S.NO	Assignment Rubies	split up	Total Marks
1)	form design	4	
	Button design	4	
	features of HTML	2	
2)	form design	4	
	validation	4	
	lay out look	2	
3)	lay out design	5	
	CSS background	5	
4)	form / design	4	
	lay out design	4	
	navigation menu	2	
5)	form design	4	
	CSS	2	
	lay out design	2	

Assignment-1

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 new version improve web development practices and enhance the functionality of web forms.

HTML:-

```
<!doctype html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>legacy web Application</title>
<meta http-equiv="content-type" content="text/html
                                charset=utf-8" />
<style type="text/css">
  .header { background-colour: #f8f8; padding: 10px; }
.content { padding: 10px; }
.footer { background-colour: #f8f8f8; padding: 10px;
                                text-align: center; }
</style>
</head>
<body>
<div class="header">
<h1> welcome to our website</h1>
<div>
</div>
<div class="content">
<form action="/submit" method="post">
<label> Name: </label>
<input type="text" name="name" /><br />
<label> email: </label>
```

```
</form>
</div>
<div class="footer">
&copy; 2024 legacy web Application
</div>
</body>
</html>
```

HTML5

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device
    -width, initial-scale=1.0">
<title> modern web Application </title>
<style>
header, footer {
 background-color: #f8f8f8;
 padding: 10px;
}
main {
 padding: 10px; }
</style>
</head>
<body>
<header>
<h1> welcome to our website </h1>
<nav>
<ul>
<li><a href="#home"> Home </a></li>
<li><a href="#about"> About </a></li>
</ul>
</nav>
```

<main>
<form action="/submit" method="post">
<label for="name">Name:</label>
<input type="text" id="name" name="name" req>
<label for="birthdate">Birthdate:</label>
<input type="date" id="birthdate" name="birthdate">
<input type="submit" value="submit">
</main>
<footer>
</footer>
</body>
</html>

output:-

welcome to our website

Name	birthday date
Janu	03-08-23
anu	07-07-25

2. Designing a user registration from for a new web application the form needs to capture essential information such as name email, and a message form users. It is crucial to implement client side validation to ensure that the layout adapts well to complete before from the submit;

```
<!Doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width
initial-scale=1.0">
<title>User Registration</title>
<style>
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 20px;
    background-color: #f4f4f4;
```

```
padding : 20px;
border-radius : 5px;
max-width : 400px;
margin : 0 auto;
box-shadow : 0 0 10px rgba (0,0,0,0.1);
}
label {
    display : block;
  margin bottom : 5px;
  font-weight : bold;
}
input, textarea {
        width : 100%;
        padding : 10px;
        margin-bottom : 15px;
        border : 1px solid #ccc;
        border-radius : 4px;
        font-size : 16px;
        }
        input [type="submit"] {
        background-color : #4CAF50;
         color : white;
         border : none;
         cursor : pointer
         transition : background-color 0.
        }
</style>
```

```
<form action="/submit" method="post" novalidate>
<h2> User Registration</h2>
<label for="name"> Name: </label>
<input type="email" id="email" name="email" required
 placeholder="example@example.com">
<label for="message"> Message: </label>
<textarea id="message" name="Message" required
 minlength="10" maxlength="500"
 placeholder="Enter your message here..">
<input type="submit" value="Register">
</form>
<script>
</body>
</html>
```

Output:-

User Registration		
Name	Email id	Message
Josh	dash75@gmail.com	Registered now

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 use CSS techiques like flexbox or Grid ensure that the layout adapts well to diffe screen sizes and maintain a consistent use experience.

HTML:-

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

```
</head>
<body>
<header class="header">
<h1> Responsive website</h1>
</header>
<nav class="nav">
<ul>
<li><a href="#">Home</a></li>
<li><a href="#">About</a></li>
<li><a href="#">services</a></li>
<li><a href="#">contact</a></li>
</ul>
</nav>
<main class="container">
<section class="content">
<h2> content Area</h2>
<p> This is the main content area. It can either fixed
fluid or responsive depending on the design require
</p>
</section>
<aside class="sidebar">
<h2> sidebar</h2>
<p> This is a sidebar that complements the ma
content.</p>
</aside>
</main>
<footer class="footer">
<p> &copy; 2024 your website</p>
</footer>
</body>
</html>
```

CSS! —

