

**CODEM BOOTCAMP TASK
WEEK – 2
JAVASCRIPT EXERCISES**

SUBMITED BY
Vasundhara P S
IT – III Year

GITHUB LINK : <https://github.com/vasujoe2005/CodemBootCamp.git>

Task 1 – Login Access System

Requirements :

- Create 3 buttons: Admin, User, Guest
- Create validateLogin(username, password)
- On button click:
- Call validateLogin()
- If valid:
 - admin → "Full access granted"
 - user → "Limited access granted"
 - guest → "Read-only access"
- Else → "Invalid credentials"

PROGRAM:

```
<!DOCTYPE html>
<html>
<head>
    <title>Login Access System</title>
    <script>
        var selectedRole="";
        function showLogin(role){
            selectedRole=role;
            document.getElementById("loginForm").style.display="block";
            document.getElementById("result").innerHTML="";
        }
        function validateLogin(){
            var username=document.getElementById("username").value;
            var password=document.getElementById("password").value;
            var message=document.getElementById("result");

            if(selectedRole=="admin" && username=="admin" && password=="123") {
                message.innerHTML="Full access granted";
            }
            else if(selectedRole=="user" && username=="user" && password=="123") {
                message.innerHTML="Limited access granted";
            }
            else if(selectedRole=="guest" && username=="guest" && password=="123") {
                message.innerHTML="Read-only access";
            }
            else{
                message.innerHTML="Invalid credentials";
            }
        }
    </script>
</head>
```

```

<body>
  <h2>Login Access System</h2>
  <!-- Role Buttons -->
  <button onclick="showLogin('admin')">Admin</button>
  <button onclick="showLogin('user')">User</button>
  <button onclick="showLogin('guest')">Guest</button>
  <br><br>
  <!-- Hidden Login Form -->
  <div id="loginForm" style="display:none;">
    Username: <input type="text" id="username"><br><br>
    Password: <input type="password" id="password"><br><br>
    <button onclick="validateLogin()">Login</button>
  </div>
  <h3 id="result"></h3>
</body>
</html>

```

OUTPUT:

The figure consists of six screenshots of a web browser window titled "Login Access System" at the URL `127.0.0.1:5500/task1.html`. Each screenshot shows a different user role or access level.

- Screenshot 1 (Top Left):** Shows the initial page with three role buttons: Admin, User, and Guest.
- Screenshot 2 (Top Middle):** Shows the page after clicking the Admin button. It displays a login form with fields for Username (admin) and Password, and a "Login" button. The status bar says "You are signed in as 23". Below the form, the message "Full access granted" is displayed.
- Screenshot 3 (Top Right):** Shows the page after clicking the User button. It displays a login form with fields for Username (user) and Password, and a "Login" button. The status bar says "You are signed in as 23". Below the form, the message "Limited access granted" is displayed.
- Screenshot 4 (Bottom Left):** Shows the page after clicking the Guest button. It displays a login form with fields for Username (guest) and Password, and a "Login" button. The status bar says "You are signed in as 23". Below the form, the message "Read-only access" is displayed.
- Screenshot 5 (Bottom Middle):** Shows the page after entering invalid credentials (Username: adminuser, Password: ...). It displays a login form and the message "Invalid credentials".
- Screenshot 6 (Bottom Right):** Shows the page after entering valid credentials (Username: admin, Password: ...). It displays a login form and the message "Full access granted".

Task 2 – Movie Ticket Permission

Requirements:

- Create 3 buttons: Adult, Teen, Child
- Each button calls `checkTicket(age, hasPermission)`
- Rules:
 - Age 13–17 → allowed only if `hasPermission == true`
 - Others → allowed normally
 - Display Allowed / Not Allowed

PROGRAM:

```
<!DOCTYPE html>
<html>
<head>
    <title>Dynamic Ticket Checker</title>
</head>
<body>
    <h2>Ticket Eligibility</h2>
    <label>Enter Age:</label>
    <input type="number" id="ageInput" placeholder="Enter age">
    <br><br>
    <label>Has Permission?</label>
    <select id="permissionInput">
        <option value="true">Yes</option>
        <option value="false">No</option>
    </select>
    <br><br>
    <button onclick="handleCheck()">Check Ticket</button>
    <h3 id="result"></h3>
    <script>
        function checkTicket(age, hasPermission) {
            if (age >= 13 && age <= 17) {
                return hasPermission==true ? "Allowed":"Not Allowed";
            } else {
                return "Allowed";
            }
        }
        function handleCheck() {
            const age=parseInt(document.getElementById("ageInput").value);
            const hasPermission=document.getElementById("permissionInput").value=="true";
            const result=checkTicket(age, hasPermission);

            document.getElementById("result").innerText=result;
        }
    </script>
</body>
</html>
```

OUTPUT:

The figure consists of three side-by-side screenshots of a web browser window. The title bar of the browser says 'Dynamic Ticket Checker'. The address bar shows the URL '127.0.0.1:5500/task2.html'. The page content is a form with the following fields:

- Ticket Eligibility**
- Enter Age:**
- Has Permission?** Yes No
- Check Ticket** (button)

In the first screenshot, the age field is empty. In the second screenshot, the age field contains '42' and the result below it says 'Allowed'. In the third screenshot, the age field contains '8' and the result below it says 'Not Allowed'.

Ticket Eligibility

Enter Age:

Has Permission?

Allowed

Ticket Eligibility

Enter Age:

Has Permission?

