

## Структура проекта FSD версии 2.1.

```
crm-app-bypass/
├── public/                                # Статические файлы, доступные напрямую
│   ├── favicon.ico                      # Иконка приложения для браузера
│   ├── logo.png                        # Логотип приложения
│   ├── manifest.json                   # Манифест для PWA (если используется)
│   └── vite.svg                         # vite иконка
├── src/                                  # Исходный код приложения
│   ├── app/                             # Глобальная конфигурация приложения
│   │   ├── entrypoint/                 # Точка входа в приложение
│   │   │   ├── App.tsx                 # Точка входа в приложение
│   │   │   └── main.tsx                 # фактическая точка входа в приложение, которая монтирует корневой React-компонент в DOM
│   │   ├── layout/                     # layout
│   │   │   ├── dashboard-layout.tsx    # Layout для страниц с дашбордом
│   │   │   ├── index.ts                # Публичный API (экспорт DashboardLayout, LoginLayout)
│   │   │   ├── login-layout.tsx        # Layout для страницы логина
│   │   │   └── README.md                # Документация (описание layouts)
│   │   ├── routes/                     # Роутинг приложения
│   │   │   ├── index.tsx               # Публичный API (экспорт роутинга)
│   │   │   └── routes.tsx              # Определение маршрутов с использованием react-router-dom
│   │   ├── stores/                     # Глобальная конфигурация хранилища
│   │   │   ├── notifications/          # Процесс управления уведомлениями
│   │   │   │   ├── _tests_/            # Тесты для процесса
│   │   │   │   │   ├── use-notifications.test.ts # Юнит-тесты для use-notifications.ts
│   │   │   │   └── hooks/              # Хуки
│   │   │   │       ├── use-notifications.ts # Хук для работы с уведомлениями
│   │   │   │       └── model/           # Бизнес-логика
│   │   │   │           ├── store.ts     # Zustand-хранилище для уведомлений
│   │   │   │           ├── index.ts     # Публичный API (экспорт хуков, store)
│   │   │   │           └── README.md    # Документация (описание уведомлений)
│   │   │   ├── theme/                  # Процесс управления темой приложения
│   │   │   │   ├── store.ts            # Zustand-хранилище для темы
│   │   │   └── styles/                  # Глобальные стили
│   │   │       ├── base.css              # Базовые стили
│   │   │       └── variables.css          # Стили
│   │   ├── error-boundary.tsx           # Компонент для обработки ошибок на уровне приложения
│   │   ├── init.tsx                     # Инициализация провайдеров (MUI ThemeProvider, QueryClientProvider)
│   │   └── README.md                     # Документация слоя app (описание маршрутов, провайдеров)
│   ├── assets/                          # Ресурсы приложения (иконки, изображения)
│   │   ├── Icon/                        # SVG-иконки
│   │   │   ├── icons/                  # Список SVG-иконок
│   │   │   │   ├── pencil-simple.svg   # Иконка карандаш
│   │   │   │   ├── task.svg            # Иконка задачи
│   │   │   │   └── user.svg             # Иконка пользователя
│   │   │   ├── Icon.tsx                 # Импорт SVG-иконки
│   │   │   └── README.md                # Документация по SVG-иконкам
│   │   ├── images/                     # Изображения
│   │   │   ├── logo.png                # Логотип приложения
│   │   │   └── react.svg                # SVG-иконка react
│   ├── data/                           ### Папка для моковых данных
│   │   └── testData.json                ### JSON-файл с тестовыми данными для разработки и тестов
│   ├── entities/                       # Бизнес-сущности
│   │   ├── employee/                   # Сущность сотрудника
│   │   │   ├── _tests_/                # Тесты для сущности
│   │   │   │   ├── normalize.test.ts   # Юнит-тесты для normalize.ts
│   │   │   │   └── model/               # Логика обработки данных
