Q1, Q2

```
<!DOCTYPE html>
<html>
<head>
  <title> Registration </title>
</head>
<body>
  <style>

  .container {
    width: 400px;
    margin: 0 auto;
    padding: 20px;
    border-radius: 5px;
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
  }

  label {
    display: block;
    margin-bottom: 5px;
  }

  input[type="text"], input[type="email"],
input[type="password"] {
    width: 100%;
    padding: 10px;
    border: 1px solid #ccc;
    border-radius: 5px;
    box-sizing: border-box;
    margin-bottom: 10px;
  }

  input[type="submit"] {
    background-color: #4CAF50;
    color: white;
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
  }

  a {
    color: #4CAF50;
    text-decoration: none;
  }
  </style>
  <div class="container">
    <h1>Register</h1>
    <form action="/" method="post">
      <label for="username">Username:</label>
      <input type="text" id="username" name="username"
required>
      <br>
      <label for="email">Email Address:</label>
      <input type="email" id="email" name="email"
required>
      <br>
      <label for="password">Password:</label>
      <input type="password" id="password"
name="password" required>
      <br>
      <label for="confirm_password">Confirm
Password:</label>
      <input type="password" id="confirm_password"
name="confirm_password" required>
      <br>

      <input type="submit" value="Register">
    </form>
  </div>
</body>
</html>
```

Q3

(P3.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="p3.css" type="text/css"?>
<student>
    <id>1</id>
    <name>John Doe</name>
    <email>john.doe@example.com</email>
    <course>Computer Science</course>
</student>
```

(p3.css)

```
student {
    margin-bottom: 15px;
    padding: 10px;
    border-bottom: 1px solid #eee;
}

email,
id,
course ,
name
    {
    color: blue;
    display: block;
  margin-top: 10px;
  font-style: italic;
  font-weight: bold;
}
```

Q4

(p4.xsl)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <html>
      <head>
        <title>Employees</title>
        <style>
          table {
            border-collapse: collapse;
            width: 200px;
          }
          th, td {
            border: 1px solid #ddd;
            padding: 8px;

          }
          th {
            color:blue;
            text-align: left;
          }
        </style>
      </head>
      <body>
        <h1>Employees</h1>
        <table>
          <thead>
            <tr>
              <th>ID</th>
              <th>Name</th>
              <th>Department</th>
            </tr>
          </thead>
          <tbody>
            <xsl:for-each
select="employees/employee">
              <tr>
                <td><xsl:value-of select="id"
/></td>
                <td>
                  <xsl:value-of select="name" />
                </td>
                <td><xsl:value-of
select="department" /></td>
              </tr>
            </xsl:for-each>
          </tbody>
        </table>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

(p4.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="employees.xsl"
type="text/xsl"?>
<employees>
  <employee>
    <id>1</id>
    <name> John doe </name>
    <department>IT</department>
  </employee>
  <employee>
    <id>2</id>
    <name>Jane Smith</name>
    <department>Marketing</department>
  </employee>
  </employees>
```

Q5.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html" indent="yes"/>

  <xsl:template match="/">
    <h1>Products</h1>
    <table>
      <thead>
        <tr>
          <th>Name</th>
          <th>Price</th>
          <th>Category</th>
        </tr>
      </thead>
      <tbody>
        <xsl:for-each select="products/product">
          <xsl:sort select="price"
order="ascending"/>  <tr>
            <td><xsl:value-of select="name"
/></td>
            <td><xsl:value-of select="price"
/></td>
            <td><xsl:value-of select="category"
/></td>
          </tr>
        </xsl:for-each>
      </tbody>
    </table>
  </xsl:template>
</xsl:stylesheet>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="xslsort.xsl"
type="text/xsl"?>
<products>
  <product>
    <name>Laptop</name>
    <price>800</price>
    <category>Electronics</category>
  </product>
  <product>
    <name>Shirt</name>
    <price>200</price>
    <category>Clothing</category>
  </product>
  <product>
    <name>Headphones</name>
    <price>500</price>
    <category>Electronics</category>
  </product>
</products>
```

Q6.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
   <xsl:output method="html" indent="yes"/>

   <xsl:template match="/">
     <h1>Students</h1>
     <ul>
       <xsl:for-each select="students/student">
         <li>
           <xsl:value-of select="name" />
           <xsl:if test="age &gt; 18">
             (Adult)
           </xsl:if>
         </li>
       </xsl:for-each>
     </ul>
   </xsl:template>
</xsl:stylesheet>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="p6.xsl" type="text/xsl"?>
<students>
  <student>
    <name>John Doe</name>
    <age>20</age>
  </student>
  <student>
    <name>Jane Smith</name>
    <age>16</age>
  </student>
</students>
```

