

1. You are updating a legacy web application to take advantage of modern web standards. The project involves transitioning from an earlier HTML version to HTML5. The goal is to leverage HTML5's new features to improve web development practices and enhance the functionality of web forms.

Update the DOCTYPE: Ensure your HTML documents start with the HTML5 doctype to leverage HTML5 features:

Use New Semantic Elements: Replace generic `<div>` and `` tags with HTML5 semantic elements to improve readability & accessibility.

- * `<headers>` for headers
- * `<nava>` for navigation skills
- * `<Section>` for sections
- * `<article>` for self-contained content
- * `<aside>` for side content
- * `<Footer>` for footers
- * `<main>` for primary content

Update Forms with New Input:

Types: HTML5 introduces new input types that enhance user experience and validation

- * `<input type = "email">` for email address
- * `<input type = "url">` for URLs
- * `<input type = "date">` for dates
- * `<input type = "number">` for numeric input

Add Form Attributes: Utilize new attributes to improve form functionality.

- ⇒ required to make fields mandatory
- ⇒ placeholder to provide hint text
- ⇒ pattern for custom validation with regex
- ⇒ auto complete to suggest or remember inputs

Implement New Form Elements: Take advantage of new form elements.

- * <datalist> to provide options for an <input>

Improve Accessibility

⇒ ARIA Roles and Attributes: use ARIA roles and attributes to improve accessibility.

Example For Transitioned Form:

```
<Form action ="/submit"
      method = "post">
    <div>
      <label
        for = "username"> Username : </label>
      <input type = "text"
            id = "Username" name = "username"
            required>
    </div>
    <div>
      <input type = "submit">
    </div>
</Form>
```

Output:

Welcome to our website

Name	Birthday date

Designing a user registration form for a new web application. The form needs to capture essential information such as name, email, and a message from users. It is crucial to implement client-side validation to ensure that the data entered is accurate and complete before the form is submitted.

HTML Structure:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title> User Registration Form </title>
<styles>
    body {
        font-family: Arial, sans-serif;
        max-width: 600px;
        margin: auto;
    }
    form {
        display: flex;
        flex-direction: column;
    }
    label {
        margin: 10px 0 5px;
    }
    input, textarea {
        padding: 8px;
        margin-bottom: 10px;
        border: 1px solid #ccc;
    }

```

```

button {
    padding: 10px;
    border: none;
    border-radius: 4px;
    background-color: #4CAF50;
    color: white;
    cursor: pointer;
}

#45a049 {
    list-style-type: none;
}

<h1> User Registration </h1>
<form id="registration-form">
    <label for="name">Name</label>
    document.getElementById("registration-form").addEventListener("submit", function(event) {
        var name = document.getElementById("name");
        if (!name || !email || !message) {
            alert("All fields are required!");
        }
    })
    <script>
        </script>
    </body>
</html>

```

output

User Registration		
Name	Email	Message

3. Designing a website that needs to function effectively across various devices and screen sizes. The design should include different types of layouts such as Fixed, Fluid and responsive. To achieve this, you need to use CSS techniques like flexbox and maintains a consistent user experience.

Assignment - 2

Fixed Layout:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset = "UTF-8">
    <meta name = "viewport"
content = "width=device-width, initial-scale=1.0">
  <style>
    body {
      margin: 0;
      padding: 0;
    }
    <div class = "container">
      <div class = "header"> Fixed Layout
        Header </div>
      <div class = "main"> Fixed width
        Content </div>
    </div>
  </style>
</html>
```

Fluid Layout:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset = "UTF-8">
    <meta name = "viewport">
    <style>
      main {
        padding: 20px;
      }
    </style>
  </head>
  <body>
    <div class = "container">
```

```
<div class="head"> &gt; fluid Layout header  
</div> output:  
<body>  
</html>
```

Responsive web design

This is the main content area

By incorporating these CSS techniques, we can create a website with fixed, fluid, and responsible layouts that adapt to various devices' screen sizes user-friendly experience.

ASSIGNMENT - 2

4. To create a webpage that offers a seamless user experience across devices, responsive design principles must be applied. This involves using fluid grids, flexible images and media queries to ensure the layout adjusts gracefully to different screensizes. Additionally, touch friendly elements and optimized images ensures fast loading times and a smooth experience on all devices.

Fluid Grids: percentage based widths

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
<meta charset="UTF-8">  
<meta name="viewport"  
content="width=device-width,  
initial-scale=1.0">  
<title> Responsive Layout with Fluid Grids  
</title>  
<style>  
body {
```

```
margin: 0;
font-family: Arial;
border-box: 1px solid black;
background-color: white;
background-size: cover;
background-position: center;

@media (max-width: 768px) {
    .container {
        grid-template-columns: 1fr;
    }

    <div class="container">
        <div class="item">content</div>
        <div class="item">content</div>
    </div>
    </body>
</html>
```