```
body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}
header, nav, main, footer {
        padding: 20px;
        background-color: #f4f4f4;
        margin-bottom: 20px;
  }
 header {
   background-color: #333;
    color: white;
    text-align: center;
 }
 nav ul {
   list-style-type: none;
    padding: 0;
    margin: 0;
    display: flex;
    justify-content: space-around;
    background-color: #444;
}
 .container-fixed {
    width: 1200px;
    margin: 0 auto;
 }
footer {
  background-color: #333;
   color: white;
```

flex-direction: column;

}

@media (min-width: 768px)

{

.nav ul {

flex-direction: row;

}

}

output:-

Responsive web design
This is the main content area. It can either fixed fluid or responsive depending on the design requirement

Assignment - 2

4. To create a webpage that offers a seamless user experience across devices, responsive design priciples must be applied this involves using fluid grids, flexible images, and media queries to ensure the layout adjusts gracefully to different screen size. for instance, on a desktop, the webpage Navigation menus should transform into a hamburger icon on smaller screens, ensuring they remain:

HTML:-

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

```
</head>
<body>
<header class="header">
<h1> Responsive website </h1>
</header>
<nav class="nav">
<div class="menu-icon" onclick="toggleMenu()">
<ul id="menu">
<li><a href='#'> Home </a></li>
<li><a href='#'> About </a></li>
<li><a href='#'> Services </a></li>
<li><a href="#"> Contact </a></li>
</ul>
</nav>
<main class="container">
<section class="content">
<h2> Content Area </h2>
<p> This is the main content area. It adjusts based
on screen size for an optimal experience. </p>
</section>
<aside class="sidebar">
<h2> Sidebar </h2>
<p> This is a sidebar that complements the main conte
</aside>
</main>
<footer>
<script>
function toggleMenu() {
if (menu.style.display === 'b')
} else {
    menu.style.display = 'block';
}
}
</script>
</body>
```

CSS:-

```
body {
    font-family: Arial, sans-serif;
        margin: 0;
        padding: 0;
        box-sizing: border-box;
}
header, nav, main, foober {
        padding: 20px;
        background-color: #f4f4f4;
        margin-bottom: 20px;
    }
    header {
        background-color: #333
        color: white;
        text-align: center;
    }
    .nav {
        background-color: #444;
        position: relative;
    }
foober {
    background-color: #333;
    color: white;
    text-align: center;
    }
@media (max-width: 768px) {
    container {
        flex-direct
```

```
nav ul {
flex-direction: column;
  display: none;
}
menu-icon{
  display: block;
}
.nav ul {
    display: none;
    position: absolute;
    top: 50px;
    right: 0;
    background-color: #444;
    width: 100%;
    }
   .nav ul li {
     text-align: center;
      padding: 10px 0;
    }
    .nav ul li a {
    padding: 10px;
  }
  }
```

output :-

Responsive web design

This is the main content area. It adjusts based on screen size for an optimal experience

5. Given a scenario (e.g. creating a blog po
a product listing), design a webpage using
appropriate elements, tables, lists and image
for optimal readability and user experience
Also describe step by step the sequence o
HTTP requests containing multiple resource
(HTML, CSS, Java script, images)

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device
width, initial-scale=1.0">
<title> My Blog </title>
<link> rel="style sheet" href="styles.css">
</head>
<body>
<header>
<h1> My Blogs </h1>
<nav>
<ul>
<li> <a href="index.html"> Home </a> </li>
<li> <a href="about.html"> About </a> </li>
<li> <a href="contact.html"> contact </li>
</ul>
</nav>
<input type:"search" place holder="search">
</header>
<main>
<article>
<h1> Blog post Title </h1>
```

 here is the introduction to the blog post
<h2> subheading </h2>
 some more content..
<blockquote> citation or highlited text</blockquotes>

