

**Department of Artificial Intelligence and Machine Learning**

**QUESTION BANK FOR VI Sem (Term: March-May 2024)**

**Web Programming Laboratory (AIL67)**

**I.A. Marks: 50 Exam Hours: 03**

**Credits: 0:0:1 Exam Marks:50**

|  |  |  |
| --- | --- | --- |
| **Sl.No** | **Questions** | **CO Mapping** |
|  | 1. Design HTML page that contains the following tags <p>, <a>, <img>, <h1>…<h6>, <div>, <color selector>, <date time selector>, <month selector>, <url selector> create a suitable profile. 2. Write JavaScript to validate the following fields of the Registration page.   First Name (Name should contain alphabets and the length should not be less than 6 characters).  Password (Password should not be less than 6 characters in length).  E-mail ID (should not contain any invalid and must follow the standard pattern name@domain.com)  Mobile Number (The phone number should contain 10 numbers) | CO1 |
|  | 1. Develop and demonstrate the usage of inline, internal, and external style sheets using CSS.(the following styles should be applied 1. Color 2. Font style 3. Font size 4. Width 5. Height 6. Background color 2. Apply responsive typography classes provided by Bootstrap to ensure consistent font sizes and line heights across different devices | CO1 |
|  | 1. Develop and demonstrate JavaScript with POP-UP boxes and functions for the following problem:   Input: A number n obtained using prompt  Output: Factorial of n number using alert   1. Design a calculator using node.js with functions add, subtract and multiply. And use the Calculator module in another Node.js file. | CO2 |
|  | **Demonstrate the following HTML5 Semantic tags** – < article >, < aside >, < details >, < figcaption >, < figure >, < footer >, < header >, < main >, < mark >, < section > **for a webpage that gives information about travel experience.** | CO2 |
|  | 1. Design a carousel component into a webpage using Bootstrap to showcase a slideshow of images or content. 2. Create modal dialogs and popovers using Bootstrap to display additional content or interactive elements. | CO2 |
|  | 1. Write a program to demonstrate BOX MODEL having the following properties    The webpage should display multiple boxes, each representing a different component of the box model (content, padding, border, margin).   Each box should have distinct colors and visible borders to clearly delineate its boundaries.   The layout should demonstrate how changes in content, padding, border, and margin affect the size and spacing of the boxes   1. Write a nodejs program that creates a simple HTTP server using Node.js. When a client sends a request to this server, it should respond with "Hello World!" and then end the response. The server listens on port 8080 for incoming requests. | CO3 |
|  | Demonstrate the following using the bootstrap   * + 1. The. the table-hover class enables a hover state (grey background on mouse-over) on table rows     2. Color spinners     3. create a basic pagination     4. images(rounded,thumbnail,square) | CO3 |
|  | Design the following static web pages required for an online book store  web site.  1) HOME PAGE: The static home page must contain three frames.  2) LOGIN PAGE  3) CATOLOGUE PAGE: The catalogue page should contain the  details of all the books available in the web site in a table.  4) REGISTRATION PAGE | CO3 |
|  | Design an HTML program that Connects to MySQL Database and perform the following actions   * + 1. INSERT entries INTO Table     2. ORDER entries BY a column     3. UPDATE Table Entries     4. DELETE Table Entries | CO3 |
|  | 1. Given an array of strings, write a function to remove all duplicate strings and return a new array with unique strings only using JavaScript. 2. Design a program that demonstrates Bootstrap Alerts messages, Bootstrap Buttons, and Bootstrap cards. | CO3 |
|  | Demonstrate an HTML document that uses Bootstrap to create a responsive webpage with a simple layout. | CO3 |
|  | Create a webpage containing 3 overlapping images using HTML, CSS and JS. Further, when the mouse is over any image, it should be on the top and fully displayed. | CO3 |
|  | Create an XML document containing data about a collection of movies. Each movie entry should include attributes such as title, director, genre, release year, and rating. Using XSLT, develop a stylesheet to transform the XML data into an HTML webpage. Apply styles to the HTML elements to display the movie information in a visually appealing format. | CO3 |
