Project Overview

This project is built with **Next.js** using the **App Router**. It supports **multiple languages** through the `next-intl` package and handles **authentication/authorization** with `next-auth`.

Please review the attached folder structure before starting development.

Key Points to Note Before Development

1. **Table Components**
    - We already have two table components:
        - `TableBasic`
        - `TableFull`
    - Please make sure to reuse these instead of creating new table implementations.
2. **Query Input**
    - For general usage of query-based inputs, you should use the `QueryInput` component.
3. **API Integration**
    - All API calls should be handled through:
        - `useCustomQuery`
        - `useCustomMutation`
    - These hooks provide consistent data fetching and mutation handling across the application.

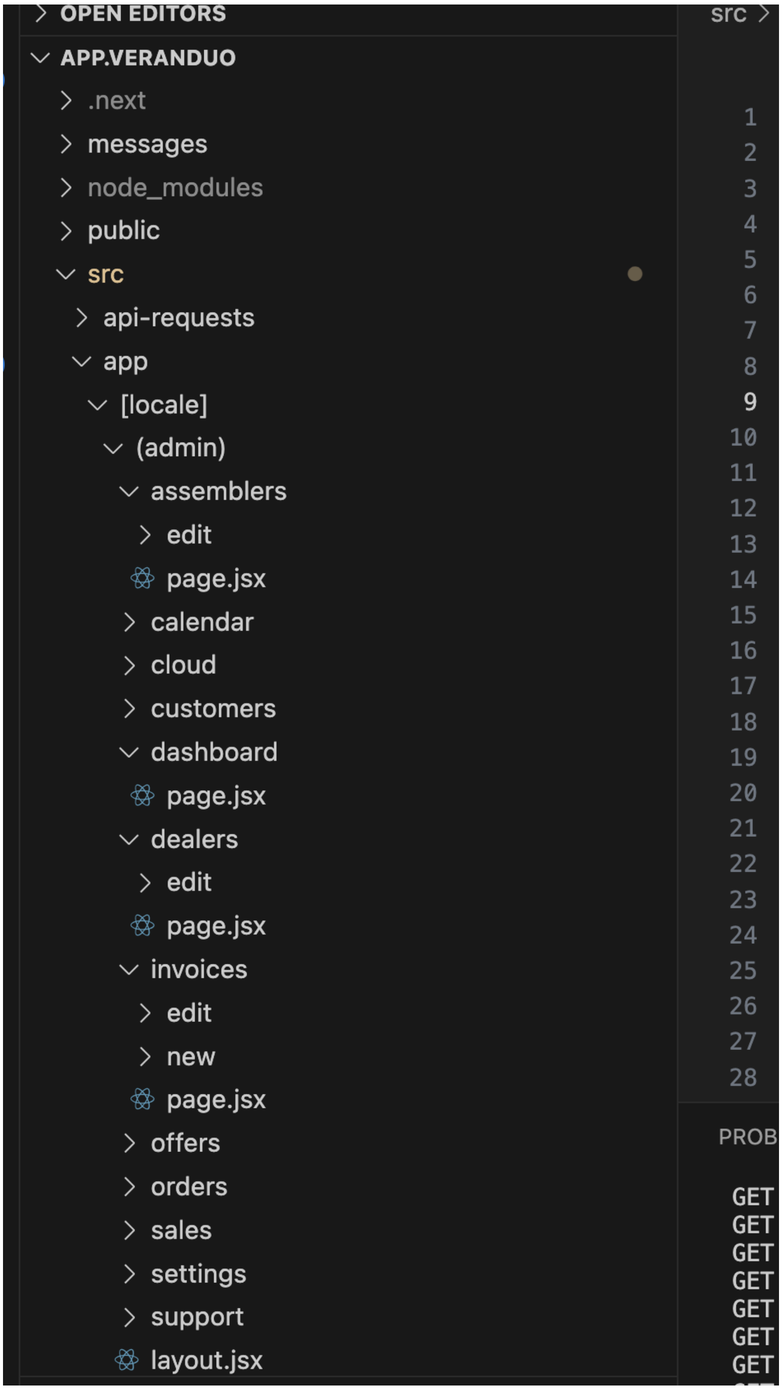
Development Guidelines

- Always check the existing components and hooks before creating new ones.

- Follow the established structure and patterns in the codebase.

- Keep localization in mind (`next-intl`) whenever you add UI text.

- Ensure that routes and protected pages are integrated correctly with `next-auth`.



```
> OPEN EDITORS
v APP.VERANDUO
  > .next
  > messages
  > node_modules
  > public
  v src
    > api-requests
    v app
      v [locale]
        v (admin)
          v assemblers
            > edit
              page.jsx
          > calendar
          > cloud
          > customers
          v dashboard
              page.jsx
          v dealers
            > edit
              page.jsx
          v invoices
            > edit
            > new
              page.jsx
          > offers
          > orders
          > sales
          > settings
          > support
          layout.jsx
```

Example code

```
<QueryInput
    queryKey={customer}
    path={customers}
    onMap={(item) => {
        return {
            label: item?.name,
            value: item?.id,
            item: item,
        };
    }}
    {...(invoice?.customer_id
        ? { initialIds: [invoice?.customer_id] }
        : {})}
    placeholder="Choose customer"
    value={customer?.id}
    onSelected={(selected) => {
        setCustomer(selected?.item);
    }}
    onChange={(e) => {
        setCustomer(e?.item);
        setInvoice({ ...invoice, customer: e?.item });
    }}
/>
```