Flexible Images:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        
    </body>
</html>
```

Output:

Responsive web design
Screen size for optimal experience

Use fluid grids for flexibility, media queries for responsive design, and techniques like flexible images and touch friendly elements to enhance usability and performance across devices.

5. Given a scenario (eg: creating a blog post, a product listing), design a webpage using appropriate elements (table lists, and images) for optimal readability and user experience (HTML, CSS, Javascript, images)

Webpage Design: Product Listing

Sample HTML layout that it includes product images, description, and pricing, laid out with tables, and lists where appropriate

HTML Structure

```
<!DOCTYPE HTML>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport"
      content="width=device-width, initial-scale=1">
    <title>Product Listing </title>
    <link rel="stylesheet"
      href="styles.css">
    <script src="script.js" defer></script>
```

```
</head>
<body>
    <header>
        <h1> product Listings </h1>
        <nav>
            <ul>
                <li>a
                    href = "#home"> Home </a> </li>
                <li>a
                    href = "#contact"> contact </a> </li>
            </ul>
        </nav>
        </header>
        <main>
            <p>Copy : 2024 my company</p>
        </main>
        <footer>
            </footer>
    </body>
</html>
```

Java Script (script.js)

```
document.addEventListener("DOMContentLoaded"
    , () =>
    console.log("document loaded and parsed");
);
```

output

My Blog
By Author Name on citation

Assignment split up

Step split up

Step	Task	Total Marks
1.	Form Design	4
1.	Button Design	4
1.	Features of HTML	2
2.	Form Design	4
2.	Validation	4
2.	Layout	2
3.	Layout design	4
3.	css Background	6
4.	Form design	4
4.	Layout design	4
4.	Navigation menu	2
5.	Form design	4
5.	css Layout	4
5.	Java script validation	2

ASSIGNMENT - 3

Implementing a feature in a web application that tracks the number of access by a client with a single session. Their session and retrieve information about the session, such as the session ID, creation time and last accessed time.

```
Import javax.servlet.ServletException
Import javax.servlet.annotation.WebServlet;
Import javax.servlet.http.HttpServlet;
Import java.io.IOException;
Import java.io.PrintWriter;
Public class TrackSession extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        HttpSession session = request.getSession(true);
        Integer accessCount = (Integer) session.getAttribute("accessCount");
        session.setAttribute("accessCount", accessCount + 1);
    }
}
```

```

if (accessCount == null) {
    accessCount = 0;
}
accessCount++;
String sessionId = session.getId();
long creationTime = long lastCreationTime as
session.getAccessedTime();
PrintWriter out = response.getWriter();
out.println("<html><body><h1>");
out.println("ch2> Session Tracking </h2>");
ID = " + session.getId() + "</p>";
Time = " + new
ACCESS = " + accessCount + "</p>";
out.println("</body></html>");
}
}

output:
<p>Session ID: 1234567890abc </p>
<p>Creation Time: Mon Sep 10 15:20:30
2024 </p>
<p>Last Accessed Time: Mon Sep 10
15:25:10 2024 </p>
<p>Number of accesses: 5 </p>

```

2. Write a scenario where you had to use JSR2 to solve a complex problem and how you went about it. How to create custom functions.

Scenario using JSTL

In a web application, you have a requirement to display a list of products with varying categories

The product list needs to be dynamically categorized into separate sessions on a web page

Solution using JSTL:

```
import javax.servlet.ServletException
import javax.servlet.annotation.WebServlet
import javax.servlet.http.HttpServlet
import java.io.IOException
import java.util.List;
import java.util.Map
import java.util.HashMap

public class
productList extends HttpServlet {
protected void
doGet(HttpServletRequest request, HttpServletResponse)
product ("laptop", "Electronics", 1000, true),
new
product ("shirt", "Clothing", 30, true),
new
product ("washing machine", "Home Appliances", 500) false
product ("shirt", "Clothing", 30, true),
}
}
```

Output:

```
<strong>laptop</strong>- $1000 - Available
<strong>shirt - $30 - Available
<strong>washing machine - $500 -
out of stock
```

3. A page of stock market quotes uses script to refresh the page every five minutes in order to ensure the latest statistics remain available.

To achieve this setup the HTML page, Add Javascript for automatic refresh, and confirm dialog

HTML and Javascript

```
<!DOCTYPE html>
<html lang="en-US">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport"
      content="width=device-width, initial-
      scale=1.0">
  <title>Stock Market Quotes</title>
  <script>
```

```
    var userResponse = confirm ("The page
      will refresh in 20 seconds. Do you need
      your time?");
```

```
    if (!userResponse) {
      window.location.reload();
    } else {
      setinterval (refreshPage, 30000);
    }
  </script>
  <h1>Stock Market Quotes</h1>
  <p>Here you can display real-time
    stock market quotes</p>
</body>
</html>
```

output

Page Display

The page will refresh in 5 seconds.

Do you need more time?

[cancel]

Geography

4. You are developing an e-commerce application that needs to integrate with an external payment gateway service. Describe the steps involved in generating the client code, invoking the service and handling

for joy

output:

put: payment responses; payment success fully

amount: 100.00 USD

ASSIGNMENT - 4.

- From a developer's perspective, discuss why JDBC is essential in building database-driven applications. provide examples of executing SQL queries using JDBC statements

Configure Data source in context.xml:

<Context>

