTPUT:



Figure 4.1: Initial Page

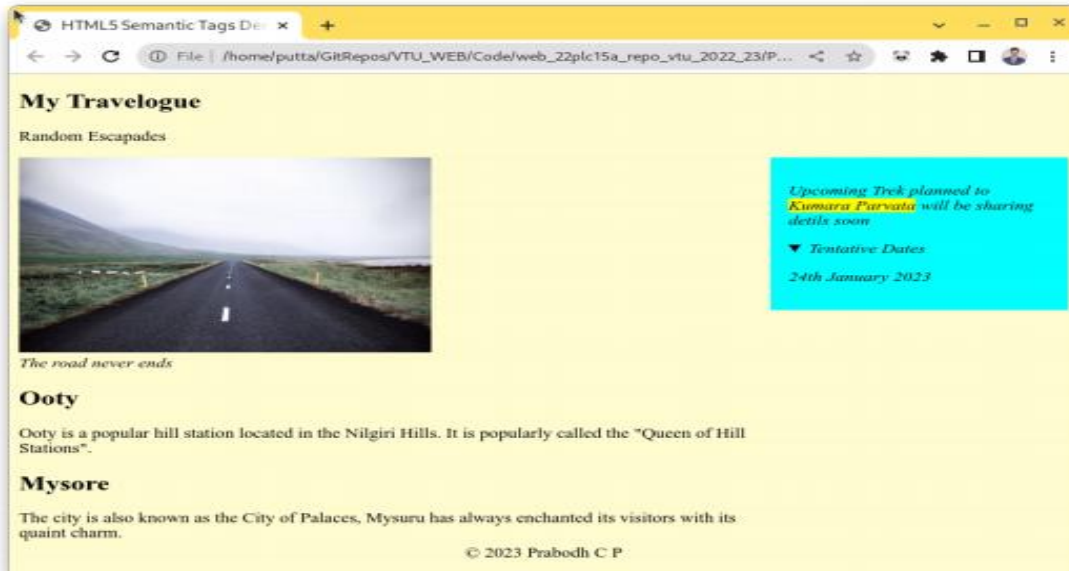


Figure 4.2: After clicking on Details

Termwork 5:

Create a class called **income**, and make it a background color of #0ff.

Create a class called **expenses**, and make it a background color of #f0f.

Create a class called **profit**, and make it a background color of #f00.

Throughout the document, any text that mentions income, expenses, or profit, attach the appropriate class to that piece of text. Further create following line of text in the same document:

~~The current price is 50 ₹~~ and new price is 40 ₹.

Theory:

- The `.class` selector selects elements with a specific class attribute.

To select elements with a specific class, write a period (.) character, followed by the name of the class.

You can also specify that only specific HTML elements should be affected by a class. To do this, start with the element name, then write the period (.) character, followed by the name of the class

CSS Syntax

```
.class {
    css declarations;
}
```

- **What are the Uses of Class Selector (".class") in CSS?**

A Class selector in CSS is used to style elements based on their class attributes.

A Class selector in CSS selects HTML elements having the specified class and applies the specified styles to those elements.

Class selectors are used to add specific styles to elements based on their specified class name.

A Class selector in CSS can be used to select multiple elements at the same time having the same class name and apply the same style to them.

● What is CSS Text Decoration?

In CSS (Cascading style sheet), we have some feature text decoration that allows us to change how the text looks on a website. With the help of text decoration, we can change the text's color and add various visual effects like underlining, overlining, line through, and bolding. It makes the text more attractive and easily readable.

We can apply these effects with the help of the text-decoration property, which can be applied to a class of elements or a single element.

In web pages, we use the text-decoration property for element to change how text or element looks. You can add visual effects like blinking, underlining, overlining, and line-through. This property applies to inline elements like anchor links, headings, and text within paragraphs.

Syntax:

Here is the syntax of text decoration in CSS:

```
/* Applying a single text decoration */

selector {

    text-decoration: value;

}

/* Applying multiple text decorations */

selector {

    text-decoration: value1 value2 value3 ...;

}
```

The various options for the text-decoration property:

- **None:** The element's text decorations are removed when this value is used, which is the default. There won't be any extra lines or styles added to the text; it will display normally.
- **Underline:** adds a line beneath the text to underline it. This is a common way to make hyperlinks (also known as anchor links) clickable.
- **Overline:** outlines the text with a line above it. With this design, a line is drawn over the text.
- **Line-through:** Insert a line through the text's center. The text has been crossed out or removed.