|  | Develop a JavaScript program to implement a simple task manager application. The application should allow users to add, delete, and mark tasks as completed | CO3 |

**Marks Distribution:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Conduction and Result** | **Write-Up (8)** | **Execution (35)** | **Viva** | **Change of Program** | **Total** |
| **50M** | **8M** | **35M** | **7M** | **-10 Marks** | **50 Marks** |

Solutions

|  |  |
| --- | --- |
| **Sl.No** | **Questions** |
|  | 1. Design HTML page that contains the following tags <p>, <a>, <img>, <h1>…<h6>, <div>, <color selector>, <date time selector>, <month selector>, <url selector>.   <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>HTML Tags Example</title>  </head>  <body>  <h1>HTML Tags Example</h1>    <p>This is a paragraph.</p>    <a href="https://www.example.com">This is a link</a>    <img src="https://via.placeholder.com/150" alt="Placeholder Image">    <h2>This is a heading 2</h2>    <h3>This is a heading 3</h3>    <h4>This is a heading 4</h4>    <h5>This is a heading 5</h5>    <h6>This is a heading 6</h6>    <div style="background-color: lightblue; padding: 10px;">  This is a div with a light blue background color.  </div>    <label for="color">Select a color:</label>  <input type="color" id="color" name="color">    <label for="date">Select a date:</label>  <input type="date" id="date" name="date">    <label for="month">Select a month:</label>  <input type="month" id="month" name="month">    <label for="url">Enter a URL:</label>  <input type="url" id="url" name="url">  </body>  </html>  Output:  [Save file as first.html]     1. Write JavaScript to validate the following fields of the Registration page.   First Name (Name should contain alphabets and the length should not be less than 6 characters).  Password (Password should not be less than 6 characters in length).  E-mail ID (should not contain any invalid and must follow the standard pattern name@domain.com)  Mobile Number (The phone number should contain 10 numbers)  <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Registration Form</title>  </head>  <body>  <h1>Registration Form</h1>  <form id="registrationForm" onsubmit="return validateForm()">  <label for="firstName">First Name:</label>  <input type="text" id="firstName" name="firstName"><br><br>    <label for="password">Password:</label>  <input type="password" id="password" name="password"><br><br>    <label for="email">E-mail ID:</label>  <input type="email" id="email" name="email"><br><br>    <label for="mobile">Mobile Number:</label>  <input type="tel" id="mobile" name="mobile"><br><br>    <input type="submit" value="Register">  </form>  <script>  function validateForm() {  var firstName = document.getElementById("firstName").value;  var password = document.getElementById("password").value;  var email = document.getElementById("email").value;  var mobile = document.getElementById("mobile").value;    // Validate First Name  if (!/^[a-zA-Z]{6,}$/.test(firstName)) {  alert("First Name should contain alphabets and the length should not be less than 6 characters.");  return false;  }    // Validate Password  if (password.length < 6) {  alert("Password should not be less than 6 characters in length.");  return false;  }    // Validate E-mail ID  if (!/^[^\s@]+@[^\s@]+\.[^\s@]+$/.test(email)) {  alert("Invalid E-mail ID. Please enter a valid E-mail ID.");  return false;  }    // Validate Mobile Number  if (!/^\d{10}$/.test(mobile)) {  alert("Mobile Number should contain 10 numbers.");  return false;  }    // All fields are valid  return true;  }  </script>  </body>  </html> |
|  | **style.css****p.left****{****text-align:left;****color:blue;****font-family:Cambria;****font-size:large;****text-indent:20px;****}****p.center****{** **text-align:center;** **text-decoration:underline;** **text-transform:uppercase;** **letter-spacing:-3px;** **word-spacing:20px;** **font-size:larger;****}****p.right****{** **text-align:right;** **color:red;** **font-family:Tahoma;** **font-size:15pt;** **text-decoration:overline;** **font-style:italic;****}****b#headline****{** **color:orange;** **font-size:22px;** **font-family:arial;** **text-decoration:underline;****}** sample.html  <html>  <head>  <style type="text/css">  body  {  background-image:url('images/cse.png');  