KTU NOTES

CST 463: Web Programming

Course Outcome 1 (CO1):

1. Construct a valid HTML document for your personal Profile registration page for a Job Site www.123Jobs.com. Add relevant HTML elements in a table, to accept a minimum of 10 different fields which includes your name, address, phone, email address, your picture, your college; your branch, fields for your personal history (Minimum 3 fields), favourite theory and practical subjects (Checkbox), Username, Password(password).

Ans.

```
<!DOCTYPE html>
<html>
 <head>
  <title>Personal Profile Registration - 123Jobs.com</title>
 </head>
 <body>
  <h1>Personal Profile Registration</h1>
  <form action="submit-profile.php" method="post">
   <table>
    <tr>
     <td>Name:</td>
     <td><input type="text" name="name"></td>
    </tr>
    <tr>
     <td>Address:</td>
     <td><input type="text" name="address"></td>
    </tr>
    <tr>
     <td>Phone:</td>
     <td><input type="tel" name="phone"></td>
    </tr>
    <tr>
     <td>Email:</td>
     <td><input type="email" name="email"></td>
    </tr>
    <tr>
     <td>Picture:</td>
     <td><input type="file" name="picture"></td>
    </tr>
    <tr>
```

```
          <td>College:</td>
          <td><input type="text" name="college"></td>
        </tr>
        <tr>
          <td>Branch:</td>
          <td><input type="text" name="branch"></td>
        </tr>
        <tr>
          <td>Personal History:</td>
          <td><textarea name="personal_history"></textarea></td>
        </tr>
        <tr>
          <td>Favourite theory subjects:</td>
          <td>
            <input type="checkbox" name="fav_theory_subjects" value="maths">Maths
            <input type="checkbox" name="fav_theory_subjects" value="physics">Physics
            <input type="checkbox" name="fav_theory_subjects" value="chemistry">Chemistry
          </td>
        </tr>
        <tr>
          <td>Favourite practical subjects:</td>
          <td>
            <input type="checkbox" name="fav_practical_subjects" value="cse">CSE
            <input type="checkbox" name="fav_practical_subjects" value="ece">ECE
            <input type="checkbox" name="fav_practical_subjects" value="mech">MECH
          </td>
        </tr>
        <tr>
          <td>Username:</td>
          <td><input type="text" name="username"></td>
        </tr>
        <tr>
          <td>Password:</td>
          <td><input type="password" name="password"></td>
        </tr>
        <tr>
          <td></td>
          <td><input type="submit" value="Submit"></td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

2. What is MIME? Give the expansion of MIME. List four examples for MIME types. State the reason why MIME type specification is necessary in a request-response transaction between a browser and server.

Ans. MIME (Multipurpose Internet Mail Extensions) is a standard that extends the format of email messages to support text in character sets other than ASCII, as well as attachments of audio, video, images, and application programs. MIME types are used to specify the format of a file in the HTTP protocol used by the World Wide Web. This is done via the "Content-Type" header in the HTTP request and response.

Examples of MIME types include:

"text/html" for HTML files
"image/jpeg" for JPEG images
"application/pdf" for PDF files
"audio/mp3" for MP3 audio files

MIME type specification is necessary in a request-response transaction between a browser and server because it allows the browser to properly interpret and display the content that the server sends in response to the browser's request. Without this specification, the browser would not know how to handle the content and might not be able to display it correctly. MIME types are also used by servers to determine how to handle a file when it is requested, such as to determine whether to send the file as an attachment or to display it within the browser. In short, MIME types are used to identify files on the Internet according to their nature and format. They are a key part of the HTTP transaction, helping to ensure that the right files are sent to the right client applications, and that those files are processed correctly.

3. What is codec? Recognize the role of controls attribute in <video> & <audio> tag in HTML. Use the COVID vaccination promotional video 'MySafety.mp4' in a web page with suitable HTML code, 'auto play' option enabled and displayed in a standard dimension 750 X500.

Ans. A codec is a device or software that compresses and decompresses digital media, such as audio and video files. Codecs are used to reduce the size of large media files, making them more manageable for storage and transmission over networks.

The controls attribute in the <video> and <audio> tags in HTML is used to enable the default set of controls for the media player, such as play/pause, volume, and seek controls.