HTML Code:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Class Selectors Demo</title>

<style>

.income {background-color: #0ff; font-style: italic;}

.expenses {background-color: #f0f;font-style: oblique;}

.profit {background-color: #ff0;font-weight: bold;}

.red{color: red;font-size: 24px;}

.strike{text-decoration: line-through; font-size: 24px;}

p {font-family: Cursive;}

</style>

</head>

<body>

<h1>Class Selectors Demo</h1>

<p>

Income, expenses, and profit are financial terms that are used to measure the

financial health of a person or a business.

</p>

<p class="income">

Income refers to the amount of money received by an individual or business from

various sources such as employment, investments, or sales of goods and services.

Income can be earned on a regular basis, such as a salary, or irregularly, such
```

as a bonus or one-time payment.

</p>

<p class="expenses">

Expenses, on the other hand, refer to the amount of money spent by an individual or business to cover their costs of living or operating. Expenses can include fixed costs such as rent or salaries, variable costs such as the cost of materials, or discretionary costs such as entertainment or travel.

</p>

<p class="profit">

Profit is the amount of money that remains after deducting expenses from income. It represents the financial gain or loss that a person or business has generated over a given period of time. A positive profit means that the income was greater than the expenses, while a negative profit means that the expenses were greater than the income.

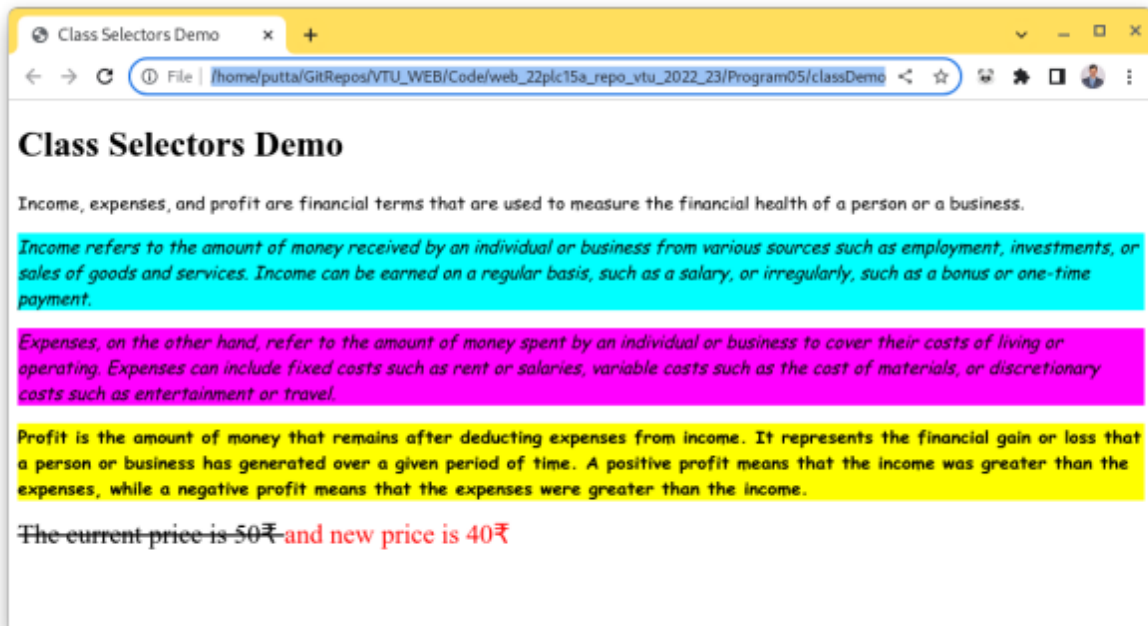
</p>

The current price is 50₹ and new price is 40₹

</body>

</html>

Output:



Termwork 6:

Change the tag `li` to have the following properties:

- ☐ A display status of inline
- ☐ A medium, double-lined, black border
- ☐ No list style type

Add the following properties to the style for `li`:

- ☐ Margin of 5px
- ☐ Padding of 10px to the top, 20px to the right, 10px to the bottom, and 20px to the left. Also demonstrate list style type with user defined image logos.

Theory :

The `<li>` HTML element is used to represent an item in a list. It must be contained in a parent element: an ordered list (`<ol>`), an unordered list (`<ul>`). Within the `<ul>` and `<menu>` elements, the list items are typically shown with bullet points.