background-repeat:no-repeat;  background-position:center center;  background-attachment:fixed;  background-color:pink;  }  a:link { text-decoration:none;color:orange; }  a:visited { text-decoration:none;color:red; }  a:hover { text-decoration:underline;color:blue; }  a:active { text-decoration:underline;color:purple; }  h3 { color:green; }  .c1{cursor:crosshair}  .c2{cursor:pointer}  .c3{cursor:move}  .c4{cursor:text}  .c5{cursor:wait}  .c6{cursor:help}  </style>  <link rel="stylesheet" type="text/css" href="style.css">  </head>  <body bgcolor="cyan">  <h1 style="color:blue;text-align:center;"> CSS (Inline, Internal and External) </h1>  <p>This Paragraph is a Not Styled</p>  <p class="left">This Paragraph is Styled by class "Left"</p>  <p class="center">This Paragraph is Styled by class "Center"</p>  <p class="right">This Paragraph is Styled by class "Right"</p>  <b>This is normal Bold</b> <br>  <b id="headline">This Bold Text is Styled </b>  <h2><b><a href=" ">This is a link</a></b></h2>  <h3 class="c1">The cursor over this element is plus sign</h3>  <h3 class="c2">The cursor over this element is a pointing hand</h3>  <h3 class="c3">The cursor over this element is a grasping hand</h3>  <h3 class="c4">The cursor over this element is a I bar</h3>  <h3 class="c5">The cursor over this element is a wait</h3>  <h3 class="c6">The cursor over this element is a question mark</h3>  </html> **Displaying XML with XSLT** XSLT (eXtensible Stylesheet Language Transformations) is the recommended style sheet language for XML.  XSLT is far more sophisticated than CSS. With XSLT you can add/remove elements and attributes to or from the output file. You can also rearrange and sort elements, perform tests and make decisions about which elements to hide and display, and a lot more.  XSLT uses XPath to find information in an XML document. **XSLT Example** We will use the following XML document:  <?xml version="1.0" encoding="UTF-8"?> <breakfast\_menu>  <food> <name>Belgian Waffles</name> <price>$5.95</price> <description>Two of our famous Belgian Waffles with plenty of real maple syrup</description> <calories>650</calories> </food>  <food> <name>Strawberry Belgian Waffles</name> <price>$7.95</price> <description>Light Belgian waffles covered with strawberries and whipped cream</description> <calories>900</calories> </food>  <food> <name>Berry-Berry Belgian Waffles</name> <price>$8.95</price> <description>Light Belgian waffles covered with an assortment of fresh berries and whipped cream</description> <calories>900</calories> </food>  <food> <name>French Toast</name> <price>$4.50</price> <description>Thick slices made from our homemade sourdough bread</description> <calories>600</calories> </food>  <food> <name>Homestyle Breakfast</name> <price>$6.95</price> <description>Two eggs, bacon or sausage, toast, and our ever-popular hash browns</description> <calories>950</calories> </food>  </breakfast\_menu>  Xslt stylesheet  <?xml version="1.0" encoding="UTF-8"?> <html xsl:version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"> <body style="font-family:Arial;font-size:12pt;background-color:#EEEEEE"> <xsl:for-each select="breakfast\_menu/food">   <div style="background-color:teal;color:white;padding:4px">     <span style="font-weight:bold"><xsl:value-of select="name"/> - </span>     <xsl:value-of select="price"/>     </div>   <div style="margin-left:20px;margin-bottom:1em;font-size:10pt">     <p>     <xsl:value-of select="description"/>     <span style="font-style:italic"> (<xsl:value-of select="calories"/> calories per serving)</span>     </p>   </div> </xsl:for-each> </body> </html> |
|  | 1. Develop and demonstrate JavaScript with POP-UP boxes and functions for the following problem:   Input: A number n obtained using prompt  Output: Factorial of n number using alert  // Get the number n from the user  var n = prompt("Enter a number:");  // Validate if n is a valid number  if (!isNaN(n) && n > 0) {  // Calculate the factorial of n  var factorial = 1;  for (var i = 1; i <= n; i++) {  factorial \*= i;  }    // Display the factorial using alert  alert("The factorial of " + n + " is " + factorial);  } else {  // Display an error message if n is not a valid number  alert("Please enter a valid positive number.");  }     1. Design a calculator using node.js with functions add, subtract and multiply. And use the Calculator module in another Node.js file.   // Returns addition of two numbers  exports.add = function (a, b) {  return a+b;  };    // Returns difference of two numbers  exports.subtract = function (a, b) {  return a-b;  };    // Returns product of two numbers  exports.multiply = function (a, b) {  return a\*b;  };  var calculator = require('./calculator');    var a=10, b=5;    console.log("Addition : "+calculator.add(a,b));  console.log("Subtraction : "+calculator.subtract(a,b));  console.log("Multiplication : "+calculator.multiply(a,b)); |