Q7.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="p7.xsl" type="text/xsl"?>
<students>
    <student>
        <name>John Doe</name>
        <grade>85</grade>
    </student>
    <student>
        <name>Jane Smith</name>
        <grade>92</grade>
    </student>
    <student>
        <name>Mark Johnson</name>
        <grade>73</grade>
    </student>
    <student>
        <name>Emily Davis</name>
        <grade>66</grade>
    </student>
</students>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">
<xsl:output method="html" indent="yes"/>

<!-- Template to match the root element and start the
transformation -->
<xsl:template match="/">
<html>
<body>
    <h2>Student Grades</h2>
    <table border="1">
        <tr bgcolor="#9acd32">
            <th>Name</th>
            <th>Grade</th>
            <th>Evaluation</th>
        </tr>
        <xsl:for-each select="students/student">
            <tr>
                <td><xsl:value-of select="name"/></td>
                <td><xsl:value-of select="grade"/></td>
                <td>
                    <xsl:choose>
                        <xsl:when test="grade &gt;= 90">
                            Excellent
                        </xsl:when>
                        <xsl:when test="grade &gt;= 75">
                            Good
                        </xsl:when>
                        <xsl:when test="grade &gt;= 50">
                            Pass
                        </xsl:when>
                        <xsl:otherwise>
                            Fail
                        </xsl:otherwise>
                    </xsl:choose>
                </td>
            </tr>
        </xsl:for-each>
    </table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Q8.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="p8.xsl" type="text/xsl"?>
<library>
    <book>
        <title>Learning XML</title>
        <author>John Doe</author>
        <year>2005</year>
        <price>39.95</price>
    </book>
    <book>
        <title>XSLT for Beginners</title>
        <author>Jane Smith</author>
        <year>2010</year>
        <price>29.95</price>
    </book>
    <book>
        <title>Advanced XML</title>
        <author>John Doe</author>
        <year>2015</year>
        <price>49.95</price>
    </book>
    <book>
        <title>XML Pocket Reference</title>
        <author>Mark Johnson</author>
        <year>2000</year>
        <price>19.95</price>
    </book>
</library>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">
<xsl:output method="html" indent="yes"/>

