raisoni EDUCATION

EXPERIMENT NO: 01

Aim: To Design and develop a simple static website for an Online Book Store using HTML, containing the following pages:

1. Home Page: Consisting of three frames (Header, Menu, and Content)
2. Login Page: For user authentication
3. Catalogue Page: Displaying book details in a tabular format.
4. Registration Page: For new user registration.

Objective: Understand a basics of HTML and Frames.
Learn to link multiple web pages together for website navigation.
Design basic forms for user input.

Theory: ① HTML (HyperText Markup Language):

- It is the standard language for creating web pages. It structures the content using elements like <html>, <head>, <body>, etc.

② Frames in HTML:

- Frames divide a web page into multiple sections, each displaying a different document.

Ex.
```
<frameset rows = "20%,80%,">
  <frame src = "header.html">
  <frameset cols = "20%, 80%">
    <frame src = "menu.html">
  </frameset>
</frameset>
```

raisoni EDUCATION

④ HTML Forms
- Used to gather user inputs like username, password, name, and email.

Ex.
```
<form>
  <input type = "text" name = "username" placeholder
                                  = "Enter Username">
  <input type = "password" name = "password"
                  placeholder = "Enter password">
</form>
```

⑤ HTML Tables:
Used to display book details in rows and columns.

Ex.
```
<table border = "1">
  <tr><th> Book ID </th><th> Title </th><th>Author </th></tr>
</table>
```

① Home Page:
Create a static home page with 3 frames.
Top frame: Website Title
left frame: Navigation Menu
Right frame: Content Area

② Login Page:
Design a login form with:
① UserName
② Password
③ Login button

raisoni
EDUCATION

③ Catalogue Page:
- Display a list of books in a table format with columns.
① Book ID ② Title ③ Author ④ Price

④ Registration Page:
Create a registration form with:
① Name ② Email ③ Password ④ Submit button.

Conclusion : In this program we complete the following concepts and implement.
Create basic static web pages using HTML
Use frames to structure a webpage
Design forms for user login and also understand the fundamentals of website development.

raisoni EDUCATION

EXPERIMENT NO: 02

Aim: To write a Javascript validation script for a registration form that checks user input fields for correctness before submitting the data.

Objective: Understand Javascript form validation techniques.
Learn to validate user input fields like Name, Email, Password, Mo. no. & Address

Theory:

① JavaScript Validation:
- Client-side validation is done using Javascript to ensure that user inputs valid data before submitting the form.
- Functions like document.getElementById() & addEventListener() are commonly used to interact with form fields and handle validation.

② Common Validation Techniques:
- Text Length Check: Ensure a field contains a minimum number of characters.
- Regular Expressions: Used to validate patterns like emails and mobile numbers.
- Empty field check: Ensure a field is not left blank.

③ Key Validation Rule Explained:
- First Name
- Password

raisoni
EDUCATION

Email ID, Mobile Number, last Name & Address.

Conclusion :

After completing the practical we will be able to validate form fields using Javascript Understand how to prevent form submission when fields are invalid. Learn to use regular expressions and DOM manipulation.

raisoni EDUCATION

EXPERIMENT NO: 03

Aim: To develop and demonstrate the usage of Inline, Internal & External Stylesheets using CSS in an HTML document to style a web page.

Objective: Understand the concept of CSS (Cascading Style Sheets)
- Learn the different types of CSS styles:
  ① Inline CSS (applied directly to an element)
  ② Internal CSS (within <style> tag in the <head> section)
  ③ External CSS (linked from an external stylesheet file.)

Theory:

What is CSS?

CSS (Cascading Style Sheets) is a language used to style the visual presentation of HTML elements. It controls properties like color, font, layout & spacing.