|  | <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8" />  <title>HTML5 Semantic Tags Demo</title>  <style>  body{  background-color: #FFFDD0;  }  aside {  float: right;  width: 25%;  background-color: cyan;  font-style: italic;  padding: 15px;  }  main {  float: left;  width: 70%;  }  footer {  position: fixed;  left: 0;  bottom: 0;  width: 100%;  text-align: center;  }  mark {  background-color: yellow;  color: black;  }  figure {  display: inline-block;  margin: auto;  }  figcaption {  font-style: italic;  }  </style>  </head>  <body>  <article>  <header>  <h1>My Travelogue</h1>  <p>Random Escapades</p>  </header>  <main>  <figure>  <img src="journey.jpeg" alt="Example Image" width="350" height="235">  <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>  <section>  <h2>Mysore</h2>  <p> The city is also known as the City of Palaces, Mysuru has always enchanted its visitors with its quaint charm.</p>  </section>  </main>  <aside>  <section>  <p>Upcoming Trek planned to <mark>Kumara Parvata</mark> will be sharing detils soon</p>  <details>  <summary>Tentative Dates</summary>  <p>24th January 2023</p>  </details>  </section>  </aside>  <footer>  <p>© 2024 megha</p>  </footer>  </article>  </body>  </html> |
|  | Design an XML file that represents a simple bookstore inventory with information about different books. Each <book> element has attributes such as category and cover and contains elements like <title>, <author>, <year>, and <price> providing specific details about each book. (MINIMUM 4 ENTRIES)  <?xml version="1.0" encoding="UTF-8"?>  <bookstore>  <book category="cooking">  <title lang="en">Everyday Italian</title>  <author>Giada De Laurentiis</author>  <year>2005</year>  <price>30.00</price>  </book>  <book category="children">  <title lang="en">Harry Potter</title>  <author>J K. Rowling</author>  <year>2005</year>  <price>29.99</price>  </book>  <book category="web">  <title lang="en">XQuery Kick Start</title>  <author>James McGovern</author>  <author>Per Bothner</author>  <author>Kurt Cagle</author>  <author>James Linn</author>  <author>Vaidyanathan Nagarajan</author>  <year>2003</year>  <price>49.99</price>  </book>  <book category="web" cover="paperback">  <title lang="en">Learning XML</title>  <author>Erik T. Ray</author>  <year>2003</year>  <price>39.95</price>  </book>  </bookstore> |
|  | 1. Write a program to demonstrate BOX MODEL having the following properties  * **Content** - The content of the box, where text and images appear * **Padding** - Clears an area around the content. The padding is transparent * **Border** - A border that goes around the padding and content * **Margin** - Clears an area outside the border. The margin is transparent   <!DOCTYPE html>  <html>  <head>  <style>  div {  background-color: lightgrey;  width: 300px;  border: 15px solid green;  padding: 50px;  margin: 20px;  }  </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 content of the box. We have added a 50px padding, 20px margin and a 15px green border. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</div>  </body>  </html>   1. Write a nodejs program that creates a simple HTTP server using Node.js. When a client sends a request to this server, it should respond with "Hello World!" and then end the response. The server listens on port 8080 for incoming requests.   var http = require('http');  http.createServer(function (req, res) {  res.whitehead(200, {'Content-Type': 'text/html'});  res.end('Hello World!');  }).listen(8080);  Note:  Save the file on your computer: C:\Users\Your Name\myfirst.js  The code tells the computer to write "Hello World!" if anyone (e.g. a web browser) tries to access your computer on port 8080. |
