

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Фронт-энд разработка

Отчет

Лабораторная работа

Выполнил:

Екушев Владислав

Группа К32402

Проверил:

Добряков Д. И.

Санкт-Петербург

2023 г.

## **Задача**

По выбранному варианту необходимо выполнить вёрстку сайта средствами HTML, CSS и Bootstrap. Продумать и реализовать основные моменты в которых необходим JS.

Вариант: Разработка интерактивного сайта для управления умным домом

## **Ход работы**

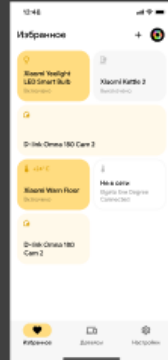
В ходе выполнения работы была видоизменена задача следующим образом: реализовать сайт по выбранному варианту средствами React, TypeScript, [MUI](#) + Emotion. Такое решение было принято потому, что эти средства разработки будут использоваться в следующих лабораторных работах. Для того чтобы упростить разработку стоит начинать работу сразу с проверенными инструментами.

### **1. Дизайн**

В первую очередь был реализован дизайн страниц приложения. Файл Figma находится [здесь](#).

## Favorites

### Favorites



### Favorites



### Device



### Device



## Frame 33

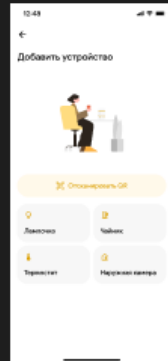
### Device



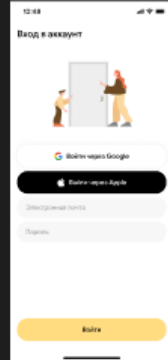
### Device



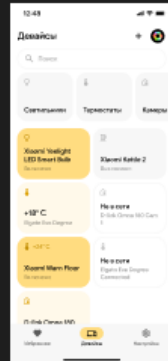
## Add Device



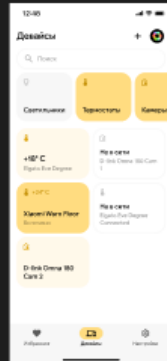
## Login



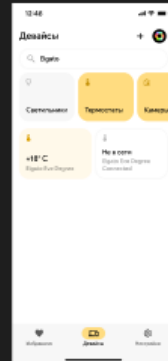
## Devices



## Devices



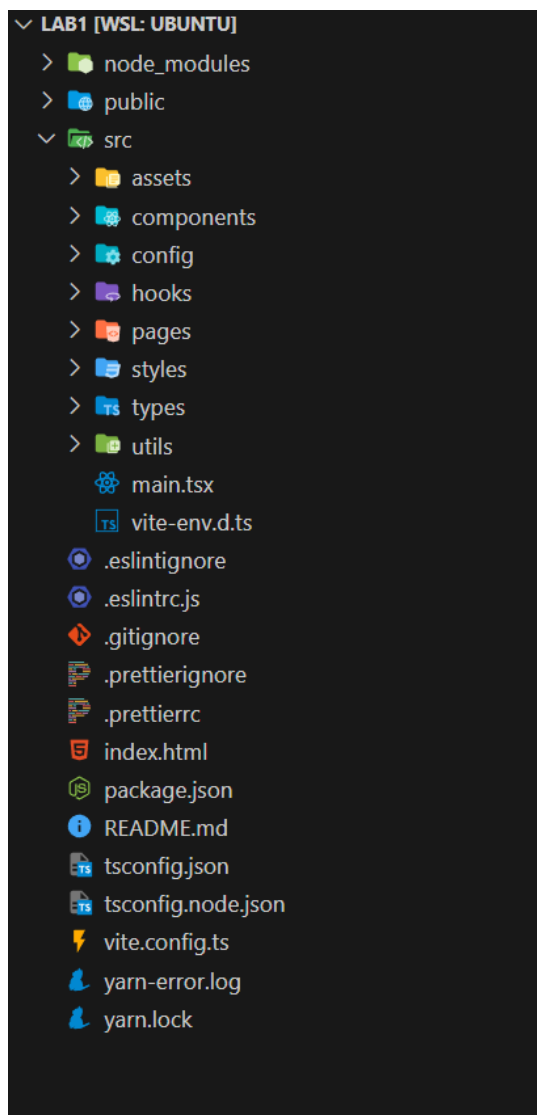
## Devices



За стиль и подход к дизайну приложения для умного дома использовались макеты умного дома от Google Home, Yandex Qazar и SberDevices.