<!-- Template to match the root element and start the
transformation -->
<xsl:template match="/">
<html>
<body>
    <h2>Books Published After 2005</h2>
    <table border="1">
        <tr bgcolor="#9acd32">
            <th>Title</th>
            <th>Author</th>
            <th>Year</th>
            <th>Price</th>
        </tr>
        <!-- Using XPath predicate to select books
published after 2005 -->
        <xsl:for-each select="library/book[year >
2005]">
            <tr>
                <td><xsl:value-of select="title"/></td>
                <td><xsl:value-of select="author"/></td>
                <td><xsl:value-of select="year"/></td>
                <td><xsl:value-of select="price"/></td>
            </tr>
        </xsl:for-each>
    </table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Q9.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="p9.xsl" type="text/xsl"?>
<company>
    <department name="Sales">
        <employee id="1">
            <name>John Doe</name>
            <position>Manager</position>
            <age>40</age>
        </employee>
        <employee id="2">
            <name>Jane Smith</name>
            <position>Salesperson</position>
            <age>30</age>
        </employee>
    </department>
    <department name="IT">
        <employee id="3">
            <name>Mike Johnson</name>
            <position>Developer</position>
            <age>25</age>
        </employee>
        <employee id="4">
            <name>Emily Davis</name>
            <position>Support</position>
            <age>28</age>
        </employee>
    </department>
</company>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">
<xsl:output method="html" indent="yes"/>
<xsl:template match="/">
<html>
<body>
<h2>Company Employees</h2>
<table border="1">
    <tr bgcolor="#9acd32">
        <th>Department</th>
        <th>Employee Name</th>
        <th>Position</th>
        <th>Employee Age</th>
        <th>Department Name (Using ancestor axis)</th>
        <th>Next Employee (Using following-sibling
axis)</th>
    </tr>
    <xsl:for-each select="//employee">
        <tr>
            <td><xsl:value-of select="../@name"/></td>
            <td><xsl:value-of select="name"/></td>
            <td><xsl:value-of select="position"/></td>
            <td><xsl:value-of select="age"/></td>
            <td><xsl:value-of
select="ancestor::department/@name"/></td>
            <td><xsl:value-of select="following-
sibling::employee[1]/name"/></td>
    </xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Q10.

```
<?xml version="1.0" encoding="UTF-8"?>
<catalog>
    <product>
        <name>Widget</name>
        <price>25.50</price>
        <quantity>100</quantity>
    </product>
    <product>
        <name>Gadget</name>
        <price>15.00</price>
        <quantity>50</quantity>
    </product>
    <product>
        <name>Doohickey</name>
        <price>7.75</price>
        <quantity>200</quantity>
    </product>
</catalog>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">
<xsl:output method="html" indent="yes"/>
<xsl:template match="/">
<html>
<body>
<h2>Product Catalog</h2>
<table border="1">
    <tr bgcolor="#9acd32">
        <th>Product Name</th>
        <th>Price</th>
        <th>Quantity</th>
        <th>Total Value</th>
        <th>Name Upper Case</th>
        <th>Name Length</th>
    </tr>
    <xsl:for-each select="catalog/product">
        <tr>
            <td><xsl:value-of select="name"/></td>
            <td><xsl:value-of
select="price"/></td>
            <td><xsl:value-of
select="quantity"/></td>
            <td><xsl:value-of select="price *
quantity"/></td>
            <td><xsl:value-of
select="translate(name,
'abcdefghijklmnopqrstuvwxyz',
'ABCDEFGHIJKLMNOPQRSTUVWXYZ')"/></td>
            <td><xsl:value-of select="string-
length(name)"/></td>
        </tr>
    </xsl:for-each>
</table>
<h3>Total Quantity: <xsl:value-of
select="sum(catalog/product/quantity)"/></h3>
<h3>Number of Products: <xsl:value-of
select="count(catalog/product)"/></h3>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Q11.

(index.js)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-
width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <p id="message">This message will be
changed.</p>
  <button onclick="changeMessage()">Change
Text</button>
  <script src="script.js"></script>
</body>
</html>
```

(script.js)

```
function changeMessage() {
    document.getElementById("message").innerHTML =
"External JavaScript changed the content!";
  }
```

Q12.

```
<html>
<head>
  <title>Document</title>
</head>
<body>

  <script>
    let mat1 = [
      [1, 1, 1, 1],
      [2, 2, 2, 2],
      [3, 3, 3, 3],
      [4, 4, 4, 4],
    ];
    let mat2 = [
      [1, 1, 1, 1],
      [2, 2, 2, 2],
      [3, 3, 3, 3],
      [4, 4, 4, 4],
    ];
    let resmat = [];
    for (let i = 0; i < mat1.length; i++) {
      let r = "";
      for (let j = 0; j < mat1[i].length; j++) {
        r += mat1[i][j] + mat2[i][j] + " ";
      }
      resmat.push(r.trim());
    }
    resmat.forEach(r => console.log(r));
  </script>
</body>
</html>
```