 point 1
 point 2
 point3

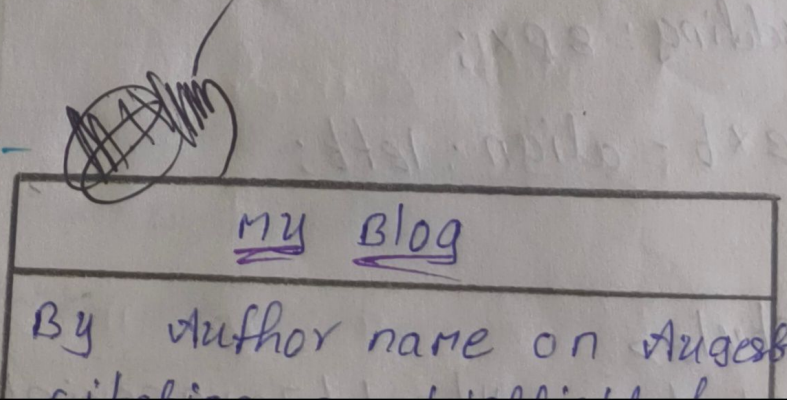
<table>

<th> feature </th>
<th> option 1 </th>
<th> option 2 </th>
</br>

<td> feature A </td>
<td> yes </td>
<td> No </td>
</br>
</table>
</article>
</main>
<aside>
<h3> Related posts </h3>

</aside>
<foober>
 © copy: 2024 my Blog . All rights reserved
 privacy policy
</foober>
</body>
</html>

output:-



Assignment

Implementing a feature in a web applicabion that tracks the no. of access by a client with a single session you need to use Javascript to manage and monitor session data.

Java servlets

```
import Java.io.Exception;
import Java.io.printwriter;
import Javax.servlet.servlet Exception;
import Javax.servlet.annotation.web servlet;
import Javax.serveb.http.Http servlet;
import Javax.serveb.http.Http session;
import Java.obil.babe;
@web servlet ("/session_bracker")
public class session Tracking servlet extends
Http servlet {
private stabic final long serial/version UID = 1L
probeibed void do get throws servlet Exception
{
response.set content bype ("bexb/html");
printwriber out = response.get writer();
HTTP session session = request.get session(bu
integer accesscount = (integer)
session.get Attribute("access count");
    if(access count == null) {
        access count = 0; }
session.set attribute("access count", access co
    long creation time
    long last Accessed Time();
```

```
out.println("<html><body>");
out.println("<h2>Session Tracking Example</h2>");
out.println("<P>Access count" + accessCount + "</P>");
out.println("<P>session id " + sessionId + "</P>");
out.println("<P>last Accessed time" + new Date() + "</P>");
out.println("</body></html>");
}
}
```

outputs:-

Session Tracking Example
Session ID : 12345ABCDG
Session created : Mon Sep 09 12:00 IST 2024
last Accessed Mon Sep 09 12:01:05 IST 2024
No. of access in this session

2. write a scenario where you had to use JSTL to solve a complex problem and how you went about it. Also elaborate the functi library in JSTL and how to create custom functions.

```
public class product {
private int id;
private string name;
private string category
private double price;
private int popularity;
public product (int id, string name, stri
    category, double price int popularit
    {
      this.id = id;
```

public int getId() { reburn id; }

Public string getname() { reburn name; }

public string get category()

public geb price()

public int geb popularity()

J.S

custom funtion

original: Hello wor

Reversed: dlrow o

<%@ taglib uri="http://java.sun.com/jsp/j
core" prefix="c" %>

<html>

<head>

<bibtle> order management </bibtle>

</head>

<body>

<h2> order lisb </h2>

<form method="geb" acbion="order.jsp">

<label for="sbabus"> filber by sbabus: </label

<select name="sbabus" id="sbabus">

<opbion value="All"> All </opbion>

<opbion value="delivered"> delivered </opbion>

</select>

<inpub bype="submib" value="filber">

</form>

<bable border="1">

</head>

```
<td> ${order.id}</td>
<td> ${order.date}</td>
<td> ${order.status}</td>
<td> ${order.amount}</td>
</tr>
</c:when>
</c:choose>
</c:forEach>
</tbody>
</table>
</body>
</html>
```

