	Rubics	Split up	Marks obtained	Total marks
1.	Form Design Button Design Features of HTML			
2.	Form Design validation, layout look and field			
3.	Layout Design, CSS background			
4.	Form Design, layout design, Navigation Menu			
5.	Form Design, CSS layout Javascript evaluation.			

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

CO1	2
CO1	3
CO1	2
CO1	3
CO1	3

27/8

Assignment - 1

CSM4399 - Internet Programming for client Server Computing

1. You are updating a legacy web application to take advantage of modern web standards. The projects involves transitioning from an earlier HTML version to HTML 5.

Solution :-

Here are some key HTML5 introduces new semantic elements that provide better structure to web pages, made more enhance web form functionality.

1. **Semantic elements :-**
to provide better structure to web pages, making them more accessible and easier to maintain.
example :- <header>, <nav>, <main>, <section>, <article>, <aside>, <footer>, etc

2. **Form Validation :-** HTML5 introduces built in form validation, which allows developers to define validation rules using attributes like required, pattern, min, max, etc...

3. **New Input Types :-** It introduces new input types like data", time, datetime, datetime-local, month, week, number, range, search, tel, url and email.

4. **Placeholder Attributes :-**
The autofocus placeholder attribute allows developers to provide a hint or example value for form fields, making it easiers for users to understand.

5. **Autofocus attributes :-**
The autofocus attribute allows developers to specify which form field should receive focus when the page loads, improving accessibility and user experience.

6. Label element :-
<label> element which allows developers to associate a label with a form field, improving accessibility and user experience

7. Fieldset and Legend element :-
<fieldset> and <legend> elements allow developers to group related form fields together, making it easier for users to understand the form structure.

8. HTML5 Validation API :- It provides a set of methods and properties that allow developers to validate form fields programmatically, making it easier to implement custom validation logic.

9. CSS3 Selectors :- It can be used in conjunction with CSS3 selectors to style form elements, providing a more visually appealing and consistent user experience.

10. Accessibility features :- It includes several accessibility features, such as ARIA attributes that make web forms more accessible to users with disabilities.

By following these steps and leveraging HTML5's 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 essen

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.

Solution:-

HTML5 form:-

```
<html>
<head>
  <title> registration form </title>
</head>
<body>
  <form id = "registration-form">
  <h2> User registration form </h2>
  <div class = "form group" >
    <label for = " name " > Name : </label>
    <input type = " text " id = "name" name =
" name " required >
  <span class = " error message " id = " name error "
                </span>
</div>
<div class = " form group " >
  <label for = "email " > Email : </label>
  <input type = " email " id = " email " name =
"email " required >
<span class = " error message " id = "email
 error " > </span> .
</div>
</form>
</body>
</html>
```

Javascript Validation :-

```
Const form = document.getElementById
        ('registration-form');
form.addEventListener ('submit', (e) => {
  e.preventDefault();
Const name = document.getElementById('nam
Const email = document.getElementById('email'
Const message = document.getElementById(
                  message');
Const nameerror = document.getElementById
        ('namerror');
Const emailerror = document.getElementById
        ('emailerror');
if (name.value.trim() === '') {
  nameerror.textcontent = 'Enter name';
  nameerror.style.display = 'block';
} else {
  nameError.style.display = 'none';
}
if(email.value.trim() === '') {
  emailerror.textContent = 'enter email';
  emailerror.style.display = 'block';
} else {
  emailError.style.display = 'none';
}
if (nameError.style.display === 'none' &&
emailError.style.display === 'none') {
  console.log ('Form submitted");
}
});
```

```
function validateEmail(email)
{
const emailRegex = /^[a-zA-Z0-9._%+-]+@[a-z
A-Z0-9.-]+\.[a-zA-Z]{2,}$/;
return emailRegex.test(email);
```