Q13.

```
<html>
<head>
  <title>Document</title>
</head>
<body>
  <script>
    // Create an object
    var person = {
      firstName: "ABC",
      lastName: "XYZ",
      age: 30,
      job: "Developer",
      fullName: function () {
        return this.firstName + " " + this.lastName;
      }
    };

    // Access and display object properties
    console.log("First Name: " + person.firstName);
    console.log("Last Name: " + person.lastName);
    console.log("Age: " + person.age);
    console.log("Job: " + person.job);

    // Call and display object method
    console.log("Full Name: " + person.fullName());
  </script>
</body>
</html>
```

Q14.

```
<html>
<head>
  <title>Document</title>
</head>
<body>
  <script>
    function divide(a, b) {
      try {
        if (b === 0) {
          throw new Error("Division by zero is not
allowed.");
        }
        var result = a / b;
        console.log("Result: " + result);
      } catch (error) {
        console.log("Error: " + error.message);
      } finally {
        console.log("Division attempt completed.");
      }
    }
    divide(10, 2);
    divide(10, 0);
  </script>
</body>
</html>
```

Q15

```
<html>
<head>
  <title>key event</title>
</head>
<body>
  <p>Press a key to enter information</p>
  <script>
    document.addEventListener("keydown", function
(event) {
      alert("You have Pressed : " + event.key);
    });
  </script>
</body>
</html>
```

Q16.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Mouse Event</title>
</head>
<body>
  <p id="mypara">
    This ia a Paragraph
  </p>
  <br><br>
  <button id="btn1">Click me</button>
  <script>
    mypara.onmouseover = (e) => {
      mypara.style.color = "red";
    }
    mypara.onmouseleave = (e) => {
      mypara.style.color = "black";
    }
    btn1.ondblclick = (e) => {
      alert("You have clicked the button twice");
    }
  </script>
</body>
</html>
```

Q17.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Events in Js</title>
</head>
<body>
    <button id="btn1" type="button"">Click me
event</button>
    <script>
      btn1.onclick = (e) =>{
        alert("You have Clicked the button")
      }
    </script>
</body>
</html>
```

Q18.

```
<!DOCTYPE html>
<html>
<body>
  <h2>The focus and blur Events</h2>
  <form id="myForm">
    <input type="text" id="myInput">
  </form>
  <script>
    myForm.addEventListener("focus",
myFocusFunction, true);
    myForm.addEventListener("blur",
myBlurFunction, true);
    function myFocusFunction() {
      document.getElementById("myInput")
.style.backgroundColor = "yellow";
    }
    function myBlurFunction() {
      document.getElementById("myInput")
.style.backgroundColor = "";
    }
  </script>
</body>
</html>
```

Q19.

```
<!DOCTYPE html>
<html>
<head>
  <title>Window Event Example</title>
</head>
<body>

<h1>Click anywhere on the window</h1>

<script>
  window.addEventListener("click", () => {
    alert("You clicked the window!");
  });
</script>

</body>
</html>
```

Q21.

(npx create-react-app wt)

```
import React, { useState } from 'react';

function App() {
  const [count, setCount] = useState(0);

  const handleClick = () => {
    setCount(count + 1);
  };

  return (
    <div className="App">
      <h1>My Simple Counter</h1>
      <p>You clicked {count} times</p>
      <button onClick={handleClick}>Click
me</button>
    </div>
  );
}

export default App;
```

