Real-Time Order Monitoring & Status Update System

UI Reference Style: Swiggy/Zepto Order Tracking Page

Step 1 – Product Listing Page

Frontend page that fetches product list from backend.

Features:

- Fetch list of products (Name, Description, Price, Stock, Image URL).
- Search bar at the top (search by name or description).
- Display products as responsive cards:
    - Image
    - Name
    - Price
    - Stock indicator (Green = Available, Red = Out of Stock)
    - “Add to Cart” button
- Cart Drawer/Sidebar:
    - List selected products
    - Quantity + Price calculation
    - Remove button
    - Total amount
- If search returns no results → show *“No products found.”*

Step 2 – Checkout & Real-Time Order Creation

Once cart has at least 1 product, navigate to checkout page.

UI Requirements:

- Show cart review
- Choose payment mode: **COD / Online (dummy)**
- Collect:
    - Customer Name

    - Phone
    - Address
- On “Place Order”:
    - Send order data to backend
    - Backend should create the order
    - A **real-time socket event** should notify Admin dashboard:
        - `newOrderPlaced` with order details

Step 3 – Customer Live Order Status Page (Real-Time)

Once order is placed, show a **real-time updating** tracking page.

Status Stages:

1. Order Placed
2. Accepted by Admin
3. Packed
4. Out for Delivery
5. Delivered

Frontend Requirements:

- Status timeline component showing the stages
- Use Socket.IO to receive updates:
    - `orderStatusUpdated` event
- Update UI instantly without refresh
- Show an animation (pulsing dot) for current stage
- Show estimated delivery time (dummy value)

Step 4 – Admin Dashboard (Real-Time Panel)

Admin Orders List Page

- Fetch all orders from backend
- Highlight newly arrived orders in blue for 5 seconds
- Real-time updates using Socket.IO:

    - `newOrderPlaced` → prepend order
    - `orderStatusUpdated` → modify existing order in UI

Features:

- Filters: Today / All / Delivered / Pending
- Search by order number or customer phone
- Click on an order to open detailed view

Order Detail Page

- Customer info
- List of items
- Total
- Current status selector (dropdown):
    - Placed
    - Accepted
    - Packed
    - Out for Delivery
    - Delivered
- On status change:
    - Update backend
    - Emit socket event to customer: `orderStatusUpdated`

Step 5 – Admin Login + Role Management

(Skip complex auth — basic login is enough)

Features:

- Admin login page (username + password, hardcoded allowed)
- Manage roles:
    - **Admin** → Full access
    - **Staff** → Can only change order status, not edit products

Step 6 – Product Management (Admin Side)

- Add/Edit/Delete products

- Fields:
  - Name
  - Description
  - Price
  - Stock Quantity
  - Image Upload (optional)
- Validation: name required, price > 0, stock ≥ 0

Non-Functional Requirements

- UI should be clean and neat (reference: Zepto / Swiggy order tracking)
- Use reusable React components
- Handle loading states & empty states
- Prevent duplicate order creation
- Orders should update in real-time without page refresh
- Proper error handling and toast notifications