Contrastingly, within the `<ol>` element, the list items are typically displayed with numbers or letters. The `<li>` tag requires a starting and end tag.

Syntax:

Department of computer Science & Engineering,
KLS's GIT, Belagavi

 List Items

CSS List-Style-Image Property, a handy tool that allows you to customize your list item markers with images. This property transforms your lists from ordinary to extraordinary, making your web pages more engaging and visually appealing.

Syntax:

list-style-image: none|url|initial|inherit;

Property Values:

none	This value specifies that no image is used as the marker. If this value is set, the marker defined in list-style-type is used instead. This is default value.
initial	This mode sets this property to its default value.
url	In this value the path to the image is used as a list-item marker.

HTML Code :

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>Tag Properties </title>
<style>
.custom {
display: inline;
border: 2px double black;
list-style-type: none;
margin: 5px;
padding-top: 10px;
padding-right: 20px;
padding-bottom: 10px;
padding-left: 20px;
```

```

}

.logo {
list-style-image: url('https://www.w3schools.com/cssref/sqpurple.gif');
margin: 15px;
}

</style>

</head>

<body>

<h2>li Tag Property modification Demo</h2>

<h3>Current Top FootBall Players</h3>

<ul>

<li class="custom">Lionel Messi</li>

<li class="custom">Kylian Mbappe</li>

<li class="custom">Lewandowski</li>

</ul>

<br>

<h2>list style type with user defined image logos</h2>

<h3>Current Top FootBall Goalkeepers</h3>

<ul class="logo">

<li>Emiliano Martinez</li>

<li>Thibaut Courtois</li>

<li>Yassine Bounou</li>

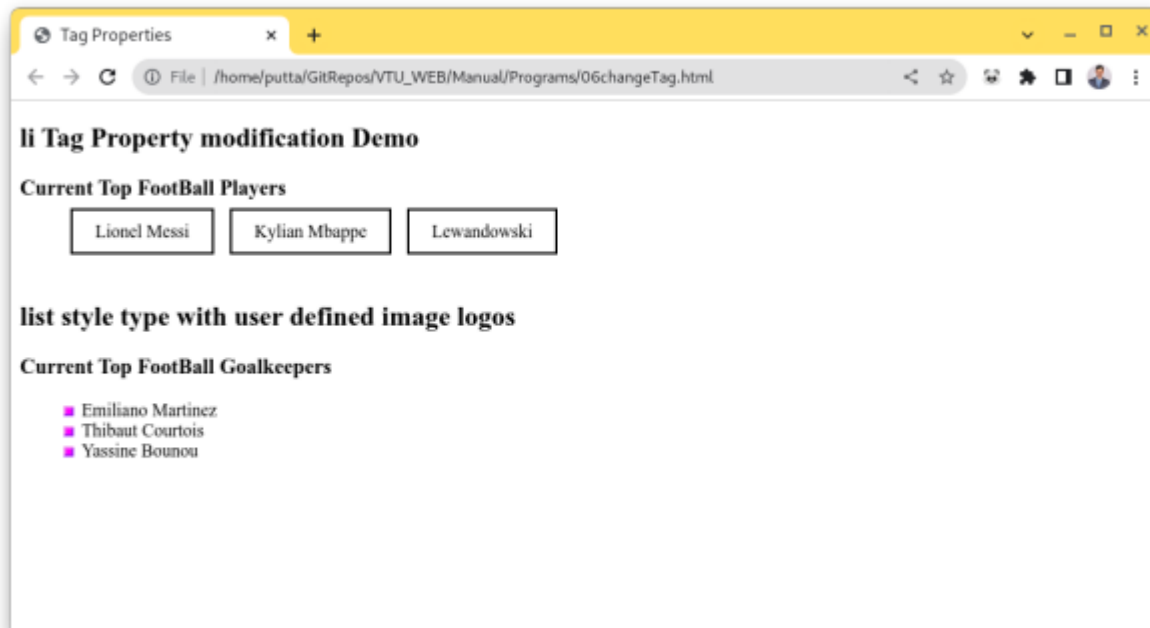
</ul>

</body>

</body>

</html>

```

OUTPUT**Termwork 7:**

Create following web page using HTML and CSS with tabular layout .

The form is titled 'Sign up Today' and is set against a light orange background. It contains the following elements:

- Name:** A text input field.
- Email:** A text input field.
- Password:** A text input field.
- Confirm password:** A text input field.
- Register:** An orange button with white text.