Not Allowed

Task 3 – Online Order Status

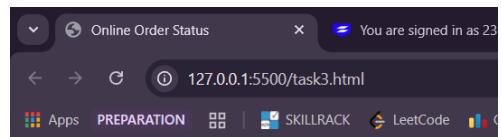
Requirements :

1. Create checkOrder(isPaid, inStock)
2. Conditions:
3. Paid + In stock → Order confirmed
4. Paid + Out of stock → Order delayed
5. Not paid → Payment required
6. Show result using alert()

PROGRAM:

```
<!DOCTYPE html>
<html>
<head>
    <title>Online Order Status</title>
</head>
<body>
    <h2>Check Order Status</h2>
    <button onclick="checkOrder(true, true)">Paid + In Stock</button>
    <button onclick="checkOrder(true, false)">Paid + Out of Stock</button>
    <button onclick="checkOrder(false, true)">Not Paid</button>
    <script>
        function checkOrder(isPaid, inStock){
            if(isPaid && inStock) {
                alert("Order Confirmed");
            }
            else if(isPaid && !inStock) {
                alert("Order Delayed");
            }
            else{
                alert("Payment Required");
            }
        }
    </script>
</body>
</html>
```

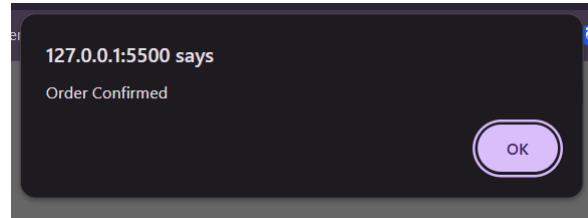
OUTPUT:



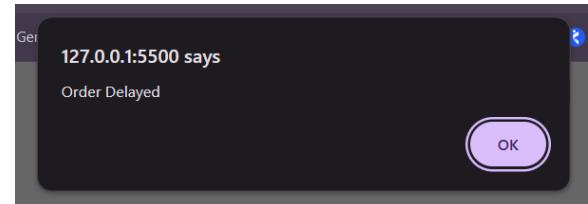
Check Order Status

[Paid + In Stock | Paid + Out of Stock | Not Paid]

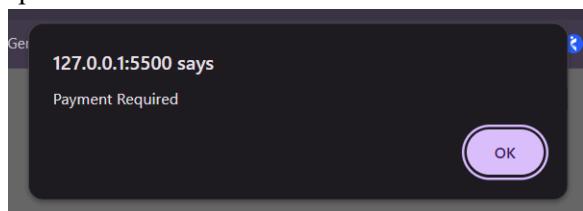
1. When Paid + In Stock button pressed:



2. When Paid + Out Of Stock button pressed:



3. When Not Paid button pressed:



Task 4 – Coin Toss Game

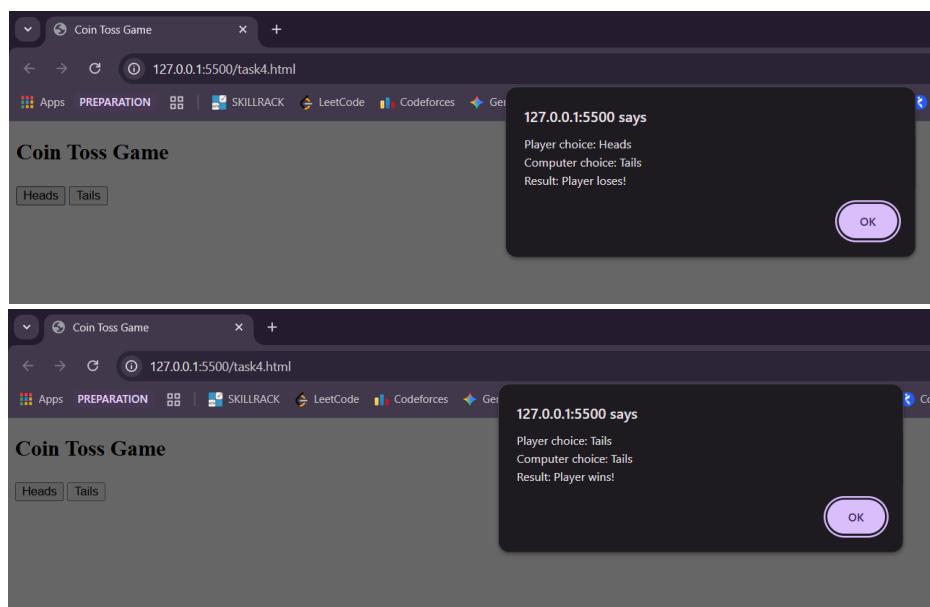
Requirements:

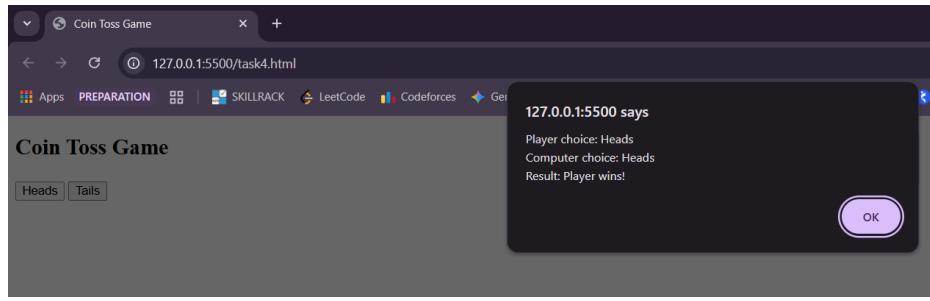
- Create Heads and Tails buttons
- On click:
 - Pass player choice to function
 - Generate random computer choice
- If both match → Player wins
- Else → Player loses
- Show using alert():
 - Player choice
 - Computer choice
 - Result

