Experiment 1 (Viva Questions)

Q1: How would you use the href attribute in HTML to create hyperlinks?

A1: The href attribute in <a> is used to create links to external pages or sections. For internal navigation, use href="#id" for anchor links. For example, Click Here links to another page.

Q2: How would you structure the HTML code to list courses?

A2: Use <ul> for an unordered list or <ol> for an ordered list, adding <li> for each course. For example: Course 1
Course 2 to list courses.

Q3: How do you use the id attribute to navigate to a section?

A3: Assign a unique id to the section (e.g., <section id="courses">). Then use an anchor link like <a href="#courses"> to navigate to that section from anywhere in the page.

Q4: How would you display faculty info using list tags?

A4: Use <ul> and <li> tags for each faculty member. For example: Dr. Smith
Prof. Johnson to list faculty names in an unordered list.

Q5: How would you link to the Library section using an id?

A5: Set an id attribute like id="library" in the section tag. Then, use Library to create a link that navigates to that section.

Experiment 2 (Viva Questions)

Q1: How would you create a webpage with two frames that divide the page into two equal rows?

A1: Use the <frameset rows="50%,50%"> tag to divide the page into two rows, with each row containing a <frame> tag to display separate content in each.

Q2: How can you divide each row into equal columns using HTML and CSS?

A2: Inside each row <frameset>, use the <frameset cols="50%,50%"> tag to split each row into two equal columns. This creates a two-column layout within each frame.

Q3: How would you fill each frame with a different background color using CSS?

A3: In the HTML content for each frame, use CSS to set the background color, such as body { background-color: lightblue; } inside the <style> tag or inline style within each frame's content.

Q4: How can you ensure that each row is divided equally on the webpage?

A4: Use equal values in the frameset rows attribute, such as rows="50%,50%", to divide the page equally between the top and bottom rows.

Q5: How does the display: flex property work in CSS to create equal columns within a row?

A5: The CSS flexbox layout allows you to use display: flex to arrange child elements in a row or column. Set justify-content to distribute space evenly, ensuring equal-width columns.

Experiment 3 (Viva Questions)

Q1: How would you use internal CSS to set an attractive background color for your hometown's webpage?

A1: Use <style> in the <head> section of your HTML file and set body { background-color: skyblue; } to apply a pleasing background color to the entire page.

Q2: How would you use internal CSS to set a different text color for headings and paragraphs?

A2: Define specific CSS styles like h1 { color: red; } and p { color: black; } within the <style> tag. This changes the text color for headers and paragraphs.

Q3: How can you add an image to your hometown's webpage using internal CSS?

A3: Use <img src="image.jpg"> to place an image on the page, then style it with CSS for size or borders. Example: img { width: 100%; } to make the image responsive.

Q4: How would you use internal CSS to set a custom font for the text on your webpage?

A4: Inside the <style> tag, use font-family: 'Arial', sans-serif; to change the font style of the body or specific elements on your webpage.

Q5: How would you use internal CSS to add styling to a navigation menu?

A5: Use internal CSS like nav { background-color: gray; padding: 10px; } to style the navigation menu, giving it a distinct background and padding.

Experiment 4 (Viva Questions)

Q1: How would you use external CSS to format your college webpage?

A1: Link an external CSS file using <link rel="stylesheet" href="styles.css"> inside the <head> section. This keeps styles separate and reusable across pages.

Q2: How would you ensure that the external CSS file is linked correctly?

A2: Ensure the file path is correct in the href attribute and that the .css file is in the right location within your project directory.

Q3: How can you use inline CSS to format individual elements directly within the HTML document?

A3: Inline CSS is applied directly to an HTML element using the style attribute. Example: Text applies styles only to that paragraph.

Q4: Explain the advantages of using external CSS over internal and inline CSS.

A4: External CSS keeps HTML clean, reduces redundancy, and allows reuse across multiple pages, making it easier to maintain. It also helps with faster loading times.

Q5: How would you use internal CSS to set a background color for the entire college webpage?

