Web Application Development Instructions

You are asked to develop this web application using .NET Core and React frontend. For the backend database, you may use SQL server. Once you finished the task, send the GitHub link of the database and solution. You are given only two days to complete the web application and you may send us the solution up to the stage you complete by the end of second day. Please follow the given instructions below.

1. Backend Architecture

The backend of the system should be developed using **.NET Core** following the recommended best practices:

1.1 Architecture Style

- **Layered (N-Tier) Architecture** or **Clean Architecture** is recommended.
- Separate core concerns into:
  - **API Layer** – Controllers, routing, authentication.
  - **Business Logic Layer (Services)** – Core application logic.
  - **Data Access Layer (Repository + Entity Framework Core)**.
  - **Domain Layer** – Entity models and interfaces.

1.2 Key Components

- **.NET Core Web API** for backend services.
- **Entity Framework Core** for ORM.
- **SQL Server** as the primary database.
- **DTOs and AutoMapper** for mapping entities to view models.
- **Dependency Injection** built into .NET Core for services and repositories.

1.3 Recommended Folder Structure

```
/ProjectName
  /API
    /Controllers
    /Models (DTOs)
  /Application
    /Interfaces
    /Services
  /Domain
    /Entities
  /Infrastructure
    /Data
    /Repositories
```

2. Frontend Architecture (React)

The frontend should be developed using **React** for UI and **Redux** for centralized state management.

2.1 Frontend Tech Stack

- **React (Functional Components)**
- **React Hooks**
- **Redux Toolkit** for state management
- **React Router** for navigation between **Home (Screen 2)** and **Sales Order (Screen 1)**
- **Axios / Fetch API** for backend communication
- **Tailwind CSS** for styling and layout (utility-first CSS framework)

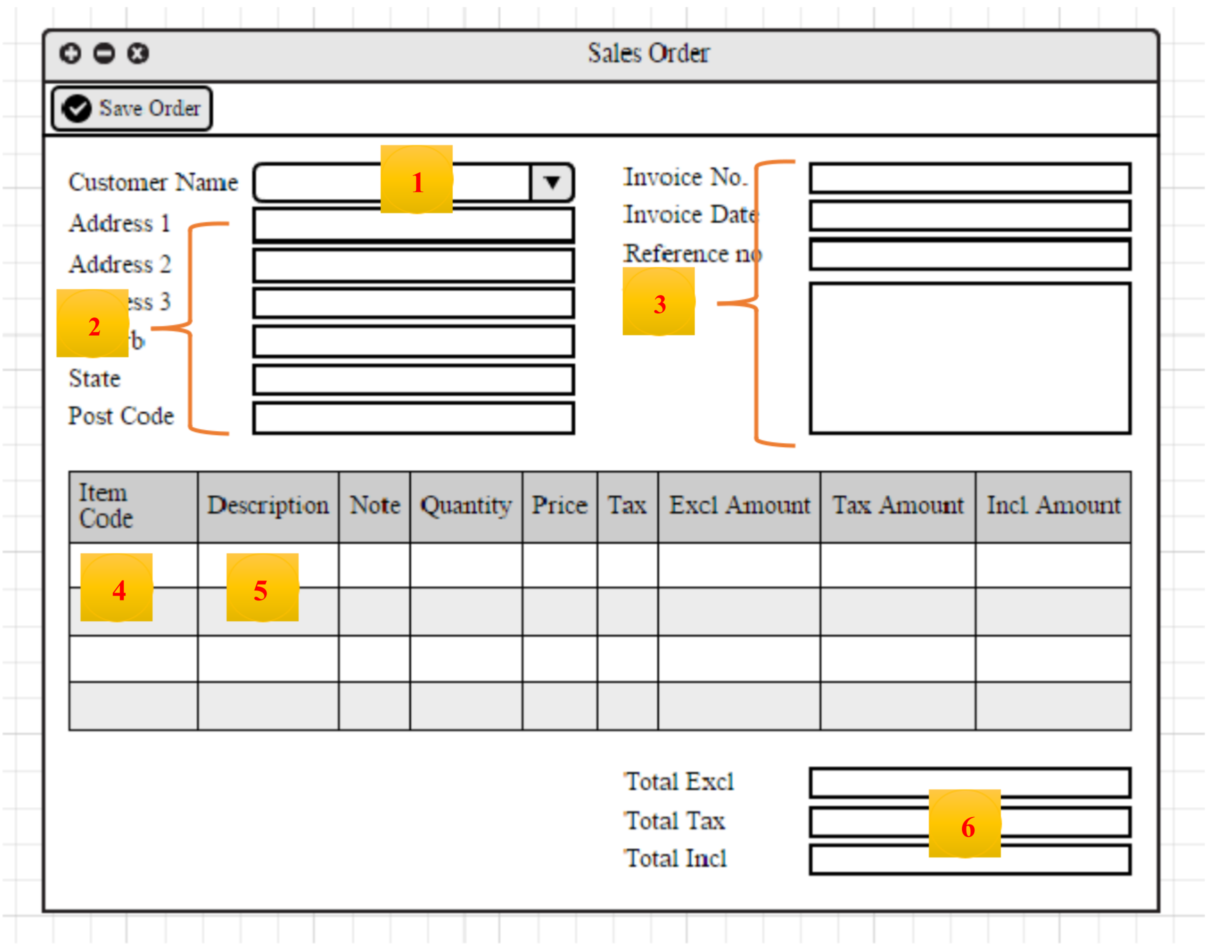
2.2 Tailwind CSS for Styling

- Use **Tailwind CSS** classes directly in JSX for fast UI development.
- Define a **consistent design system** (colors, spacing, fonts) via `tailwind.config.js`.
- Create **reusable components** for:
  - Form controls (inputs, dropdowns, buttons) used in **Screen 1**.
  - Tables/grids for listing orders in **Screen 2**.
- Ensure responsive design so both screens render properly on different screen sizes.
- Use utility classes for:
  - Layout: `flex`, `grid`, `gap-*`, `p-*`, `m-*`.
  - Typography: `text-sm`, `text-lg`, `font-semibold`.
  - Colors: `bg-gray-100`, `bg-white`, `text-gray-700`, `border-gray-300`.

