SAVEETHA SCHOOL OF ENGINEERING
SIMATS, CHENNAI

SET -2

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.	10	CO1	2
2	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.	10	CO1	3
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 or Grid to ensure that the layout adapts well to different screen sizes and maintains a consistent user experience.	10	CO1	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 screen sizes. For instance, on a desktop, the webpage might display multiple columns with detailed content, while on a smartphone, the same content could stack vertically for easy scrolling. Navigation menus should transform into a hamburger icon on smaller screens, ensuring they remain accessible without taking up too much space. Additionally, touch-friendly elements and optimized images ensure fast loading times and a smooth experience on all devices.	10	CO1	3
5	Given a scenario (e.g., creating a blog post, a product listing), design a webpage using appropriate elements, tables, lists, and images for optimal readability and user experience. Also, describe step-by-step the sequence of HTTP requests and responses that occur when a user accesses a webpage containing multiple resources (HTML, CSS, JavaScript, images).	10	CO1	3

Assignment-1

1. Your updating a leg web application to take advantage of modern web standards. The projects involve transitioning from an earlier HTML version.

Solution :

Some key HTML 5 introduces new sematic elements that provide better structure to web pages, made more enhance web form functionality

1. **sematic elements :**
   to provide better structure to web pages, making them more accessible and easier to maintain ex: <header>, <nav>, <main>, <section>, <article>; <aside>, <footer> etc.,
2. **Form validation:** HTML 5 introduces built in form validation, which allows developers to define valigation rules. required, pattern, min, max .. etc..,
3. **New input types:** It Introduces new input types like data, time, datatime, month, week etc.,
4. **place holder attributes:** The atributes allows developers to provide a hint (or) example values from feilds, making easier to users.
5. **Autofocus attributes:** The auto famous attribute allows developers to specify feild should receive focus when the page loads.
6. **label element:** <label> element which allows developers to associate a label with a form feild, improving accessibility and user experience.

7. feildset and legend element: <fieldset> and <legend> elements allows developers to group related form feilds togeother making it easier for users.

8. HTML 5 validation API: It provides a set of methods and properties that allow developers to validata form feild programmatically, making it easier to implement custom validate logic.

9. CSS3 Selectors: It can be used in conjuction with css3 selectors to style form elements providing a more visually appealing.

10. Accessiblity features: It includes several accesibility features, such as ARIA attribute that make web forms more accible to users.

By following these steps and leverging HTML'5 new features, you can improve web development practices and enhance the functionality of web forms.

(2) Designing a user registration form for a new web application. the form needs to capture essential information such as name, email..

HTML form:

```
<html>
<head>
<title> registration form </title>
</head>
<body>
```

```
<form id = "registration form"
<h2> user regi form </h2>
<dir class = " form group">
<label for = " name> Name : </label>
<input type = "text" id = "name" reduired>
<span class = " error message" id = "nameerror"
                                    </span>
</dir>
<dir class = "form group">
<label for = " email " > Email : </label>
<input types = "email " id = "email" name = "email"
  reduired>
<span class = " error message"  id = "email </span>
</dir>
</ body >
</html >
```

Javascript validation :

```
const form = document . get ElementById ("reg form")
form . acklevent listener ('submit', le) => { e . prevent Default
const name = document . get ElementById ( home)
const email = document . getElementById ('email'),
const message = document . getElementById ('message')
const nameerror = document . get ElementById
                                ('nameerror')
const emailerror = document . get ElementById
                                ('emailerror')
 if(name , value . trim ( ) = = = ' ') {
name . error . text content = 'enter name' ;
```

```
nameerroy.textcontent='enter name';
nameerror.style.display='block',
}else{
nameerror.style.display:'none',
}
if(email.value.trim()===''){
emailerror.style=display===='none'
console.log('form submitted");
}
});
function validateEmail(email)
{
const emailReg=/^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]
+/[a-zA-Z]{2,}$/;
return emailreg.text(email);
```

