1 Write a js code to validate a form
Name -fuelds can't be empty
Afe-number should be valid
Password
Confirm password

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

<form id="myForm" onsubmit="return validateForm()">
  <label for="name">Name:</label><br>
  <input type="text" id="name" name="name"><br>
  <label for="age">Age:</label><br>
  <input type="text" id="age" name="age"><br>
  <label for="password">Password:</label><br>
  <input type="password" id="password" name="password"><br>
  <label for="confirmPassword">Confirm Password:</label><br>
  <input type="password" id="confirmPassword"
name="confirmPassword"><br><br>
  <input type="submit" value="Submit">
</form>

<script>
function validateForm() {
  var name = document.getElementById('name').value;
  var age = document.getElementById('age').value;
  var password = document.getElementById('password').value;
  var confirmPassword = document.getElementById('confirmPassword').value;

  // Check if name field is empty
  if (name == "") {
    alert("Name must be filled out");
    return false;
```

```
  }

  // Check if age is a valid number
  if (isNaN(age) || age == "") {
    alert("Age must be a valid number");
    return false;
  }

  // Check if password field is empty
  if (password == "") {
    alert("Password must be filled out");
    return false;
  }

  // Check if confirm password matches password
  if (confirmPassword != password) {
    alert("Passwords do not match");
    return false;
  }

  return true;
}
</script>

</body>
</html>
```

2)write a js code that collects following info from student -
Usn
Branch-dropdown
Name
Marks obtained in3 subject
Calculate percentage of marks and display all entered details along with percentage

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

```
<title>Student Information</title>
</head>
<body>

<h2>Enter Student Information</h2>

<form id="studentForm">
  <label for="usn">USN:</label><br>
  <input type="text" id="usn" name="usn"><br><br>

  <label for="branch">Branch:</label><br>
  <select id="branch" name="branch">
    <option value="CSE">Computer Science and Engineering</option>
    <option value="ECE">Electronics and Communication Engineering</option>
    <option value="ME">Mechanical Engineering</option>
    <option value="CE">Civil Engineering</option>
  </select><br><br>

  <label for="name">Name:</label><br>
  <input type="text" id="name" name="name"><br><br>

  <label for="subject1">Marks in Subject 1:</label><br>
  <input type="number" id="subject1" name="subject1"><br><br>

  <label for="subject2">Marks in Subject 2:</label><br>
  <input type="number" id="subject2" name="subject2"><br><br>

  <label for="subject3">Marks in Subject 3:</label><br>
  <input type="number" id="subject3" name="subject3"><br><br>

  <input type="button" value="Submit" onclick="calculatePercentage()">
</form>

<div id="result"></div>

<script>
function calculatePercentage() {
  var usn = document.getElementById('usn').value;
  var branch = document.getElementById('branch').value;
  var name = document.getElementById('name').value;
```

```
  var subject1 = parseInt(document.getElementById('subject1').value);
  var subject2 = parseInt(document.getElementById('subject2').value);
  var subject3 = parseInt(document.getElementById('subject3').value);

  var totalMarks = subject1 + subject2 + subject3;
  var percentage = (totalMarks / 3).toFixed(2);

  var result = "USN: " + usn + "<br>" +
            "Branch: " + branch + "<br>" +
            "Name: " + name + "<br>" +
            "Marks in Subject 1: " + subject1 + "<br>" +
            "Marks in Subject 2: " + subject2 + "<br>" +
            "Marks in Subject 3: " + subject3 + "<br>" +
            "Percentage: " + percentage + "%";

  document.getElementById('result').innerHTML = result;
}
</script>

</body>
</html>
```

3) what are different ways of declaring variables and give suitable examples

Using `var`: `var` is the most common way to declare variables in JavaScript. Variables declared with `var` have function scope or global scope, depending on where they are declared.
Example:

```
var age = 25;
var name = "John";
```

Using `let`: Introduced in ECMAScript 6 (ES6), `let` allows you to declare block-scoped variables. Variables declared with `let` are limited to the block (enclosed by curly braces) in which they are defined.
Example:

```
let x = 10;
let y = "Hello";
```

Using `const`: Also introduced in ES6, `const` allows you to declare variables whose values cannot be reassigned. However, it's important to

note that **`const`** does not make the variable immutable; it just prevents reassignment of the variable itself.
Example:
const PI = 3.14;
const name = "Alice";
Using Object Destructuring: You can declare variables by destructuring values from objects.
Example:
const person = { name: "John", age: 30 };
const { name, age } = person;

Using Array Destructuring: Similar to object destructuring, you can declare variables by destructuring values from arrays.
Example:
const [first, second, third] = [1, 2, 3];
Global Variables: Variables declared without any keyword (e.g., **`var`**, **`let`**, **`const`**) become global variables.
Example:
myGlobalVariable = "Hello";

4) devlope a js code that computes square of a number whenever a square button is clicked and similarly calculate cube of the number

5) outline p<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Square and Cube Calculator</title>
</head>
<body>

<h2>Enter a Number</h2>