│   │   │   │       ├── normalize.ts     # Нормализация данных сотрудника
│   │   │   │       ├── index.ts         # Публичный API (экспорт типов, нормализации)
│   │   │   │       ├── README.md        # Документация (описание сущности)
│   │   │   │       └── types.ts         # Типы сотрудника (Employee)
│   │   ├── non-compliance/            # Сущность несоответствия
│   │   │   ├── model/                  # Логика обработки данных
│   │   │   │   ├── normalize.ts        # Нормализация данных несоответствия
│   │   │   │   ├── types.ts            # Типы данных для несоответствий
│   │   │   ├── index.ts                # Публичный API (экспорт типов, нормализации)
│   │   │   └── README.md                # Документация (описание сущности)
│   │   ├── object/                     # Сущность объекта
│   │   │   ├── _tests_/                # Тесты для сущности
│   │   │   │   ├── normalize.test.ts   # Юнит-тесты для normalize.ts
│   │   │   │   └── model/               # Логика обработки данных
│   │   │   │       ├── normalize.ts     # Нормализация данных объекта
│   │   │   │       ├── index.ts         # Публичный API (экспорт типов, нормализации)
│   │   │   │       ├── README.md        # Документация (описание сущности)
│   │   │   │       └── types.ts         # Типы объекта (Object)
│   │   └── object-type/                # Сущность типа объекта
│   │       ├── _tests_/                # Тесты для сущности
│   │       │   ├── normalize.test.ts   # Юнит-тесты для normalize.ts
│   │       └── model/                  # Логика обработки данных
```

└─ normalize.ts	# Нормализация данных объекта
└─ index.ts	# Публичный API (экспорт типов, нормализации)
└─ README.md	# Документация (описание сущности)
└─ types.ts	# Типы объекта (Object-type)
parameter/	# Сущность параметра
└─ _tests_/_	# Тесты для сущности
└─ └─ normalize.test.ts	# Юнит-тесты для normalize.ts
└─ model/	# Логика обработки данных
└─ └─ normalize.ts	# Нормализация данных параметра
└─ index.ts	# Публичный API (экспорт типов, нормализации)
└─ README.md	# Документация (описание сущности)
└─ types.ts	# Типы параметра (Parameter)
task/	# Сущность задачи
└─ _tests_/_	# Тесты для сущности
└─ └─ normalize.test.ts	# Юнит-тесты для normalize.ts
└─ model/	# Логика обработки данных
└─ └─ api.ts	# API Нормализация данных задачи
└─ └─ hooks.ts	# Хуки Нормализация данных задачи
└─ └─ normalize.ts	# Нормализация данных задачи
└─ index.ts	# Публичный API (экспорт типов, нормализации)
└─ README.md	# Документация (описание сущности)
└─ types.ts	# Типы задачи (Task)
user/	# Сущность пользователя
└─ @x/	# x
└─ api/	# API для user
└─ └─ requests/	# запросы
└─ └─ └─ get-profile.request.ts	# get-profile.request.ts
└─ └─ index.ts	# Публичный API (запросов)
└─ └─ routes.ts	# роутинг
└─ model/	# Логика обработки данных
└─ └─ normalize.ts	# Нормализация данных пользователя
└─ └─ index.ts	# Публичный API (моделей)
└─ └─ user.model.ts	# модели
└─ └─ user.store.ts	# store
└─ ui/	# UI-компоненты
└─ └─ Avatar/	# Аватар пользователя
└─ └─ └─ Avatar.module.css	# Стили для аватара
└─ └─ └─ Avatar.tsx	# Аватар
└─ └─ index.ts	# Публичный API (ui компонента)
└─ index.ts	# Публичный API (экспорт типов, нормализации)
└─ README.md	# Документация (описание сущности)
└─ types.ts	# Типы пользователя (User)
└─ index.ts	# Публичный API (entities)
features/	# Функциональные возможности (фичи)
└─ auth/	# Фича авторизации
└─ └─ _tests_/_	# Тесты для фичи
└─ └─ └─ login-form.test.tsx	# Юнит-тесты для login-form.tsx
└─ └─ └─ use-auth.test.ts	# Юнит-тесты для use-auth.ts
└─ hooks/	# Хуки фичи
└─ └─ use-auth.ts	# Хук для управления авторизацией
└─ model/	# Бизнес-логика
└─ └─ store.ts	# Zustand-хранилище для состояния авторизации
└─ ui/	# UI-компоненты
└─ └─ login-form.tsx	# Компонент формы логина
└─ index.ts	# Публичный API (экспорт компонентов, хуков)
└─ README.md	# Документация (описание авторизации, API)
└─ types.ts	# Типы для авторизации (например, User, Credentials)
calendar/	# Фича для календаря