В настоящее время не был реализован экран добавления девайса и не прорисован экран настроек, так как они будут задействованы в следующих лабораторных работах. Также, на макетах нет модального окна при нажатии на аватар пользователя, он в свою очередь был скопирован из приложений экосистемы Google.

Структура исходников приложения:



*public* – папка для статики

*src/assets* – папка для статики, которую обрабатывает Vite

*src/components* – компоненты приложения, часто используемые блоки

*src/config* – ключи приложения, константы, роуты

*src/hooks* – часто используемые хуки для компонентов

*src/pages* – роуты приложения, отдельные страницы, собираемые из блоков

*src/styles* – критичные стили приложения, которые нужно загрузить в первую очередь

*src/types* – типы данных

*src/utils* – вспомогательные функции

## **2. Исследование сферы**

Для того, чтобы создать масштабируемое и многофункциональное приложение, следует разобраться в его сфере применения (то есть, посмотреть у конкурентов). У Яндекса есть [спецификация](#) о том, как они хранят и обновляют информацию об умных устройствах и их способностях. По их образу была создана типизация для девайсов в данном проекте.

```

Device.ts

export enum DeviceType {
    LIGHT_BULB = 'light_bulb',
    KETTLE = 'kettle',
    THERMOSTAT = 'thermostat',
    CAMERA_OUTDOOR = 'camera_outdoor',
}

export enum DeviceState {
    ONLINE,
    OFFLINE,
}

export enum DeviceCapabilityType {
    COLOR_SETTING = 'color_setting',
    ON_OFF = 'on_off',
    RANGE = 'range',
    VIDEO_STREAM = 'video_stream',
}

export interface BasicDevice {
    id: string
    name: string
    state: DeviceState
    capabilities: {
        [Type in DeviceCapabilityType]?: DeviceCapabilityByType<Type>
    }
}

```

В приведённом выше коде показано, как формируется тип “базового” умного устройства. Ему задан уникальный ID, имя, состояние онлайн/оффлайн и набор способностей. В данный момент поддерживается четыре способности для устройства:

- COLOR\_SETTING – управление цветом в hsv
- ON\_OFF – переключатель вкл/выкл
- RANGE – переключение по заданному промежутку с определённым шагом

- VIDEO\_STREAM – стриминг видео, например, с наружной камеры

В приложении реализована поддержка четырех устройств: лампочка, термостат, чайник и камера. Но, с помощью такой структуры с использованием способностей, можно реализовать поддержку множества умных устройств, так как их возможности зачастую сводятся к этому набору из четырех способностей. Из нереализованных способностей осталось только задание специального сценария работы (например, включить режим дискотеки для лампочки).