```
<input type="number" id="numberInput" placeholder="Enter a
number">
<button onclick="calculateSquare()">Square</button>
<button onclick="calculateCube()">Cube</button>

<div id="result"></div>

<script>
function calculateSquare() {
  var number = document.getElementById("numberInput").value;
  var square = number * number;
  document.getElementById("result").innerHTML = "Square of " +
number + " is: " + square;
}

function calculateCube() {
  var number = document.getElementById("numberInput").value;
  var cube = number * number * number;
  document.getElementById("result").innerHTML = "Cube of " + number
+ " is: " + cube;
}
</script>

</body>
</html>
```

5 Outline prrocess of establishing a connection to my SQL database and inserted data using php

Establishing a connection to a MySQL database and inserting data using PHP involves several steps. Below is an outline of the process:

1. **Set up MySQL Database**:
   - Ensure that you have MySQL installed on your server or local machine.
   - Create a database and the necessary tables using MySQL commands or a graphical interface like phpMyAdmin.
2. **PHP Configuration**:
   - Make sure PHP is installed on your server or local machine.

- Ensure that PHP is configured properly to work with MySQL. This usually involves enabling the MySQL extension in the PHP configuration file (`php.ini`). For newer PHP versions, you might need to use the MySQLi or PDO extension.

3. **Establish Connection**:
   - Use PHP to establish a connection to the MySQL database using appropriate credentials (hostname, username, password, and database name).
4. **Insert Data**:
   - Prepare an SQL INSERT statement with the data you want to insert into the database.
   - Execute the INSERT statement using PHP and the established database connection.

Here's a simple example code demonstrating these steps:

```
<?php
// Step 1: Database credentials
$hostname = "localhost"; // or your server hostname
$username = "your_username"; // your MySQL username
$password = "your_password"; // your MySQL password
$database = "your_database"; // your MySQL database name

// Step 2: Establish connection
$connection = mysqli_connect($hostname, $username, $password,
$database);

// Check connection
if (!$connection) {
    die("Connection failed: " . mysqli_connect_error());
}

// Step 3: Insert data
$name = "John Doe";
$email = "john@example.com";
$age = 30;
```

```
$sql = "INSERT INTO users (name, email, age) VALUES ('$name', '$email',
$age)";

if (mysqli_query($connection, $sql)) {
    echo "New record inserted successfully";
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($connection);
}

// Step 4: Close connection
mysqli_close($connection);
?>
```

6) explain concepts of state and props in react js

State:

- State is a JavaScript object that stores data relevant to a component.
- It represents the current condition of the component.
- Whenever the state of a component changes, React re-renders the component to reflect the updated state.
- State is mutable and can be updated using the `setState()` method provided by React.
- State is typically initialized in the constructor of a class component using `this.state`.

Example:

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

class Counter extends Component {
  constructor(props) {
    super(props);
    this.state = {
      count: 0
    };
  }
```

```
  incrementCount() {
    this.setState({ count: this.state.count + 1 });
  }

  render() {
    return (
      <div>
        <p>Count: {this.state.count}</p>
        <button onClick={() =>
this.incrementCount()}>Increment</button>
      </div>
    );
  }
}

export default Counter;
```

Props:

- Props (short for properties) are a way to pass data from parent components to child components.
- Props are immutable, meaning they cannot be changed by the child components.
- Parent components can pass any type of data as props to their child components, including strings, numbers, functions, or even other components.
- Props are passed as attributes to components when they are instantiated.

