1.Description:

This program creates a travel blog webpage that uses HTML5 semantic tags for better structure and readability. The <header> tag defines the page heading, while the <nav> tag contains the navigation menu for moving to different sections. The <main> tag holds the central content, and inside it, <article> is used to represent a self-contained story. The <section> tag groups related content inside the article, and <figure> with <figcaption> is used to insert an image along with a descriptive caption. The <aside> tag contains tips related to the main topic, and <footer> displays copyright details. The <div> tag is a generic container, while <span> is used for inline styling or grouping text inside other elements.

**Program :**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>HTML5 Semantic elements Demo</title>

<style> body {

font-family: arial, sans-serif; margin:0;

}

header,nav,footer{ background-colour:black; colour:black; padding:1em;

} nav a{ colour:black; margin:0 10px; text-decoration:none;

}

main{ display:flex; padding:20px;

} article{ flex:3; padding:20px; background-colour:black;

} aside { flex:1;

padding:20px; background-color:black;

}

figure { margin:0; text-align:center;

}

figcaption{ font-style:italic; font-size:0.9em;

}

footer{ text-align:center;

}

</style>

</head>

<body>

<head>

<h1>My blog website</h1>

</head>

<nav>

<a href="home.html">Home</a>

<a href="">Articles</a>

<a href="#">Gallery</a>

<a href="#">Contact</a>

</nav>

<main>

<article>

<h2>Understanding HTMLS semantic Tags</h2>

<p><span style="font-weight:bold;">HTML5</span>introduced semantic elements that make your code more readable and accessible.<p>

<figure>

<img src="C:\Users\garag\OneDrive\Desktop\HTML\2b\Sushma.jpg" height="100" width="100" alt="HTMLS Illustration">

<figcaption>figure:Illustration of HTMLS layout</figcaption>

</figure>

<section>

<h3>why use semantic Tags?</h3>

<p>Semantic tags help search engines and screen reader understand the the strcture of your webpage.</p>

</section>

<aside>

<h3>Related Posts</h3>

<u1>

<li><a href="#">Introduction to HTML</a></li>

<li><a href="#">CSS Basics</a></li>

<li><a href="#">JavaScript for Beginners</a></li>

</u1>

</aside>

</main>

<footer>

<p>&copy; 2025 my blog.All right reserved.</p>

</footer>

</body>

</html>

2.**DESCRIPTION:**

The purpose of this HTML program is to demonstrate how audio and video files can be embedded into a web page using the <audio> and <video> tags. In modern web development, multimedia elements such as music, sound effects, or videos play a vital role in making websites more interactive and engaging. The <audio> tag in HTML allows developers to insert audio files directly into a webpage, with attributes like controls, autoplay, and loop to give users options such as play, pause, and volume adjustment. Similarly, the <video> tag enables embedding video files and provides controls for playing, pausing, adjusting volume, and even fullscreen display. Both tags support multiple formats such as MP3, OGG, MP4, and WebM to ensure cross-browser compatibility. By writing an HTML program with these tags, we can easily integrate background music, podcasts, or instructional videos into a webpage, making it more dynamic, user-friendly, and visually appealing without requiring external plugins.

**PROGRAM:**

<html>

<body>

<h1>The audio element</h1>

<p>Click on the play button to play a sound:</p>

<audio controls>

<source src="C:\Users\garag\Downloads\I'm Giving Up - Everet Almond.mp3"type="audio/ogg"> your browser does not support the audio element

</audio>

<h2>The vedio element</h2>

<p>Click on the play button to play a vedio:</p>

<video controls>

<source src="C:\Users\garag\OneDrive\Desktop\College\Eng ppt.mp4"type="video/ogg"> your browser does not support the video element

</video>

</body>

</html>

**3.Description:**

CSS (Cascading Style Sheets) can be applied to HTML elements in three main ways: inline, internal, and external CSS. Inline CSS is applied directly within an element’s style attribute in the HTML code itself. It is useful for making quick, small changes to a single element, such as changing the color or font of a specific heading or paragraph, but it is not recommended for larger projects because it mixes content with presentation and becomes hard to maintain. Internal CSS, on the other hand, is defined within a <style> tag placed inside the <head> section of the HTML file. This method allows you to style multiple elements on the same webpage consistently without writing styles repeatedly for each element. It is generally used when styling is required for only one webpage. The most efficient and commonly used method is External CSS, where all style rules are written in a separate .css file, and this file is linked to the HTML document using the <link> tag inside the <head>. External CSS is ideal for larger websites with multiple pages because it helps maintain a consistent look across all pages and makes updates easier since changes made in the external CSS file automatically reflect in all linked pages. In summary, inline CSS is suitable for single elements, internal CSS works well for styling one entire page, and external CSS is the best practice for large projects as it separates design from content and improves maintainability.

**Program:**

**External css:**

<!Doctype>

<html>

<head>

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

</head>

<body bgcolor="Lavender">

<h1><b>This is a heading</b></h1>

<p><b>This is a paragraph</b></p>

</body>

</html>

**Internal css:**

<html>

<head> <style> body{ background-color:blue; }

h1{ background-color:pink;

}

p{

background-color:green;

}

</style>

</head>

<body>

<h1>This is a heading</h1>

<p>This is a pragraph.</p>

</body>

</html>

**Iniline css:**

<html>

<body>

<h1 style="color:skyblue;text-align:center;">This is a heading</h1>

<p style="color:red;">This is a paragraph.</p>

</body>

</html>