Пример задания типа устройства:

```
Device.ts

export type DeviceCapabilityOnOff = {
  type: DeviceCapabilityType.ON_OFF
  state: {
    instance: 'on'
    value: boolean
  }
}

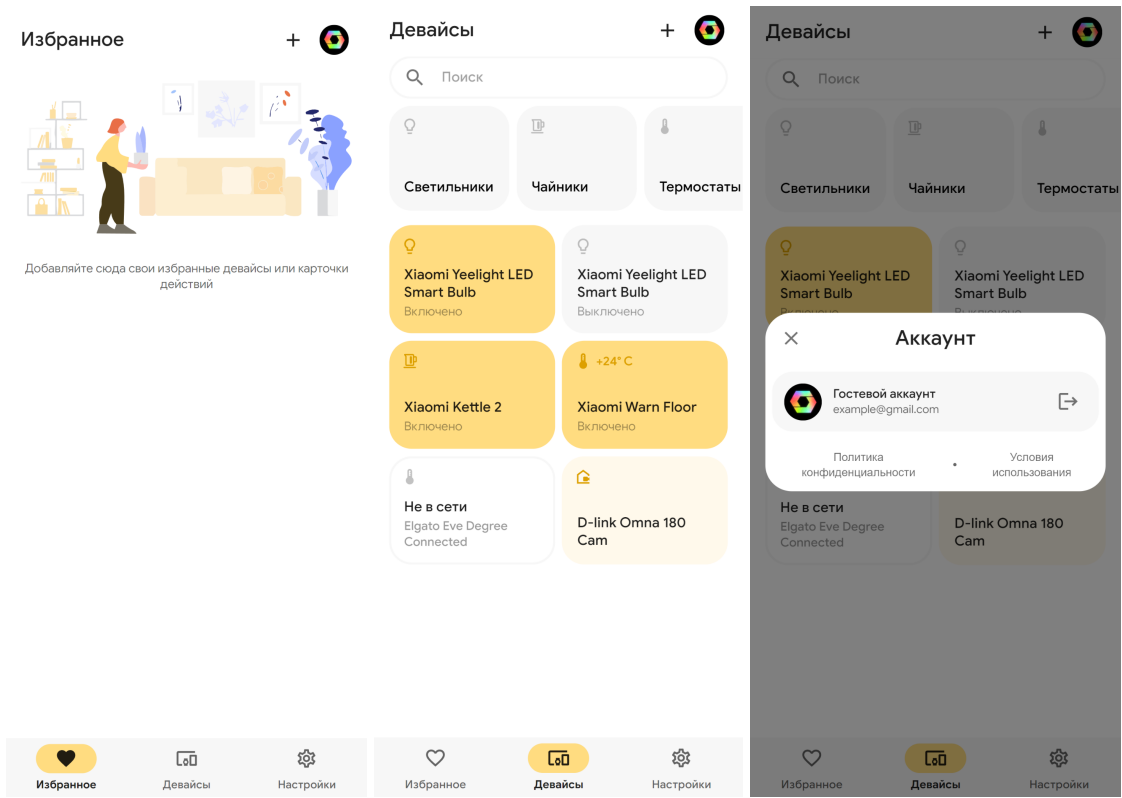
export type DeviceCapabilityColorSetting = {
  type: DeviceCapabilityType.COLOR_SETTING
  state:
    | {
      instance: 'hsv'
      value: {
        h: number
        s: number
        v: number
      }
    }
    | {
      instance: 'temperature_k'
      value: number
    }
}

export interface LightBulb extends BasicDevice {
  type: DeviceType.LIGHT_BULB
  capabilities: {
    [DeviceCapabilityType.ON_OFF]: DeviceCapabilityOnOff
    [DeviceCapabilityType.COLOR_SETTING]: DeviceCapabilityColorSetting
  }
}
```

Видно, что у умной лампочки есть две способности: включение и выключение, а также выбор цвета по HSV или по температуре цвета в кельвинах.



### 3. Создание интерфейса



Вход в аккаунт

Illustration of a person standing in front of a door. Below the illustration is a form with two input fields: 'v.ekushev@vk.team' and 'Пароль'.

Войти

Зарегистрироваться

Регистрация

Illustration of a person standing in front of a door. Below the illustration is a form with two input fields: 'v.ekushev@vk.team' and '.....'.

