Используемые библиотеки:

* react-chartjs-2
* react-router-dom
* sass
* uuid

Структура проекта (/client)

client/

│── node\_modules/

│── public/

│── src/

│ ├── assets/

│ │ ├── fonts/

│ │ ├── icons/

│ │ ├── images/

│ │ │ ├── logo\_2.svg

│ │ │ ├── logo.svg

│ ├── components/

│ │ ├── footer/

│ │ ├── header/

│ │ ├── home/

│ │ ├── left\_panel/

│ │ ├── modules/

│ │ ├── register/

│ │ ├── signin/

│ │ ├── user/

│ │ │ ├── completing\_task\_1/

│ │ │ ├── completing\_task\_2/

│ │ │ ├── main/

│ │ │ ├── module/

│ │ │ ├── modules/

│ │ │ ├── settings/

│ │ │ ├── task/

│ │ │ ├── terminal/

│ │ │ ├── user\_top/

│ ├── pages/

│ │ ├── home/

│ │ ├── modules/

│ │ ├── register/

│ │ ├── signin/

│ │ ├── user/

│ │ │ ├── completing\_task\_1/

│ │ │ ├── completing\_task\_2/

│ │ │ ├── main/

│ │ │ ├── module/

│ │ │ ├── modules/

│ │ │ ├── settings/

│ │ │ ├── task/

│ │ │ ├── terminal/

│ ├── scss/

│── App.js

│── index.js

│── .gitignore

│── package-lock.json

│── package.json

│── README.md

Проект представляет собой клиентскую часть веб-приложения, разработанную на **React** с использованием **React Router** для управления маршрутами.

**🔹 Технологический стек:**

* **React 19** – основа фронтенда
* **React Router** – маршрутизация
* **SCSS** – стилизация
* **useState, useEffect** – управление состоянием и побочные эффекты

**🔹 Структура проекта:**

* **src/components/** – переиспользуемые компоненты (header, footer, left\_panel и т. д.).
* **src/pages/** – страницы, разбитые на:
  + **Публичные** (home, signin, register, modules).
  + **Пользовательские** (user/main, user/modules, user/module/:id, user/task/:id, user/settings, user/terminal).
* **src/assets/** – шрифты, иконки и изображения.
* **src/scss/** – стилизация.

**🔹 Основная логика в App.js:**

1. Определены **скрываемые маршруты** (hiddenRoutes), на которых не отображаются Header и Footer.
2. **Глобальное состояние openMenu** контролирует открытие бокового меню и блокирует прокрутку при его активации.
3. **Динамические маршруты** поддерживают загрузку модулей и заданий через :id.
4. **Передача setOpenMenu в дочерние компоненты** позволяет управлять меню из различных страниц.

**🔹 Как дополнять проект:**

* Добавляя новые **публичные страницы**, их нужно просто прописать в Routes.
* При добавлении **новых пользовательских разделов** – добавить их в hiddenRoutes, если на них не должен отображаться Header и Footer.
* Для новых **динамических маршрутов** (/user/task/:id) – убедиться, что они корректно принимают параметры.
* При необходимости добавить **глобальные состояния**, можно использовать useContext или Redux.