

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

**ОТЧЕТ
ПО ДОМАШНЕЙ РАБОТЕ №3**

Выполнила: Лапшина Екатерина

Группа: K3440

Проверил: Добряков Д. И.

Санкт-Петербург

2025 г.

Задание

Выполнить темизацию ранее реализованного сайта. Добавить к текущему варианту сайта дополнительную тему, в итоге должно получиться либо: светлая и тёмная с ориентиром на пользовательские настройки, либо две кастомные темы с переключателем через JS.

Проект: Сервис для аренды недвижимости.

Ход работы

Для реализации темизации были определены CSS-переменные в корневом селекторе `:root` для светлой темы и в селекторе `[data-bs-theme="dark"]` для тёмной темы. Это позволяет легко переключать цветовые схемы, изменяя атрибут у корневого элемента.

1. Определение CSS-переменных

В файле `css/style.css` были добавлены CSS-переменные для обеих тем. Переменные определяют основные цвета интерфейса: цвет фона, текста, заголовков, карточек и границ элементов.

```
:root {
  --primary-color: #0d6efd;
  --secondary-color: #6c757d;
  --bg-color: #ffffff;
  --text-color: #212529;
  --header-bg: #212529;
  --footer-bg: #f8f9fa;
  --card-bg: #ffffff;
  --border-color: rgba(0, 0, 0, 0.175);
}

[data-bs-theme="dark"] {
  --bg-color: #212529;
  --text-color: #f8f9fa;
  --header-bg: #1a1d20;
  --footer-bg: #2c3035;
  --card-bg: #2c3035;
  --border-color: #495057;
}

body {
  background-color: var(--bg-color);
  color: var(--text-color);
```

```
    transition: background-color 0.3s ease, color 0.3s ease;
  }
```

Использование CSS-переменных позволяет централизованно управлять цветовой схемой всего сайта. Плавный переход между темами обеспечивается свойством `transition`.

2. Реализация переключателя тем

Был создан файл `js/theme.js`, который автоматически добавляет UI переключателя на страницу и обрабатывает логику переключения. Скрипт проверяет сохраненные настройки пользователя в `localStorage` и системные предпочтения через медиа-запрос `prefers-color-scheme`.

```
document.addEventListener('DOMContentLoaded', () => {
  // Внедрение переключателя темы в DOM
  const togglerHTML = `
    <div class="theme-switch-wrapper">
      <span class="theme-icon">☀️ </span>
      <label class="theme-switch" for="checkbox">
        <input type="checkbox" id="checkbox" />
        <div class="slider round"></div>
      </label>
      <span class="theme-icon">🌙 </span>
    </div>
  `;
  document.body.insertAdjacentHTML('beforeend', togglerHTML);

  const toggleSwitch = document.querySelector( '.theme-switch
    input[type="checkbox"] '
  );
  const currentTheme = localStorage.getItem('theme');

  // Применение сохраненной темы
  if (currentTheme) {
    document.documentElement.setAttribute('data-bs-theme', currentTheme);
    if (currentTheme === 'dark') {
      toggleSwitch.checked = true;
    }
  } else {
    // Проверка системных настроек
    if (window.matchMedia && window.matchMedia('(prefers-color-scheme:
      dark)').matches) {
      document.documentElement.setAttribute('data-bs-theme', 'dark');
      toggleSwitch.checked = true;
    }
  }
}
```

```

    // Обработчик переключения темы function switchTheme(e) {
    if (e.target.checked) {
        document.documentElement.setAttribute('data-bs-theme', 'dark');
        localStorage.setItem('theme', 'dark');
    } else {
        document.documentElement.setAttribute('data-bs-theme', 'light');
        localStorage.setItem('theme', 'light');
    }
}

toggleSwitch.addEventListener('change', switchTheme, false));

```

Переключатель динамически добавляется на страницу при загрузке документа. При первом посещении сайта проверяются системные настройки пользователя, и если они указывают на предпочтение тёмной темы, она активируется автоматически. Выбор пользователя сохраняется в локальном хранилище браузера и применяется при последующих посещениях сайта.

3. Стилизация переключателя

Для переключателя были добавлены стили, позиционирующие его в правом нижнем углу экрана в виде фиксированного элемента. Переключатель имеет форму ползунка с иконками солнца и луны для интуитивного понимания функционала.

```

.theme-switch-wrapper
{ display: flex;
  align-items: center;
  position: fixed;
  bottom: 20px;
  right: 20px;
  z-index: 1000;
  background: var(--card-bg);
  padding: 10px 15px;
  border-radius: 25px;
  box-shadow: 0 2px 8px rgba(0, 0, 0, 0.2);
}

.theme-icon {
  font-size: 20px;
  margin: 0 8px;
}

.theme-switch {
  position:
  relative; width:

```

```

    50px; height:
    24px;
}

.theme-switch input {
    opacity: 0;
    width: 0;
    height: 0;
}

.slider {
    position:
    absolute; cursor:
    pointer; top: 0;
    left: 0;
    right: 0;
    bottom: 0;
    background-color: #ccc;
    transition: 0.4s;
}

.slider.round {
    border-radius: 24px;
}

.slider.round:before {
    border-radius: 50%;
    content: "";
    position: absolute;
    height: 18px;
    width: 18px;
    left: 3px;
    bottom: 3px;
    background-color: white;
    transition: 0.4s;
}

input:checked + .slider {
    background-color: #0d6efd;
}

input:checked + .slider:before {
    transform: translateX(26px);
}

```

Стили обеспечивают плавную анимацию переключения и адаптивность к выбранной теме за счет использования CSS-переменных.