└─ _tests_/_	# Тесты для фичи
└─ └─ README.md	# Документация для тестирования
└─ hooks/	# Хуки фичи
└─ └─ useCalendar.ts	# Хук для управления календаря
└─ index.ts	# Публичный API (экспорт компонентов, хуков)
└─ README.md	# Документация (описание календаря, API)
└─ types.ts	# Типы для календаря
object-type-filters/	# Модальные окна фильтрации
└─ _tests_/_	# Тесты для фичи
└─ └─ README.md	# Документация для тестирования
└─ model/	# Состояние фильтров (Zustand)
└─ └─ store.ts	# Zustand-хранилище для состояния Модального окна фильтрации
└─ ui/	# UI компонентов фильтрации
└─ index.ts	# Публичный API (экспорт компонентов, хуков)
└─ README.md	# Документация (описание Модального окна фильтрации, API)
└─ types.ts	# Типы для Модального окна фильтрации
object-type-edit/	# Модальное редактирование
└─ _tests_/_	# Тесты для фичи
└─ └─ README.md	# Документация для тестирования
└─ model/	# Состояние фильтров (Zustand)
└─ └─ store.ts	# Zustand-хранилище для состояния Модального окна редактирование
└─ ui/	# UI компонентов фильтрации
└─ index.ts	# Публичный API (экспорт компонентов, хуков)
└─ README.md	# Документация (описание Модального окна редактирование, API)

```

└─ types.ts                # Типы для Модального окна редактирование

tasks/                    # Фича задач
└─ _tests_/              # Тесты для фичи
  └─ task-list.test.tsx  # Юнит-тесты для task-list.tsx
  └─ use-tasks.test.ts   # Юнит-тесты для use-tasks.ts
└─ components/          # Компоненты
  └─ Chat.tsx           # Компонент для переписки (чат)
└─ hooks/              # Хуки
  └─ use-task.ts        # Хук для работы с одной задачей
  └─ use-tasks.ts       # Хук для получения списка задач
└─ model/              # Бизнес-логика
  └─ filter.ts          # Логика фильтрации задач
  └─ store.ts           # Zustand-хранилище для задач
└─ task-form/          # Форма создания задания
  └─ api/              # API
    └─ task.ts         # API запрос задания
  └─ model/            # Бизнес-логика
    └─ task-schemas.ts # Zustand-хранилище для задач
└─ ui/                # UI-компоненты
  └─ chat.tsx          # Компонент чата для задачи
  └─ task-filter-form.tsx # Компонент формы фильтрации задач
  └─ task-list.tsx     # Компонент списка задач
└─ index.ts           # Публичный API (экспорт компонентов, хуков)
└─ README.md          # Документация (описание функционала, API)
└─ types.ts           # Типы для задач и фильтров

user-list/               # Фича списка пользователей
└─ _tests_/             # Тесты для фичи
  └─ filter-form.test.tsx # Юнит-тесты для filter-form.tsx
  └─ use-user-list.test.ts # Юнит-тесты для use-user-list.ts
└─ hooks/              # Хуки
  └─ use-employee-list.ts # Хук для получения списка сотрудников
  └─ use-user-list.ts     # Хук для получения списка пользователей
└─ model/              # Бизнес-логика
  └─ filter.ts          # Логика фильтрации пользователей
  └─ store.ts           # Zustand-хранилище для списка пользователей
└─ ui/                # UI-компоненты
  └─ edit-button.tsx    # Компонент кнопки редактирования
  └─ filter-form.tsx    # Компонент формы фильтрации
└─ index.ts           # Публичный API (экспорт компонентов, хуков)
└─ README.md          # Документация (описание функционала, API)
└─ types.ts           # Типы для пользователей и фильтров

pages/                   # Страницы приложения, соответствующие маршрутам
└─ calendar/            # Страница календаря
  └─ _tests_/          # Тесты для страницы
    └─ ui.test.tsx     # Юнит-тесты для ui.tsx
  └─ ui/              # ui компоненты страниц
    └─ CalendarPage/  # страница Календарь работ
      └─ CalendarPage.tsx # Календарь работ
      └─ README.md       # Документация (описание функционала Календарь работ)
    └─ CheckModal/    # Открытие модального окна
      └─ CheckModal.tsx # Календарь работ
    └─ index.ts       # Публичный API (экспорт ui страниц)
  └─ index.ts         # Публичный API (экспорт CalendarPage)