Output: Registration form

Name: _____

Email: _____

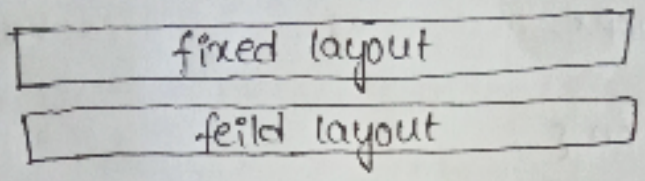
message: _____

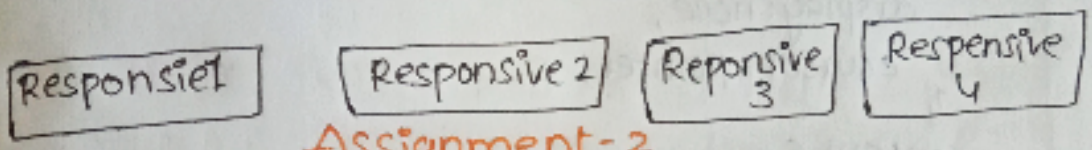
Submit

3. Designing a web page that offers the Function effecting across various devices and screen size. the design include diff types of layouts.

```
<html>
<head>
<title> Responsive layout </title>
<style>
fixed containe {
width: 800 px; margin: 0 auto; background
colour, # fufufu;
```

```
}
feild - container {
 width = 100 %. ; Padding :20 px;
 background = Colour # e2e2e2;
 }
</body>
</ html>
```





Assignment-2

4. To create a web page that refers a seamless user experience across device, respective design principles must be applied.

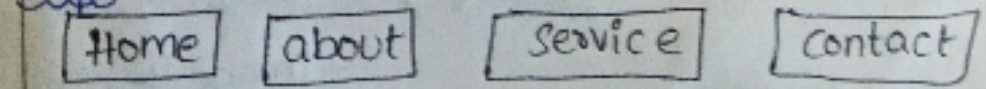
```
<html>
<head>
<title> Responsive webpage </titel>
<style>
<body>{
  font-family: Arial, sans-serit;
  margin 0; padding 0; box-sizing;
  border-box;
}
header {
 backgrount-color : #333;
color:while; padding: 10px ; text-design:
 center;
}
nav {
 display :flex;
```

```
Justify-content: space-around;
background-color: #444;
Padding: 10px;
}
nav a {
color: white; text-decoration: none;
Padding: 10px;
}
menu-icon {
  display: none;
  cursor: pointer;
}
content {
 display: grid;
 grid template-columns: repeat (3, 1fr);
 gap: 10px;
 padding: 20 px;
}
Content div {
   background-color: #fufufu;
   Padding: 20px;
   }.
   @media (max-width: 768px) {
     content {
     grid-template-columns: 1fr;
   }
   nav.archive a {
    display: block;
    width: 100%;
    padding: 10 px;
   }
```

```
</style>
</head>
<body>
  <header>
  <h1> Responsive webpage </h1>
  </header>
  <nav>
  <dir class = "menu-icon" onclick = "toggle
        menu()" > ≡ minu </dir>
<a href = "#home" > Home</a>
<a href = "#about" > about </a>
<a href = "#services" > services </a>
</nav>
<div class = "content" >
 <div>
  <h2> Coloumn</h2>
  <p> content-goes here. </p>
 </div>
 <script>
  function togglemenu () {
  const nav: document.query selector ('nav');
  nav.classlist toggle ('active');
  }
  </script>
  </body>
  </html>
```

Responsive webpage

Output:



Column1
Content goes here.