4. Результат

В результате выполнения работы была реализована полноценная поддержка светлой и тёмной тем. Тема автоматически определяется на основе системных настроек пользователя при первом посещении и сохраняется при последующих визитах. Переключатель расположен в удобном месте и имеет интуитивно понятный интерфейс.

Также была улучшена читаемость элементов интерфейса в тёмной теме путем замены статичного класса `bg-light` на адаптивный `bg-body-secondary`, который автоматически подстраивается под текущую тему.

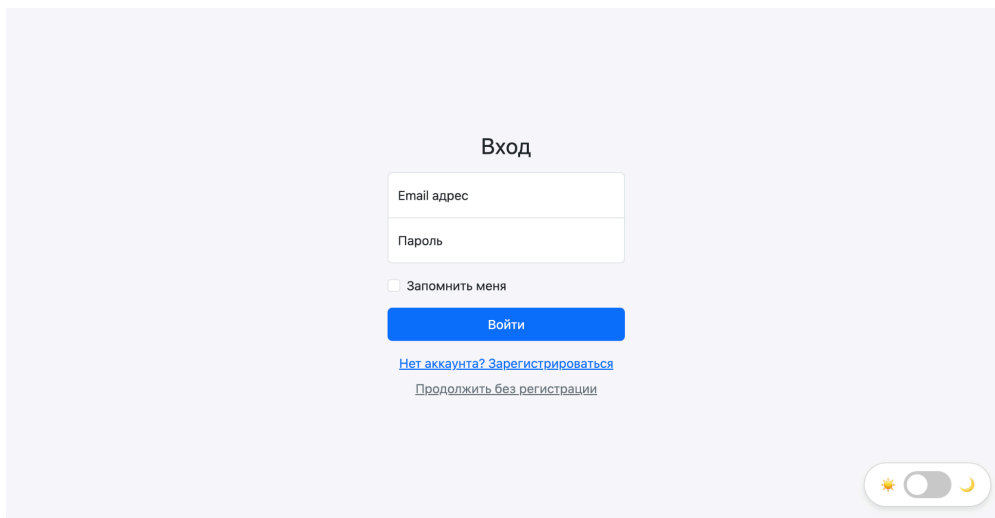


Рисунок 1 – Внешний вид сайта в светлой теме

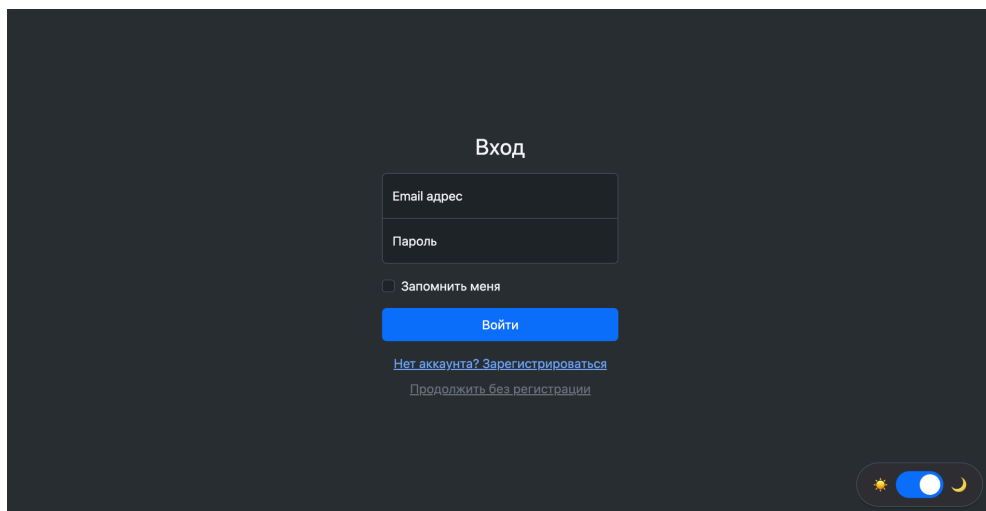


Рисунок 2 – Внешний вид сайта в тёмной теме

Вывод

В ходе выполнения домашней работы была успешно реализована темизация сайта для сервиса аренды недвижимости с использованием современных веб-технологий. Применение CSS-переменных позволило создать гибкую систему управления цветовой схемой, а JavaScript обеспечил интерактивность и сохранение пользовательских предпочтений.

Реализовано переключение между светлой и тёмной темами с автоматическим определением системных настроек пользователя и сохранением выбора в локальном хранилище браузера. Использован современный подход с атрибутами данных `data-bs-theme`, совместимый с Bootstrap 5, что обеспечивает масштабируемость и поддерживаемость кода.

Полученные навыки работы с CSS-переменными, локальным хранилищем и медиа-запросами являются важными для разработки современных адаптивных веб-приложений, ориентированных на удобство пользователей.