└─ chat/              # Страница переписки пользователей (Чат)
  └─ _tests_/         # Тесты для страницы
    └─ ui.test.tsx    # Юнит-тесты для ui.tsx
  └─ ui/             # ui компоненты страниц
    └─ ChatPage/     # страница переписки (Чат)
      └─ ChatPage.tsx # Календарь работ
      └─ README.md    # Документация (описание функционала переписки (Чат))
    └─ index.ts      # Публичный API (экспорт ui страниц)
  └─ index.ts        # Публичный API (экспорт ChatPage)

└─ check-logs/        # Страницы журнала проверок
  └─ ui/             # ui компоненты страниц
    └─ CheckLogsPage/ # страница журнала проверок
      └─ CheckLogsPage.tsx # журнал проверок
      └─ README.md       # Документация (описание функционала журнала проверок)
    └─ index.ts       # Публичный API (экспорт ui страниц)
  └─ index.ts        # Публичный API (экспорт CheckLogsPage)

└─ dashboard/         # Страница дашборда
  └─ _tests_/         # Тесты для страницы
    └─ ui.test.tsx    # Юнит-тесты для ui.tsx
  └─ ui/             # ui компоненты страниц
    └─ DashboardPage/ # страница переписки (Дашбورد)
      └─ AppUsageChart.tsx # AppUsageChart
      └─ DashboardPage.tsx # Дашбورد (отображает статистику, графики)
      └─ README.md       # Документация (описание функционала Дашборда)
    └─ index.ts       # Публичный API (экспорт ui страниц)
  └─ index.ts        # Публичный API (экспорт DashboardPage)

└─ defect-logs/       # Страницы журнала дефектов
  └─ ui/             # ui компоненты страниц
    └─ DefectLogsPage/ # страница журнала дефектов

```

```

├── DefectLogsPage.tsx # журнал дефектов
├── README.md # Документация (описание функционала журнала дефектов)
├── index.ts # Публичный API (экспорт ui страниц)
└── index.ts # Публичный API (экспорт DefectLogsPage)

help/ # Страница помощи
├── _tests_/ # Тесты для страницы
├── ui.test.tsx # Юнит-тесты для ui.tsx
├── ui/ # ui компоненты страниц
├── HelpPage/ # страница помощи
├── HelpPage.tsx # Форма помощи
├── README.md # Документация (описание функционала помощи)
├── index.ts # Публичный API (экспорт ui страниц)
└── index.ts # Публичный API (экспорт HelpPage)

instructions/ # Страница инструкции
├── _tests_/ # Тесты для страницы
├── ui.test.tsx # Юнит-тесты для ui.tsx
├── ui/ # ui компоненты страниц
├── InstructionsPage/ # страница Инструкции
├── InstructionsPage.tsx # инструкции
├── README.md # Документация (описание функционала инструкции)
├── index.ts # Публичный API (экспорт ui страниц)
└── index.ts # Публичный API (экспорт InstructionsPage)

login/ # Страница авторизации
├── _tests_/ # Тесты для страницы
├── ui.test.tsx # Юнит-тесты для ui.tsx
├── ui/ # ui компоненты страниц
├── LoginPage/ # страница авторизации
├── LoginPage.tsx # форма авторизации
├── README.md # Документация (описание функционала авторизации)
├── index.ts # Публичный API (экспорт ui страниц)
└── index.ts # Публичный API (экспорт LoginPage)

non-compliance/ # Страница список несоответствий
├── _tests_/ # Тесты для страницы
├── ui.test.tsx # Юнит-тесты для ui.tsx
├── ui/ # ui компоненты страниц
├── NonCompliancePage/ # Страница список несоответствий
├── NonCompliancePage.tsx # список несоответствий
├── README.md # Документация (описание функционала список несоответствий)
├── index.ts # Публичный API (экспорт ui страниц)
└── index.ts # Публичный API (экспорт NonCompliancePage)

object/ # Страница объекта
├── _tests_/ # Тесты для страницы
├── ui.test.tsx # Юнит-тесты для ui.tsx
├── ui/ # ui компоненты страниц
├── ObjectPage/ # Страница объекта
├── ObjectPage.tsx # Объект
├── README.md # Документация (описание функционала объекта)
├── index.ts # Публичный API (экспорт ui страниц)
└── index.ts # Публичный API (экспорт ObjectPage)

object-type/ # Страница тип объекта
├── _tests_/ # Тесты для страницы
├── ui.test.tsx # Юнит-тесты для ui.tsx
├── ui/ # ui компоненты страниц
├── ObjectTypePage/ # Страница тип объекта
├── ObjectTypePage.tsx # тип объекта