5. Given a scenario (eg: blog post) design a webpage using appropriate eliminate, tables etc.,

```
<html>
<head>
<title> my Blog Post </title>
< line rel = stylished ` href = " styles . css" >
</head>
<body>
<header>
<h1> my awesome Blog </h1>
<nav>
<ul>
<li> <a href = " index - html " >Home</a> </li>
<li><a href = " about . html "> about </a> </li>
<li> <a href = " blog . html "> Blog </a></li>
<li> <a href = " contact . html " > contact </a> </li>
</ul>
</nav>
</header>
<article.
<h2> the title of my Blog post </h2>
<p> published on < time datetime = " 2024- 08 - 27 ,
Aug 27, 2024 </time> by <a herf = " about. html ">
Author name </a> </p>
<img src . " blog . image . jpg " alt = " A discribes
image related to the blog post " class = " feathred
- image ">
<p> this is the inhoduction to my blog post
```

It's engaging and gives a brief overview of what the post is about.

<h3> subheading </h3>

 Here is some detailed content about the first topic for the blog post.

<h3> subheading 2 </h3>

 more detailist content.

<table>

<caption > compaision table </caption>

<thread>

<tr>

<th> Feature </th>

<th> option A </th>

<th> option B </th>

</tr>

</thread>

</body>

<tr>

<td> price </td>

<td> $100 </td>

<td> $150 </td>

</tr>

</head>

</body>

<tr>

<td> performance </td>

<td> Good </td>

<td> Excellent </td>

</tr>

<tr>

```
    <td> support </td>
    <td> 24/7 </td>
    <td> Business Hours </td>
  </tr>
 </body>
</table>
 <h3> subheading 3 </h3>
 <p> this section might includes a list </p>.
 <ul>
  <li> key point one. </li>
  <li> key point two. </li>
  <li> key point three </li>
 </ul>
<p> condusion: summarize the blog pot
  </p>
</article>
  <footer>
  <p> &copy; 2024 my Awesome Blog. </p>
  </footer>
  <script src=" script.js" > </script>
  </body>
  </html>
```

Output:

My Awesome blog

- Home
- About
- Blog
- contact

The title of my blog post.

published on Aug 27, 2024 by Author Name

a discriptive image related to the blog plot

this is the introduction to my blog post.

Subheading1 :

- Here is some detailed content

Subheading 2:

More detailed about content.

Comparison table :

Feature	option A	option B
Price	$100	$150
Perfomance	Good	excellent
Support	24/7	Businesshours

Subheading 3:

- Keypoint 1
- Keypoint 2
- Keypoint 3

conclusion : Summarize the blog post

Rubics	Split up	Marks obtained	Total Marks
1. From Design But			
Button Design			
features of HTML			
(2) form design			
validation			
lay out look			
(3) lay out design			
css background			
(4) form design			
layout design			
navigation			
(5) form design			
css			
layout design			
Java script			

ASSIGNMENT-3

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

```
import
javax.servlet.ServletException
import
javax.servlet.http.Http servlet;
import
javax.servlet.http.HttpservletRequest.
import
java.Servlet.http.Http servletResponse;
import
java.servlet.http.HTTP session;
import java.io.Ioexception;
import java.io.printwrites;
Public class track.session servlet extends
    Http servlet {
      @override
      protected void
doget (Http servlet Request request, Http
 Servlet Response response)
throws servletException
  Io exception{
response.set content type(" text/html");
Httpsession session = request.get.session(true);
Integer accesscount = (Integer)
```

```
Session.getAttribute("accesscount")
if(access count == null){
    accesscount = 0;
}
accesscount++;
String sessionId = session.getId();
long creationTime = long lastCreationTime();
Session.getAccessedTime();
Printwriter out = response.getwriter();
out.println("<html><body>");
out.println("<h2>session Tracking </h2>
I," +session Id +" </P2" );
Time :"+new Acess :" + accesscount +"</P>");
out.println("</body></html>");
}
}
```

Output

<P> session ID : 1234567890abc </P>

<P> creation Time: Mon Sep 10 15:120:30 2024 </P>

<P> last accessed time: mon sep 10 15:25:10 2024 </P>

<P> Number of Access : 5 </P>

2) write a scenario where you had to use <ST< to slove a complex problem and how you went about it. How to create custom functions

Scenario using JSL <:

In a web application, you have a requirement do display a list of products with varying the catagories.

The product list needs to be dynamically catoga -rised into saperate session on a web page.

