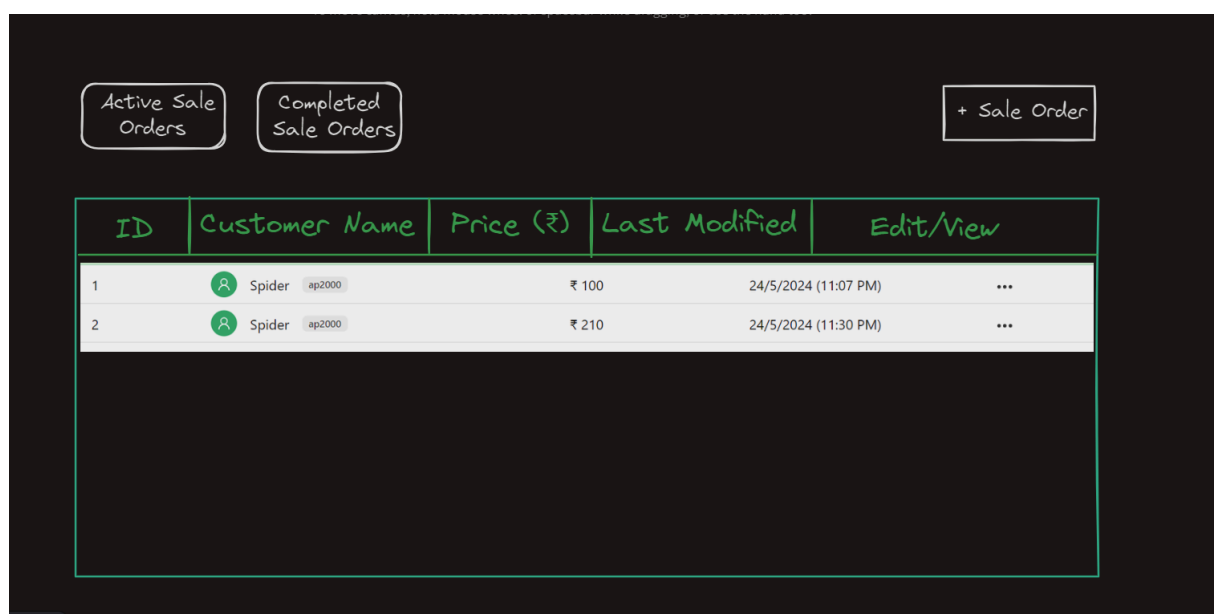


Background

A consumer goods manufacturing company has stocks in its product inventory. whenever it sells it to retailers or distributors, it needs to generate a sale order for record-keeping and compliance purposes. As a part of the assignment, you are required to build only the frontend part of the simple sale order management application. Below are all the specifications -

1. Single Page Application with a Login Page (dummy username and password). An unauthenticated user should be redirected to the login page
2. Dark Theme Toggle Switch. (The theme should persist on reloads)
3. One tab for the **Active Sale Orders**. Another For the **Completed Sale Orders**.
4. Below in the image, **+ Sale order** button is a modal trigger. The content inside the modal should be the form to **create a sale order**.
5. Below in the image, the active sale order row contains (**triple horizontal dots icon/ ...**), which is another **modal trigger**, for editing that active sale order. (Initially, the Form must be prefilled with the details). A similar pattern should be followed in the Completed Sale Order table, but the form will be **Read-Only**.



Wireframe for page(s)

All Products *

Product 22 ×

Product 5

Stocked Product I

Benoit Saint Denis

Anonymous Product

Stocked Tea I

Stocked Tea II

Enter selling rate

Enter Quantity

104 Item(s) Remaining

2. SKU 227 (234 Kg)

Rate: ₹ 324

Selling Rate

Total Items

Enter selling rate

Enter Quantity

46 Item(s) Remaining

UI In Sync w/ form state

Product Multi select Snippet of sale order form

Schema

1. CUSTOMER SCHEMA :

```
{
```

```
"id": 9,
"customer": 11908,
"customer_profile": {
  "id": 11908,
  "name": "Ram",
  "color": [
    182,
    73,
    99
  ],
  "email": "jesus_christ@church.com",
  "pincode": "Mumbai",
  "location_name": "Mumbai, Maharashtra, India",
  "type": "C",
  "profile_pic": null,
  "gst": ""
},
}
```

2. PRODUCT SCHEMA

```
{
  "id": 209,
  "display_id": 8,
  "owner": 1079,
  "name": "New Product",
  "category": "The god of War",
  "characteristics": "New Product Characteristics",
  "features": "",
  "brand": "New Product Brand",
  "sku": [
    {
      "id": 248,
      "selling_price": 54,
      "max_retail_price": 44,
```

```

        "amount": 33,
        "unit": "kg",
        "quantity_in_inventory": 0,
        "product": 209
    },
    {
        "id": 247,
        "selling_price": 32,
        "max_retail_price": 32,
        "amount": 33,
        "unit": "kg",
        "quantity_in_inventory": 0,
        "product": 209
    },
    {
        "id": 246,
        "selling_price": 23,
        "max_retail_price": 21,
        "amount": 22,
        "unit": "kg",
        "quantity_in_inventory": 1,
        "product": 209
    }
],
"updated_on": "2024-05-24T12:46:41.995873Z",
"adding_date": "2024-05-24T12:46:41.995828Z"
}

```

3. SALE ORDER FORM SCHEMA (Payload)

```

{
  "customer_id": 11908,
  "items": [

```

```
{
  "sku_id": 220,
  "price": 12,
  "quantity": 12
} ],
"paid": false,
"invoice_no": "Invoice - 1212121",
"invoice_date": "7/5/2024"
}
```

Guidelines:

1. Form should be equipped with validation rules. Once submitted, a new sale order must be created and the active sale order UI must be updated (synchronously without a refresh)
2. Use a Date Picker for invoice_date form field

Required tech stack:

1. React 18+ (with optimization hooks)
2. React Router DOM (latest version) for routing purposes.
3. Tanstack react query for managing server state. For all the events, the api calls should be mimicked.
4. React Hook Form for managing form state
5. Chakra-UI (component UI Library) for styling
6. Chakra MultiSelect (or any other that's compatible with Chakra-UI)

SUBMISSION GUIDELINES:

1. Video (Google Drive link) of the working application with all the major features well highlighted.
2. Github repository link (public).