Theory :**HTML Forms**

An HTML form is used to collect user input. The user input is most often sent to a server for processing.

The HTML `<form>` element is used to create an HTML form for user input. The `<form>` element is a container for different types of input elements, such as: text fields, checkboxes, radio buttons, submit buttons, etc.

Tag	Description
<code><form></code>	It defines an HTML form to enter inputs by the user side.
<code><input></code>	It defines an input control.
<code><textarea></code>	It defines a multi-line input control.
<code><label></code>	It defines a label for an input element.
<code><fieldset></code>	It groups the related element in a form.
<code><legend></code>	It defines a caption for a <code><fieldset></code> element.
<code><select></code>	It defines a drop-down list.
<code><optgroup></code>	It defines a group of related options in a drop-down list.
<code><option></code>	It defines an option in a drop-down list.
<code><button></code>	It defines a clickable button.

- **Radio Buttons:** The radio button is used to select one option from multiple options. It is used for selection of gender, quiz questions etc.

If you use one name for all the radio buttons, only one radio button can be selected at a time.

Using radio buttons for multiple options, you can only choose a single option at a time.

- **Checkbox Control**

The checkbox control is used to check multiple options from given checkboxes.

- **Submit button control**

HTML **<input type="submit">** are used to add a submit button on web page. When user clicks on submit button, then form get submit to the server.

- **HTML <form> method Attribute**

The **HTML <form> method Attribute** is used to specify the HTTP method used to send data while submitting the form. There are two kinds of HTTP methods, which are **GET** and **POST**. The method attribute can be used with the **<form>** element.

Syntax:

```
<form method="get | post">
```

Attribute Values:

Attribute Values	Description
GET	Form values are visible in the address bar, limited to 3000 characters, suitable for non-sensitive data.
POST	Form values not visible in the address bar, have no size limit, are secure for sensitive data, not bookmarkable

Code :

```
<!DOCTYPE html>

<html>

<head>


<title>Sign Up</title>


<style>


body {


font-family: Arial, sans-serif;


background-color: #f2f2f2;


}


.container {


width: 500px;


margin: 0 auto;


padding: 20px;


background-color: #F7E7CE;
```

```
border-radius: 5px;
box-shadow: 0 0 10px rgba(0,0,0,0.3);
}
table {
width: 100%;
}
th, td {

padding: 10px;


}
th {
text-align: left;
background-color: #f2f2f2;
}
input[type=text], input[type=password], input[type=email] {
width: 100%;
padding: 8px;
margin: 8px 0;
border: 1px solid #ccc;
border-radius: 4px;
box-sizing: border-box;
}
button[type=submit] {
background-color: #FFA500;
```

```
color: #fff;

padding: 10px 20px;

border: none;


border-radius: 4px;

cursor: pointer;

}

</style>

</head>

<body>

<div class="container">

<h1>Sign up Today</h1>

<form>


<table>

<tr>


<td><label for="username">Name:</label> <br>

<input type="text" id="username" name="username" required></td>

</tr>

<tr>

<td><label for="email">Email:</label> <br>

<input type="email" id="email" name="email" required></td>

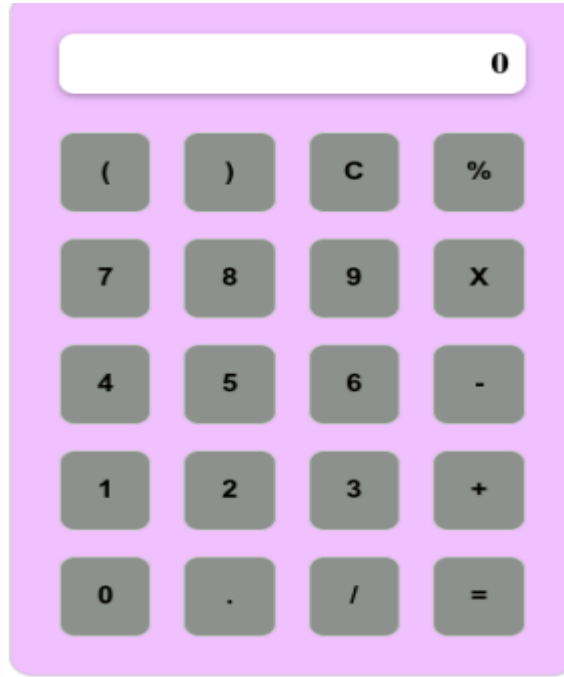
</tr>
```

```
<tr>
<td><label for="password">Password:</label> <br>
<input type="password" id="password" name="password" required></
td>
</tr>
<tr>
<td><label for="password">Confirm password:</label> <br>
<input type="password" id="password" name="password" required></
td>

</tr>
<tr>
<td colspan="2"><button type="submit">Register</button></td>
</tr>
</table>
</form>
</div>
</body>
</html>
```

Termwork 8:

Create following calculator interface with HTML and CSS.