PROGRAM:

```
<!DOCTYPE html>
<html>
<head>
    <title>Coin Toss Game</title>
    <script>
        function tossCoin(playerChoice) {
            var randomNumber=Math.random();
            var computerChoice;
            if(randomNumber < 0.5) {
                computerChoice="Heads";
            }else{
                computerChoice="Tails";
            }
            var result;
            if(playerChoice==computerChoice) {
                result="Player wins!";
            }else{
                result="Player loses!";
            }
            alert(
                "Player choice: " + playerChoice +
                "\nComputer choice: " + computerChoice +
                "\nResult: " + result
            );
        }
    </script>
</head>
<body>
    <h2>Coin Toss Game</h2>
    <button onclick="tossCoin('Heads')">Heads</button>
    <button onclick="tossCoin('Tails')">Tails</button>
</body>
</html>
```

OUTPUT:





Task 5 – Order & Discount Processor

Requirements :

- Create three buttons to place Small, Medium, and Large orders.
- Each button calls processOrder(orderAmount, isMember, couponCode)
- Create processOrder():
- Validate orderAmount (must be a number > 0).
- Apply discounts:
 - ≥ 3000 or member $\rightarrow 20\%$
 - $\geq 1500 + \text{"SAVE10"} \rightarrow 10\%$
 - Empty coupon $\rightarrow 0\%$
 - Else $\rightarrow 5\%$
- Apply free delivery for final amount ≥ 2000 , else add ₹100.
- Create validateCoupon(couponCode):
- Use console.log() to display order amount, discount %, final amount, delivery status, and coupon validity.

PROGRAM:

```
<!DOCTYPE html>
<html>
<head>
    <title>Order & Discount Processor</title>
    <script>
        function validateCoupon(couponCode) {
            return couponCode=="SAVE10";
        }
        function processOrder() {
            var orderAmount=Number(document.getElementById("amount").value);
            var isMember=document.getElementById("member").checked;
            var couponCode=document.getElementById("coupon").value;
            if(isNaN(orderAmount) || orderAmount<=0) {
                console.log("Invalid order amount");
                return;
            }
            var orderType;
            if(orderAmount < 1500) {
                orderType="Small Order";
            }
            else if(orderAmount < 3000) {
                orderType="Medium Order";
            }
            else{
                orderType="Large Order";
            }
        }
    </script>
</head>
<body>
    <input type="checkbox" id="member" checked="checked"/> Is Member?
    <input type="text" id="amount" value="1000" /> Order Amount
    <input type="text" id="coupon" value="SAVE10" /> Coupon Code
    <input type="button" value="Place Order" onclick="processOrder()"/>
</body>
</html>
```

```

var discount=0;
if(orderAmount>=3000 || isMember==true) {
    discount=20;
}
else if(orderAmount>=1500 && validateCoupon(couponCode)) {
    discount=10;
}
else if(couponCode=="") {
    discount=0;
}
else{
    discount=5;
}
var finalAmount=orderAmount-(orderAmount * discount / 100);
var deliveryStatus;
if(finalAmount>=2000) {
    deliveryStatus="Free Delivery";
}else{
    finalAmount=finalAmount+100;
    deliveryStatus="₹100 Delivery Charge Added";
}
console.log("Order Type:", orderType);
console.log("Order Amount: ₹"+orderAmount);
console.log("Member Status:", isMember ? "Member" : "Not a Member");
console.log("Delivery Status:", deliveryStatus);
console.log("Discount Applied: "+discount+"%");
console.log("Final Amount: ₹"+finalAmount);
console.log("Coupon Valid:", validateCoupon(couponCode));
}
</script>
</head>
<body>
<h2>Order & Discount Processor</h2>
Enter Order Amount: <input type="number" id="amount"><br><br>
Member: <input type="checkbox" id="member"> <br><br>
Coupon Code: <input type="text" id="coupon"><br><br>
<button onclick="processOrder()">Place Order</button>
</body>
</html>

```

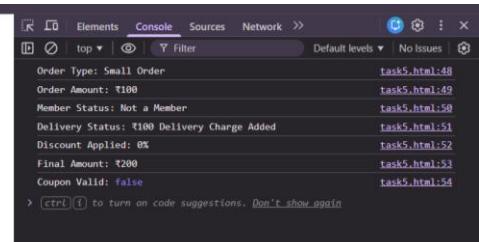
OUTPUT:

Order & Discount Processor

Enter Order Amount:

Member:

Coupon Code:

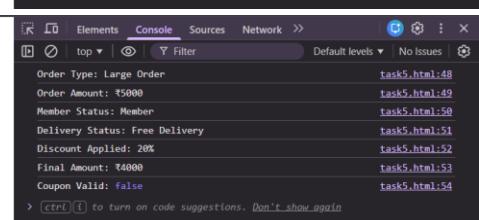


Order & Discount Processor

Enter Order Amount:

Member:

Coupon Code:



```

Order & Discount Processor
Enter Order Amount: 2000
Member: 
Coupon Code: SAVE10
Place Order

Order Type: Medium Order          task5.html:48
Order Amount: 2000                task5.html:49
Member Status: Member            task5.html:50
Delivery Status: T100 Delivery Charge Added task5.html:51
Discount Applied: 20%             task5.html:52
Final Amount: 1700                task5.html:53
Coupon Valid: true               task5.html:54

> [ctrl] [i] to turn on code suggestions. Don't show again

```

Task 6 – Simple Billing System

Requirements :

- Create Buttons: Regular, Member
- Call calculateBill(billAmount, isMember)
- If $\text{billAmount} \leq 0 \rightarrow \text{log error \& stop}$
- Discounts:
 - Member $\rightarrow 10\%$
 - Bill $\geq 1000 \rightarrow 5\%$
 - Else $\rightarrow 0\%$
- Gift:
 - Final $\geq 1500 \rightarrow \text{Free gift}$
- Display all details

