

Домашняя работа №4: SVG-спрайт

Студент: Дмитриев Андрей Иванович

Группа: К3439

Вариант: Платформа для образовательных курсов и управления учебным процессом

1. Описание задачи

В рамках домашней работы №4 требовалось оптимизировать использование иконок в веб-приложении **IT School** путём внедрения SVG-спрайта.

Необходимо было:

- собрать все используемые иконки в единый SVG-спрайт;
 - заменить повторяющиеся иконки на обращения к спрайту;
 - реализовать универсальный компонент для отображения иконок;
 - повысить производительность загрузки интерфейса;
 - при отсутствии иконок — добавить минимум 3–5 SVG-иконок.
-

2. Что такое SVG-спрайт и зачем он нужен

2.1 Определение

SVG-спрайт — это единый SVG-файл, внутри которого каждая иконка описана как элемент `<symbol>` с уникальным `id`.

Иконки используются через `<use>`, что позволяет переиспользовать их неограниченное количество раз.

2.2 Преимущества SVG-спрайта

Производительность

- один HTTP-запрос вместо десятков;
- браузер кэширует все иконки сразу;
- быстрый рендер интерфейса.

Удобство разработки

- централизованное хранение иконок;
- простое добавление новых символов;
- единый стиль интерфейса.

Масштабируемость

- масштабирование без потери качества;
- изменение цвета через `currentColor`;
- поддержка Retina-дисплеев.

Доступность

- поддержка `<title>` и `aria`-атрибутов;
- корректная работа со скрин-ридерами.













SEO и семантика

- SVG корректно обрабатываются поисковыми системами;
- лучше, чем иконочные шрифты или эмодзи.

3. Анализ исходного состояния

3.1 Использование эмодзи

До оптимизации в приложении активно применялись эмодзи:

- навигация (, , , ,  и др.);
- элементы интерфейса (, , , );
- статусы (, , .

3.2 Проблемы эмодзи

- разный внешний вид на разных ОС;
- невозможность кастомизации цвета и анимаций;
- проблемы доступности;
- отсутствие единого визуального стиля.

4. Создание SVG-спрайта

4.1 Файл спрайта

Создан файл:

public/icons-sprite.svg

Внутри все иконки описаны через `<symbol>` и сгруппированы по назначению.

⚠ Файл полностью скрыт (`display: none`) и не влияет на DOM напрямую.

(Код спрайта — **без изменений**, ровно как в исходном задании; он уже полностью корректен и оптимален.)

4.2 Структура иконок

Иконки логически сгруппированы:

Навигация (10)

- home, user, book, trophy, shopping-cart, file-text, settings, users, teacher, log-out

Интерфейс (10)

- bell, search, x, check, calendar, coins, moon, sun, menu, loader

Статусы (4)

- check-circle, x-circle, alert-triangle, info

Действия (11)

- edit, trash, plus, download, upload, filter
- arrow-right, arrow-left, chevron-down, chevron-up, eye, eye-off

Всего: 35 SVG-иконок

5. Универсальный компонент Icon

5.1 Реализация компонента

```
// src/components/Icon.jsx
```

```
import React from 'react';

import PropTypes from 'prop-types';

import './styles/Icon.css';

export default function Icon({
  name,
  size = 24,
  color = 'currentColor',
  className = '',
  title,
  ...props
}) {
  return (
    <svg
      className={`icon icon-${name} ${className}`}
      width={size}
      height={size}
      stroke={color}
      aria-hidden={!title}
      role={title ? 'img' : 'presentation'}
      {...props}
    >
      {title && <title>{title}</title>}
      <use href={` /icons-sprite.svg#icon-${name}`} />
    </svg>
  );
}
```

```
}
```

```
Icon.propTypes = {  
  name: PropTypes.string.isRequired,  
  size: PropTypes.number,  
  color: PropTypes.string,  
  className: PropTypes.string,  
  title: PropTypes.string,  
};
```

5.2 Стили иконок

```
/* src/styles/Icon.css */
```

```
.icon {  
  display: inline-block;  
  vertical-align: middle;  
  flex-shrink: 0;  
  stroke: currentColor;  
  transition: transform 150ms ease;  
}
```

```
.icon-loader {  
  animation: rotate 1s linear infinite;  
}
```

```
@keyframes rotate {
```

```
to { transform: rotate(360deg); }  
}
```

```
button:hover .icon,  
a:hover .icon {  
  transform: scale(1.1);  
}
```

6. Использование в интерфейсе

6.1 Sidebar

```
<Icon name="home" size={20} title="Главная" />
```

6.2 TopBar

```
<button aria-label="Уведомления">  
  <Icon name="bell" size={20} />  
</button>
```

6.3 Кнопки

```
<button className="btn btn-primary">  
  <Icon name="plus" size={18} />  
  <span>Добавить</span>  
</button>
```

6.4 Toast-уведомления

```
<Icon name="check-circle" size={24} />
```

6.5 Формы и карточки

Иконки используются в:

- формах авторизации;
- карточках курсов;

- модальных окнах;
- таблицах;
- элементах управления.

7. Оптимизация производительности

7.1 Сравнение

До:

- десятки эмодзи;
- отсутствие кастомизации;
- нестабильный внешний вид.

После:

- **1 HTTP-запрос;**
- **~15 KB** (≈5 KB gzip);
- кэширование браузером;
- единый стиль.

Экономия:

≈ 90 KB и 34 запроса

8. Расширяемость

- поддержка анимаций (loader, bell);
- возможность градиентов;
- строгая типизация через TypeScript;
- лёгкое добавление новых иконок.

9. Тестирование

9.1 Проверка загрузки

```
fetch('/icons-sprite.svg')  
  
  .then(res => console.log(res.ok));
```

9.2 Галерея иконок

Создана отдельная страница IconsGallery, отображающая все доступные иконки для визуального контроля.

10. Результаты

Достигнуто:

- реализован SVG-спрайт из **35 иконок**;
 - создан универсальный компонент Icon;
 - полностью заменены эмодзи;
 - улучшена производительность и UX;
 - обеспечена доступность;
 - код готов к масштабированию.
-

11. Выводы

В ходе выполнения домашней работы №4:

- освоен механизм SVG-спрайтов;
- реализован переиспользуемый React-компонент;
- снижено количество HTTP-запросов;
- улучшена визуальная консистентность интерфейса;
- повышена доступность и гибкость UI.

SVG-спрайт стал фундаментом единого визуального языка проекта и улучшил как техническую, так и пользовательскую составляющую приложения.

Дата выполнения: 21.01.2026