

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Домашняя работа 3: CSS-переменные, темизация сайта
средствами CSS

Выполнил:
Фирсов Илья
Группа К3441

Проверил:
Добряков Д. И.

Санкт-Петербург

2023 г.

Задача

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.

Ход работы

1. Система CSS-переменных для темизации

Для темизации интерфейса была использована схема с CSS Custom Properties: все ключевые параметры оформления вынесены в переменные, чтобы переключение темы происходило без переписывания стилей компонентов.

Пример (базовая тема по умолчанию):

```
:root {  
  
  --bg-main: #0f1729;  
  
  --bg-alt: #101828;  
  
  --card: rgba(255, 255, 255, 0.06);  
  
  --stroke: rgba(255, 255, 255, 0.12);  
  
  --accent: #00f2c3;  
  
  --text: #e9ecf4;  
  
  --muted: #9ea7be;  
  
  --shadow: 0 20px 60px rgba(9, 12, 26, 0.45);  
  
  color-scheme: dark;  
  
}
```

Дальше эти переменные используются в стилях элементов:

```
body {  
  
  background: var(--bg-main);  
  
  color: var(--text);  
  
}  
  
.glass-panel {  
  
  background: var(--card);  
  
  border: 1px solid var(--stroke);  
  
  box-shadow: var(--shadow);  
  
}
```

2. Описание светлой темы через data-атрибут

Дополнительная тема реализована через атрибут `data-theme="light"` на корневом элементе. Это позволяет менять значения переменных одной строкой и автоматически “перекрашивать” весь интерфейс.

Пример (светлая тема):

```
:root[data-theme="light"] {  
  
  --bg-main: #f6f8ff;  
  
  --bg-alt: #eef1fb;  
  
  --card: rgba(255, 255, 255, 0.9);  
  
  --stroke: rgba(10, 15, 30, 0.08);  
  
  --accent: #00c4a0;  
  
  --text: #0e1422;  
  
  --muted: #445066;  
  
  --shadow: 0 12px 36px rgba(12, 20, 32, 0.12);  
}
```

```
color-scheme: light;

}
```

Смена темы не требует отдельных CSS-файлов: меняется только набор переменных, а компоненты продолжают использовать `var(...)`.

3. Автовыбор темы по системным настройкам (`prefers-color-scheme`)

При первом открытии сайт выбирает тему автоматически — в зависимости от системной настройки пользователя (тёмная/светлая). Это сделано через `matchMedia`.

```
const prefersDark = window.matchMedia('(prefers-color-scheme:
dark)').matches;

const savedTheme = localStorage.getItem(THEME_KEY);

const initialTheme = savedTheme || (prefersDark ? 'dark' : 'light');

setTheme(initialTheme);
```

То есть логика такая:

если пользователь уже выбирал тему - берём из `localStorage`

если нет - ориентируемся на `prefers-color-scheme`

4. Ручной переключатель темы и сохранение выбора

В шапке страницы добавлена кнопка переключения темы. При клике:

- меняется `data-theme` на `<html>`
- выбор сохраняется в `localStorage`
- обновляются подписи и ARIA-атрибуты кнопки

Кнопка в HTML:

```
<button type="button" class="btn btn-outline-light theme-toggle"
```

```
data-theme-toggle aria-pressed="false">
```

Светлая

```
</button>
```

Логика переключения:

```
function setTheme(theme) {
```

```
  document.documentElement.setAttribute('data-theme', theme);
```

```
  localStorage.setItem(THEME_KEY, theme);
```

```
  updateThemeToggleButtons(theme);
```

```
}
```

```
document.querySelectorAll('[data-theme-toggle]').forEach((btn) => {
```

```
  btn.addEventListener('click', () => {
```

```
    const cur = document.documentElement.getAttribute('data-theme') ||  
    initialTheme;
```

```
    setTheme(cur === 'dark' ? 'light' : 'dark');
```

```
  });
```

```
});
```

5. Доступность переключателя и динамических блоков

Для улучшения доступности добавлены атрибуты для ключевых мест:

Skip-link для пропуска навигации:

```
<a class="skip-link" href="#mainContent">Перейти к основному  
содержимому</a>
```

Кнопка темы сообщает своё состояние:

```
btn.setAttribute('aria-pressed', theme === 'dark' ? 'true' : 'false');
```

```
btn.setAttribute('aria-label', `Переключить тему: сейчас ${theme === 'dark' ? 'тёмная' : 'светлая'}`);
```

Карточка деталей вакансии объявляет обновления скринридеру:

```
<div class="glass-panel p-4 mt-4" id="vacancyDetailCard"
  aria-live="polite" aria-atomic="true">
</div>
```

Для блока с деталями выбранной вакансии добавлены атрибуты `aria-live` и `aria-atomic`, что позволяет экранным дикторам корректно озвучивать изменения содержимого.

```
<div id="vacancyDetailCard"
  aria-live="polite"
  aria-atomic="true">
</div>
```

При выборе вакансии содержимое блока обновляется, и пользователь с ассистивными технологиями получает уведомление об изменениях.