OUTPUT:-

Registration Form

Name : [ ]
Email : [ ]
Message : [ ]
Submit

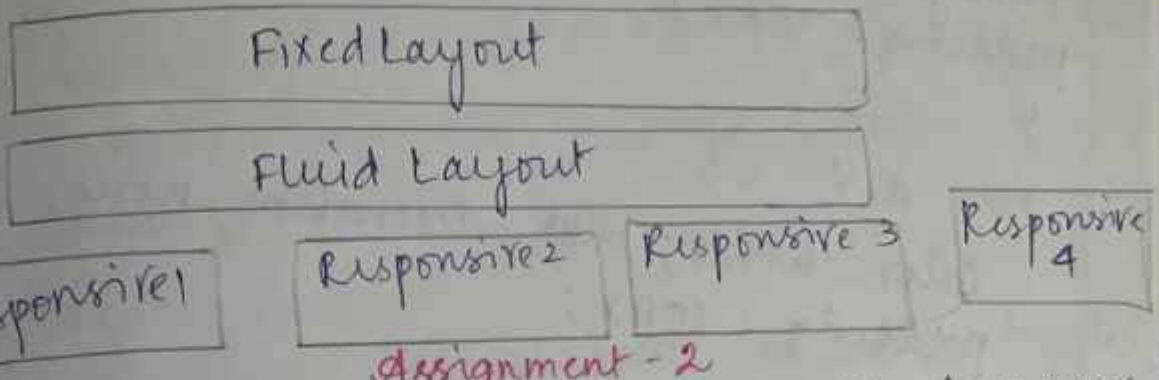
3). Designing a web page that offers the function effectively across various devices and screen size. The design should include different types of layouts such as fixed, fluid and responsive.

Solution:-

```
<html>
<head>
    <title> Responsive Layout </title>
<style>
  .fixed container {
  width : 800px; margin : 0 auto; background
- color : #f4f4f4;
}
.fluid-container {
width : 100%; padding: 20px;
background-color : #e2e2e2;
}
```

```
.responsive-grid {
  display: grid;
  grid-template-columns: repeat(auto-fit,
minmax(200px, 1fr));
  gap: 10px;
  background-color: #ddd;
  padding: 10px;
}
.item {
  background-color: #4CAF50;
  padding: 20px;
  text-align: center;
  color: white;
}
</style>
</head>
<body>
  <div class="fixed container">
  <div class="item">Fixed Layout</div>
</div>
<div class="fluid container">
  <div class="item">Fluid Layout</div>
</div>
<div class="responsive-grid">
  <div class="item">Responsive 1</div>
  <div class="item">Responsive 2</div>
  <div class="item">Responsive 3</div>
  <div class="item">Responsive 4</div>
```

</body>
</html> O/P



Assignment - 2

1). To create a webpage that offers a seamless user experience across devices, responsive design principles must be applied. Additionally, touch friendly elements and optimized images ensure fast loading times and a smooth experience on all devices.

