WEB TECHNOLOGY

MOBILE APPLICATION

ASHWADH A

AIML 'A'

LAB EXPERIMENT-1

231501022

LAB EXPERIMENT : 01

1 HTML &amp; CSS

1. Create a web page to embed a map along with hot spot, frames &amp; links.

AIM : To create a web page that embeds a map with interactive hot spots, uses frames for layout, and includes hyperlinks for navigation.

ALGORITHM :

Code :

1 Create the HTML structure:

* Create a basic HTML page with required tags (<html>, <head>,

<body>).

* Add a <frameset> to divide the page into frames.
* Include a <map> tag to create the clickable hotspots over the embedded map.
* Define the map's hotspots using <area> tags.
* Use a <link> to specify a stylesheet for styling. 2 Embed a Map:
* Use the <iframe> tag to embed a map (e.g., Google Maps).
* Link the map to specific coordinates or location. 3 Add Hotspots:
* Use the <map> and <area> tags to define areas that will act as clickable hotspots.
* Each hotspot should link to a different page or section on the site.

4 Style the Web Page:

* Apply inline, internal, and external CSS for styling the layout, hotspots, and other elements.

4 Test the Page:

* Test the page for proper functionality of the map, hotspots, and frames

Code : Html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial- scale=1.0">

<title>Map with Hotspots</title>

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

<style>

body {

font-family: Arial, sans-serif;

}

</style>

</head>

<body>

<h1>Welcome to the Map Page</h1>

<frameset rows="30%, 70%">

<!-- Top frame with map-->

<frame src="map.html" name="mapFrame">

<!-- Bottom frame with hotspots-->

<frame src="hotspots.html" name="linkFrame">

</frameset>

<p>This page contains a map and clickable hotspots.</p>

<!-- Image map for hotspots-->

<img src="map.jpg" usemap="#map" alt="Map" width="500" height="300">

<map name="map">

<area shape="rect" coords="50,50,150,150" alt="Spot 1" href="https://[www.example.com"](http://www.example.com/) target="linkFrame">

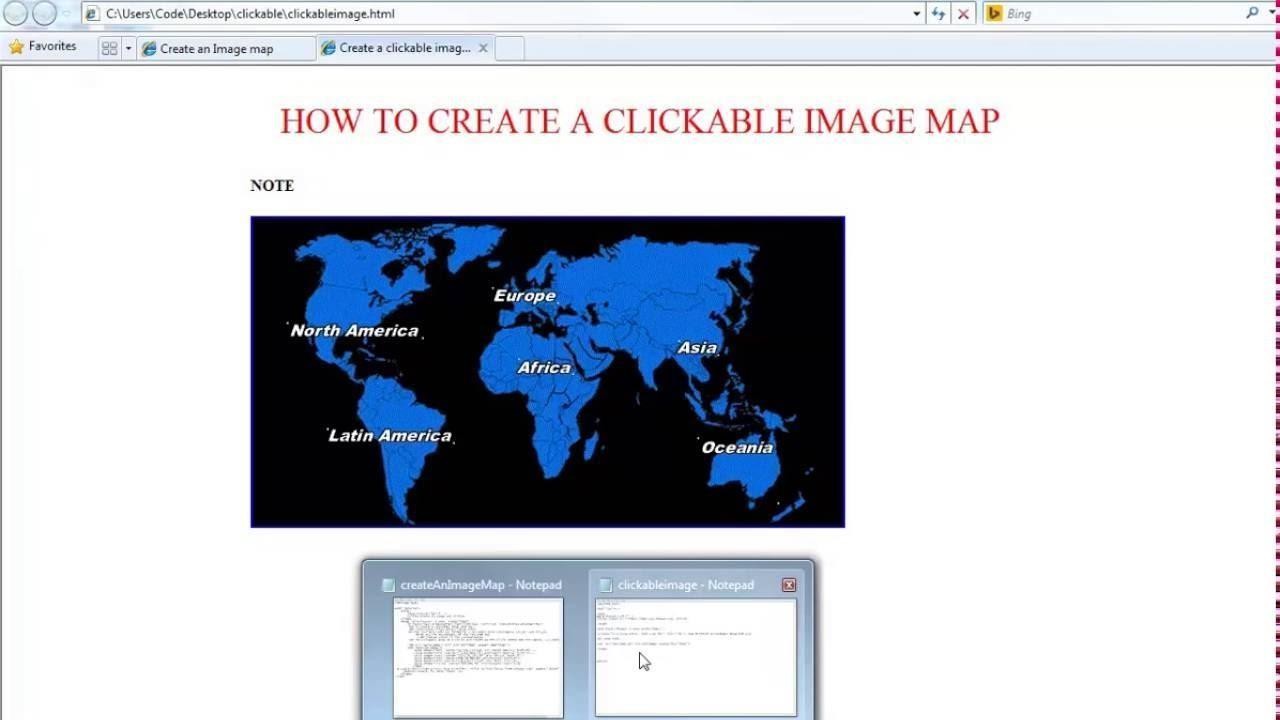
<area shape="circle" coords="200,100,50" alt="Spot 2" href="https://[www.example2.com"](http://www.example2.com/) target="linkFrame">

</map>

</body>

</html>

OUTPUT :



RESULT:

By creating a web page with embedded, external, and inline CSS, we can effectively manage the styling of different elements of the page. This demonstrates how different CSS methods can be combined for a comprehensive styling approach.