Solution using JST< :

```
import javax.servlet.ServlexException;
import javax.servlet.annotation.webservlet;
import javax.servlet.http.HTTP servlet;
import javax.servlet.Request;
import java.util.<list;
import java.util.map;
import java.util.Hashmap;
Public class
Product <ist servlet extends HTTP servlet {
    protected void
do get (HttpServeletRequest request, HTTP servelet)
Product ("laptop","Elections",1000, true),
        new
Product ("shirt","clothing",30 true),
        new
Product ("washing machine","Home appliance",500);
        false
Product <list.jsp").forward request, response);
}
```

Output:

<strong <aptop < /strong >- $ 1000 - Available

<clothing > shirt - $30 - Available

<Home appliance > washing machine - $500 -
Out of stock.

(3) A page of stock market quotes uses Screpto refresh the page every five minutes in order to ensure the largest statistics remain available.

TO achieve this setup the HTML page add Javascript for Automatic Refresh and confirm Dialog.

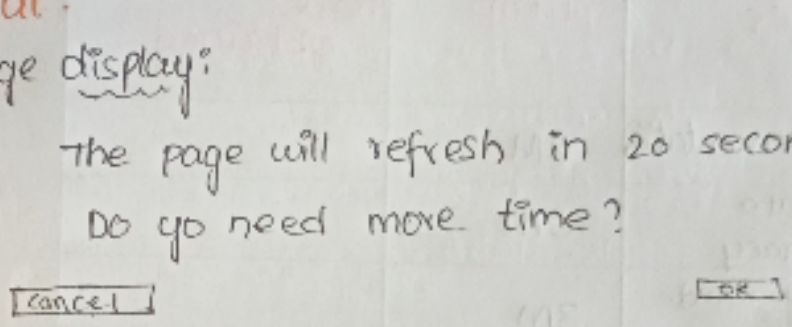
HTML and Javascript Code:

```
<!DOCTYPE html>
<html lang= "en">
<head>
    <meta Charset =" UTF-8">
    <meta name = "viewport"
content = " width " = device-width , initial -scale=1.0">
<title > stock market Quotes > /< title > <screpto>
    Var user Response = confirm The page will
refresh in 20 seconds · Do you need more time?);
    if(!user Response ){
      window · location · reload();
    }else{
      settimeout (refresh page , 30000);
    }
}, 20000);
```

```
window.location.reload();
    3, refresh Interval);
<h1> stock market quotes </h1>
<p> Here you can display ref-time
stock market quotes </p>
</body>
</html>
```

Output:

Page display:



(4) you are developing an e-commerce application that needs to integrate with an external payment gateway service. Descrime the steps involved in generating the client code.

steps to integrate a payment Gateway using WSDL:

For Java:

```
import java.package.name.payment service;
import .Package.name.payment service port type;
Public class payment client {
    Public static void main (string[] args){
Paymentservice service = new payment
                        service();
String response = port.process payment
    ("amout", "currency", "payment Details");
```

```
        System.out.println("Responce:" + responce);
    }
}
```

Output:

Payment response : payment Sucessful for amount 100.00 usp

	Rubrics	Splitup	marks obtained	total Marks
(1)	Code implementation	8M		
	Session Data accuracy	5M		
	Efficiency and clarity	3M		
	Explanation	4M		
(2)	Scenario Explanation	6M		
	Function library Explanation	5M		
	custom Functions	5M		
	Clarity and organisation	4M		
(3)	Script Functionality	8M		
	User Interaction Design	5M		
	Code effeciency	4M		
	Explanation	3M		
(4)	understanding of WSDL	6M		
	client code Generation	6M		
	Error Handling	4M		
	Clarity and depth	4M		

Assignment - 4

1. From a developer's presentive, discuss why JDBC is essential in building database-driven applications. Provide examples of executing SQL queries.

```
<context>
    <Resource name = "jdbc / myDataSource"
auth = "container"
type = " javax.sql.DataSource "
            maxTotal = "20"
            maxIdle = "10"
maxWaitMillis = "1000"
 username = " dbuser"
 password = " dbpassword "
driverClassName = "com.mysql.cj.jdbc.Driver"
url = "jdbc:mysql://localhost
 />
  </content>
```