Solution :-

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

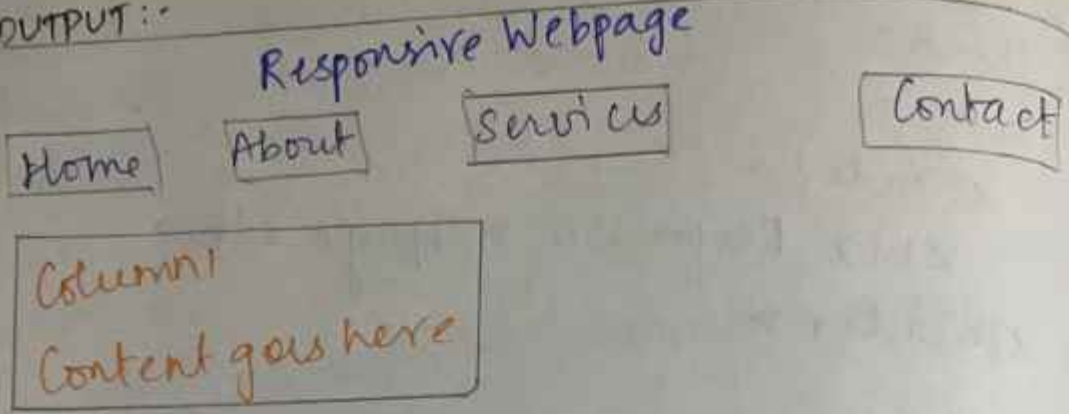
```
justify-content: space-around;
background-color: #444;
padding: 10px;
}
nav a {
  color: white; text-decoration: none;
  padding: 10px;
}
.menu-icon {
  display: none;
  cursor: pointer;
}
.content {
  display: grid;
  grid template-columns: repeat(3, 1fr);
  gap: 10px;
  padding: 20px;
}
.content div {
  background-color: #f4f4f4;
  padding: 20px;
}
@media (max-width: 768px) {
  .content {
    grid-template-columns: 1fr;
  }
nav.active a {
  display: block;
  width: 100%;
  padding: 10px;
}
}
```

```
</style>
</head>
<body>
  <header>
    <h1> Responsive Webpage </h1>
  </header>

<nav>
  <div class = "menu-icon" onclick = "toggle
      Menu()"> ☰ Menu </div>
<a href = "#home"> Home </a>
<a href = "#about"> About </a>
<a href = "#services"> Services </a>
<a href = "#contact"> Contact </a>

</nav>
<div class = "content">
  <div>
    <h2> Column 1 </h2>
    <p> content goes here. </p>
  </div>
<script>
  function toggleMenu() {
  const nav = document.querySelector('nav');
  nav.classList.toggle('active');
}
</script>
</body>
</html>
```

OUTPUT :-



5). Given a scenario (eg: blog post). design a webpage using appropriate elements, tables, lists and images for optimal readability and user experience. (HTTP, CSS, images etc.)

Solution :-

html :-

```
<html>
<head>
  <title> My Blog Post </title>
  <link rel = "stylesheet" 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>
 Published on <time datetime="2024-08-27">
Aug 27, 2024 </time> by
Author Name
<img src="blog-image.jpg" alt="A describes
image related to the blog post" class="featured
-image">
 This is the introduction to my blog post.
It's engaging and gives a brief overview of
what the post is about.

<h3> Subheading 1 </h3>
 Here is some detailed content about the
first topic of the blog post.
<h3> Subheading 2 </h3>
 More detailed content.

<table>
<caption> Comparison Table </caption>
<thead>
<tr>
<th> Feature </th>
<th> Option A </th>
<th> Option B </th>
</tr>
</thead>
<tbody>
<tr>
<td>Price</td>
<td> $100 </td>
<td> $150</td>
</tr>
</thead>
<tbody>

```
<tr>
  <td> Performance </td>
  <td> Good </td>
  <td> Excellent </td>
</tr>
<tr>
  <td> Support </td>
  <td> 24/7 </td>
  <td> Business Hours </td>
</tr>
</tbody>
</table>
<h3> Subheading 3 </h3>
<p> This section might include a list. </p>
<ul>
  <li> key point one. </li>
  <li> key point two. </li>
  <li> key point three. </li>
</ul>
<p> Conclusion : Summarize the blog po
</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 post

This is the introduction to my blog post.

Subheading 1

Here is some detailed content

Subheading 2

More detailed about content.

Comparison table

Feature	Option A	Option B
Price	$100	$150
Performance	Good	excellent
Support	24/7	Business Hours.

Subheading 3

- key point one.
- key point two.
- key point three.

Conclusion : Summarize the blog post.

