Milestone Assignment

Q 1. Password Validator.

Answer:-

```
// Prompt the user to enter their password
const userPassword = prompt('Please enter your password:');

// Prompt the user to confirm their password
const confirmPassword = prompt('Please confirm your password:');

// Check if the entered password and the confirmed password match
if (userPassword === confirmPassword) {
  console.log('Password validation Successful');
} else {
  console.log('Password didn\'t match. Password validation
unsuccessful');
}
```

Q 2. Calculator.

Answer:-

```
function calculator(num1, num2, operator) {
  let result;

  switch (operator) {
    case '+':
```

```
      result = num1 + num2;
      break;
    case '-':
      result = num1 - num2;
      break;
    case '*':
      result = num1 * num2;
      break;
    case '/':
      if (num2 === 0) {
        console.log('Error: Division by zero is not allowed.');
        return;
      }
      result = num1 / num2;
      break;
    default:
      console.log('Invalid operator');
      return;
  }

  console.log(`The result of the operation ${num1} ${operator} ${num2}
is ${result}`);
}

// Test cases
calculator(5, 3, '+'); // Output: The result of the operation 5 + 3 is 8
calculator(5, 3, '-'); // Output: The result of the operation 5 - 3 is 2
calculator(5, 3, '*'); // Output: The result of the operation 5 * 3 is 15
calculator(5, 3, '/'); // Output: The result of the operation 5 / 3 is
1.6666666666666667
calculator(5, 3, '%'); // Output: Invalid operator
```

Q 3. Color Mixer.

Answer:-

```
function mixColors(color1, color2) {
  // Convert input to lowercase for case-insensitive comparison
  color1 = color1.toLowerCase();
  color2 = color2.toLowerCase();

  // Use a switch statement to determine the resulting color
  switch (color1 + ' ' + color2) {
    case 'red blue':
    case 'blue red':
      return 'purple';
    case 'red yellow':
    case 'yellow red':
      return 'orange';
    case 'blue yellow':
    case 'yellow blue':
      return 'green';
    default:
      return 'Invalid color combination';
  }
}

// Example usage:
console.log(mixColors('Red', 'BLUE')); // Output: purple
console.log(mixColors('Yellow', 'green')); // Output: Invalid color
combination
```

Q 4. Highest Marks.

Answer:-

```
// Array of marks scored by five students
const marks = [85, 92, 78, 95, 88];

// Initialize highest score and its index
let highestScore = marks[0];
let highestScoreIndex = 0;

// Iterate through the array using a for loop
for (let i = 1; i < marks.length; i++) {
  // Use a ternary operator to check if the current score is higher than the highest score
  marks[i] > highestScore ? (highestScore = marks[i], highestScoreIndex = i) : null;
}

// Display the highest score and the index of the student who scored it
console.log(`The highest score is ${highestScore} and was scored by student
${highestScoreIndex + 1}.`);
```

Q 5. Capitalize.

Answer:-

```
// Function to capitalize the first letter of a string if it's lowercase
function capitalizeFirstLetter(string) {
  return string.charAt(0).toUpperCase() + string.slice(1);
}
```

```
// Example usage:
const userName = 'arzo raja';
const capitalizedName = capitalizeFirstLetter(userName);
console.log(capitalizedName); // Output: 'Arzo raja'
```

Q 6. Vowel Counter.

Answer:-

```
// Function to count the number of vowels in a string
function countVowels(string) {
    // Initialize a counter variable
    let count = 0;

    // Iterate through the string using a for loop
    for (let i = 0; i < string.length; i++) {
      // Check if the current character is a vowel
      if ('aeiou'.indexOf(string[i].toLowerCase()) !== -1) {
        // If it is, increment the counter
        count++;
      }
    }

    // Return the final count
    return count;
  }

  // Example usage:
  const name = 'Akbar ali';
  const vowelCount = countVowels(name);
```

```
console.log(`The number of vowels in '${name}' is ${vowelCount}.`);
```

Q 7 Remove Duplicates.

Answer:-

```
// Function to remove duplicates from an array
function removeDuplicates(array) {
    // Initialize a result array and a hash set
    const result = [];
    const hashSet = new Set();

    // Iterate through the input array using a for loop
    for (let i = 0; i < array.length; i++) {
      // If the current item is not in the hash set, add it to the result array
and the hash set
      if (!hashSet.has(array[i])) {
        result.push(array[i]);
        hashSet.add(array[i]);
      }
    }

    // Return the result array, which contains no duplicates
    return result;
  }

  // Example usage:
  const cartItems = ['apple', 'banana', 'Akbar', 'Akbar', 'apple', 'orange',
'banana'];
  const uniqueCartItems = removeDuplicates(cartItems);
```

```
console.log(uniqueCartItems); // Output: ['apple', 'banana', 'orange']
```

Q 8. Inverted right-angled triangle pattern with asterisks.