PROGRAM:

```

<!DOCTYPE html>
<html>
<head>
  <title>Simple Billing System</title>
  <script>

    function calculateBill(isMember) {
      var billAmount=Number(document.getElementById("bill").value);
      if(isNaN(billAmount) || billAmount<=0) {
        console.log("Error: Invalid bill amount");
        return;
      }
      var discountPercent=0;
      if(isMember==true) {
        discountPercent=10;
      }
      else if(billAmount>=1000) {
        discountPercent=5;
      }
      else{
        discountPercent=0;
      }
      var discountAmount=(billAmount * discountPercent) / 100;
      var finalAmount=billAmount-discountAmount;
      var giftStatus;
      if(finalAmount>=1500) {
        giftStatus="Free Gift Eligible";
      }else{
        giftStatus="No Free Gift";
      }
    }
  </script>
</head>
<body>
  <input type="text" id="bill" value="2000" />
  <input checked="" type="checkbox" /> Member
  <input type="text" value="SAVE10" />
  <input type="button" value="Place Order" />
  <div>Final Amount: <span id="finalAmount"></span></div>
  <div>Gift Status: <span id="giftStatus"></span></div>
</body>

```

```

        console.log("Bill Amount: ₹"+billAmount);
        console.log("Customer Type:", isMember ? "Member" : "Regular");
        console.log("Discount Applied: "+discountPercent+"%");
        console.log("Discount Amount: ₹"+discountAmount);
        console.log("Final Amount: ₹"+finalAmount);
        console.log("Gift Status:", giftStatus);

    }
</script>
</head>
<body>
    <h2>Simple Billing System</h2>
    Enter Bill Amount: <input type="number" id="bill"><br><br>
    <button onclick="calculateBill(false)">Regular</button>
    <button onclick="calculateBill(true)">Member</button>
</body>
</html>

```

OUTPUT:

Simple Billing System

Enter Bill Amount:

Bill Amount: ₹5000
Customer Type: Member
Discount Applied: 10%
Discount Amount: ₹500
Final Amount: ₹4500
Gift Status: Free Gift Eligible

Simple Billing System

Enter Bill Amount:

Bill Amount: ₹5000
Customer Type: Regular
Discount Applied: 5%
Discount Amount: ₹250
Final Amount: ₹4750
Gift Status: Free Gift Eligible

Simple Billing System

Enter Bill Amount:

Bill Amount: ₹1500
Customer Type: Regular
Discount Applied: 5%
Discount Amount: ₹75
Final Amount: ₹1425
Gift Status: No Free Gift

Task 7 – Travel Eligibility Checker

Requirements :

- Create checkTravelEligibility(age, ticketType, hasID).
- If age $\leq 0 \rightarrow$ "Invalid age" and return.
- If ticketType not "normal" or "priority" \rightarrow "Invalid ticket type" and return.
- Eligibility:
 - age ≥ 18 and hasID \rightarrow Eligible
 - age < 18 and "priority" ticket \rightarrow Eligible with guardian
 - Else \rightarrow Not eligible
- Fare:
 - Base fare = ₹200
 - "priority" \rightarrow add ₹50
- Log eligibility, fare, ticket type, age check

PROGRAM:

```
<!DOCTYPE html>
<html>
<head>
    <title>Travel Eligibility Checker</title>
    <script>
        function checkTravelEligibility() {
            var age=Number(document.getElementById("age").value);
            var ticketType=document.getElementById("ticketType").value.toLowerCase();
            var hasID=document.getElementById("hasID").checked;
            if(age<=0 || isNaN(age)) {
                console.log("Invalid age");
                return;
            }
            if(ticketType !== "normal" && ticketType !== "priority") {
                console.log("Invalid ticket type");
                return;
            }
            var eligibility;
            if(age>=18 && hasID) {
                eligibility="Eligible";
            }
            else if(age < 18 && ticketType==="priority") {
                eligibility="Eligible with Guardian";
            }
            else{
                eligibility="Not Eligible";
            }
            var fare=200;

            if(ticketType==="priority") {
                fare += 50;
            }
            console.log("Age:", age);
            console.log("Ticket Type:", ticketType);
            console.log("Has ID:", hasID ? "Yes" : "No");
            console.log("Eligibility:", eligibility);
            console.log("Total Fare: ₹"+fare);
        }
    </script>
</head>
<body>
    <h2>Travel Eligibility Checker</h2>
    Enter Age: <input type="number" id="age"><br><br>
    Ticket Type:
    <select id="ticketType">
        <option value="normal">Normal</option>
        <option value="priority">Priority</option>
    </select>
    <br><br>
    Has Valid ID:<input type="checkbox" id="hasID"> <br><br>
    <button onclick="checkTravelEligibility()">Check Eligibility</button>
</body>
</html>
```

OUTPUT:

The figure displays four separate browser windows, each showing a "Travel Eligibility Checker" form and its associated JavaScript console output.