```
<Resource name="jdbc/myDataSource"
  auth="Container"
  type="javax.sql.DataSource"
  maxTotal="20"
  maxIdle="10"
  maxWaitMillis="10000"
  username="dbuser"
  password="dppassword"
  driverClassName="com.mysql.jdbc.JDBC4Driver"
  url="jdbc:mysql://localhost:3306/mydata"
  />
```

<Context>

Callable Statement:

```
Import java.sql.CallableStatement
import java.sql.Connection
import java.sql.SQLException
import java.sql.Types
import java.sql.Statement
public class CallableStatementExample
public void callStoredProcedure 'procedure int employee'
try {
    Connection conn = DatabaseUtil.getConnection();
    String sql = "call getEmployeeName(?, ?)";
    Statement stmt = conn.createStatement();
    ResultSet rs = stmt.executeQuery(sql);
    while (rs.next()) {
        System.out.println(rs.getString("EmployeeName"));
    }
}
```

```
System.out.println("Employee Name: " +  
    employee.getName());
```

```
catch (SQLException e)  
    e.printStackTrace();
```

4
4

Output:

Employee ID: 1, Name: John

Employee ID: 2, Name: John

Employee ID: 3, Name: Emily

Rows Updated: 1

Employee Name: John

2. Describe the life cycle of phases of a JSP page. Explain the significance of each phase in the JSP execution process. Discuss the different ways to embed Java code within a Java JSP pages.

Life cycle of phases of a JSP page:

1. Translation phase:

Description: The JSP page is translated into a Java servlet by the JSP engine. e.g. HTML mixed with JSP tags.

Significance: This ensures that the JSP content is converted into a form that the Java servlet container can execute.

2. compilation phase:

Description: The servlet container initializes the servlet instance.

Significance: Compilation ensures that the JSP is converted into executable.

3. Initialization phase:

Description: The servlet container initializes the servlet instance.

Significance: Initialization gets up any resources the JSP might need such as

4. Requesting processing phase

Description: The servlet processes incoming client requests by calling the service()

Significance: This phase is where the dynamic content generation occurs

5. Destroy phase:

Description: The servlet container destroys the servlet instance the destroy()

Significance: This phase ensures that resources are properly released

3. You need to develop a php program that generates a chessboard using HTML tables. The table should have a total width of 30px provide the code for this program

PHP code:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset = "UTF-8">
    <meta name = "viewport">
      content = "width=device-width, initial-scale=1"
```

<styles

table {

border-collapse: collapse;

width: 400px;

height: 400px;

td {

width: 80px;

height: 30px

For (\$row = 0; \$row < 8; \$row++) {

echo "<tr>";

For (\$col = 0; \$col < 8; \$col++) {

echo "<td>";

9.5

</td>

</tr>

</table>

</body>

</html>

Output:

[w][B][w][w][B][w][B]

[B][w][B][w][B][B][w]

[w][B][w][B][w][w][B]

[B][w][B][w][B][w][B]

[w][B][w][B][w][B][w]

[B][w][w][B][w][B][w]

- (4) You are developing a PHP application that reads content from a text file and uses regular expression to extract email address and phone number describe the steps & provide code for the application.

PHP code for the application

```
<?php  
$textfile path = "input.txt"  
$xmlPath = "output.xml";  
$textContent = file_get_contents($textfile path);  
$emailPattern = '/([a-zA-Z0-9-]+@[a-zA-Z0-9-]+\.[a-zA-Z]{2,})/  
$phonePattern = '/(\b\d{10}\b)/';  
preg-match-all($emailPattern, $textContent,  
$emails);  
preg-match-all($phonePattern, $textContent,  
$phones);  
$xml = new  
SimpleXMLElement('<Data/>' );  
$emailsElements = $xml->add('child'  
'email');  
SimpleXMLElement('<.Data/>' );  
$phoneElement = $xml->add('child'  
'phone number');  
echo "Data extracted and saved to  
XML file successfully";  
?>
```

Output:

Email: support@example.com

Phone number: 1234567890

Assignment -3	SPLIT UP	Total marks
1. code implementation	8	
1. session data accuracy	5	
1. efficiency & clarity	3	
1. explanation	4	
2. Scenario Explanation	6	
2. Function	5	
2. clarity & organization	4	
3. script	8	
3. Interaction Design	5	
3. code efficiency	4	
3. explanation	5	
4. wsol	6	
4. code	6	
4. error	4	
4. depth	4	

Assignment 4

	Splitup	Total Marks
1	Explanation of JDBC Connection pooling SQL queries Statement Type	5 6 5 4
2	Life cycle Java code Adv & Dis depth	6 5 5 4
3	Java code HTML Table Logic Cells Explanation	8 5 4 3
4	Java code Pattern extraction XML file DTO vs XML	8 5 4 3