Types of CSS Styles:

① Inline CSS: Applied directly within an HTML element.
<h1 style = "color": blue;"> Welcome</h1>

② Internal CSS: Within inside the <style> tag

within <head>
Ex. <style> h1 { color: blue; } </style>

External css:
Stored in an external file and linked using <link>
Ex. <link rel = "stylesheet" href = "style.css" >

Code Implementation:

HTML File (index.html)

```
<html lang="en">
<head>
    <meta charset = "UTF-8">
    <title> Css Styles Demonstration </title>
    <link rel = "stylesheet" href = "style.css">
    <style>
        p {
            color : green;
            font size : 18px;
          }
    </style>
</head>
<body>
    <h1 style = "color : blue;"> Welcome </h1>
    <p> This is a paragraph </p>
    <button> Click me </button>
</body>
</html>
```

css file (style.css)

En.

```
button {
    background-color : orage ;
    color : white;
    padding : 10px 20px;
    border : none;
}
```

Conclusion : In this practical we ~~will~~ complete the following concept as follows:

- Understand the differences and use cases of Inline, Internal and External css.
- Learn how to structure style effectively in web page.
- Appreciate the importance of seperating content and style in modern web development

raisoni EDUCATION

EXPERIMENT NO : 04

Aim : To develop and demonstrate Javascript pop-up boxes (alert, prompt, confirm) and functions to perform the following operations:
1) Display Current Date in a text box on button click.
2) Calculate factorial of a number entered using prompt().
3) Display Multiplication Table of number from 1 to 10 using prompt() and alert()
4) Add Numbers by taking input using promt() & confirm(), & display the sum using alert().

Objective : Understand the use of Javascript Pop-up Boxes.
- alert(): Display the message.
- prompt(): Take input from the user.
- confirm(): Ask for user confirmation (Yes/No)

. Learn Javascript event handling using the onclick() function.

Theory :

JavaScript Pop-Up Boxes:

JavaScript provides three types of pop-up boxes to interact with users:

1) Alert Box (alert()) - Displays a simple message to the user.

raisoni EDUCATION

2) Prompt Box (prompt()) - Accepts user input and returns it as a string.

3) Confirmation Box (confirm()) - Asks for confirmation with OK/ Cancel options

* Javascript Functions:

- Functions allow us to group code and execute it whenever needed.

Eg. function greet() {
    alert("Hello");
}

* Javascript Event Handling
(onclick() is used to execute Javascript when a user clicks a button

- It helps trigger functions dynamically.

Conclusion: Through this we learned:
How to use Javascript pop-up boxes (alert), prompt(), and confirm().
- How to handle user input dynamically.
- How to use function and onclick events to trigger Javascript actions.

raisoni EDUCATION

EXPERIMENT NO: 05

Aim: Write an HTML page that contains a selection Box with a list of 5 countries. When the user selects a country it capital should be printed next in the list. Add CSS to customize the properties of the font of the capital. (Color, bold and font size)

Objective: The Display a selection box with five countries.

Show the capital city of the selected country dynamically.

- Apply CSS styling to customize the appearance of the capital name.

Theory:

1) HTML Dropdown (<select>)

- The <select> tag is used to create a dropdown list.

- The <option> tags define the list items in the dropdown.

- Using Javascript, we can detect the selected value and display its corresponding capital.

2) Javascript Event Handling.

The onchange event detects when a user selects an option in the dropdown.

raisoni
EDUCATION

3) Css styling for Text Appearance.

- The capital city's font properties are customized using Css.

Color : Change text color
Bold : Use font-weight : bold;
Font size : set with font-size

Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title> Country and Capital Selector </title>
  <style>
    body {
      font-family : Arial;
      text-align : center;
      margin-top : 50px;
    }
    select {
        padding : 8px;
        font-size : 16px;
      }
  </style>
  <script>
    function showCapital() {
      var country = document.getElementById
      ("countrySelect").value;
      var capital = "";
```

raisoni EDUCATION

```
switch (country){
    case "USA" :
        capital = "Washington, D.C." ;
    break;
  case "UK" :
    capital = "London";
    break;
 }
 }
 </script>
</head>
<body>
 <h2> Select a country to see its Capital </h2>