├── README.md # Документация (описание функционала типа объекта)
├── index.ts # Публичный API (экспорт ui страниц)
└── index.ts # Публичный API (экспорт ObjectTypePage)

objects/ # Страница список объектов
├── _tests_/ # Тесты для страницы
├── ui.test.tsx # Юнит-тесты для ui.tsx
├── ui/ # ui компоненты страниц
├── ObjectsPage/ # Страница список объектов
├── ObjectsPage.tsx # список объектов
├── README.md # Документация (описание функционала списка объектов)
├── index.ts # Публичный API (экспорт ui страниц)
└── index.ts # Публичный API (экспорт ObjectsPage)

parameters/ # Страница список параметров
├── _tests_/ # Тесты для страницы
├── ui.test.tsx # Юнит-тесты для ui.tsx
├── ui/ # ui компоненты страниц
├── ParametersPage/ # Страница список параметров
├── ParametersPage.tsx # список параметров
├── README.md # Документация (описание функционала список параметров)
├── index.ts # Публичный API (экспорт ui страниц)
└── index.ts # Публичный API (экспорт ParametersPage)

settings/ # Страница настроек
├── _tests_/ # Тесты для страницы
├── ui.test.tsx # Юнит-тесты для ui.tsx
└── ui/ # ui компоненты страниц

```

SettingsPage/	# Страница настроек
SettingsPage.tsx	# Настройки
README.md	# Документация (описание функционала настроек)
index.ts	# Публичный API (экспорт ui страниц)
index.ts	# Публичный API (экспорт SettingsPage)
task/	# Страницы задание
control/	# Страница контроля задач
tests_/	# Тесты для страницы
ui.test.tsx	# Юнит-тесты для ui.tsx
ui/	# ui компоненты страниц
TaskPage/	# Страница задания
TaskPage.tsx	# Задание
README.md	# Документация (описание функционала задания)
index.ts	# Публичный API (экспорт ui страниц)
index.ts	# Публичный API (экспорт TaskPage)
task-control/	# Страницы для работы с задачами (контроль задач)
ui/	# ui компоненты страниц
TaskControlPage/	# Страница Контроль задач
TaskControlPage.tsx	# Контроль задач
README.md	# Документация (описание функционала контроля задач)
index.ts	# Публичный API (экспорт ui страниц)
index.ts	# Публичный API (экспорт CreateTaskPage, TaskControlPage, TaskViewPage)
task-create/	# Страницы для работы с задачами (создания задачи)
ui/	# ui компоненты страниц
CreateTaskPage/	# Страница создания задачи
CreateTaskPage.tsx	# Создание задание
README.md	# Документация (описание функционала создания задачи)
index.ts	# Публичный API (экспорт ui страниц)
index.ts	# Публичный API (экспорт CreateTaskPage, TaskControlPage, TaskViewPage)
task-view/	# Страницы для работы с задачами (просмотр задач)
ui/	# ui компоненты страниц
TaskViewPage/	# Страница просмотра задачи
TaskViewPage.tsx	# Просмотр задачи
README.md	# Документация (описание функционала просмотра задачи)
index.ts	# Публичный API (экспорт ui страниц)
index.ts	# Публичный API (экспорт CreateTaskPage, TaskControlPage, TaskViewPage)
user-profile/	# Страница профиля пользователя
ui/	# ui компоненты страниц
ProfilePage/	# страница Профиль пользователя
ProfilePage.tsx	# Профиль пользователей
README.md	# Документация (описание функционала профиля пользователя)
index.ts	# Публичный API (экспорт ui страниц)
index.ts	# Публичный API (экспорт ProfilePage)
users-employees/	# Страница списка сотрудников
ui/	# ui компоненты страниц
EmployeesPage/	# страница Список сотрудников
EmployeesPage.tsx	# Список сотрудников
README.md	# Документация (описание функционала списка сотрудников)
index.ts	# Публичный API (экспорт ui страниц)
index.ts	# Публичный API (экспорт EmployeesPage)
users-list/	# Страница списка пользователей
ui/	# ui компоненты страниц
UserPage/	# страница Список пользователей
README.md	# Документация (описание функционала списка пользователей)
UserPage.tsx	# Список пользователей
index.ts	# Публичный API (экспорт ui страниц)
index.ts	# Публичный API (экспорт UsersPage)
index.ts	# Публичный API (экспорт страниц)
shared/	# Общие утилиты и компоненты
api/	# Централизованные API-клиенты
auth/	# API для авторизации
client.ts	# Axios-клиент для запросов авторизации
index.ts	# Публичный API (экспорт authApi)
calendar/	# API для календаря