Example:

```
import React from 'react';

const WelcomeMessage = (props) => {
  return <h1>Welcome, {props.name}!</h1>;
}

const App = () => {
  return <WelcomeMessage name="John" />;
```

```
}

export default App;
```

7) write a php program to accept roll no , students name and display values after submission

```
<!DOCTYPE html>
<html>
<head>
    <title>Student Information</title>
</head>
<body>

    <h2>Enter Student Information</h2>

    <form method="post">
        <label for="rollNo">Roll Number:</label><br>
        <input type="text" id="rollNo" name="rollNo"><br><br>

        <label for="studentName">Student Name:</label><br>
        <input type="text" id="studentName"
name="studentName"><br><br>

        <input type="submit" name="submit" value="Submit">
    </form>

    <?php
    // Check if form is submitted
    if (isset($_POST['submit'])) {
        // Retrieve data from form
        $rollNo = $_POST['rollNo'];
        $studentName = $_POST['studentName'];

        // Display submitted values
        echo "<h2>Submitted Information</h2>";
        echo "<p>Roll
```

8) create a react js program to greet .

```
import React from 'react';
import ReactDOM from 'react-dom';

class Greeting extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      name: '',
      greeting: 'Hello'
    };
    this.handleChange = this.handleChange.bind(this);
  }

  handleChange(event) {
    this.setState({ name: event.target.value });
  }

  render() {
    return (
      <div>
        <h2>Welcome to Greeting App</h2>
        <label htmlFor="nameInput">Enter your name: </label>
        <input
          id="nameInput"
          type="text"
          value={this.state.name}
          onChange={this.handleChange}
        />
        <button onClick={() => this.setState({ greeting: 'Hello' })}>Say
Hello</button>
        <button onClick={() => this.setState({ greeting: 'Hi' })}>Say
Hi</button>
        <button onClick={() => this.setState({ greeting: 'Good morning'
})}>Say Good Morning</button>
        <p>{this.state.greeting}, {this.state.name || 'Stranger'}!</p>
```

```
      </div>
    );
  }
}

ReactDOM.render(<Greeting />, document.getElementById('root'));
```

9)develop a js function which validate e-mail field in a form and ensure that it follows proper email format

```
function validateEmail(email) {
  // Regular expression pattern to match email format
  const emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

  // Test if the email matches the pattern
  return emailPattern.test(email);
}

// Example usage:
const email = "example@example.com";
if (validateEmail(email)) {
  console.log("Email is valid.");
} else {
  console.log("Email is not valid.");
}
```

10)write a php script to connect to MySQL database and fetch record from a table. Displaying them in a webpage.

```
<!DOCTYPE html>
<html>
<head>
    <title>Display Records</title>
</head>
<body>

    <h2>Records from Database</h2>
```

```
    <table border="1">
        <tr>
            <th>ID</th>
            <th>Name</th>
            <th>Email</th>
        </tr>

        <?php
        // Database connection settings
        $hostname = "localhost"; // or your server hostname
        $username = "your_username"; // your MySQL username
        $password = "your_password"; // your MySQL password
        $database = "your_database"; // your MySQL database name

        // Create connection
        $conn = new mysqli($hostname, $username, $password,
$database);

        // Check connection
        if ($conn->connect_error) {
            die("Connection failed: " . $conn->connect_error);
        }

        // Fetch records from table
        $sql = "SELECT id, name, email FROM your_table";
        $result = $conn->query($sql);

        if ($result->num_rows > 0) {
            // Output data of each row
            while($row = $result->fetch_assoc()) {
                echo "<tr>";
                echo "<td>".$row["id"]."</td>";
                echo "<td>".$row["name"]."</td>";
                echo "<td>".$row["email"]."</td>";
                echo "</tr>";
            }
        } else {
```

```
            echo "<tr><td colspan='3'>No records found</td></tr>";
        }

        // Close connection
        $conn->close();
        ?>

    </table>

</body>
</html>
```

11) create a react js function that uses hooks to manage form inputs for a login form including user name and password.

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

function LoginForm() {
  // State variables for username and password
  const [username, setUsername] = useState('');
  const [password, setPassword] = useState('');

  // Function to handle username input change
  const handleUsernameChange = (event) => {
    setUsername(event.target.value);
  };

  // Function to handle password input change
  const handlePasswordChange = (event) => {
    setPassword(event.target.value);
  };

  // Function to handle form submission
  const handleSubmit = (event) => {
    event.preventDefault();
    // Perform login logic here (e.g., send login request to server)
    console.log("Username:", username);
```

```
    console.log("Password:", password);
    // Reset the form after submission
    setUsername('');
    setPassword('');
  };

  return (
    <form onSubmit={handleSubmit}>
      <h2>Login</h2>
      <div>
        <label htmlFor="username">Username:</label>
        <input
          type="text"
          id="username"
          value={username}
          onChange={handleUsernameChange}
          required
        />
      </div>
      <div>
        <label htmlFor="password">Password:</label>
        <input
          type="password"
          id="password"
          value={password}
          onChange={handlePasswordChange}
          required
        />
      </div>
      <button type="submit">Login</button>
    </form>
  );
}

export default LoginForm;
```