Q20.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Form Validation Example</title>
    <style>
        .error {
            color: red;
        }
    </style>
    <script>
        function validateForm() {
            let valid = true;
            const name =
document.forms["myForm"]["name"].value;
            const email =
document.forms["myForm"]["email"].value;
            const age =
document.forms["myForm"]["age"].value;
            const nameError =
document.getElementById("nameError");
            const emailError =
document.getElementById("emailError");
            const ageError =
document.getElementById("ageError");

            nameError.innerHTML = "";
            emailError.innerHTML = "";
            ageError.innerHTML = "";

            // Validate name
            if (name === "") {
                nameError.innerHTML = "Name is required";
                valid = false;
            } else if (name.length < 3) {
                nameError.innerHTML = "Name must be at least
3 characters long";
                valid = false;
            }

            // Validate email
            const emailPattern =
/^[^\s@]+@[^\s@]+\.[^\s@]+$/;
            if (email === "") {
                emailError.innerHTML = "Email is required";
                valid = false;
            } else if (!emailPattern.test(email)) {
                emailError.innerHTML = "Invalid email
format";
                valid = false;
            }
            return valid;
        }
    </script>
</head>
<body>
    <h2>Form Validation Example</h2>
    <form name="myForm" onsubmit="return validateForm()"
method="post">
        <div>
            <label for="name">Name:</label>
            <input type="text" id="name" name="name">
            <span class="error" id="nameError"></span>
        </div>
        <div>
            <label for="email">Email:</label>
            <input type="text" id="email" name="email">
            <span class="error" id="emailError"></span>
        </div>
        <div>
            <label for="age">Age:</label>
            <input type="text" id="age" name="age">
            <span class="error" id="ageError"></span>
        </div>
        <div>
            <input type="submit" value="Submit">
        </div>
    </form>
</body>
</html>
```

(npm start)

Q22.

```php
<?php
    $student_one = array("Maths"=>90,
"chemistry"=>89, "Biology"=>86,
"Physics"=>76);

    echo "Marks for student one is = \n";
    echo "Chemistry : ".
$student_one["chemistry"],"\n";
    echo "Biology : ".
$student_one["Biology"],"\n";
    echo "Physics : ".
$student_one["Physics"],"\n";

?>
```

Q23.

```php
<?php
    $arr = array(5,8,9,2,3,0,1);
    sort($arr);
    print_r($arr);
    echo "\n";

    $array1 = array("1" => "aakash", "2" =>
"rishav", "3" => "gaurav");
    $array2 = array("1" => "shyam", "2" =>
"rishi", "5" => "rishav");
    $array3 = array("1" => "aakash", "4" =>
"raghav", "2" => "ravi");

    print_r(array_intersect_key($array1, $array2,
$array3));

    print_r(count($arr));
    echo "\n";

    print_r(array_push($arr,89,78));
    print_r($arr);

    print_r(array_pop($arr));
    print_r($arr);

    print_r(array_search("aakash",$array1));

    $array = array("vasu","sakshi","kashish");
    $name1 = array(1 => "rohit");
    $name2 = array(0 => "abhay");

    $res_array = array_replace($array, $name1,
$name2);
    print_r($res_array);
?>
```

Q24.

```php
<?php
class Fruit {
  // Properties
  public $name;

  // Methods
  function set_name($name) {
    $this->name = $name;
  }
  function get_name() {
    return $this->name;
  }
}

$apple = new Fruit();
$banana = new Fruit();
$apple->set_name('Apple');
$banana->set_name('Banana');

echo $apple->get_name();
echo "<br>";
echo $banana->get_name();
?>
```

Q25.

(p25.html)

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Simple Form</title>
</head>
<body>
    <form action="server.php" method="post">
        <label for="name">Name:</label>
        <input type="text" id="name" name="name"
required>
        <br>
        <label for="email">Email:</label>
        <input type="email" id="email"
name="email" required>
        <br>
        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

(server.php)