client.ts	# Axios-клиент для запросов календаря
index.ts	# Публичный API (экспорт календаря)
dashboard/	# API для дашборда
client.ts	# Axios-клиент для запросов дашборда
index.ts	# Публичный API (экспорт dashboardApi)
employee/	# API для сотрудников
client.ts	# Axios-клиент для запросов сотрудников
index.ts	# Публичный API (экспорт getEmployeesByDepartment)
help/	# API для помощи
client.ts	# Axios-клиент для запросов помощи
index.ts	# Публичный API (экспорт helpApi)
non-compliance/	# API для несоответствий
client.ts	# Axios-клиент для запросов несоответствий
index.ts	# Публичный API (экспорт nonComplianceApi)
object/	# API для объектов
client.ts	# Axios-клиент для запросов объектов
index.ts	# Публичный API (экспорт objectApi)

types.ts	# типы для объекта
object-type/	# API для типа объекта
client.ts	# Axios-клиент для запросов типа объекта
index.ts	# Публичный API (экспорт objectTypeApi)
types.ts	# Тип (objectTypeApi)
parameter/	# API для параметров
client.ts	# Axios-клиент для запросов параметров
index.ts	# Публичный API (экспорт parameterApi)
settings/	# API для настроек
client.ts	# Axios-клиент для запросов настроек
index.ts	# Публичный API (экспорт settingsApi)
task/	# API для задач
employee/	# API employee для задач
client.ts	# Axios-клиент для запросов задач
index.ts	# Публичный API (экспорт employeeApi)
types.ts	# Тип (employeeApi)
history/	# API history для история проверок объекта
client.ts	# Axios-клиент для запросов задач
index.ts	# Публичный API (экспорт taskHistoryApi)
types.ts	# Тип (TaskHistoryItem)
non-compliance/	# API non-compliance для задач
client.ts	# Axios-клиент для запросов задач
index.ts	# Публичный API (экспорт nonComplianceApi)
types.ts	# Тип (nonComplianceApi)
object/	# API object для задач
client.ts	# Axios-клиент для запросов задач
index.ts	# Публичный API (экспорт getParametersAndObjectType)
types.ts	# Тип (getParametersAndObjectType)
client.ts	# Axios-клиент для запросов задач
index.ts	# Публичный API (экспорт taskApi)
types.ts	# Тип (taskApi)
user/	# API для пользователей
client.ts	# Axios-клиент для запросов пользователей
index.ts	# Публичный API (экспорт userApi)
access-token.inteceptor.ts	# access-token
axios.ts	# Глобальная конфигурация Axios (интерцепторы, базовый URL)
index.ts	# Публичный API (экспорт всех клиентов)
config/	# Конфигурации приложения
env.ts	# env
flags.ts	# flags
index.ts	# Публичный API (экспорт конфигураций)
navigation.ts	# Конфигурация навигации (пункты меню, роли)
theme.ts	# MUI-тема для стилизации приложения
hooks/	# Общие хуки
tests_/	# Тесты для хуков
use-debounce.test.ts	# Юнит-тесты для use-debounce.ts
use-user.test.ts	# Юнит-тесты для use-user.ts
index.ts	# Публичный API (экспорт хуков)
use-debounce.ts	# Хук для дебаунсинга значений
use-user.ts	# Хук для получения данных текущего пользователя
lib/	# Общие утилиты
tests_/	# Тесты для утилит
error-handler.test.ts	# Юнит-тесты для error-handler.ts
normalize.test.ts	# Юнит-тесты для normalize.ts
auth.ts	# авторизация
error-handler.ts	# Утилита для обработки ошибок API
index.ts	# Публичный API (экспорт утилит)
normalize.ts	# Общие функции нормализации данных
schemas.ts	# Zod-схемы для валидации форм и API
storage.ts	#
useDebounce.ts	# useDebounce
routes/	# роутинг
routes.ts	# Роутинг
types/	# Общие типы, используемые в приложении
api.ts	# Типы для API-ответов (например, Response<T>)
index.ts	# Публичный API (экспорт типов)
ui/	# Общие UI-компоненты
tests_/	# Тесты для компонентов
Button.test.tsx	# Юнит-тесты для Button.tsx
Input.test.tsx	# Юнит-тесты для Input.tsx
Button.tsx	# Компонент универсальной кнопки
ErrorBoundary.tsx	# Компонент универсальной вывода ошибки
index.ts	# Публичный API (экспорт Button, Input)
Input.tsx	# Компонент универсального поля ввода
Logo.tsx	# Компонент универсального отображения логотипа
widgets/	# Переиспользуемые UI-компоненты
page-header/	# Компонент заголовка страницы
index.ts	# Публичный API (экспорт TitlePage)
README.md	# Документация (описание заголовка страницы)
ui.tsx	# UI-компонент (отображение заголовка страницы)
header/	# Компонент шапки

	├─ _tests_/	# Тесты для компонента
	├─ ui.test.tsx	# Юнит-тесты для ui.tsx
	├─ index.ts	# Публичный API (экспорт Header)
	├─ README.md	# Документация (описание шапки)
	├─ types.ts	# типы (описание шапки)
	├─ ui.tsx	# UI-компонент (отображение аватара, меню)
	├─ non-compliance/	
	├─ add-new-non-compliance-modal/	
	├─ AddNewNonComplianceModal.tsx	# Модальное окно для добавления
	├─ index.ts	
	├─ edit-non-compliance-modal/	
	├─ EditNonComplianceModal.tsx	# Модальное окно для редактирования
	├─ index.ts	
	├─ filter-non-compliance-modal/	
	├─ FilterNonComplianceModal.tsx	# Модальное окно для фильтрации
	├─ index.ts	
	├─ non-compliance-table/	
	├─ NonComplianceTable.tsx	# Компонент таблицы несоответствий
	├─ types.ts	# Типы данных для таблицы
	├─ index.ts	# Публичный API (экспорт non-compliance)
	├─ object/	# Компоненты для объекта
	├─ object-history-table/	# Таблица проверки объекта
	├─ _tests_/	# Тесты для компонента
	├─ ui.test.tsx	# Юнит-тесты для ui.tsx
	├─ index.ts	# Публичный API (экспорт таблицы)
	├─ README.md	# Документация (описание таблицы)
	├─ types.ts	# типы (типы данных в таблице)
	├─ ui.tsx	# UI-компонент (Таблица, которая использует CustomTable и логика отображения)
	├─ object-info-card/	# Карточка информация об объекте
	├─ _tests_/	# Тесты для компонента
	├─ ui.test.tsx	# Юнит-тесты для ui.tsx
	├─ index.ts	# Публичный API (экспорт информационной карточки по объекту)
	├─ README.md	# Документация (описание информационной карточки по объекту)
	├─ ui.tsx	# UI-компонент (информационная карточка по объекту и логика отображения)
	├─ object-modal/	# модальное окно отображения объекта
	├─ _tests_/	# Тесты для компонента
	├─ ui.test.tsx	# Юнит-тесты для ui.tsx
	├─ index.ts	# Публичный API (экспорт модального окна по объекту)
	├─ README.md	# Документация (описание модального окна по объекту)
	├─ types.ts	# типы (типы данных в модального окна по объекту)
	├─ ui.tsx	# UI-компонент (модальное окно по объекту и логика отображения)
	├─ object-type/	# Компоненты для объекта
	├─ add-object-type-modal/	# Модалка добавления нового типа объекта
	├─ _tests_/	# Тесты для компонента
	├─ ui.test.tsx	# Юнит-тесты для ui.tsx
	├─ index.ts	# Публичный API (экспорт Нового типа объекта)
	├─ README.md	# Документация (описание Нового типа объекта)
	├─ types.ts	# типы (описание Нового типа объекта)
	├─ ui.tsx	# UI-компонент (отображение Нового типа объекта)
	├─ object-type-table/	# Таблица типов объектов
	├─ _tests_/	# Тесты для компонента
	├─ ui.test.tsx	# Юнит-тесты для ui.tsx
	├─ model/	# Zustand store + API hooks
	├─ store.ts	# Zustand store (выбранные типы объектов)
	├─ index.ts	# Публичный API (экспорт таблицы)
	├─ README.md	# Документация (описание таблицы)
	├─ types.ts	# типы (описание таблицы)
	├─ ui.tsx	# UI-компонент (Таблица типов объектов и логика отображения)
	├─ parameters/	# Компоненты для параметров проверки объекта
	├─ add-new-parameter-modal/	# Модалка добавления нового параметра
	├─ _tests_/	# Тесты для компонента
	├─ ui.test.tsx	# Юнит-тесты для ui.tsx
	├─ model/	# Тесты для компонента
	├─ store.ts	# Zustand store (выбранные несоответствия)
	├─ index.ts	# Публичный API (экспорт нового параметра)
	├─ README.md	# Документация (описание типа нового параметра и несоответствий)
	├─ types.ts	# Типы нового параметра и несоответствий
	├─ ui.tsx	# UI-компонент (Компонент Нового параметра)
	├─ add-parameter-modal/	# Модалка добавления параметра, который уже есть в базе данных к списку типа объектов или
объекту	├─ _tests_/	# Тесты для компонента
	├─ ui.test.tsx	# Юнит-тесты для ui.tsx
	├─ model/	# Тесты для компонента
	├─ store.ts	# Zustand store (выбранные несоответствия)
	├─ index.ts	# Публичный API (экспорт AddParameterModal)
	├─ README.md	# Документация (описание типа параметров и несоответствий)
	├─ types.ts	# Типы параметров и несоответствий