Output:-

order list

order ID	Date	Status	Amount
1002	2024-9-8	pending	150.00
1003	2024-9-9	pending	300.00

3. To implement the described functionality for refreshing a stock market quotes page dialog appearing 20 seconds before the refr

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title> stock market Quotes </title>
<script>
```

```
function refreshPage() {
    location.reload();
}
setTimeout(()=>{
const confirmRefresh = confirm("the page will
refresh in 20 seconds");
 if(!confirmRefresh){
     refreshPage();
   }else{
 alert("page refresh cancelled");
}
},25000);
</script>
<body>
<h1> stock market quotes </h1>
</body>
</html>
```

output:-

page display

stock market quotes

- Apple Inc (AAPL) $150.00
Microsoft Corp (MSFT) : $250.00
Alphabet Inc (Google) : $200.00

confirmation Dialog

the page will refresh in 20 sec

[ok] [cancel]

Alert:

page refresh cancelled

[OK]

Page Refresh

Stock Market Quotes

Apple - Inc (APPL) : $152.00

Alphabet Inc (GOOGL) : $2850.00

4) To integrate an external payment gateway service into your e-commerce application using WSDL file.

Generate client code from WSDL

wls import -keep -s src -d bin -p com.example payment. verbose http://example.com/payment gateway

Integrate Generate code into Application

- include Generated code
- configure service endPoint

Handle Response and Errors

Check response

if (response.isSuccessful()) {

} else {

// Handle payment failure

}

Exception Handling.

```
try {
payment Response = port.processPayment
} catch {
    Handle (soap faults (e.g) invalid requ
} catch {
    // Handle connectivity or configuration
}
```

Output :-

Successfull Payment:

Payment successful Transaction Id

payment failure (e.g Invalid card details)

Payment failed error : Invalid card details.

soap fault

Payment failed due to soap fault : Inva

Request format.

Assignment-4

1. Why JDBC is essential in Building database-driver Applications.

configure database source

```
<Resource name = "JdClMyOB"
          auth = "container"
          type = "javax.sql.DataSource"
          max total = "20"
          max wait millis = "10000"
          username = "dbuser"
          password = "dbpassword"
          driverClassName = "com.mysql.cj.jdbc
                             Driver"
          url = "jdbc:mysql://localhost:3306/
                 mydatabase" />
```

lookup DataSource in Java code using JNDI

```
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.sql.DataSource;
public static Connection getConnection()
                    throws Exception {
DataSource ds = (DataSource) initContext
                    .lookup("java:/comp/
    return ds.getConnection();
}
```

Executing sql queries using JDBC statements

① using a statement

```
try connection conn = database util.get
    connection();
  statement stmt = conn.create statement();
string query = "select * from users";
Result set rs = stmt.execute query(query);
while (rs.next()) {
system.out.println("user ID" + rs.get int
        ("id") + " + Name" + rs.get string
```

using a callable statement for stored procedures:-

```
try (connection conn = database.util.get
connection callable statement cstmt = conn.
prepare call("{ call get user By Id(?)}")){
istmt.set Int(1)) {
Result set rs = cstmt.execute query();
while (rs.next()) {
system.out.println("user ID" trs.get Int(1)
}
}
```

Executing sql queries using JDBC statements

① using a statement

```
try connection conn = database obj.getConnection();
    Statement stmt = conn.createStatement();
String query = "select * from users";
ResultSet rs = stmt.executeQuery(query);
while (rs.next()) {
System.out.println("userID" + rs.getInt("id") + " " + Name" + rs.getString
```

using a callable statement for stored procedures:-