Callable Statement:

```
import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.SQLException;
import java.sql.Types;
Public class CallabStatement Example {
public void CallStored procedure (int employeeId){
    String sql = " {call getEmployeeName
                (?,?)};
try (Connection conn = DataBase utility.get
            Connection();
employee.setName = cstmt.getstring(2);
```

```
        System.out.println("Employee Name:" + employeeName);
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
}
```

Output:

```
Employee ID: 1, Name: John
Employee ID: 2, Name: Jane
Employee ID: 3, Name: Emily
Rows updated: 1
Employee Name: John
```

(2) Describe the life cycle of phases of JSP Page Explain the significance of each phase in the JSP execution.

Life cycle of JSP Page:

1. Translation page:

Description: The JSP page is translate int a java servlet by the JSP engine (e.g: HTML mixed with JSP tags)

Significance: This phase ensure that the JSP content is converted into a form that the Java Servelet containers can execute.

Compilation phase:

Description: The java source code generated from the translation phase.

Significance: compilation ensures that the jsp is converted into executable

3. **Initilization page:**

The servlet container initializes the servelet instance.

Initilization sets up any resource the Jsp might need such as db.

4. **Requesting processing phase:**

The servelet process incoming client requests by calling the service();

This phase is where the dynamic content generation occurs.

5. **Destroy phase:**

The servelet contains destroy the servelet instance the destroy()

This phase ensures that resource are properly displayed

(3) You need to develop a PHP program that generates chessboard that generates table should have a total width of this program.

PHP code:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="view port"
Content="width=device-width, intial-scale=1.0
<title>chessboard</title>
</head>
<body>
```

```
table {
    border : collapse : collapse;
    width : 400 px;
    height : 400px;
}
td {
    width : 300px;
    height : 30 px;
for ($ row = 0 ; $row < 8 ; $row++) {
        echo " <tr> "
    for ($ col = 0 ; $ col < 2 ; $ col++) {
        echo " </tr> ";
    }
}>
</table>
</body>
</html>
```

* w - white
* B - Black.

Output:

[w][B] [w][B] [w] [B]
[B] [w] [B] [w] [B] [w]
[w] [B] [w] [B] [w] [B]
[B] [w] [B] [w] [B] [w]
[w] [B] [w] [B] [w] [B]
[B] [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 specific such as steps provided code.

PHP Code:

```
<? PHP
$ textfilepath = " input . txt ";
$ xmlfilepath = " Output-xml ";
$ textcontent = file- get- contents ( $ textfile path)
    0-9 . - ] + ) . [a-z A-Z] {2,4} /i;
$ phone pattren = ' /\b\d{10}\b/ ';
preg-match -all ($ email pattern , $ text content
                                    $ phones),
```

```
preg_match_all ($phone_pattern, $text_content,
                $phones);

$xml = new
simpleXMLElement ('<Data>');
$emails Element = $xml -> addChild ('Emails');
simpleXMLElements ('<Data1>');
$phonesElements = $xml -> addchild ('Phone
                  Numbers');
echo "Data extracted and saved to xml file
sucessfully.";
?>
```

Output:

Email : support @example.com

phone number : 1234567890

	Rubrics	marks split up	marks obtained	Total marks
(1)	explanation of JDBC	5M		
	connection pooling	6M		
	SQL Queries	5M		
	Statement types	4M		
(2)	binding Javacode	5M		
	lifecycle phases explanation	6M		
	Advantages & Disadvantages	5M		
	Clarity and Depth	4M		
3	code implementation	3M		
	HTML Table structure	5M		
	Alternatic colours logic	4M		
	Explanation	3M		
4.	Code Implementation	7M		
	pattern extraction	5M		
	XML file generation	4M		
	DTD vs XML schema comparision	3M		