

# Functions And IF Statements

## 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"**

## 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**

## 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()

## 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

## 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:
  - $\geq 3000$  or member → 20%
  - $\geq 1500$  + "SAVE10" → 10%
  - Empty coupon → 0%
  - Else → 5%

- Apply free delivery for final amount  $\geq 2000$ , else add ₹100.
- Create `validateCoupon(couponCode)`:
- Use `console.log()` to display order amount, discount %, final amount, delivery status, and coupon validity.

## Task 6 – Simple Billing System

Requirements :

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


## 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  $\geq 18$  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

## 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

## 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()

## 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.

### 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().

### 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.

### 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().

### Task 14 – Notification Filter

#### Requirements:

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

## Task 15 – Payment Status Checker

### Requirements:

Create `checkPaymentStatus(amount, isPaid, paymentMethod)`.

- If `amount <= 0` → "Invalid payment amount"
- Else if `isPaid` is false → "Payment Pending"
- Else if `paymentMethod` is "UPI" or "Card" → "Payment Successful"
- Else → "Invalid payment method"
- Display: Payment status , Amount and Payment Method