2.3 Recommended Folder Structure

```
/src
  /components
  /pages
  /redux
    /slices
    store.js
  /services
  /hooks
  /utils
```

Screen 1



1. 'Customer name' dropdown menu should list down the customers (customer name) from 'Client' table in the database.

2. Once user selected a one customer, address fields should fill automatically according to the selected customers.

3. These fields can be filled (type) by user as wish.

4. Item code column should be dropdown menu and it should populate with item codes.

5. Description also should be dropdown menu and it should populate with item Description. User can select an item from Item code or from description. After that user can enter note, quantity and tax rate. Once user entered the quantity and tax rate you should calculate,

   Excl Amount = Quantity * Price

   Tax Amount = Excl Amount * Tax Rate /100

   Incl Amount = Excl Amount + Tax Amount

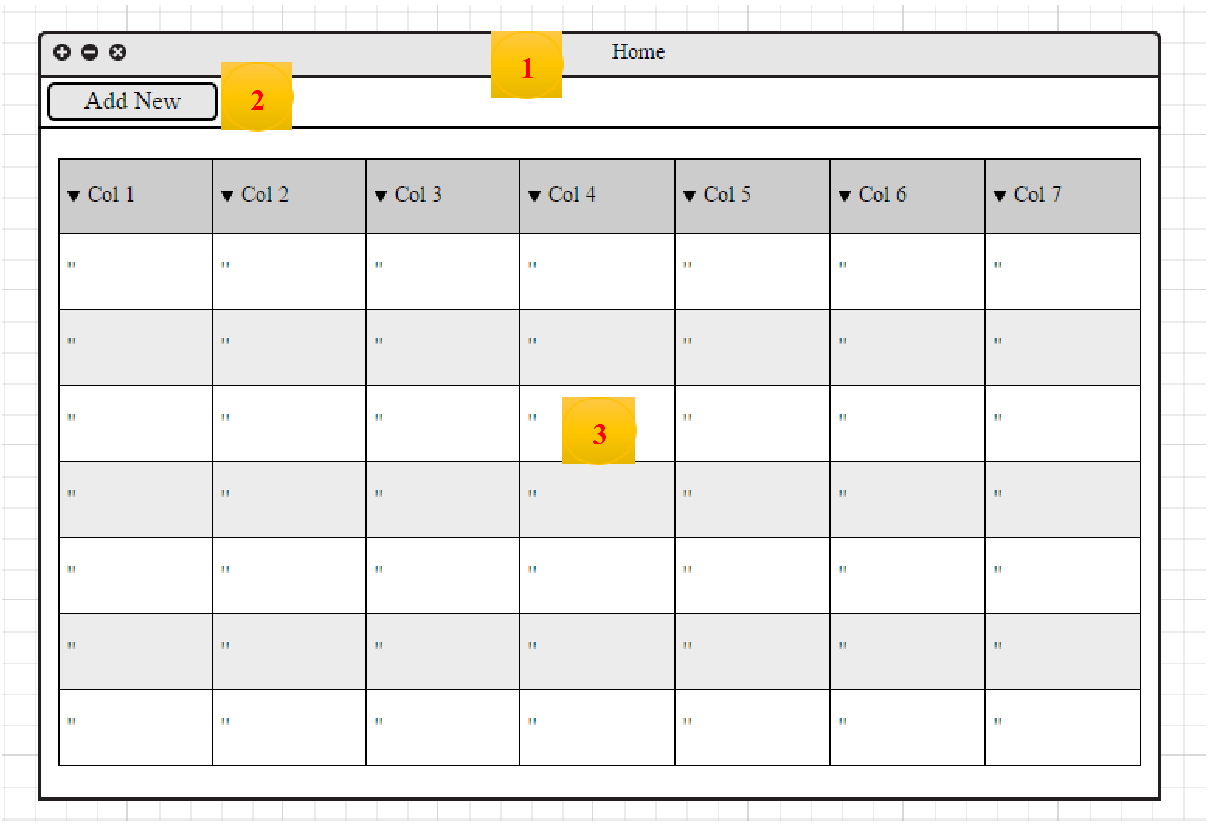
   Likewise, user can add multiples items for each order.

6. Here you should calculate each line total value and display in these fields.

7. You have to create required table(s) to save order details accordingly.

8. Once the order is saved, you can add print option to print each sales order (any reporting tools).

Screen 2



1. This home screen should be the first screen that open when we run the web application.

2. Once user clicks Add new button you can open the sales order screen (previous screen)

3. There should be list of added orders and you can define columns in the grid as you wish. User can also double click on a one order (row) you should open the screen 2 again with the added data. In addition, there should be option to edit the order and save again.