1. Create a Web Page Using Embedded, External, and Inline CSS Files

Aim:

To create a web page with a combination of embedded, external, and inline CSS to style elements such as headers, paragraphs, and links.

Algorithm:

* 1. Create the HTML structure:
     + Create a basic HTML page structure (<html>,

<head>, <body>).

* + - Add a header, paragraph, and link elements that will be styled.
  1. Add Inline CSS:
     + Apply CSS styling directly to specific HTML elements using the style attribute.
  2. Add Internal CSS:
     + Include CSS styles within the <style> tag inside the <head> section of the HTML document.
  3. Link to an External CSS File:
     + Create a separate .css file and link it to the HTML document using the <link> tag.
  4. Test the Page:
     + Test if the inline, internal, and external CSS are being applied correctly to different elements.

Code :

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device- width, initial-scale=1.0">

<title>CSS Styling Example</title>

<link rel="stylesheet" type="text/css" href="styles.css"> <!-- External CSS-->

<style>

/\* Internal CSS \*/

h1 {

color: blue;

text-align: center;

}

p {

font-size: 18px; color: darkgreen;

}

</style>

</head>

<body>

<h1>This is a Heading</h1>

<p>This is a paragraph styled using internal CSS.</p>

<a href="https[://www.example.com"](http://www.example.com/) style="color: red;">Click here to visit Example.com</a> <!-- Inline CSS

-->

</body>

</html>

styles.css (External CSS) Example:

/\* External CSS \*/ body {

font-family: Arial, sans-serif; background-color: #f0f0f0;

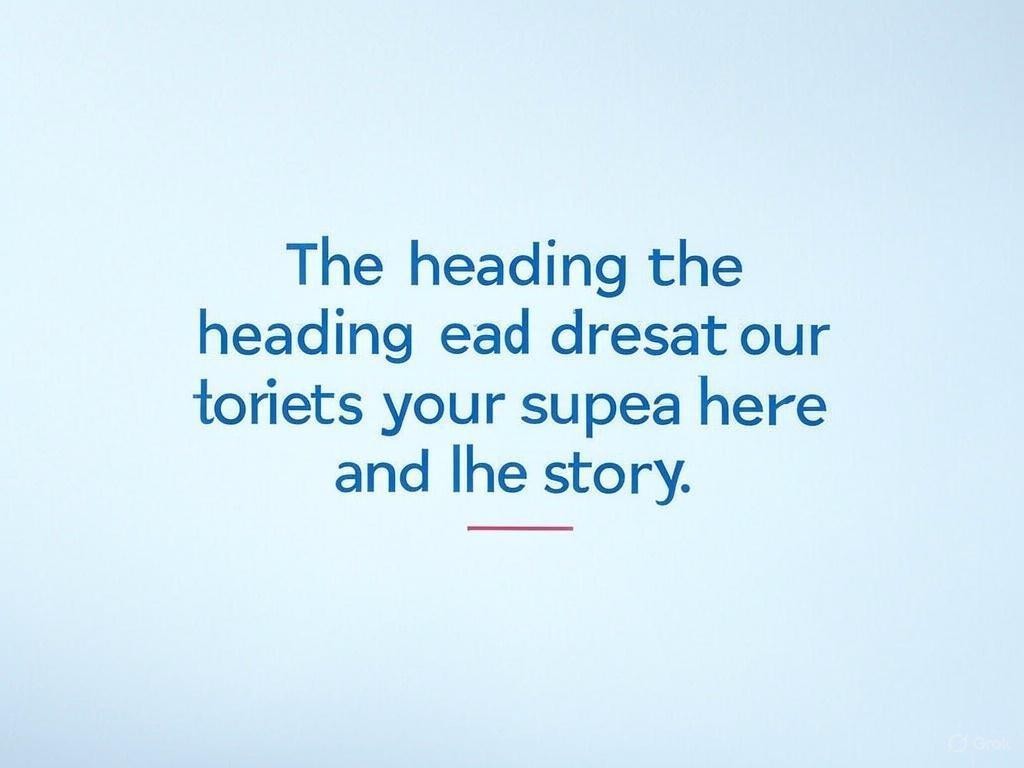
}

a {

text-decoration: none; font-weight: bold;

}

Output :



.

Result:

* By creating a web page with embedded, external, and inline CSS, we can effectively manage the styling of different elements of the page. This demonstrates how different CSS methods can be combined for a comprehensive styling approach.