- User Input 1:** Age: 17, Ticket Type: Normal, Has Valid ID: checked. Console log:
 - Age: 17
 - Ticket Type: normal
 - Has ID: Yes
 - Eligibility: Not Eligible
 - Total Fare: £200
- User Input 2:** Age: 17, Ticket Type: Priority, Has Valid ID: checked. Console log:
 - Age: 17
 - Ticket Type: priority
 - Has ID: Yes
 - Eligibility: Eligible with Guardian
 - Total Fare: £250
- User Input 3:** Age: 50, Ticket Type: Normal, Has Valid ID: unchecked. Console log:
 - Age: 50
 - Ticket Type: normal
 - Has ID: No
 - Eligibility: Not Eligible
 - Total Fare: £200
- User Input 4:** Age: 28, Ticket Type: Normal, Has Valid ID: checked. Console log:
 - Age: 28
 - Ticket Type: normal
 - Has ID: Yes
 - Eligibility: Eligible
 - Total Fare: £200

Task 8 – Message Priority Formatter

Requirements:

- Create processMessage(message, isUrgent).
- Validate: If message is empty → "Message cannot be empty" and return.
- Create booleans:
 - hasWarningWord → message includes "error" or "fail".
 - isHighPriority → isUrgent OR hasWarningWord.
- Message formatting:
 - If high priority → uppercase + " IMPORTANT: "
 - Else → lowercase + "INFO: "
- Display: Final message and High-priority status

PROGRAM:

```
<!DOCTYPE html>
<html>
<head>
    <title>Message Priority Formatter</title>
    <script>
        function processMessage() {
            var message=document.getElementById("message").value;
            var isUrgent=document.getElementById("urgent").checked;
            if(message.trim() === "") {
                console.log("Message cannot be empty");
                return;
            }
        }
    </script>

```

```

var lowerMessage=message.toLowerCase();
var hasWarningWord=lowerMessage.includes("error")||lowerMessage.includes("fail");
var isHighPriority=isUrgent || hasWarningWord;
var finalMessage;
if(isHighPriority) {
    finalMessage="IMPORTANT: "+message.toUpperCase();
} else{
    finalMessage="INFO: "+message.toLowerCase();
}
console.log("Original Message:", message);
console.log("Urgent:", isUrgent ? "Yes" : "No");
console.log("Contains Warning Word:", hasWarningWord);
console.log("High Priority:", isHighPriority ? "Yes" : "No");
console.log("Final Message:", finalMessage);
}

</script>
</head>
<body>
<h2>Message Priority Formatter</h2>
Enter Message:<br><textarea id="message" rows="4" cols="40"></textarea>
<br><br>
Mark as Urgent:<input type="checkbox" id="urgent">
<br><br>
<button onclick="processMessage()">Process Message</button>
</body>
</html>

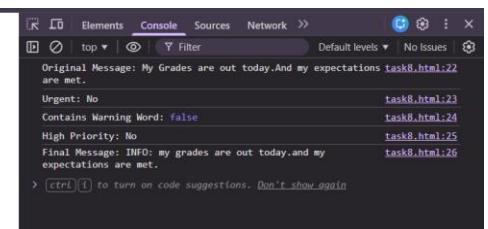
```

OUTPUT:

Message Priority Formatter

Enter Message:
My Grades are out today. And my expectations are met.

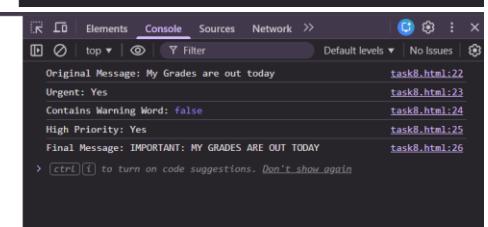
Mark as Urgent:



Message Priority Formatter

Enter Message:
My Grades are out today

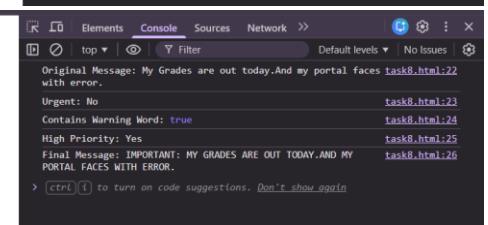
Mark as Urgent:



Message Priority Formatter

Enter Message:
My Grades are out today. And my portal faces with error.

Mark as Urgent:



Task 9 – Message Intent Detection

Requirements:

- Create a function detectIntent(message).
- Validate empty or spaces → Invalid
- Flags
 - isQuestion → message contains ?
 - isComplaint → message contains "problem" or "not working"
- Intent
 - Question + complaint → Urgent request
 - Complaint → Complaint
 - Question → Question
 - Else → Other
- Log intent and original message using console.log()

PROGRAM:

```
<!DOCTYPE html>
<html>
<head>
    <title>Message Intent Detection</title>
    <script>
        function detectIntent() {
            var message=document.getElementById("message").value;
            if(message.trim() === "") {
                console.log("Invalid message");
                return;
            }
            var lowerMessage=message.toLowerCase();
            var isQuestion=lowerMessage.includes "?";
            var isComplaint=
                lowerMessage.includes("problem") ||
                lowerMessage.includes("not working");

            var intent;
            if(isQuestion && isComplaint) {
                intent="Urgent Request";
            }
            else if(isComplaint) {
                intent="Complaint";
            }
            else if(isQuestion) {
                intent="Question";
            }
            else{
                intent="Other";
            }
            console.log("Original Message:", message);
            console.log("Intent:", intent);
        }
    </script>
</head>
```

```

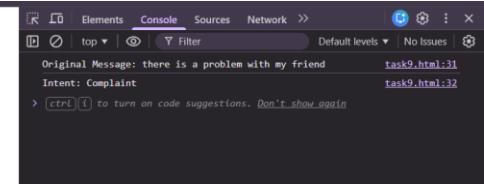
<body>
  <h2>Message Intent Detection</h2>
  Enter Message:<br><textarea id="message" rows="4" cols="40"></textarea>
  <br><br>
  <button onclick="detectIntent()">Detect Intent</button>
</body>
</html>

```

OUTPUT:

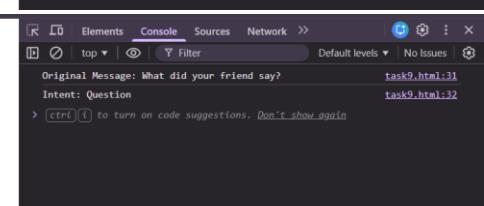
Message Intent Detection

Enter Message:



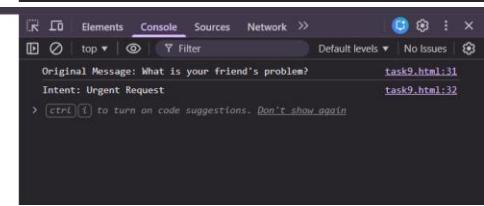
Message Intent Detection

Enter Message:



Message Intent Detection

Enter Message:



Task 10 – Account Security Check

Requirements:

- Check account access using password status, OTP verification, and account block status.
- Allow login only if all conditions are valid.
- Display login success or failure message.

PROGRAM:

```

<!DOCTYPE html>
<html>
<head>
  <title>Account Security Check</title>
  <script>
    function checkAccountSecurity() {
      var passwordStatus=document.getElementById("passwordStatus").value;
      var otpStatus=document.getElementById("otpStatus").value;
      var accountStatus=document.getElementById("accountStatus").value;
      var result=document.getElementById("result");
      if(passwordStatus==="correct" && otpStatus==="verified" && accountStatus==="active"){
        result.innerHTML="Login Successful";
      }
      else{
        result.innerHTML="Login Failed";
      }
    }
  </script>
</head>

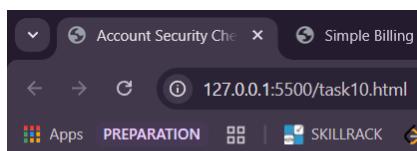
```

```

<body>
  <h2>Account Security Check</h2>
  Password Status:
  <select id="passwordStatus">
    <option value="correct">Correct</option>
    <option value="wrong">Wrong</option>
  </select>
  <br><br>
  OTP Verification:
  <select id="otpStatus">
    <option value="verified">Verified</option>
    <option value="notVerified">Not Verified</option>
  </select>
  <br><br>
  Account Status:
  <select id="accountStatus">
    <option value="active">Active</option>
    <option value="blocked">Blocked</option>
  </select>
  <br><br>
  <button onclick="checkAccountSecurity()">Login</button>
  <h3 id="result"></h3>
</body>
</html>

```

OUTPUT:



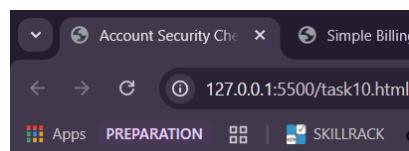
Account Security Check

Password Status:

OTP Verification:

Account Status:

Login Successful



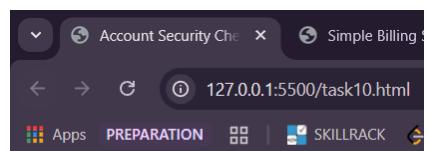
Account Security Check

Password Status:

OTP Verification:

Account Status:

Login Failed



Account Security Check

Password Status:

OTP Verification:

Account Status:

Login Failed

Task 11 – Weather Action Advisor

Requirements:

- Check weather condition (rainy/sunny/etc.) and temperature.
- Suggest an action (carry umbrella, stay hydrated, etc.).
- Display the suggestion using alert().

PROGRAM:

```
<!DOCTYPE html>
<html>
<head>
    <title>Weather Action Advisor</title>
    <script>
        function checkWeather(){
            var weather=document.getElementById("weather").value.toLowerCase();
            var temperature=Number(document.getElementById("temp").value);
            var suggestion="";
            if(weather==="rainy") {
                suggestion="Carry an umbrella.";
            }
            else if(weather==="sunny") {
                suggestion="Wear sunglasses.";
            }
            else if(weather==="cold") {
                suggestion="Wear warm clothes.";
            }
            else{
                suggestion="Have a good day.";
            }
            if(temperature > 35) {
                suggestion += "Stay hydrated.";
            }
            else if(temperature < 15) {
                suggestion += "Keep yourself warm.";
            }
            else{
                suggestion += "Weather looks comfortable.";
            }
            alert("Weather: "+weather+"\nTemperature: "+temperature+"°C"+`\nSuggestion: "+suggestion);
        }
    </script>
</head>
<body>
    <h2>Weather Action Advisor</h2>
    Weather Condition:
    <select id="weather">
        <option value="sunny">Sunny</option>
        <option value="rainy">Rainy</option>
        <option value="cold">Cold</option>
        <option value="cloudy">Cloudy</option>
    </select>
    <br><br>
```

Temperature (°C):

```
<input type="number" id="temp">
<br><br>
<button onclick="checkWeather()">Get Advice</button>
</body>
</html>
```

OUTPUT:

The screenshots show four separate runs of the "Weather Action Advisor" application, each with different input values and resulting output messages.

- Screenshot 1 (Top):** Weather Condition: Sunny, Temperature (°C): 25. The message says: "127.0.0.1:5500 says Weather: sunny Temperature: 25°C Suggestion: Wear sunglasses. Weather looks comfortable." An "OK" button is visible.
- Screenshot 2 (Second from top):** Weather Condition: Cold, Temperature (°C): 10. The message says: "127.0.0.1:5500 says Weather: cold Temperature: 10°C Suggestion: Wear warm clothes. Keep yourself warm." An "OK" button is visible.
- Screenshot 3 (Third from top):** Weather Condition: Rainy, Temperature (°C): 12. The message says: "127.0.0.1:5500 says Weather: rainy Temperature: 12°C Suggestion: Carry an umbrella. Keep yourself warm." An "OK" button is visible.
- Screenshot 4 (Bottom):** Weather Condition: Cloudy, Temperature (°C): 25. The message says: "127.0.0.1:5500 says Weather: cloudy Temperature: 25°C Suggestion: Have a good day. Weather looks comfortable." An "OK" button is visible.