? 7

OUTPUT ::

Email : support@example.com

phone Numbers : 1234567891

ASSIGNMENT - 3

	Rubrics	split up	marks obtained	Total marks	Sign
1.	Code implementation	8M			
	session data accuracy	5m			
	Efficiency & clarity	3m			
	Explanation	4m			
2.	Scenario explanation	6m			
	function library	5m			
	custom functions	5m			
	clarity & org	4m			
3.	script function	8m			
	user Interaction	5m			
	Code efficiency	4m			
4.	• Explanation	3m			
4.	client code generation	6m			
	error handling	4m			
	understanding of wspl	6m			
	Clarity and depth	4m			

ASSIGNMENT - 3

1. Implementing a feature in a web application that tracks the number of access by a client within a single session.

```
import
javax.servlet servlet exception;
import
javax.servlet.annotation.webservlet;
import
javax.servlet.http.Httpservlet;
import
javax.servlet.http.HttpservletResponse;
import
java.servlet.http.Http servelt Request;
import
java.servlet.http.HttpSession;
import java.io.Ioexception;
import java.io.printwriter;
public class Tracksession Servlet extends
                Http Servlet {
    @override
    protected void
doget (HttpservletRequest request, Http servletResponse
                                    response)
throws servletexception,
IoException {
response.setcontenttype("text/html");
Httpsession session = request.getsession(true);
Integer accesscount = (Integer)
session.get Attribute ("access Count");
if (accesscount == null) {
    accesscount = 0;
}
accesscount++;
```

```
String sessionId = session.getId();
long creationTime = long lastcreationtime();
session.getAccessedTime();
PrintWriter out = response.getWriter();
out.println("<html><body>");
ID = " + sessionId + " </p> );
Time : "+new
Access : " + accesscount + " </p>" );
out.println("</body></html>");
}
}
```

OUTPUT :-

```
<p> session ID : 123456 ab </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 Access : 5 </p>
```

2). Write a scenario where you had to use JSTL to solve a complex problem and how you went about it.

Scenario using JSTL :-

```
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.io.IoException;
import java.util.List;
import java.util.Map;
import java.util.HashMap;
public class
productlistservlet extends HttpServlet {
```

```
protected void
doget (HttpServlet Request request. Http servlet)
product ("Laptop", "Electronics", 100, true),
new product ("shirt", "clothing", 30, true),
new product ("Washing Machine", "Home
    Appliances", 500) false
);
product list.jsp").forward(request, response);
}
}
```

OUTPUT :-

<Strong> Laptop </Strong> -$1000 - Available

<clothing> shirt - $30 - Available

<Home Appliances> Washing machine - $500 - out of Stock.

3. A page of stock market quotes uses script to refresh the page every 5 min in order to ensure the latest statistics remain available.

```
<html>
<head>
<title> Stock Market </title>
<script>
    var user response = confirm("The page
will refresh in 30 sec");
if(!user response) {
  window.location.reload();
} else {
  setTimeout(refresh page, 30000);
}
```

window.location.reload();
}, refreshInterval);
</h1> Stock Market Quotes </h1>
 Here you can display real-time
Stock market quote s
</body>
</html>

OUTPUT :-

Page Display

The page will refresh in 20 sec

Cancel [OK]

4. You are developing an e-commerce app that needs to integrate with an external payment gateway service.

```
import your.package.name.paymentService;
import.package.name.paymentServicePortTyp
public class paymentClient {
public static void main (String[] args) {
  paymentService = new paymentService();
String response = port.processPayment("amou
        currency", "paymentdetails");
System.out.println("Response :" + response)
}
}
```

OUTPUT :-

Payment Response : Payment Successfully for 100.000 USD.

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

```
<Context>
    <Resource name = "jdbc/myDataSource"
    auth = "Container"
type = "javax.sql.DataSource" maxTotal = "20"
        maxIdle = "10"
maxWaitMillis = "1000"
username = "dbuser"
password = "dbpwd"
driverClassName = "com.mysql.cj.jdbc.Driver"
url = "jdbc:mysql://localhost:3306/mydb".
/>
</Context>
```

Callable Statement

```
import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.SQLException;
import java.sql.Types;
public class CallableStatementExample {
public void CallStoreProcedure (int employeeId)
 String sql = "{ Call getEmployeeName
                    (?)}";
try Connection conn = DatabaseUtility.get
                    Connection();
employeeName = cstmt.getString(2);
System.out.println("employeeName: " +
                    employeeName);
} catch (SQLException) {
    e.printStackTrace();
```

}
}
}

OUTPUT :-

employeeID : 1, Name: John
employeeID : 2, Name: Jane
employeeID = 3, Name : Emdy.