**Theory:**

In this article, we are going to discuss how to develop a Real-Time Calculator using HTML, CSS. Usually, if we observe any real-time calculator we know that it has –

- A grid of numbers (0-9 and 00).
- Some basic arithmetic operators (+, -, /, x, %).
- And some symbols for special operations such as (clear, backspace, equal)

Use of HTML

In this program, we are using HTML to create content for the calculator UI; that means we are creating boxes, input fields, buttons, etc.

Use of CSS

We are using CSS for managing the content of HTML like the content color, width, height, font size, padding, margin, etc.

Use of JavaScript built-in methods.

In a calculator, it is established that there are different buttons and all these buttons have different functions. For example, the + button performs the addition operation, and – performs the subtraction, and it is possible to assign these operations to these buttons using JavaScript.

- The <link> tag defines the relationship between the current document and an external resource. The <link> element is an empty element, it contains attributes only.

```
<head>
  <link rel="stylesheet" href="styles.css">
</head>
```

rel attribute is required. Specifies the relationship between the current document and the linked document .

href attribute specifies the location of the linked document.

Code:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title></title>

<link rel="stylesheet" type="text/css" href="calcstyle.css">

</head>

<body>

<div class="calculator">

<div class="display">

<p id="result">0</p>

</div>
```

```
<div class="buttons">

<button onclick="appendToDisplay('(')">(</button>

<button onclick="appendToDisplay(')')">)</button>

<button onclick="clearDisplay()">C</button>

<button onclick="appendToDisplay('%')">%</button>

<button onclick="appendToDisplay('7')">7</button>
<button onclick="appendToDisplay('8')">8</button>

<button onclick="appendToDisplay('9')">9</button>

<button onclick="appendToDisplay('*')">X</button>

<button onclick="appendToDisplay('4')">4</button>

<button onclick="appendToDisplay('5')">5</button>

<button onclick="appendToDisplay('6')">6</button>

<button onclick="appendToDisplay('-')">-</button>
```

```
<button onclick="appendToDisplay('1')">1</button>
```

```
<button onclick="appendToDisplay('2')">2</button>
```

```
<button onclick="appendToDisplay('3')">3</button>
```

```
<button onclick="appendToDisplay('+')">+</button>
```

```
<button onclick="appendToDisplay('0')">0</button>
```

```
<button onclick="appendToDisplay('.')">.</button>
```

```
<button onclick="appendToDisplay('/')">/</button>
```

```
<button onclick="evaluate()">=</button>
```

```
</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

CSS Code

```
.calculator {
```

```
display: flex;
```

```
flex-direction: column;
```

```
width: 350px;
```

```
margin: 10px;
```

```
border: 1px solid #ccc;
```

```
border-radius: 15px;
```

```
background-color: #F0C0FF;
```

```
}
```

```
.display {
```

```
background-color: #fff;
```

```
border-radius: 10px;
```

```
box-shadow: 0px 2px 5px rgba(0, 0, 0, 0.3);
```

```
display: flex;
```

```
justify-content: flex-end;
```

```
align-items: center;
```

```
padding: 10px;
```

```
margin-left: 30px;
```

```
margin-right: 30px;
```

```
margin-top: 30px;
```

```
}
```

```
.buttons {
```

```
display: grid;
```

```
grid-template-columns: repeat(4, 1fr);
```

```
padding: 20px;
```

```
}
```

```
button {
```

```
padding: 20px;
```

```
background-color: #8D918D;
```

```
border: 1px solid #ccc;
```

```
border-radius: 10px;
```

```
cursor: pointer;
```

```
margin: 10px;
```

```
font-size: 18px;
```

```
font-weight: bold;
```

```
}
```

```
button:hover {
```

```
background-color: #d9d9d9;
```

```
}
```

```
button:active {
```

```
background-color: #bfbfbf;
```

```
}
```

```
#result {
```

```
margin: 0;
```

```
font-size: 24px;
```

```
font-weight: bold;
```

```
}
```

Output:

