

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Фронт-энд разработка

Отчет

Домашняя работа №3

Выполнил:

Екушев Владислав

Группа К32402

Проверил:

Добряков Д. И.

Санкт-Петербург

2023 г.

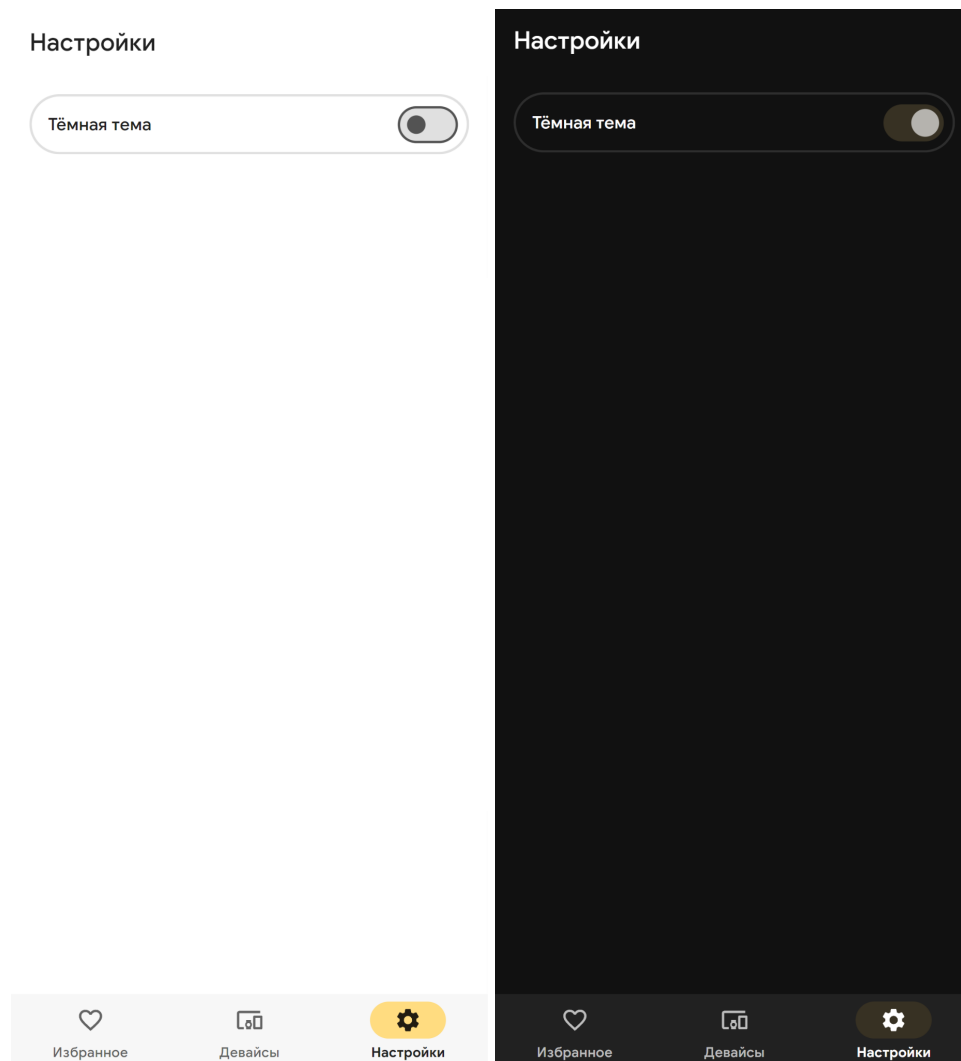
## Задача

Выполнить темизацию ранее реализованного сайта

Вариант: Разработка интерактивного сайта для управления умным домом

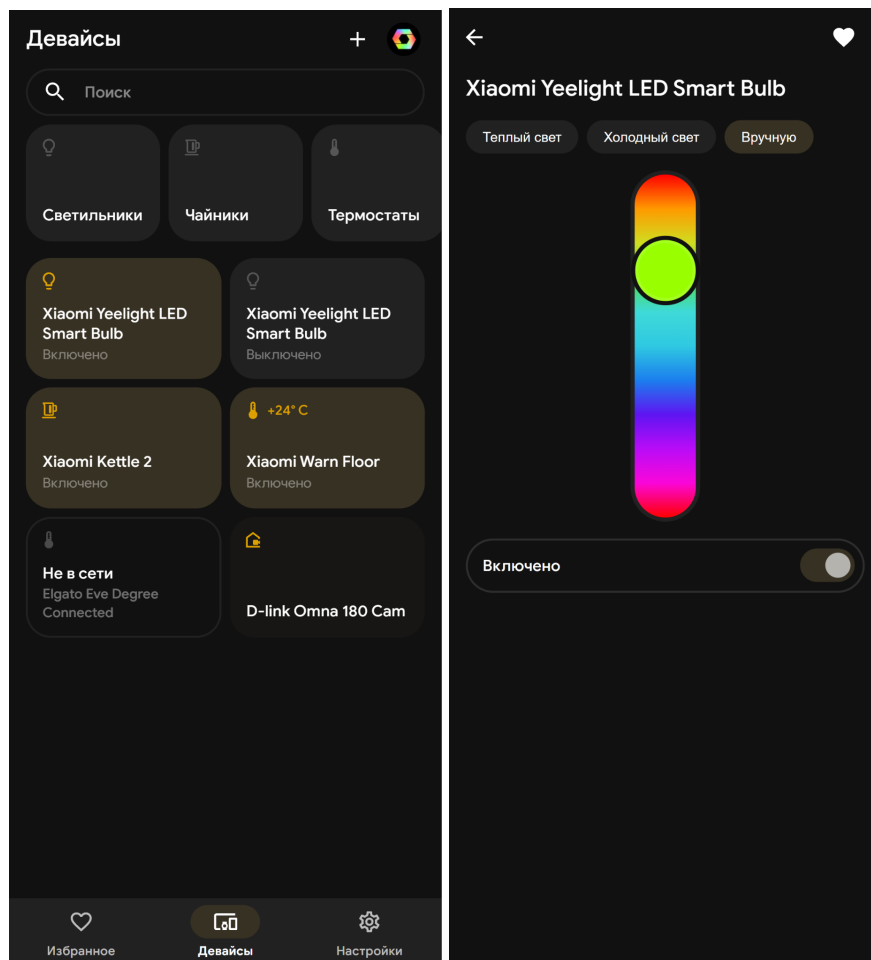
## Ход работы

Добавлен переключатель тем на страницу настроек.



Так как изначально использовалась библиотека MUI, в которой есть поддержка создания палитры для приложения, добавить новую тему было довольно легко. Нужно только следить за тем, чтобы в коде приложения везде использовались токены из палитры, а не просто цвета.

Итоговый результат темной темы:



Слайс, в котором хранится информация о текущей теме:

```
app.ts
labs > K33402 > Екушев Влад > lab1 > src > store > app.ts > ...
You, last week | 1 author (You)
1 import { createSlice, PayloadAction } from '@reduxjs/toolkit'
2 import { Theme } from 'src/types/Theme'
3
4 type AppState = {
5   theme: Theme
6 }
7
8 const initialState: AppState = {
9   theme: Theme.LIGHT,
10 }
11
12 const slice = createSlice({
13   name: 'app',
14   initialState,
15   reducers: {
16     setTheme: (state, { payload }: PayloadAction<Theme>) => {
17       state.theme = payload
18     },
19   },
20 })
21
22 export const { setTheme } = slice.actions
23
24 export default slice.reducer
25
```

Код компонента App, который оборачивает все страницы в контекст темы через ThemeProvider:

```
74 export const App: React.FC = () => {
75   const route = useRoute()
76   const user = useAppSelector((store) => store.auth.user)
77   const userFetchState = useAppSelector((store) => store.auth.state)