```
<!DOCTYPE html>
<html>
  <head>
    <title>COVID Vaccination Promotional Video</title>
  </head>
  <body>
    <h1>COVID Vaccination Promotional Video</h1>
    <video width="750" height="500" autoplay controls>
      <source src="MySafety.mp4" type="video/mp4">
      Your browser does not support the video tag.
    </video>
  </body>
</html>
```

This code will embed the video "MySafety.mp4" on the webpage. The auto play attribute is set on the <video> tag which will automatically start playing the video when the page loads. The width and height attributes set the dimension of the video to 750x500 pixels. The controls attribute is set on the <video> tag which will enable the default set of controls for the media player. It is worth noting that the auto play feature can be blocked by some browsers and devices and also, it's not recommended to use it as it could cause an annoying experience for the user, and also it could use a lot of data and battery. It's always best to give the user the option to start the video.

Course Outcome 2 (CO2):

1. Organize a sample web page for the event 'Raagam2021' at your campus and use embedded Style sheets to apply a minimum 5 styles. State the Style Specification format of embedded style sheets.

Ans. <!DOCTYPE html>

```
<html>
  <head>
    <title>Raagam2021 - Campus Event</title>
    <style type="text/css">
      /* Style 1: Change background color of the body */
      body {
        background-color: #F5F5DC;
      }
      /* Style 2: Add padding and margin to the header */
      h1 {
        padding: 20px;
        margin: 0;
      }
      /* Style 3: Change font color of the event details */
      .event-details {
```

```
      color: #00008B;
    }
    /* Style 4: Add border to the event schedule table */
    .schedule-table {
      border: 1px solid #00008B;
    }
    /* Style 5: Change cursor to pointer for the registration button */
    .registration-button {
      cursor: pointer;
    }
  </style>
</head>
<body>
  <h1>Raagam2021</h1>
  <div class="event-details">
    <p>Date: 15th January, 2021</p>
    <p>Time: 9am to 5pm</p>
    <p>Location: Main Auditorium, XYZ Campus</p>
  </div>
  <h2>Event Schedule</h2>
  <table class="schedule-table">
    <tr>
      <th
```

The style specification format of embedded style sheets in HTML is CSS (Cascading Style Sheets). CSS is a stylesheet language used to describe the presentation of a document written in a markup language. It is most commonly used to style HTML and XHTML documents, but can also be applied to other types of documents, such as XML. In an embedded style sheet, CSS styles are added directly to the HTML document using the <style> tag, which should be placed in the <head> of the HTML document. The styles are then applied to the elements in the HTML document according to the selectors, properties, and values specified in the CSS styles.

The format of a CSS rule is as follows:

```
selector {
  property: value;
}
```

For example:

```
body {
  background-color: #F5F5DC;
}
```

This rule will change the background colour of the entire page to beige. The body is the selector, background-colour is the property, and #F5F5DC is the value.

2. Write CSS style rules to implement the following in a web page:

- **to display the content of hyperlinks with yellow background colour and in italics**
- **to display the contents of unordered lists in bold and in Arial font**
- **to display a background image titled “birds.jpg” with no tiling.**

Ans. Here are the CSS style rules to implement the following in a web page:

a. To display the content of hyperlinks with yellow background colour and in italics:

```
a {
   background-color: yellow;
   font-style: italic;
}
```

b. To display the contents of unordered lists in bold and in Arial font:

```
ul {
   font-weight: bold;
   font-family: Arial;
}
```

c. To display a background image titled "birds.jpg" with no tiling:

```
body {
   background-image: url("birds.jpg");
   background-repeat: no-repeat;
}
```

3. Write the code for an HTML document with embedded JavaScript scripts, which initially displays a paragraph with text "Welcome" and a button titled "Click". When the button is clicked, the message "Hello from JavaScript" in bold should replace the paragraph text.

Ans.