A5: Use <style> in the <head> section with body { background-color: lightblue; } to set a consistent background color across the entire page.

Experiment 5 (Viva Questions)

Q1: How would you create an HTML form to take an integer input from the user?

A1: Use <input type="number" id="num"> in an HTML form to accept integer input from the user, ensuring only numbers can be entered.

Q2: How would you use JavaScript to check if the entered number is odd or even?

A2: In JavaScript, use a simple condition like if (num % 2 == 0) alert("Even"); else alert("Odd") to check whether the number is divisible by 2.

Q3: How can you ensure that the user enters an integer and not a decimal or a non-numeric value?

A3: Use the <input type="number"> attribute in HTML, and check with JavaScript if the input is a valid integer using Number.isInteger().

Q4: How would you modify the JavaScript function to display the result on the webpage instead of using an alert box?

A4: Instead of using alert(), use document.getElementById("result").innerText = "Even" to display the result directly in a designated area on the page.

Q5: How would you handle the case where the user enters a non-integer value or leaves the input field empty?

A5: Use JavaScript validation to check if the input is empty or not a number, and show an error message like "Please enter a valid number."

Experiment 6 (Viva Questions)

Q1: What HTML elements are used to create the form fields?

A1: Use <input> for text, email, and number fields, <textarea> for multi-line text, and <select> for dropdowns within a <form> tag to create form fields.

Q2: What JavaScript function is called when the button is clicked?

A2: You can use the `onclick` event handler in the button, e.g., <button onclick="validateForm()">Submit</button>, which calls the validateForm() JavaScript function.

Q3: How is the combined information displayed to the user?

A3: Use JavaScript to combine form values, e.g., name = document.getElementById("name").value, then display the result in a text field using document.getElementById("output").value.

Q4: How can you implement validation to ensure the email field contains a valid email address?

A4: Use HTML5 input type="email" for basic validation or add a custom regex in JavaScript to check the email format before submitting the form.

Q5: How can you restrict the input in the mobile number field to accept only numeric values?

A5: Use <input type="tel" pattern="\d{10}" to restrict the input to only 10 digits, and ensure the field only accepts numbers.

Experiment 7 (Viva Questions)

Q1: What is the root element of the XML document?

A1: The root element is the top-level element that wraps all other elements. For example, <students> could be the root element that contains multiple <student> entries.

Q2: What elements are used to represent each student's information?

A2: Use <student> tags for each entry, with child elements like <name>, <email>, <mobile> to store specific data about the student.

Q3: How can you add another student’s information to this XML file?

A3: Copy the <student> block and add new data inside it. Update the values for name, email, and other relevant details for the new student.

Q4: How can you modify an existing student’s information in this XML file?

A4: Open the XML file and update the values inside the corresponding <student> tags to modify existing student information.

Q5: How can you remove a student’s information from this XML file?

A5: To remove a student, simply delete the entire <student>...</student> block from the XML file.

Experiment 8 (Viva Questions)

Q1: How does the PHP script open the text file for reading?

A1: Use fopen("file.txt", "r") to open the file in read mode, allowing the PHP script to read its contents line by line.

Q2: What function is used to split each line into an array using ":" as the delimiter?

A2: The PHP function explode(":", $line) is used to split the line into an array where each part (name, password, email) is separated by a colon.

Q3: What does the feof() function do?

A3: The feof() function checks if the end of the file has been reached while reading the file, helping to prevent errors when trying to read past the end.

Q4: How can you improve the script to handle empty lines or lines with incorrect formatting?

A4: Use PHP's trim() function to remove unnecessary spaces and check for line validity before processing the data. Ignore any empty lines.

Q5: How can you add CSS styling to the HTML table to improve its appearance?

A5: Use <style> or an external CSS file to apply styles to the table, such as borders, background color, padding, and text alignment for a better presentation.

Experiment 9 (Viva Questions)

Q1: How does the PHP script retrieve the username and password entered by the user?