Answer:-

```
// Function to print inverted right-angled triangle pattern
function printInvertedTriangle(i) {
  // Check if input is a positive integer
  if (i < 1 || !Number.isInteger(i)) {
    console.log("Please provide a positive integer as input.");
    return;
  }

  // Loop through the number of rows
  for (let row = i; row > 0; row--) {
    // Create a string of asterisks with the current row length
    let rowString = "*".padStart(row, "*");

    // Print the row string
    console.log(rowString);
  }
}

// Example usage:
printInvertedTriangle(6);
```

Q 9 . check for divisibility.

Answer:-

```
// Function to print numbers divisible by 3 but not by 2
function printNumbers(arr) {
  // Loop through the array elements
  for (let i = 0; i < arr.length; i++) {
    // Check if the number is divisible by 3 and not by 2
    if (arr[i] % 3 === 0 && arr[i] % 2 !== 0) {
      // Print the number
      console.log(arr[i]);
    } else {
      // If not, continue to the next iteration
      continue;
    }
  }
}

// Example usage:
const numbers = [12, 15, 18, 21, 24, 27];
printNumbers(numbers);
```

Q 10. Correct a bug.

Answer:-

```
// Function to double the quantity of each item in the cart array
function doubleCartQuantity(cart) {
  // Loop through the elements in the cart array
  for (let i = 0; i < cart.length; i++) {
    // Double the quantity of the current item
    cart[i] *= 2;
  }
}

// Example usage:
const cart = [1, 2, 3, 4, 5];
doubleCartQuantity(cart);
console.log(cart); // Output: [2, 4, 6, 8, 10]
```

Q 11. Unit Converter.

Answer:-

```
// Function to convert Celsius to Fahrenheit
function celsiusToFahrenheit(celsius) {
  // Calculate Fahrenheit from Celsius
  const fahrenheit = (celsius * 9) / 5 + 32;

  // Return the result
  return fahrenheit;
```

```
}

// Example usage:
const celsius = 25;
const fahrenheit = celsiusToFahrenheit(celsius);
console.log(`${celsius}°C is equal to ${fahrenheit}°F`);
```

Q 12. Calculate rental cost.

Answer:-

```
// Function to calculate rental cost
function calculateRentalCost(daysRented, carType) {
  // Define the rental costs
  const RENTAL_COSTS = {
    economy: 4000,
    midsize: 10000,
    luxury: 20000,
  };

  // Check if the carType is valid
  if (!Object.keys(RENTAL_COSTS).includes(carType)) {
    console.error(`Invalid car type. Please use economy, midsize, or
luxury.`);
    return;
  }

  // Calculate the rental cost for the given car type
  const rentalCost = RENTAL_COSTS[carType];
```

```
  // Calculate the total cost
  const totalCost = rentalCost * daysRented;

  // Return the total cost
  return totalCost;
}

// Example usage:
const daysRented = 5;
const carType = 'midsize';
const rentalCost = calculateRentalCost(daysRented, carType);
console.log(`The rental cost for ${daysRented} days in ${carType} is Rs.
${rentalCost}/-`);
```

Q 13. Bill snipping.

Answer:-

```
function calculateBill(dishCost, peopleSharing) {
    if (typeof dishCost !== 'number' || typeof peopleSharing !== 'number') {
      throw new Error('Both dishCost and peopleSharing must be
numbers');
    }

    const totalBill = dishCost * peopleSharing;
    const billPerPerson = totalBill / peopleSharing;

    return { totalBill, billPerPerson };
  }
  const dishCost = 20; // cost of each dish in dollars
const peopleSharing = 4; // number of people sharing the dish
```

```
const bill = calculateBill(dishCost, peopleSharing);
console.log(bill);
```

Q 14. Calculate the final order price.

Answer:-

```
const calculateFinalOrderPrice = (cart) => {
   if (!Array.isArray(cart)) {
     throw new Error('The cart must be an array');
   }

   const totalCost = cart.reduce((acc, item) => {
     if (typeof item.unitPrice !== 'number' || typeof item.quantity !==
'number') {
       throw new Error('Each item in the cart must have a unitPrice and
quantity property that are numbers');
     }
     return acc + item.unitPrice * item.quantity;
   }, 0);

   return totalCost;
 };
 const cart = [
   { unitPrice: 10, quantity: 2 },
   { unitPrice: 20, quantity: 3 },
   { unitPrice: 30, quantity: 1 },
 ];
```

```
const finalOrderPrice = calculateFinalOrderPrice(cart);
console.log(finalOrderPrice);
```

Q 15. Calculate the percentage of the discount.

