

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Фронт-энд разработка

Отчет

Домашняя работа №3

Выполнил:

Рыбалко Олег

K33392

Проверил:

Добряков Д. И.

Санкт-Петербург

2023 г.

## Задание

Выполнить темизацию ранее реализованного сайта.

## Решение

В файле *index.scss* определим переменные для различных тем нашего веб-сайта.

```
src > index.scss > .btn-close
1  ∨ :root[data-theme="light"] {
2    --fg-color: black;
3    --bg-color: white;
4    --modal-bg-color: #212529;
5    --outline-color: black;
6    --profile-image-bg-color: #212529;
7  }
8
9  ∨ :root[data-theme="dark"] {
10   --fg-color: #f8f9fa;
11   --bg-color: black;
12   --modal-bg-color: #212529;
13   --outline-color: white;
14   --profile-image-bg-color: #f8f9fa;
15 }
```

Создадим реакт-контекст для того, чтобы задавать и получать данные о текущей теме из компонентов

```
1  import React from 'react'
2
3  ∨ export const ThemeContext = React.createContext<{
4    theme?: string
5    setTheme?: React.Dispatch<React.SetStateAction<string>>
6  }>({})
7
8
```

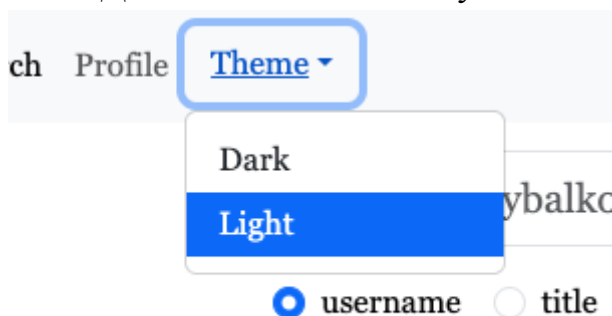
Далее создадим файл *enums.ts* в который добавим перечисление для тем

```
1  export enum Theme {
2    dark = 'dark',
3    light = 'light',
4  }
5
6
```

Также нам необходимо создать провайдер контекста, который будет хранить данные в стейте реакта и получать данные о текущей теме из localStorage.

```
1  import React, { ReactNode } from 'react'
2  import { ThemeContext } from '@/contexts/ThemeContext'
3  import { Theme } from '@/enums'
4
5  const getTheme = () => {
6    const theme = `${window?.localStorage?.getItem('theme')}`
7    if (Object.values(Theme).includes(theme as Theme)) return theme
8
9    const userMedia = window.matchMedia('(prefers-color-scheme: light)')
10   if (userMedia.matches) return Theme.light
11
12   return Theme.dark
13 }
14
15 const ThemeProvider = ({ children }: { children: ReactNode }) => {
16   const [theme, setTheme] = React.useState(getTheme)
17
18   React.useEffect(() => {
19     document.documentElement.dataset.theme = theme
20     localStorage.setItem('theme', theme)
21   }, [theme])
22
23   return (
24     <ThemeContext.Provider value={{ theme, setTheme }}>
25       {children}
26     </ThemeContext.Provider>
27   )
28 }
29
30 export default ThemeProvider
31
32
```

Для изменения темы будем использовать bootstrap выпадающее меню



Для того, чтобы тема успешно обновлялась, компонент навбара обернем в компонент получателя нашего контекста

```
return (  
  <ThemeContext.Consumer>  
    ({ { theme, setTheme } }) => (  
      <Navbar theme={theme} setTheme={setTheme} />  
    )  
  )
```

Затем в scss-файлах воспользуемся переменными для стилей и все будет работать

```
.profileImage {  
  border: none;  
  border-radius: 50%;  
  width: 200px;  
  background-color: var(--profile-image-bg-color);  
}
```

```
17 body {  
18   font-family: Georgia, serif;  
19   background: var(--bg-color);  
20   color: var(--fg-color);  
21 }  
22  
23 p {  
24   color: var(--fg-color);  
25 }  
26  
27 .modal-content {  
28   background: var(--modal-bg-color);  
29 }
```

## **Вывод**

После выполнения данной домашней работы я научился делать темизацию веб-сайта при помощи css-переменных