**Workflow Steps**

1. **User Signup/Login**:
   * User signs up or logs in on the React.js frontend.
   * React.js sends the signup/login request to the Node.js server.
   * Node.js interacts with Firebase Authentication to manage user accounts and sessions.
   * Firebase Authentication responds to Node.js, which then sends the response back to React.js.
   * User is authenticated and logged in on the frontend.
2. **Slot Reservation**:
   * Authenticated user selects a parking slot to reserve on the React.js frontend.
   * React.js sends the reservation request to the Node.js server.
   * Node.js checks the slot availability using sensor data from Firebase.
   * If the slot is available, Node.js updates the Firebase database with the reservation details.
   * Node.js sends the reservation confirmation back to React.js.
   * React.js displays the reservation confirmation to the user.
3. **Car Arrival**:
   * The car arrives within the buffer time (15 minutes).
   * The sensor detects the car's presence and sends the data to the Node.js server via the Arduino.
   * Node.js checks if the slot is reserved.
   * If the slot is reserved, Node.js updates the Firebase database with the car's arrival time and status.
   * If the slot is not reserved, Node.js updates the Firebase database to show the slot as full but not reserved (no ticket generation).
4. **Real-Time Updates**:
   * Node.js uses Socket.IO to send real-time updates to the React.js frontend.
   * React.js displays real-time slot status and updates to the user.
5. **Car Departure and Ticket Generation**:
   * The sensor detects the car's departure and sends the data to the Node.js server via the Arduino.
   * Node.js checks if the slot was reserved.
   * If the slot was reserved, Node.js calculates the parking duration based on the arrival and departure times.
   * Node.js generates a parking ticket based on the hours parked.
   * Node.js updates the Firebase database with the ticket details.
   * Node.js processes the payment using Stripe.
   * Stripe responds with the payment status.
   * Node.js updates the payment status in Firebase.
   * Node.js sends the ticket and payment confirmation back to React.js.
   * React.js displays the ticket and payment confirmation to the user.