```
<!DOCTYPE html>
<html>
<head>
  <title>JavaScript Example</title>
</head>
<body>
  <p id="welcome">Welcome</p>
  <button onclick="sayHello()">Click</button>
  <script>
    function sayHello() {
      document.getElementById("welcome").innerHTML = "<b>Hello from JavaScript</b>";
    }
  </script>
</body>
</html>
```

Course Outcome 3 (CO3):

1. Write a PHP program to store the name and roll no of 10 students in an Associative Array and Use foreach loop to process the array and Perform asort, rsort and ksort in the array. Illustrate with suitable output data

Ans.

```
<?php
  // Creating an associative array to store name and roll no of 10 students
  $students = array(
    array("name" => "John", "roll" => 1),
    array("name" => "Mike", "roll" => 2),
    array("name" => "Sara", "roll" => 3),
    array("name" => "Mark", "roll" => 4),
    array("name" => "Julia", "roll" => 5),
    array("name" => "Laura", "roll" => 6),
    array("name" => "David", "roll" => 7),
    array("name" => "Emily", "roll" => 8),
    array("name" => "Adam", "roll" => 9),
    array("name" => "Amy", "roll" => 10)
  );

  // Using asort() to sort the array by values
  asort($students);
  echo "Array sorted by values (asort) : <br>";
  foreach ($students as $student) {
    echo $student["name"] . " - " . $student["roll"] . "<br>";
  }

  // Using rsort() to sort the array in reverse order
  rsort($students);
  echo "<br>Array sorted in reverse order (rsort) : <br>";
  foreach ($students as $student) {
    echo $student["name"] . " - " . $student["roll"] . "<br>";
  }

  // Using ksort() to sort the array by keys
  ksort($students);
  echo "<br>Array sorted by keys (ksort) : <br>";
  foreach ($students as $student) {
    echo $student["name"] . " - " . $student["roll"] . "<br>";
  }
?>
```

The output of the above PHP program would produce:

```
Array sorted by values (asort) :
Adam - 9
Amy - 10
David - 7
Emily - 8
Julia - 5
John - 1
Laura - 6
Mark - 4
Mike - 2
Sara - 3

Array sorted in reverse order (rsort) :
Sara - 3
Mike - 2
Mark - 4
Laura - 6
John - 1
Julia - 5
Emily - 8
David - 7
Amy - 10
```

```
Adam - 9
```

Array sorted by keys (ksort) :

```
Adam - 9
Amy - 10
David - 7
Emily - 8
Julia - 5
John - 1
Laura - 6
Mark - 4
Mike - 2
Sara – 3
```

2. Design an HTML page which enters a given number, write a PHP program to display a message indicating, whether the number is odd or even, when clicking on the submit button.

Ans. HTML page which enters a given number:

```
<!DOCTYPE html>
<html>
<head>
  <title>Odd or Even</title>
</head>
<body>
  <form action="submit.php" method="post">
    Enter a number: <input type="text" name="num"><br>
    <input type="submit" value="Submit">
  </form>
</body>
</html>
```

PHP program to display a message indicating, whether the number is odd or even:

```
<?php
  if(isset($_POST['num'])) {
    $num = $_POST['num'];
    if($num % 2 == 0) {
      echo "The number is even";
    } else {
      echo "The number is odd";
    }
  }
?>
```

In this HTML page, there is a form with a single text input field for a number and a submit button. When the user enters a number and clicks the submit button, the form data is sent to the PHP script "submit.php" which receives the entered number and checks if it is odd or even using the modulus operator and prints the message accordingly.

3. Write a PHP program to compute the sum of the positive integers up to 100 using do While.

Ans.
```
<?php
  $sum = 0;
  $i = 1;
  do {
    $sum += $i;
    $i++;
  } while ($i <= 100);
  echo "The sum of positive integers up to 100 is: $sum";
?>
```

In this case, the output will be the sum of positive integers up to 100 is: 5050

In this PHP program, a variable $sum is initialized with 0, and a variable $i is initialized with 1. The do-while loop runs until the condition $i <= 100 is true. In each iteration of the loop, the value of $i is added to the $sum variable and $i is incremented by 1. After the loop has finished executing, the final value of $sum variable will be the sum of all positive integers up to 100.

Course Outcome 4 (CO4):

1. Write a PHP form handling program to verify the user authentication credentials of a web page using MySQL connection and store the user id value as a Session variable if the user id is valid.

Ans.
```
<?php
  session_start(); // Starting a session
  // Connect to MySQL
  $con = mysqli_connect("hostname", "username", "password", "database_name");
  if(!$con) {
    die("Error connecting to database: " . mysqli_connect_error());
  }
  if(isset($_POST['submit'])) {
    $userid = $_POST['userid'];
    $password = $_POST['password'];
    // Retrieve user data from the database
    $query = "SELECT * FROM users WHERE userid='$userid' AND password='$password'";
    $result = mysqli_query($con, $query);
    if(mysqli_num_rows($result) > 0) {
      // Store userid in a session variable
      $_SESSION['userid'] = $userid;
      // Redirect the user to the protected page
      header("Location: protected_page.php");
    } else {
      echo "Invalid userid or password";
    }
  }
?>

<form method="post">
  User ID: <input type="text" name="userid"><br>
  Password: <input type="password" name="password"><br>
  <input type="submit" name="submit" value="Submit">
</form>
```

This PHP program uses the session_start() function to start a session, then connects to a MySQL database using the mysqli_connect() function. The program checks if the form has been submitted using the isset($_POST['submit']) condition. If the form has been submitted, the program retrieves the userid and password values from the $_POST array, and uses them to query the users table in the MySQL database. If the query returns any rows, it means the userid and password are valid, so the userid is stored in a session variable using the $_SESSION array and the user is redirected to the protected_page.php. If the query returns no rows, it means the userid and password are invalid, so an error message is displayed.

2. Create a valid HTML document for yourself, including your name, address, and email address. Also add your college; your major and the course. Perform form handling in PHP and process the output using POST method.

Ans. <!DOCTYPE html>

```
<html>
<head>
  <title>Contact Form</title>
</head>
```

```
<body>
  <form action="submit.php" method="post">
    Name: <input type="text" name="name"><br>
    Address: <input type="text" name="address"><br>
    Email: <input type="email" name="email"><br>
    College: <input type="text" name="college"><br>
    Major: <input type="text" name="major"><br>
    Course: <input type="text" name="course"><br>
    <input type="submit" value="Submit">
  </form>
</body>
</html>
```

```
<?php
  if (isset($_POST['name'])) {
    $name = $_POST['name'];
    $address = $_POST['address'];
    $email = $_POST['email'];
    $college = $_POST['college'];
    $major = $_POST['major'];
    $course = $_POST['course'];
    echo "Name: " . $name . "<br>";
    echo "Address: " . $address . "<br>";
    echo "Email: " . $email . "<br>";
    echo "College: " . $college . "<br>";
    echo "Major: " . $major . "<br>";
    echo "Course: " . $course . "<br>";
  }
?>
```

This is a basic example of a contact form, where the user can enter their name, address, email, college, major and course. Once the user submits the form, the form data is sent to the PHP script "submit.php" using the POST method. The PHP script receives the form data using the $_POST array and processes it by printing the values of the form fields on the screen.

3. Write an embedded PHP script which displays the factorial of all numbers from 1 to 10 in a table in the web page. The factorial should be calculated and returned from a function. The table headings should be "Number" and "Factorial".

Ans.

