

## Задание 4: Клиент – Работа с формами и аутентификацией

### Цель:

Познакомиться с работой менеджеров форм и аутентификацией

### Технические требования

#### 1. Стек технологий:

- React + TypeScript
- React Router v6 (или любой другой инструмент для маршрутизации)
- State-менеджер: любой на выбор (Redux Toolkit / Zustand / MobX и т.д.) либо вообще без него, можно попробовать tanstack Query
- Сборщик проекта: любой на выбор (Vite / Webpack / Next.js и т.д.)
- UI-фреймворк: любой на выбор (Ant Design / Material UI / AdmiralDS), возможно самостоятельная разработка компонентов без использования UI-фреймворка
- Инструменты стилизации: на ваш выбор (CSS Modules / styled-components / tailwind и т.д.)
- Менеджер форм formik/react hook form, либо оба на разных страницах

#### 2. Особенности реализации:

##### Маршруты

- Главная страница с таблицей и списком сущностей: /
- Страница создания юзера /user/create (форма создания)
- Страница редактирования юзера
- Страница логина /login

#### 3. Требования к реализации:

- Слева находится меню навигации, клик по элементу навигации ведет на страницу создания юзера
- В верхней части экрана находится хедер
- Хедер и навигация часть лэяута и присутствуют на всех страницах кроме страницы логина
- При создании юзера открывается страница с формой, форма отправляется по кнопке, после успешного создания редирект на главную страницу
- При редактировании юзера открывается форма с уже заполненными полями которые можно редактировать, пароль и почта не редактируются
- Главная страница это таблица с обязательными столбцами id и email – остальные по желанию

Бонусные требования:

Сделать валидацию полей

Сделать два варианта создания с разными менеджерами в таком случае страницы создания называем по менеджеру - /formik

Сделать кнопку для удаления сущности в таблице

Два поля для пароля – пароль и подтверждение с проверкой  
Полное имя заполняется автоматически при вводе имени и фамилии, но доступно для редактирования

#### 4. Рекомендации

- ✓ Для старта проекта можно использовать бойлерплейт из прошлых заданий
- ✓ Для работы с api подготовлен бэк <https://github.com/Pardeg/forms-server>
- ✓ Дефолтная учетка email - [admin@inno.tech](mailto:admin@inno.tech) password - admin
- ✓ Установка согласно readme.md для настройки прокси в vite можно использовать



```
server: {
  allowedHosts: ['localhost'],
  port: 5173,
  proxy: {
    '/api': {
      changeOrigin: true,
      secure: false,
      target: 'http://localhost:4000',
    },
  },
  strictPort: true,
},
```

Для webpask или аналогов можно посмотреть в документации

- ✓ Сваггер с описанием доступен по localhost:4000/api
- ✓ Токен передается через cookies
- ✓ Рекомендуются использовать auth/me для определения текущего статуса и перенаправлять в случае 401 ошибки на страницу логина

#### 5. Описание валидации полей post api/v1/users

Название	Тип	Обязательность	Дополнительно
----------	-----	----------------	---------------

name	string	да	Максимальная длина 64
surName	string	да	Максимальная длина 64
password	string	да	
fullName	string	да	Максимальная длина 130
email	string	да	Проверка на regex для email
birthDate	Пример 2025-07- 16T20:58:15.998Z	нет	
telephone	string Пример +79991231231	нет	Проверка на regex для телефона
employment	string Делаем селектом	нет	
userAgreement	boolean Делаем чекбоксом	нет	