6. Корректировка элементов hero-секции при смене темы

Для hero-секции были учтены возможные конфликты между кастомной темизацией и стандартными классами Bootstrap (`text-white`, `text-white-75`).

В светлой теме цвета текста переопределяются вручную:

```
:root[data-theme="light"] .hero h1,
:root[data-theme="light"] .hero p {
```

```
color: var(--text);  
  
}  
  
:root[data-theme="light"] .hero .text-white,  
:root[data-theme="light"] .hero .text-white-75 {  
    color: var(--text) !important;  
  
}
```

7. Какие именно категории токенов есть и зачем

Для удобства поддержки и расширения темизации CSS-переменные были логически сгруппированы по назначению.

Фон/поверхности: --bg-main, --bg-alt, --card, --card-soft

Границы/сетка: --stroke

Текст: --text, --muted, --text-soft

Акценты: --accent, --accent-2

Компонентные токены: --topbar-bg, --badge-bg, --badge-border, --pill-text, --tag-text

UI-геометрия: --radius, --shadow, --topbar-h

Пример из CSS:

```
:root {  
    --bg-main: #0f1729;  
    --card: rgba(255, 255, 255, 0.06);  
    --stroke: rgba(255, 255, 255, 0.12);  
    --accent: #00f2c3;  
    --text: #e9ecf4;
```

```
--radius: 18px;  
  
--shadow: 0 20px 60px rgba(9, 12, 26, 0.45);  
  
}
```

8. Итоговая интеграция темизации в компоненты (включая Bootstrap)

Темизация была встроена так, чтобы работать поверх Bootstrap-компонентов: кнопки, формы и панели используют переменные, поэтому выглядят одинаково в обеих темах.

Например, кастомизация кнопок через переменные:

```
.btn-outline-light {  
  
  color: var(--text);  
  
  border-color: var(--stroke);  
  
  background: var(--card);  
  
}  
  
.btn-outline-light:hover {  
  
  color: var(--contrast-text);  
  
  background: var(--badge-bg);  
  
  border-color: var(--accent);  
  
}
```

Формы также подхватывают тему:

```
.form-control,  
.form-select {  
  
  background: rgba(255, 255, 255, 0.06);  
  
  border: 1px solid var(--stroke);  
  
}
```



```
color: var(--text);  
  
}
```

На рисунках показаны получившиеся светлая и темная темы:

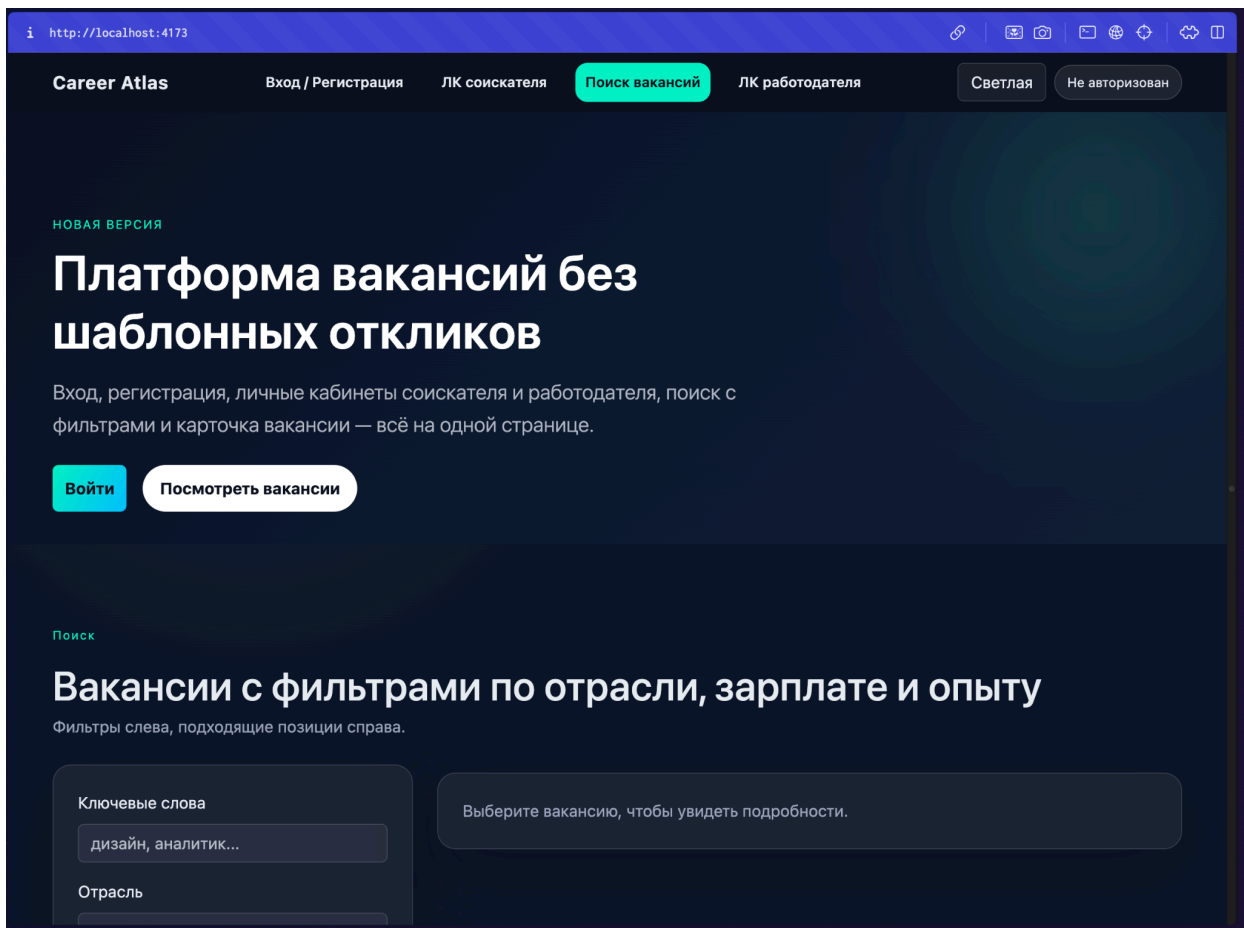


Рисунок 1 - Темная тема

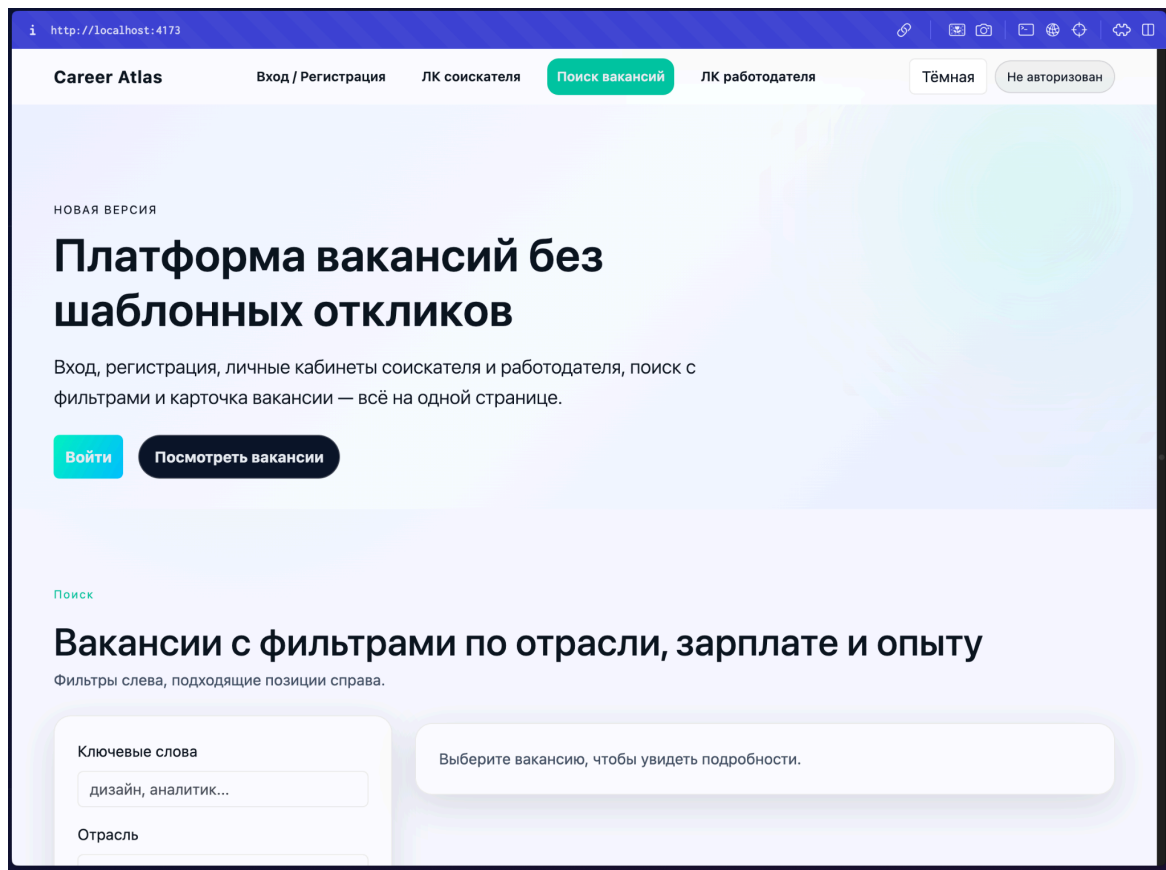


Рисунок 2 - Светлая тема

Вывод

В ходе работы реализована темизация сайта на основе CSS-переменных: создан набор базовых переменных, добавлена альтернативная (светлая) тема через `data-theme`, настроен авто-выбор темы через `prefers-color-scheme`, а также ручной переключатель с сохранением состояния в `localStorage`.