<select id = "countrySelect" onchange = "showCapital()">
   <option value = "" > -- Select country -- </option>
   <option value = "USA"> USA </option>
   <option value : "UK"> UK </option>
</select>
 <p id = "capitalDisplay" ></p>
</body>
</html>
```

Conclusion: Through this implementation, we successfully.

Used HTML <select> to create a dropdown list Applied Javascript event handling to display the selected country's capital dynamically. Styled the capital text using CSS properties like color, bold and font size.

raisoni EDUCATION

EXPERIMENT NO: 06

Aim: Develop and demonstrate PHP script for the following problems.
- Write a PHP script to find out the sum of the individual Digits
- Write a PHP script to check whether the given number is palindrome or not.

Objective:
- To understand PHP's number manipulation techniques.
- To implement control structures and loops for solving numerical problems.
- To enhance problem-solving skills using PHP

Theory:

1) Sum of Individual digits:
- A number consists of digits, and we can extract each digit using modulus (%) and integer division (floor() or /).
- By interating through each digit and summing them, we obtain the sum of individual digits.

2) Palindrome Check:
- A palindrome is a number that remains the same where reversed (e.g. 121, 1331)

raisoni EDUCATION

- To check, we reverse the number and compare it with the original.
- If both are equal, the number is a palindrome.
- PHP Code Implementation:

```
<?
function sumOfDigits ($num) {
    $sum = 0;
    while ($num > 0) {
        $sum += $sum % 10;
        $num = (int) ($num / 10);
    }
    return $sum;
} // function to check if a number is palindrome
function isPalindrom ($num) {
    $original = $num;
    $reverse = 0;

while ($num > 0) {
    $digit = $num % 10
    $reverse = ($reverse * 10) + $digit;
    $num = (int)($num / 10);
}
return $original == $reverse;
}

// Example input:

$number = 121;
echo "Sum of digits of $number :". sumOfDigits ($number).
"<br>";
```

```
if (isPalindrome($number)) {
    echo "$number is a Palindrome.";
} else {
    echo "$number is not Palindrome.";
}
?>
```

Conclusion:

- We successfully implemented a PHP script to calculate the sum of individual digits of a number.
- We developed a function to check if a number is a palindrome by reversing it and comparing it with the original.
- This exercise helped in understanding PHP loops, arithmetic operations, and control structures for problem-solving.

raisoni EDUCATION

EXPERIMENT NO : 07

Aim : Write a XML program to validate student details (Roll No, Name, college & branch) using DTD.

Objective :

- To understand the structure of XML and how data is represented in a hierarchical format
- To learn how to define rules for XML documents using DTD.
- To validate XML document against a DTD to ensure data consistency and correctness.

Theory

- XML is widely used format for storing and transporting structured data.
- DTD (Document Type Definition) defines the structure and rules of an XML document, such as element names, attributes, and nesting.
- The XML document must adhere to the rules defined in the DTD to be considered valid.

Types of DTD :

1) Internal DTD : Defined within the same XML document.

raisoni
EDUCATION

2) External DTD: Defined in a seperate file and linked to the XML document.

XML and DTD code Implementation.

1) Student XML Document (students.xml)