2. Describe the lifecycle of phases of JSP page explain the significance of each phase in the JSP execution process.

i) Translation phase :-
The JSP page is translated in a Java servlet by the JSP engine (eg :- HTML mixed with JSP tags)
significance :- this phase ensures that the JSP content is converted to form that the java servlet container can execute.

ii) Compilation phase :- The java source code generated from the translation phase.
significance :- Compilation ensures that the JSP is converted into executable.

iii) Initialization phase :- The servlet container initializes the servlet instance
significance :- Initialization sets up any resources the JSP might read such as DB.

iv) Processing phase :- The servlet process incoming client requests by calling the service().
significance :- This phase is where the dynamic content generation occurs.

v) Destroy phase :- The servlet process in 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 :-

```
<html>
<head>
<title> chessboard </title>
<style>
table {
    border-collapse : collapse;
    width : 400px;
    height : 400px;
}
td {
    width : 30pxs ;
    height : 30px;
}
for ( $row = 0 ; $row < 8 ; $row++)
{
    echo " <tr> " ;
for ( $col = 0 ; $col < 8 ; $col++)
{
    echo " </tr> " ;
}
?>
</table>
```

</body>
</html>

OUTPUT :-

[W][B][W][B][W][B][W][B]
[B][W][B][W][B][W][B][W]
[W][B][W][B][W][B][W][B]
[B][W][B][W][B][W][B][W].

* W represents a white cells
* B represents a black cells.

4. You are developing a php applications that make content from a text file and uses regular expressions to provide code for the applications.

php code :-

```
<?php
$textfilepath = 'input.txt';
$xmlfilepath = 'output.txt';
$textcontent = file_get_contents($
        textFilepath);
$email pattern = '/ (a-z A-Z 0-9._-]
        -)+@[a-z a-z 0-9.-]+[
    (a-z A-Z] {2,}$/i';
$phone pattern = '/\b\d{10}\b/';
preg_match_all ($email pattern,
        $textcontent, $email);
Preg_match_all ($phone pattern, $text
        content, $phones);
```

```
$xml = new SimpleXMLElement('<Data/>');
$emailElement = $xml->addChild('Emails');
SimpleXMLElement('<Data/>');
$phonesElement = $xml->addChild('
                    phoneNumbers');
echo "Data extracted and saved to
  xml file successfully";
?>
```

OUTPUT :-

Email : support@example.com

phoneNumbers : 123456971

ASSIGNMENT - 3

	Rubrics	split up	marks obtained	Total marks	sign
1.	Code implementation	8M			
	session data accuracy	5m			
	Efficiency & clarity	3m			
	Explanation	4m			
2.	Scenario explanation	6m			
	Function library	5m			
	custom functions	5m			
	clarity & org	4m			
3.	script Function	8m			
	user Interaction	5m			
	Code efficiency	4m			
4.	• Explanation	3m			
4.	client code generation	6m			
	error handling	4m			
	understanding of WSDL	6m			
	clarity and depth	4m			

ASSIGNMENT - 4

	Rubrics	split up	marks obtained	Total marks	Sign
1.	explanation of JDBC	5m			
	connection polling	6m			
	SQL queries	5m			
	Statement types	4m			
2.	Javacode	5m			
	lifecycle of JSP	6m			
	Ad & dis	5m			
	Clarity & depth	4m			
3.	Code implementation	8m			
	HTML table st	5m			
	Alternate color	4m			
	explanation	3m			
4.	Code Implementation	8m			
	Patter extraction	5m			
	xml file generation	4m			
	DTP vs xml schema	3m			