Answer:-

```
const calculateDiscountPercentage = (originalPrice, discountedPrice) =>
{
   if (originalPrice <= 0 || discountedPrice < 0 || discountedPrice >=
originalPrice) {
     throw new Error('Invalid input');
   }

   const discountAmount = originalPrice - discountedPrice;
   const discountPercentage = (discountAmount / originalPrice) * 100;

   return Math.round(discountPercentage * 100) / 100;
 };
 const originalPrice = 1000;
const discountedPrice = 80;

const discountPercentage = calculateDiscountPercentage(originalPrice,
discountedPrice);
console.log(discountPercentage);
```

Q 16. Generate a random number.

Answer:-

```
(() => {
   const randomNumber = Math.floor(Math.random() * 100) + 1;
   console.log(`Random number generated: ${randomNumber}`);
 })();
```

Q 17. Build a banking application.

Answer:-

```
const customer = {
   name: 'Akbar ali',
   balance: 10000,

   deposit: function (amount) {
     if (typeof amount !== 'number' || amount <= 0) {
       throw new Error('Invalid deposit amount');
     }
     this.balance += amount;
     console.log(`Deposit successful! New balance: $${this.balance}`);
   },

   withdraw: function (amount) {
     if (typeof amount !== 'number' || amount <= 0) {
       throw new Error('Invalid withdrawal amount');
     }
     if (this.balance < amount) {
       throw new Error('Insufficient balance');
     }
     this.balance -= amount;
```

```
    console.log(`Withdrawal successful! New balance: ${this.balance}`);
  },
};

// Example usage:
customer.deposit(500);
customer.withdraw(299);
```

Q 18. Change text on button click.

Answer:-

```
<!DOCTYPE html>
<html>
<head>
  <title>Toggle Heading Text</title>
  <style>
    h1 {
      font-size: 2em;
      margin-bottom: 0.5em;
    }
    button {
      font-size: 1em;
      padding: 0.5em 1em;
      cursor: pointer;
    }
  </style>
</head>
<body>
  <h1 id="heading">The most affordable learning platform</h1>
  <button id="toggle-button">Toggle text</button>

  <script>
    const heading = document.getElementById('heading');
    const toggleButton = document.getElementById('toggle-button');
```

```
    toggleButton.addEventListener('click', () => {
      if (heading.textContent === 'The most affordable learning platform') {
        heading.textContent = 'PW Skills';
      } else {
        heading.textContent = 'The most affordable learning platform';
      }
    });
  </script>
</body>
</html>
```

Q 19. Validate password.

Answer:-

```
<!DOCTYPE html>
<html>
<head>
  <title>Login Form</title>
  <style>
    .error {
      color: red;
    }
    .success {
      color: green;
    }
  </style>
</head>
<body>
  <form id="login-form">
    <label for="email">Email:</label><br>
    <input type="text" id="email" name="email"><br>
```

```
    <label for="password">Password:</label><br>
    <input type="password" id="password" name="password"><br>
    <input type="submit" value="Submit">
  </form>
  <p id="message"></p>

  <script>
    const emailInput = document.getElementById('email');
    const passwordInput = document.getElementById('password');
    const message = document.getElementById('message');

    const loginForm = document.getElementById('login-form');
    loginForm.addEventListener('submit', (event) => {
      event.preventDefault();

      const email = emailInput.value;
      const password = passwordInput.value;

      const emailRegex = /\S+@\S+\.\S+/;
      const emailValid = emailRegex.test(email);
      const passwordValid = password.length >= 8;

      if (emailValid && passwordValid) {
        message.textContent = 'Valid email and password!';
        message.classList.add('success');
        message.classList.remove('error');
      } else {
        message.textContent = 'Invalid email or password!';
        message.classList.add('error');
        message.classList.remove('success');
      }
    });
  </script>
</body>
</html>
```

Q 20. Dynamically adding list items to an ordered list.

Answer:-

```
<!DOCTYPE html>
<html>
<head>
  <title>JavaScript Dynamic List Example</title>
</head>
<body>
  <button id="add-button">Add Item</button>
  <ol id="list"></ol>

  <script>
    // Pre-existing array of list items
    const items = [
      "HTML",
      "CSS",
      "JavaScript",
      "React",
      "Angular",
      "Vue.js",
      "Node.js",
      "Express.js",
      "MongoDB",
      "SQL"
    ];

    // Button click event handler
    document.getElementById("add-button").addEventListener("click", ()
=> {
      // Check if any remaining items are available in the array
      if (items.length > 0) {
        // Add the next item to the list
        const list = document.getElementById("list");
        const newListItem = document.createElement("li");
```

```
        newListItem.textContent = items.shift(); // Remove the first item
from the array
        list.appendChild(newListItem);
      } else {
        // Display a message indicating that all items have been added
        alert("All items have been added!");
      }
    });
  </script>
</body>
</html>
```