```
try (connection conn = database.obj.getConnection
connection callableStatement cstmt = conn.prepareCall("{call getUserByID()}"){
cstmt.getInt()) {
ResultSet rs = cstmt.executeQuery();
while (rs.next()) {
System.out.println("user ID" trs.getInt);
}
}
```

the code used

outputs

outputs:-

statement example output:-

user ID : 1, Name: vishnu

user ID : 2, name: priya

prepared statement

user ID : 1, name: vishnu

callable statement

user ID : 1, name: vishnu.

2. lifecycle phases of a JSP page

① Translation phase

② compilation phase

③ initialization phase

④ Request processing phase

⑤ Destruction phase

① Scriptlets

<%. int sum = 5+10; %>

<P> the sum is <%= sum %> </P>

output:-

the sum is 15

② declarations

<%! int add (int a, int b) { return a+b; } %>

<P> the result is <%= add (3,7) %> </P>

output:-

the result is 10

③ Expressions

<P> current time: <% new Java.util.Date

Advantages: usefull for declaring reasonable methods and variables across multiple requests

Disadvantages:- can clutter JSP with Java code, leading to poor separation concerns.

③ Generates a chess board using HTML tables. width of 400px (total) and each cell to be height and width of 30px

PHP code:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title> Chessboard </title>
<style>
table {
    width: 400px;
    boarder-collapse: collapse;
}
td {
    width: 30px;
    height: 30px;
}
</style>
```

```
</body>
</table>
<?php
//loop for 8 rows for ($row=0; $row<8;
                    $row++){
echo "<br>";
} else {
  echo="<td style='background-#000+black
                    ></td>";
}}
echo "</br>";
}
?>
</table>
</body>
</html>
```

Output:-

[ ] [#] [ ] [#] [ ] [#] [ ] [#]
[#] [ ] [#] [ ] [#] [ ] [#] [ ]
[ ] [#] [ ] [#] [ ] [#] [ ] [#]
[#] [ ] [#] [ ] [#] [ ] [#] [ ]
[ ] [#] [ ] [#] [ ] [#] [ ] [#]
[#] [ ] [#] [ ] [#] [ ] [#] [ ]
[ ] [#] [ ] [#] [ ] [#] [ ] [#]
[#] [ ] [#] [ ] [#] [ ] [#] [ ]

4. PHP Application to extract data and store in XML

Steps:-

Read content from a text file.

Extract patterns using Regular

```
<?php
$filename = "inpub.txt";
$content = file_get_content($filename);
preg_match_all("/[a-zA-Z0-9._+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}/",
$content, $emails);
preg_matches_all("/\+?[0-9]{3,4}[\s]?[0-9]{3,4}[\s]?[0-9]{3,4}/",
$content, $phones);
$xml = new simpleXMLElement("<data></data>");
foreach($emails[0] as $email){
$email_element = $xml->addChild('email');
}
$phone_element = $xml->addChild('phone');
foreach($phones[0] as $phone){
$phone_element->addChild('phone', $phone);
}
$xml->asXML("outpub.xml");
echo "Data extracted and saved to output.xml";
?>
```

outpub:-

```
<data>
<emails>
<email>example@example.com</email>
<email>example@example.com</email>
</emails>
<phones>
```

<phones>+123-456-7920</phone>
<phone>987-654-3210</phone>
</phones>
</data>

Assignment:-3

S.No	Assigment Rubrics	split up		total mark
1.	code implementation	8 M		
	session data Accuracy	5 M		
	Efficiency and clarity	3 M		
	Explanation	4 M		
2.	using JSTL senario Explanation	6 M		
	function library Management	5 M		
	custom function	5 M		
	clarity and organi	4 M		
3.	stock market page script functionality	8 M		
	user intraction design	5 M		
	code efficiency	4 M		
	Explanation	3 M		
4.	understanding of WDSL	6 M		
	client code Gene	6 M		
	Error Hand - dority	8 M		

Assignment - 4

S.No	Assignment Rubrics	Split up	
1.	Explanation of JDBC	5M	
	connection pooling	6M	
	SQL Queries	5M	
	statement type	4M	
2.	lifecycle phase of Explanation	6M	
	Embedding Java code	5M	
	Advantages & disadvantages	5M	
	clarity & depth	4M	
3.	code implementation	8M	
	HTML table structure	5M	
	Alternating colours logic	4M	
	Explanation	3M	
4.	code implementation	8M	
	pattern Extraction	5M	
	XML file Generation	4M	
	DTD vs XML schema comparision	3M	