A1: The PHP script uses the $_POST superglobal to retrieve the values entered in the login form. Example: $username = $_POST['username'];.

Q2: How does the script read the stored usernames and passwords from the file?

A2: The PHP script opens the file using fopen(), reads it line by line with fgets(), and uses explode(":", $line) to split each line into username and password.

Q3: What function is used to split each line of the file into an array using ":" as the delimiter?

A3: The explode(":", $line) function is used to split each line at the colon (:) into an array of username, password, and email.

Q4: How does the script display a message indicating whether the login attempt was successful or not?

A4: Use echo or print to output success or failure messages based on whether the entered username and password match the stored values.

Q5: How can you enhance the security of the login process?

A5: To enhance security, use password hashing functions like password_hash() to store passwords securely and avoid storing them in plain text.

Experiment 10 (Viva Questions)

Q1: Explain the process of connecting PHP to a MySQL database.

A1: Use mysqli_connect("localhost", "username", "password", "database") to connect to a MySQL database, then run queries using mysqli_query() to interact with the database.

Q2: How do you create a MySQL database and table for storing user information?

A2: Use SQL commands like CREATE DATABASE database_name and CREATE TABLE table_name (columns) to create a new database and table in MySQL for storing user information.

Q3: How do you insert user information into a MySQL table using PHP?

A3: Use an INSERT INTO SQL query like INSERT INTO users (name, email) VALUES ('$name', '$email'); and execute it with mysqli_query().

Q4: Explain the use of the mysqli_query() function in PHP for executing MySQL queries.

A4: The mysqli_query() function is used to execute SQL queries from PHP to interact with the MySQL database. It returns the result of the query or true if successful.

Q5: How can you secure user input before storing it in a MySQL database using PHP?

A5: Use mysqli_real_escape_string() to escape special characters in user input and prevent SQL injection attacks.

Experiment 11 (Viva Questions)

Q1: How to design a simple calculator using JavaScript?

A1: Create HTML buttons for digits and operations (addition, subtraction, etc.). Use JavaScript functions to perform operations when buttons are clicked, such as addition (num1 + num2). Display results using the document.getElementById() method to update the display.

Q2: How to handle user input for the calculator?

A2: Use input fields or buttons for digits, and associate each button with a JavaScript function to update a display area. For example, use button.onclick = function() { display.value += this.innerText; } to add the button's value to the display.

Q3: How to perform basic arithmetic operations in the calculator?

A3: Define JavaScript functions for each operation, such as addition, subtraction, multiplication, and division. Each function will take the input from the display, perform the operation, and update the result.

Q4: How to clear the display in the calculator?

A4: Add a "Clear" button with an onclick event that sets the display value to an empty string: document.getElementById("display").value = "" to reset the calculator.

Q5: How to make the calculator interactive with the user input?

A5: Use JavaScript to listen for button clicks or keyboard input and perform calculations in real time. Update the display dynamically after each calculation using the .innerText or .value methods.

Experiment 12 (Viva Questions)

Q1: How to design a quiz website using JavaScript?

A1: Create HTML elements for questions and answer options (radio buttons or checkboxes). Use JavaScript to capture the selected answers and compare them with the correct answers stored in the script.

Q2: How to calculate the score based on the quiz answers?

A2: After the user submits the quiz, use JavaScript to check if each selected answer is correct. For each correct answer, increment a score variable, then display the total score on the screen.

Q3: How to display the results of the quiz to the user?

A3: After calculating the score, use JavaScript to display the result on the page by updating an HTML element's innerText or innerHTML. Example: document.getElementById("result").innerText = "Your score is: " + score.

Q4: How to ensure that the user has answered all questions before submitting?

A4: Use JavaScript to check if each question has a selected answer before allowing the user to submit. If any question is unanswered, prompt the user to complete it before proceeding.

Q5: How to make the quiz interactive and visually appealing?

A5: Style the quiz with CSS to create an engaging layout. Add animations or transitions for buttons, and use JavaScript to provide feedback after each answer (correct/incorrect) and highlight the correct answers.