Task 12 – Password Strength Validator

Requirements:

- Create a function to check password content (length, characters).
- Classify password as Weak, Medium, or Strong.
- Display the strength result.

PROGRAM:

```
<!DOCTYPE html>
<html>
<head>
    <title>Password Strength Validator</title>
    <script>
        function checkPassword() {
            var password = document.getElementById("password").value;
            var result = document.getElementById("result");
            var lengthCheck = password.length >= 8;
            var upperCheck = /[A-Z]/.test(password);
            var lowerCheck = /[a-z]/.test(password);
            var numberCheck = /[0-9]/.test(password);
            var specialCheck = /[!@#$%^&*]/.test(password);
            var noSpaceCheck = /\s/.test(password);
            var score = 0;
            if (lengthCheck) score++;
            if (upperCheck) score++;
            if (lowerCheck) score++;
            if (numberCheck) score++;
            if (specialCheck) score++;
            if (noSpaceCheck) score++;
            var strength;
            if (score <= 2) {
                strength = "Weak";
                result.style.color = "red";
            }
            else if (score <= 5) {
                strength = "Medium";
                result.style.color = "orange";
            }
            else {
                strength = "Strong";
                result.style.color = "green";
            }
            result.innerHTML = "Password Strength: " + strength;
        }
    </script>
</head>
<body>
    <h2>Password Strength Validator</h2>
    Enter Password:
    <input type="text" id="password" onkeyup="checkPassword()">
    <br><br>
    <h3 id="result"></h3>
</body>
</html>
```

OUTPUT:

The figure consists of three side-by-side screenshots of a web browser window. Each window has a title bar 'Password Strength Validator' and a URL bar '127.0.0.1:5500/task12.html'.
1. In the first screenshot, the input field contains 'vasu'. Below it, the text 'Password Strength: Weak' is displayed in red.
2. In the second screenshot, the input field contains 'VasuJoe@77'. Below it, the text 'Password Strength: Medium' is displayed in orange.
3. In the third screenshot, the input field contains 'VasuJoe@77'. Below it, the text 'Password Strength: Strong' is displayed in green.

Task 13 – Appointment Confirmation System

Requirements:

- Create checkAppointment(dateAvailable, userConfirmed).
 - If dateAvailable is false → "Appointment Rejected"
 - Else if userConfirmed is false → "Appointment Rejected"
 - Else → "Appointment Confirmed"
- Display Confirmed or Rejected status using alert().

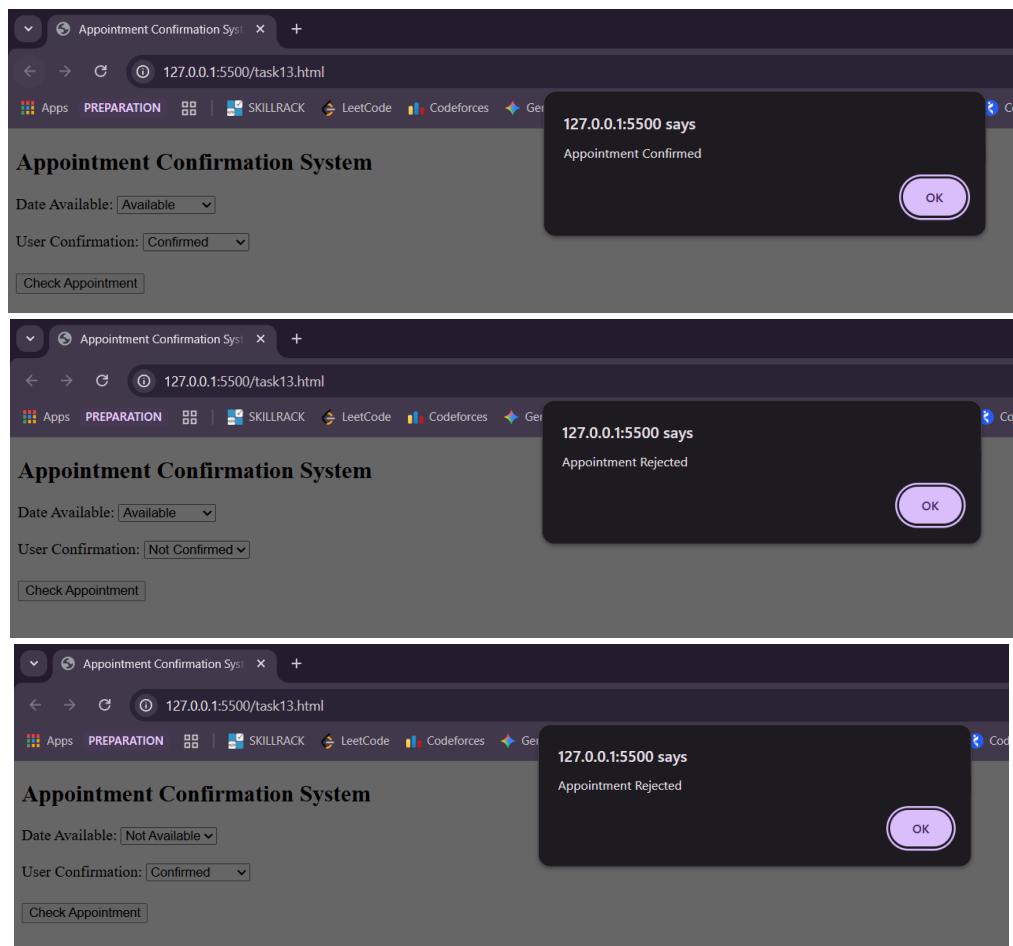
PROGRAM:

```
<!DOCTYPE html>
<html>
<head>
    <title>Appointment Confirmation System</title>
    <script>
        function checkAppointment(){
            var dateAvailable=document.getElementById("dateAvailable").value;
            var userConfirmed=document.getElementById("userConfirmed").value;
            if(dateAvailable==="false") {
                alert("Appointment Rejected");
            }
            else if(userConfirmed==="false") {
                alert("Appointment Rejected");
            }
            else{
                alert("Appointment Confirmed");
            }
        }
    </script>
</head>
<body>
    <h2>Appointment Confirmation System</h2>
    Date Available:
    <select id="dateAvailable">
        <option value="true">Available</option>
        <option value="false">Not Available</option>
    </select>
    <br><br>
```