```
<!DOCTYPE html>
<html>
<head>
  <title>Factorials</title>
</head>
<body>
  <table>
    <tr>
      <th>Number</th>
      <th>Factorial</th>
    </tr>
    <?php
      function factorial($n) {
        if ($n == 0) {
          return 1;
        }
        return $n * factorial($n – 1);
      }
      for ($i = 1; $i <= 10; $i++) {
        echo “<tr>”;
        echo “<td>” . $i . “</td>”;
        echo “<td>” . factorial($i) . “</td>”;
        echo “</tr>”;
      }
    ?>
  </table>
</body>
</html>
```

This is an example of a simple PHP script that displays the factorial of all numbers from 1 to 10 in an HTML table. It uses a function called factorial() to calculate the factorial of a given number. The function uses a recursive approach to calculate the factorial by multiplying the number by the factorial of the number – 1 until the number reaches 0. Then it returns 1. This function is called for all numbers from 1 to 10 inside a for loop. The for loop is used to create a new row in the table for each number and the corresponding factorial. The output will be a table with two columns, “Number” and “Factorial” and 10 rows, one for each number from 1 to 10.

Course Outcome 5 (CO5):

1. What is Route Model Binding in Laravel? Which types of route model binding are supported in Laravel?

Ans. Route model binding is a feature in Laravel that allows the framework to automatically inject an instance of a model class based on the value of a route parameter. This is done by specifying a type-hint for a controller method's parameter and using a specific naming convention for the route parameter.

There are two types of route model binding supported in Laravel: implicit and explicit.

Implicit Route Model Binding: This type of binding automatically matches the value of the route parameter to the primary key of the model class and returns an instance of that model. The naming convention for the route parameter must match the name of the primary key column in the database table.

Explicit Route Model Binding: This type of binding requires that you specify the method to be used to retrieve the model instance. This allows you to use a custom column other than the primary key for matching the route parameter to a model.

In both types of binding, Laravel will automatically handle the case when the model is not found and will throw an HTTP 404 exception. In summary, Route Model Binding in Laravel is a powerful feature that allows you to easily retrieve models from the database based on a route parameter and automatically handle cases where the model is not found.

2. Explain how Laravel performs route handling using routes calling controller methods.

Ans. In Laravel, routes are used to define the URL patterns that the application should respond to, and the controller methods that should be called when a user visits those URLs.

When a user makes a request to a specific URL, Laravel's routing component checks the list of defined routes to find a matching pattern. If a matching pattern is found, the corresponding controller method is executed.

When a request is made, Laravel's routing component will match the requested URL and HTTP method with the defined routes, and call the appropriate controller method. The controller method can then handle the request and return a response to the user.

Laravel also supports passing parameters to controller methods through the URL. In summary, Laravel's routing component handles the mapping of URLs to controller methods, allowing developers to easily define how the application should respond to different types of requests and providing a simple way to pass parameters to controllers through the URL.

3. List the data types used in JSON? Explain the use of parse() and stringify() functions in JSON with examples.

Ans. In JSON, there are several data types that can be used:

String: a sequence of characters, enclosed in double quotes.
Number: a decimal number, which can be an integer or a floating-point number.
Boolean: a true or false value.
Array: a collection of values, enclosed in square brackets.
Object: a collection of key-value pairs, enclosed in curly braces.
Null: a null value.

The JSON.parse() function is used to convert a JSON string into a JavaScript object. It takes a JSON string as an argument and returns a JavaScript object that can be used to access the data in the JSON string.

For example:

```
let jsonString = '{"name":"John", "age":30, "city":"New York"}';
let obj = JSON.parse(jsonString);
console.log(obj.name); // Output: "John"
```

In this example, the JSON.parse() function is used to convert the JSON string '{"name":"John", "age":30, "city":"New York"}' into a JavaScript object. Then, the obj.name will return "John"

The JSON.stringify() function is used to convert a JavaScript object into a JSON string. It takes a JavaScript object as an argument and returns a JSON string that can be stored or transmitted.

For example:

```
let obj = {name: "John", age: 30, city: "New York"};
let jsonString = JSON.stringify(obj);
console.log(jsonString); // Output: '{"name":"John","age":30,"city":"New York"}'
```

In this example, the JSON.stringify() function is used to convert the JavaScript object {name: "John", age: 30, city: "New York"} into a JSON string. The function returns a string '{"name":"John","age":30,"city":"New York"}'

In summary, JSON is a lightweight data-interchange format that uses human-readable text to transmit data objects consisting of attribute-value pairs and array data types. The parse() and stringify() functions are used to convert JSON string to JavaScript object and JavaScript object to JSON string respectively.