	├─ ui.tsx	# UI-компонент (Компонент AddParameterModal)
	├─ edit-parameter-modal/	# Модалка редактирования параметра
	├─ _tests_/	# Тесты для компонента

```

├── ui.test.tsx # Юнит-тесты для ui.tsx
├── model/ # Тесты для компонента
├── store.ts # Zustand store (выбранные несоответствия)
├── index.ts # Публичный API (экспорт EditParameterModal)
├── README.md # Документация (описание типа параметров и несоответствий)
├── types.ts # Типы параметров и несоответствий
├── ui.tsx # UI-компонент (Компонент EditParameterModal)

├── parameters-table/ # Таблица отображения параметров проверки объектов
├── _tests_/ # Тесты для компонента
├── ui.test.tsx # Юнит-тесты для ui.tsx
├── index.ts # Публичный API (экспорт таблицы)
├── README.md # Документация (описание таблицы)
├── types.ts # типы (описание таблицы)
├── ui.tsx # UI-компонент (Таблица отображения параметров проверки объектов и логика отображения)

├── sidebar/ # Компонент бокового меню
├── _tests_/ # Тесты для компонента
├── ui.test.tsx # Юнит-тесты для ui.tsx
├── index.ts # Публичный API (экспорт Sidebar)
├── README.md # Документация (описание меню)
├── ui.tsx # UI-компонент (навигация)

├── table/ # Компонент таблицы
├── _tests_/ # Тесты для компонента
├── ui.test.tsx # Юнит-тесты для ui.tsx
├── model/ # Бизнес-логика
├── store.ts # Zustand-хранилище для кастомного элемента таблицы
├── index.ts # Публичный API (экспорт Table)
├── README.md # Документация (описание таблицы)
├── ui.tsx # UI-компонент (таблица с сортировкой, пагинацией)

├── task/ # Компоненты для задания
├── object-task-table/ # Таблица заданий по проверке объекта
├── _tests_/ # Тесты для компонента
├── ui.test.tsx # Юнит-тесты для ui.tsx
├── index.ts # Публичный API (экспорт таблицы)
├── README.md # Документация (описание таблицы заданий по проверке объекта)
├── types.ts # типы (описание таблицы)
├── ui.tsx # UI-компонент (таблица заданий по проверке объекта и логика отображения)

├── task-chat/ # Блок чата (переписка по заданию)
├── _tests_/ # Тесты для компонента
├── ui.test.tsx # Юнит-тесты для ui.tsx
├── index.ts # Публичный API (экспорт блока чата)
├── README.md # Документация (описание блока чата)
├── ui.tsx # UI-компонент (Блок чата (переписка по заданию) TaskChat)

├── task-management/ # Блок управления (скачать/загрузить XML)
├── _tests_/ # Тесты для компонента
├── ui.test.tsx # Юнит-тесты для ui.tsx
├── index.ts # Публичный API (экспорт блока управления)
├── README.md # Документация (описание блока управления)
├── ui.tsx # UI-компонент (Блок управления (скачать/загрузить XML) TaskManagementBlock)

├── task-parameters-table/ # Таблица параметров проверки
├── _tests_/ # Тесты для компонента
├── ui.test.tsx # Юнит-тесты для ui.tsx
├── index.ts # Публичный API (экспорт таблицы параметров проверки)
├── README.md # Документация (описание таблицы параметров проверки)
├── ui.tsx # UI-компонент (Таблица параметров проверки TaskParametersTable)

├── vite-env.d.ts ### Файл с объявлениями типов для Vite

├── tests/ # Глобальные тесты (интеграционные, e2e)
├── integration/ # Интеграционные тесты
├── auth.test.ts # Тесты авторизации
├── tasks.test.ts # Тесты работы с задачами
├── setup.ts # Настройка тестовой среды (Jest)

├── .env # Переменные окружения для продакшена (API_URL и т.д.)
├── .env.test # Переменные окружения для тестов
├── .gitignore # Игнорируемые файлы (node_modules, dist)
├── eslint.config.js # Конфигурация ESLint (правила линтинга, FSD)
├── index.html # index страница
├── jest.config.js # jest.config.js
├── package-lock.json # package-lock.json
├── package.json # Зависимости и скрипты проекта
├── README.md # Общая документация проекта (установка, запуск)
├── tsconfig.app.json # Конфигурация TypeScript для приложения
├── tsconfig.json # Основная конфигурация TypeScript
├── tsconfig.node.json # Конфигурация TypeScript для Node.js
├── tsconfig.tests.json # Конфигурация TypeScript для tests.json
├── vite.config.ts # Конфигурация Vite (алиасы, плагины)

```