User Confirmation:

```
<select id="userConfirmed">
    <option value="true">Confirmed</option>
    <option value="false">Not Confirmed</option>
</select>
<br><br>
<button onclick="checkAppointment()">Check Appointment</button>
</body>
</html>
```

OUTPUT:



Task 14 – Notification Filter

Requirements:

- Check notification message content and urgency flag.
- Classify notification as Important or Normal.
- Display notification type with formatted message.

PROGRAM:

```
<!DOCTYPE html>
<html>
<head>
    <title>Notification Filter</title>
    <script>
        function filterNotification() {
            var message=document.getElementById("message").value;
            var isUrgent=document.getElementById("urgent").checked;
            var result=document.getElementById("result");
            var lowerMessage=message.toLowerCase();
            var hasImportantWord=lowerMessage.includes("error")||lowerMessage.includes("alert")||lowerMessage.includes("urgent");
            var isImportant=isUrgent||hasImportantWord;
            var formattedMessage;
            if(isImportant){
                formattedMessage="IMPORTANT: "+message.toUpperCase();
            }else{
                formattedMessage="NORMAL: "+message;
            }
            result.innerHTML ="<strong>Notification Type:</strong> "+(isImportant ? "Important" :
            "Normal")+"<br><br>"+formattedMessage;
        }
    </script>
</head>
<body>
    <h2>Notification Filter</h2>
    Enter Notification Message:<br>
    <textarea id="message" rows="4" cols="40"></textarea>
    <br><br>
    Mark as Urgent:
    <input type="checkbox" id="urgent">
    <br><br>
    <button onclick="filterNotification()">Check Notification</button>
    <br><br>
    <div id="result"></div>
</body>
</html>
```

OUTPUT:

Notification Filter

Enter Notification Message:
this is an error message

Mark as Urgent:

Notification Type: Important

IMPORTANT: THIS IS AN ERROR MESSAGE

Notification Filter

Enter Notification Message:
this is a happy message

Mark as Urgent:

Notification Type: Normal

NORMAL: this is a happy message

Notification Filter

Enter Notification Message:
this is a sad message

Mark as Urgent:

Notification Type: Important

IMPORTANT: THIS IS A SAD MESSAGE

Task 15 – Payment Status Checker

Requirements:

- Create checkPaymentStatus(amount, isPaid, paymentMethod).
- If amount $\leq 0 \rightarrow$ "Invalid payment amount"
- Else if isPaid is false \rightarrow "Payment Pending"
- Else if paymentMethod is "UPI" or "Card" \rightarrow "Payment Successful"
- Else \rightarrow "Invalid payment method"
- Display: Payment status , Amount and Payment Method

PROGRAM:

```
<!DOCTYPE html>
<html>
<head>
    <title>Payment Status Checker</title>
    <script>
        function checkPaymentStatus(){
            var amount=Number(document.getElementById("amount").value);
            var isPaid=document.getElementById("isPaid").value;
            var paymentMethod=document.getElementById("method").value;
            var result=document.getElementById("result");
            var status;
            if(isNaN(amount)||amount <= 0){
                status="Invalid payment amount";
            }
            else if(isPaid==="false"){
                status="Payment Pending";
            }
            else if(paymentMethod==="UPI"||paymentMethod==="Card"){
                status="Payment Successful";
            }
            else{
                status="Invalid payment method";
            }
            result.innerHTML=<strong>Payment Status:</strong> "+status+"<br><strong>Amount:</strong>
            ₹"+amount+"<br><strong>Payment Method:</strong> "+paymentMethod;
        }
    </script>
</head>
<body>
    <h2>Payment Status Checker</h2>
    Enter Amount: <input type="number" id="amount"><br><br>
    Payment Done:
    <select id="isPaid">
        <option value="true">Yes</option>
        <option value="false">No</option>
    </select>
    <br><br>
    Payment Method:
    <select id="method">
        <option value="UPI">UPI</option>
        <option value="Card">Card</option>
        <option value="Cash">Cash</option>
    </select>
    <br><br>
```

```
<button onclick="checkPaymentStatus()">Check Payment</button>
<br><br>
<div id="result"></div>
</body>
</html>
```

OUTPUT:

Payment Status Checker

Enter Amount:

Payment Done:

Payment Method:

Payment Status: Payment Successful
Amount: ₹5000

Payment Method: UPI

Payment Status Checker

Enter Amount:

Payment Done:

Payment Method:

Payment Status: Invalid payment method

Amount: ₹5000

Payment Method: Cash

Payment Status Checker

Enter Amount:

Payment Done:

Payment Method:

Payment Status: Payment Pending

Amount: ₹5000

Payment Method: Card