|  | Demonstrate the following using the bootstrap   * + 1. The. the table-hover class enables a hover state (grey background on mouse-over) on table rows     2. Colour spinners     3. create a basic pagination     4. images(rounded,thumbnail,square)   <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Bootstrap Features Demo</title>  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/css/bootstrap.min.css" rel="stylesheet">  </head>  <body>  <!-- 1. Table with table-hover class -->  <div class="container mt-5">  <h2>Table with Hover Effect</h2>  <table class="table table-hover">  <thead>  <tr>  <th>#</th>  <th>First Name</th>  <th>Last Name</th>  </tr>  </thead>  <tbody>  <tr>  <td>1</td>  <td>John</td>  <td>Doe</td>  </tr>  <tr>  <td>2</td>  <td>Jane</td>  <td>Smith</td>  </tr>  </tbody>  </table>  </div>  <!-- 2. Color Spinners -->  <div class="container mt-5">  <h2>Color Spinners</h2>  <div class="spinner-border text-primary" role="status">  <span class="visually-hidden">Loading...</span>  </div>  <div class="spinner-border text-secondary" role="status">  <span class="visually-hidden">Loading...</span>  </div>  <div class="spinner-border text-success" role="status">  <span class="visually-hidden">Loading...</span>  </div>  </div>  <!-- 3. Basic Pagination -->  <div class="container mt-5">  <h2>Basic Pagination</h2>  <nav aria-label="Page navigation example">  <ul class="pagination">  <li class="page-item"><a class="page-link" href="#">Previous</a></li>  <li class="page-item"><a class="page-link" href="#">1</a></li>  <li class="page-item"><a class="page-link" href="#">2</a></li>  <li class="page-item"><a class="page-link" href="#">3</a></li>  <li class="page-item"><a class="page-link" href="#">Next</a></li>  </ul>  </nav>  </div>  <!-- 4. Images (rounded, thumbnail, square) -->  <div class="container mt-5">  <h2>Images</h2>  <img src="https://via.placeholder.com/150" class="img-fluid rounded mt-2" alt="Rounded Image">  <img src="https://via.placeholder.com/150" class="img-fluid thumbnail mt-2" alt="Thumbnail Image">  <img src="https://via.placeholder.com/150" class="img-fluid square mt-2" alt="Square Image">  </div>  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/js/bootstrap.bundle.min.js"></script>  </body>  </html> |
|  | Design A web application that lists all cookies stored in the browser by clicking the “List Cookies” button. Add cookies if necessary.  Demonstrate to add Google Maps to your web page.  <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Google Maps Example</title>  <style>  /\* Set the size of the map container \*/  #map {  height: 400px;  width: 100%;  }  </style>  </head>  <body>  <h1>My Google Map</h1>  <!-- Add a div element to contain the map -->  <div id="map"></div>  <!-- Include the Google Maps JavaScript API -->  <script src="https://maps.googleapis.com/maps/api/js?key=YOUR\_API\_KEY"></script>  <script>  // Initialize and display the map  function initMap() {  // Specify the coordinates of the center of the map  var center = { lat: 40.7128, lng: -74.006 };  // Create a new map object  var map = new google.maps.Map(document.getElementById('map'), {  center: center,  zoom: 12 // Specify the initial zoom level  });  // Add a marker at the center of the map  var marker = new google.maps.Marker({  position: center,  map: map,  title: 'New York City'  });  }  </script>  <!-- Call the initMap function to initialize the map -->  <script>  initMap();  </script>  </body>  </html> |
|  | Design an HTML program that Connects to MySQL Database and perform the following actions using node.js   * + 1. INSERT entries INTO Table     2. ORDER entries BY a column     3. UPDATE Table Entries     4. DELETE Table Entries   / include node fs module  var fs = require('fs');  var data ='Learn Node FS module';    // write File function with filename, content and call back function  fs.writeFile('newfile.txt', data, function (err) {  if (err) throw err;  console.log('File is created successfully.');  });  $ node createFileExample.js  File is created successfully.  // include file system module  var fs = require('fs');    // read file sample.html  fs.readFile('sample.html',  // callback function that is called when reading file is done  function(err, data) {  if (err) throw err;  // data is a buffer containing file content  console.log(data.toString('utf8'))  });  $ node readFileExample.js  <html>  <body>  <h1>Header</h1>  <p>I have learned to read a file in Node.js.</p>  </body>  </html> |