```php
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = $_POST['name'];
    $email = $_POST['email'];
    echo "Name: " . $name . "<br>";
    echo "Email: " . $email;
} else {
    echo "Form not submitted.";}
?>
```

Q26.
(index.html)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>GET Form Example</title>
</head>
<body>
    <form action="server.php" method="get">
        <label for="name">Name:</label>
        <input type="text" id="name" name="name">
        <br><br>
        <label for="email">Email:</label>
        <input type="email" id="email" name="email">
        <br><br>
        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

(server.php)

```
<?php
if (isset($_GET['name']) && isset($_GET['email'])) {
    $name = $_GET['name'];
    $email = $_GET['email'];

    echo "Name: " . $name . "<br>";
    echo "Email: " . $email . "<br>";
} else {
    echo "Form data not submitted!";
}
?>
```

Q27.
(index.html)

```
<html>
    <body>
        <form action="p27.php" method="post"
enctype="multipart/form-data">
            select file:
            <input type="file" name="fileToUpload"/>
            <input type="submit" value="Upload Image"
name="submit"/>
        </form>
    </body>
</html>
```

(server.php)

```
<?php
        $target="c:/";
        if(move_uploaded_file($_FILES['fileToUpload']
['tmp_name'], $target.$_FILES['fileToUpload']['name']))
        {
            echo "File uploaded successfully!";
        }
        else
        {
            echo "Sorry, File not uploaded, please try
again!";
        }
    ?>
```

Q28.
(p1.php)

```
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
$_SESSION["favcolor"] = "green";
$_SESSION["favanimal"] = "cat";
echo "Session variables are set.";
?>

</body>
</html>
```

(p2.php)

```
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
echo "Favorite color is " . $_SESSION["favcolor"]
. ".<br>";
echo "Favorite animal is " .
$_SESSION["favanimal"] . ".";
?>

</body>
</html>
```

Q29.

```
<?php
$value = "Kashish Jadhav";
setcookie("my_cookie", $value, time() + (86400 *
30), "/");

if(isset($_COOKIE["my_cookie"])) {
    $stored_value = $_COOKIE["my_cookie"];
    echo "Value retrieved from cookie: " .
$stored_value;
} else {
    echo "Cookie not set.";
}
?>
```

Q30.
(index.html)

```
<html>
<head>
    <title>Insert values in database</title>
</head>
<body>

    <form action="server.php" method="POST">
    <label for="name">Name: </label>
    <input type="text" name="name"
id="name"><br><br>

    <label for="email">Email: </label>
    <input type="email" name="email"
id="email"><br><br>

    <label for="pass">Password: </label>
    <input type="password" name="pass"
id="pass"><br><br>

    <button type="submit">Submit</button>
    </form>
</body>
</html>
```

(server.php)

```
<?php
    $mycon = mysqli_connect("localhost", "root",
"", "user");
    if(isset($_POST["name"])){
        $name = $_POST["name"];
        $email = $_POST["email"];
        $pass = $_POST["pass"];
        $sql_query = "INSERT INTO user (`name`,
`email`, `pass`) VALUES ('$name', '$email',
'$pass')";
        $sql_exec = mysqli_query($mycon,
$sql_query);
        echo "<h2>Hi, $name</h2> </br>";
        echo "<h3>Data Saved Successfully </br>";
    }
    else{
        echo "<h3>Error While sending Data </br>";
    }
?>
```

Q31.

```
<?php
    $con = mysqli_connect("localhost", "root", "",
"user");
    $query = "SELECT * FROM user";
    $data = mysqli_query($con,$query);
    $total = mysqli_num_rows($data);

    if($total != 0){
    ?>
        <table>
            <tr>
                <th>Roll No</th>
                <th>Name</th>
                <th>Class</th>
            </tr>

    <?php
        while($result = mysqli_fetch_assoc($data)){
            echo "<tr>
            <td>".$result['rollno']."</td>
            <td>".$result['studentname']."</td>
            <td>".$result['class']."</td>
        </tr>";
        }
    }
    else{
        echo "Table has no records";
    }
?>
```