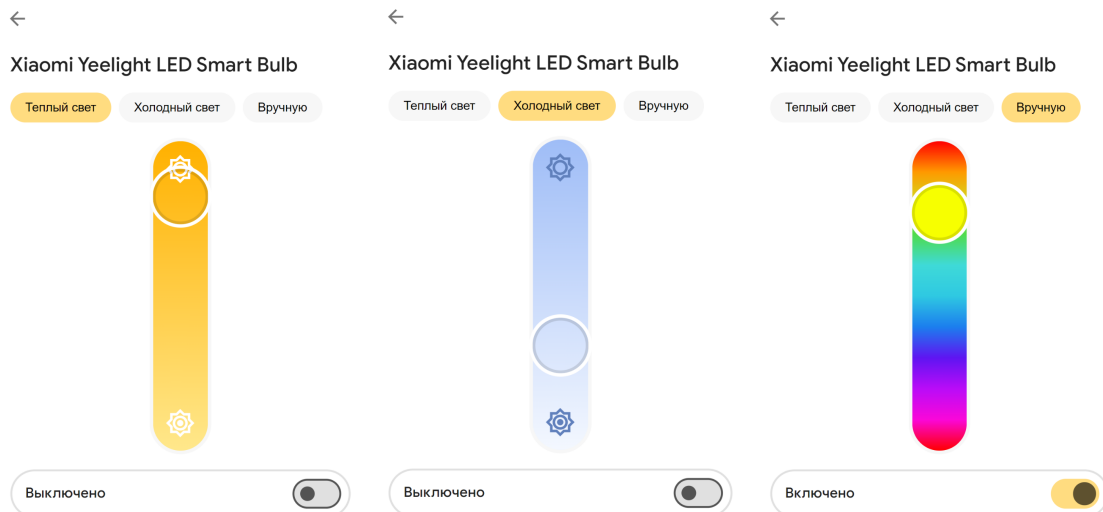
Регистрация

Войти с существующим аккаунтом

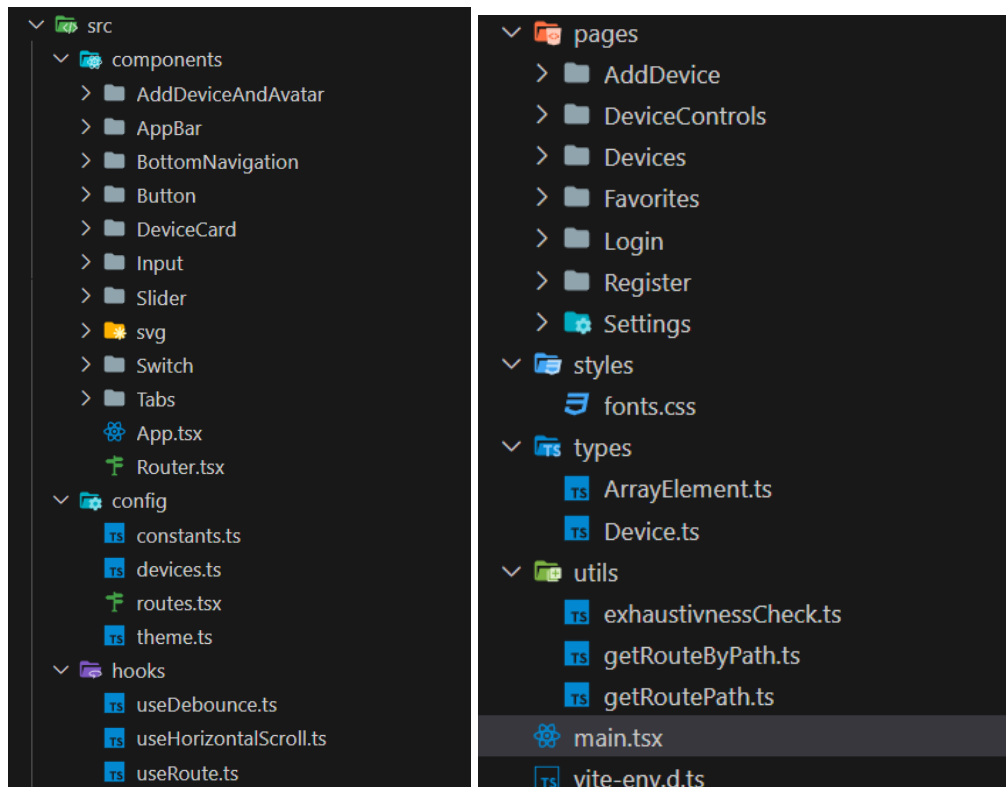
Настройки

Illustration of a person standing in front of a door. Below the illustration is a form with a single input field: '.....'. Below the form is a button 'Настройки'.

Страница в разработке



Итоговое состояние папки исходников:



Были выделены компоненты, в каждой странице также есть подэлементы, которые вынесены в свои файлы. Для данных в *src/config/devices.ts* записана фейковая информация о добавленных в умный дом девайсах.

Проработаны анимации для кнопок в BottomNavigation, карточек, просто кнопок. Планируется добавить переходы между экранами с анимацией, вдохновлённой Android 13, с использованием библиотеки Framer Motion (из неё по большей части будет главным использование компонента AnimatePresence, который анимирует удаляющиеся и добавляющиеся элементы из VDOM).

## **Вывод**

В ходе лабораторной работы был прорисован первоначальный дизайн приложения, продумана логика работы умного дома и его устройств, а также создана основа для приложения в виде выбранных фреймворков и библиотек и базовым каркасом фронтенда.