|  | Design a program that demonstrates Bootstrap Alerts messages, Bootstrap Buttons, and Bootstrap cards.  <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Bootstrap Features Demo</title>  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/css/bootstrap.min.css" rel="stylesheet">  </head>  <body>  <!-- Bootstrap Alert Messages -->  <div class="container mt-5">  <div class="alert alert-primary" role="alert">  This is a primary alert—check it out!  </div>  <div class="alert alert-secondary" role="alert">  This is a secondary alert—check it out!  </div>  <div class="alert alert-success" role="alert">  This is a success alert—check it out!  </div>  <div class="alert alert-danger" role="alert">  This is a danger alert—check it out!  </div>  <div class="alert alert-warning" role="alert">  This is a warning alert—check it out!  </div>  <div class="alert alert-info" role="alert">  This is a info alert—check it out!  </div>  <div class="alert alert-light" role="alert">  This is a light alert—check it out!  </div>  <div class="alert alert-dark" role="alert">  This is a dark alert—check it out!  </div>  </div>  <!-- Bootstrap Buttons -->  <div class="container mt-5">  <button type="button" class="btn btn-primary">Primary Button</button>  <button type="button" class="btn btn-secondary">Secondary Button</button>  <button type="button" class="btn btn-success">Success Button</button>  <button type="button" class="btn btn-danger">Danger Button</button>  <button type="button" class="btn btn-warning">Warning Button</button>  <button type="button" class="btn btn-info">Info Button</button>  <button type="button" class="btn btn-light">Light Button</button>  <button type="button" class="btn btn-dark">Dark Button</button>  </div>  <!-- Bootstrap Cards -->  <div class="container mt-5">  <div class="card" style="width: 18rem;">  <img src="https://via.placeholder.com/150" class="card-img-top" alt="...">  <div class="card-body">  <h5 class="card-title">Card Title</h5>  <p class="card-text">Some quick example text to build on the card title and make up the bulk of the card's content.</p>  <a href="#" class="btn btn-primary">Go somewhere</a>  </div>  </div>  </div>  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/js/bootstrap.bundle.min.js"></script>  </body>  </html> |
|  | Demonstrate an HTML document that uses Bootstrap to create a responsive webpage with a simple layout.  <!DOCTYPE html>  <html lang="en">  <head>  <title>Bootstrap Example</title>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/css/bootstrap.min.css" rel="stylesheet">  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/js/bootstrap.bundle.min.js"></script>  </head>  <body>  <div class="bg-primary text-white p-5 text-center">  <h1>My First Bootstrap Page</h1>  <p>Resize this page to see the responsive effect!</p>  </div>    <div class="container">  <div class="row">  <div class="col-sm-4">      <h2>London</h2>      <p>London is the most populous city in the United Kingdom,  with a metropolitan area of over 13 million inhabitants.</p>  </div>  <div class="col-sm-4">      <h2>Paris</h2>      <p>The Paris area is one of the largest population centers in Europe,  with more than 12 million inhabitants.</p>  </div>  <div class="col-sm-4">      <h2>Tokyo</h2>      <p>Tokyo is the center of the Greater Tokyo Area,  and the most populous metropolitan area in the world.</p>  </div>  </div>  </div>  </body>  </html> |
|  | Create a webpage containing 3 overlapping images using HTML, CSS and JS. Further when the mouse is over any image, it should be on the top and fully displayed.  <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="utf-8">  <meta name="author" content="Putta" >  <title>Animal Stacking</title>  <style>  h1 {text-align: center;}  .dog {  position: absolute;  left: 10%; top: 10%;  z-index: 0;  }  .cat {  position: absolute;  left: 30%; top: 30%;  z-index: 1;  }  .horse {  position: absolute;  left: 50%; top: 50%;  z-index: 2;  }  </style>  <script>  var topIndex = 2;  function moveToTop(picture) {  picture.style.zIndex = ++topIndex;  }  </script>  </head>  <body>  <h1>Image Overlap Demo</h1>  <div id="image-container">  <img id="dog" class="dog" src="dog.jpg" onmouseover="moveToTop(this)" width="400" height="300">  <img id="cat" class="cat" src="cat.jpg" onmouseover="moveToTop(this)" width="400" height="300">  <img id="horse" class="horse" src="horse.jpg" onmouseover="moveToTop(this)" width="400" height="300">  </div>  </body>  </html> |