78   const themeName = useAppSelector((store) => store.app.theme)
79   const navigate = useNavigate()
80   const dispatch = useAppDispatch()
81   const theme = THEME_NAME_TO_MUI_THEME[themeName]
82   const shouldRenderSpinner =
83     userFetchState === FetchingState.PENDING ||
84     userFetchState === FetchingState.IDLE
85
86   > React.useEffect(() => { ...
87     }, [])
88
89   > React.useEffect(() => { ...
90     }, [navigate, user, userFetchState])
91
100
101   return (
102     <ThemeProvider theme={theme}>
103   > <...
104     </>
105   </ThemeProvider>
106 )
107 }
108
```

## Конфигурация тем:

```
import { createTheme, Theme as MUITheme } from '@mui/material'
import { CSSProperties } from 'react'
import { BUTTON_MAX_WIDTH } from './constants'
import { Theme } from 'src/types/Theme'

declare module '@mui/material' {
  interface TypeText {
    hint: string
  }
  interface Mixins {
    button: CSSProperties
  }
}

export const lightTheme = createTheme({
  palette: {
    mode: 'light',
    primary: {
      main: '#FFDC80',
      dark: '#DDA000',
    },
    secondary: {
      main: '#8AA1FF',
      light: '#BFCAFF',
    },
    // ...
  },
})

export const darkTheme = createTheme({
  palette: {
    mode: 'dark',
    primary: {
      main: '#373123',
      dark: '#DDA000',
    },
    secondary: {
      main: '#8AA1FF',
      light: '#BFCAFF',
    },
    // ...
  },
})

export const THEME_NAME_TO_MUI_THEME: Record<Theme, MUITheme> =
{ [Theme.LIGHT]: lightTheme,
  [Theme.DARK]: darkTheme,
}
```

## **Вывод**

В ходе лабораторной работы был создан дизайн для темной темы в приложении, создан стор для хранения информации о текущей теме и был реализован функционал динамического переключения темы.