```
<? xml version = "1.0" encoding = "UTF-8" ?>
<! DOCTYPE students SYSTEM "students.dtd" >
<students>
    <Student>
      <rollno> 101 </rollNO>
      <name> John Doe </name>
      <college> XYZ University </college>
      <branch> computer scienc</branch>
    </Student>
    <student>
      <rollNO> 102 </rollNO>
      <name> Jane Smith </name>
      <college> ABC Institute </college>
      <branch> Electrical Engineering </branch>
    </student>
  </students>
```

Explaination: The XML document (students.xml) contains student details, including rollno, name, college, and branch.

The DTD file (students.dtd) ensure that.

raisoni
EDUCATION

- The <student> element contains one or more <student> element (student+).
- Each <student> contains <rollNo>, <name>, <college>, & <branch>.

Conclusion:

- We successfully created an XML document to store student details.
- The XML document was validated using an external DTD, ensuring proper structure and data integrity.
- This implementation helps in maintaining data consistency, validation, and structured storage in XML-Based applications.

raisoni EDUCATION

EXPERIMENT NO : 08

Aim : Implement the web applications using
a) PHP b) Servlets c) JSP.

Objective :

- To understand and implement server-side programming using PHP, servlets, and JSP.
- To explore request handling, form submission and database interaction using different web technologies.
- To compare different approaches to developing dynamic web applications.
- To enhance problem-solving skills for developing scalable and efficient web applications.

Theory :

1) PHP (Hypertext Preprocessor) :

- PHP is widely-used server-side scripting language embedded in HTML.
- It interacts with databases like MySQL and supports session management, file handling, and form validation.
- PHP scripts run on the server and generate dynamic content for web pages.

raisoni EDUCATION

Ex. Simple PHP web application

```
<?php
 echo "Welcome to PHP Web Application!";
?>
```

2) Servlets (Java Servlets):
Servlets are Java-based web components that run on a web server.
They handle HTTP requests and responses and interact with databases.
Unlike PHP, servlets require compilation and deployment in a servlet container like Tomcat

Ex.

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServlet
    Response response) throws ServletException, IOException
    {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<h1> Welcome to Java Servlets !</h1>");
    }
}
```

3) JSP (JavaServer Pages):

- JSP is an extension of Servlets, allowing Java code to be embedded with HTML using <%...%> tags.
- It supports custom tags, Expression Language (EL), and JavaBeans for better code organization.

Ex. Simple JSP web Application:

```
<%@ page language = "java" contentType = "text/html" %>
<html>
<head>
 <title> JSP Example </title>
</head>
<body>
 <h1> Welcome to JSP web Application ! </h1>
</body>
</html>
```

Conclusion:

We implemented dynamic web applications using PHP, Servlets, and JSP.
- PHP is easy to use, but Java-based technologies offer scalability and security.
- Understanding these technologies helps in developing modern, interactive, and database-driven web applications.

raisoni EDUCATION

EXPERIMENT NO : 09

Aim : Implement the web applications with Database using a) PHP b) Servlet & c) JSP.

Objective :

- To understand how server-side programming interacts with a database.
- To learn how to perform CRUD (Create, Read, Update, Delete) operations using PHP, Servlets, and JSP.
- To compare different methods for handling user input, queries, and displaying data dynamically.
- To implement a simple user registration and retrieval system using a database.

Theory :

1) PHP with database (MySQL) :

- PHP can connect to MySQLi or PDO extensions.
- Queries are executed using mysqli_query() or prepare()

Eg.

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "testdb";

// Create Connection
$conn = new mysqli ($servername, $username, $password,
                    $dbnam);
```

```
if ($conn->connect_error) {
    die ("Connection failed:" . $conn->connect_error);
}
// Insert Data

$sql = "INSERT INTO students (name, email) VALUES ('John
    Doe', 'john@example.com')";
$conn->query ($sql);

//Retrive data
$result = $conn->query ("SELECT * FROM students");

while ($row = $result->fetch_assoc()) {
    echo "Name:" . $row ["name"] . ", Email:" .
    $row ["email"] . "<br>";
}
$conn->close();
?>
```

2> Java Servlets with Database

- Java servlets use JDBC (Java Database Connectivity) to interact with a database.
- The DriverManager class loads the MySQL and SQL queries are executed using PreparedStatement

3> JSP with Database:

JSP can use JDBC just like servlets but allows embedding Java code directly in HTML

It follows the mvc pattern for better seperation of logic and UI.

Conclusion :

We successfully implemented database-driven web applications using PHP, servlets and JSP.

- PHP provides simplicity and flexibility for small to medium-sized applications.

- Servlets and JSP offer scalability and security, making them suitable for enterprise applications.