

Дополнительная лабораторная работа 1: Отделение стилей от компонентов (CSS Modules)

Цель работы

Научиться выносить стили из компонентов React в отдельные CSS-файлы с использованием **CSS Modules**, что обеспечивает:

- **Чистоту кода** компонентов (отделение логики от представления)
- **Локальную область видимости** стилей (стили не конфликтуют между разными компонентами)
- **Удобство поддержки** и повторного использования стилей

Что такое CSS Modules?

CSS Modules - это подход, при котором CSS-классы автоматически получают уникальные имена при компиляции, что предотвращает конфликты стилей между разными компонентами.

Как это работает:

Обычный CSS (проблема):

```
css
/* styles.css */
.button { background: blue; }
tsx
// Компонент A
import './styles.css';
<button className="button">Кнопка A</button>

// Компонент B (случайно использовали то же имя класса)
import './other-styles.css';
<button className="button">Кнопка B</button> // Стили перезапишутся!
```

CSS Modules (решение):

```
css
/* Button.module.css */
.button { background: blue; }
tsx
// Компонент A
import styles from './Button.module.css';
<button className={styles.button}>Кнопка A</button>
// Результат: <button class="Button_button_1x2y3z">Кнопка A</button>
```

```
// Компонент B (то же имя класса, но другой файл)
import styles from './OtherButton.module.css';
<button className={styles.button}>Кнопка B</button>
// Результат: <button class="OtherButton_button_4a5b6c">Кнопка B</button>
// Имена разные → стили не конфликтуют!
```

Шаг 1: Создание файлов стилей для страниц

Ожидаемый результат

После выполнения шага у вас будет структура файлов:

```
text
src/
├─ pages/
│   ├─ HomePage.tsx
│   ├─ HomePage.module.css ← новый файл
│   ├─ GalleryPage.tsx
│   ├─ GalleryPage.module.css ← новый файл
│   ├─ AddIdeaPage.tsx
│   ├─ AddIdeaPage.module.css ← новый файл
│   ├─ EditIdeaPage.tsx
│   └─ EditIdeaPage.module.css ← новый файл
├─ components/
│   ├─ Header.tsx
│   └─ Header.module.css ← новый файл
```

Действия

Создайте файлы стилей:

```
powershell
cd src
cd pages
New-Item -Path HomePage.module.css -ItemType File
New-Item -Path GalleryPage.module.css -ItemType File
New-Item -Path AddIdeaPage.module.css -ItemType File
New-Item -Path EditIdeaPage.module.css -ItemType File
cd ../components
New-Item -Path Header.module.css -ItemType File
```

Шаг 2: Стили для Header (шапка сайта)

Файл `src/components/Header.module.css`

css

```
/* Header.module.css - стили для компонента шапки */
```

```
/* Контейнер шапки */
```

```
.header {  
  background-color: #282c34;  
  padding: 10px 20px;  
  margin-bottom: 20px;  
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);  
  position: sticky;  
  top: 0;  
  z-index: 100;  
}
```

```
/* Навигационное меню */
```

```
.nav {  
  display: flex;  
  gap: 20px;  
  max-width: 1200px;  
  margin: 0 auto;  
}
```

```
/* Ссылки навигации */
```

```
.link {  
  color: white;  
  text-decoration: none;  
  font-size: 16px;  
  padding: 8px 12px;  
  border-radius: 4px;  
  transition: background-color 0.3s ease;  
}
```

```
/* Эффект при наведении на ссылку */
```

```
.link:hover {  
  background-color: #3a4049;  
  text-decoration: none;  
}
```

```
/* Активная ссылка (опционально, потребует доработки компонента) */
```

```
.linkActive {  
  background-color: #007bff;  
  color: white;  
}
```

```

/* Контейнер логотипа/бренда */
.brand {
  font-size: 20px;
  font-weight: bold;
  margin-right: auto;
  color: #61dafb;
}

/* Адаптивность для мобильных устройств */
@media (max-width: 600px) {
  .nav {
    flex-direction: column;
    gap: 8px;
  }

  .brand {
    margin-right: 0;
    text-align: center;
  }
}

```

Файл `src/components/Header.tsx` (обновленный)

```

tsx
import { Link, useLocation } from 'react-router-dom';
import styles from './Header.module.css'; // ← импорт CSS Module

export function Header() {
  const location = useLocation(); // для определения активной ссылки

  // Функция для определения активного класса
  const getLinkClass = (path: string) => {
    return location.pathname === path ? `${styles.link} ${styles.linkActive}` : styles.link;
  };

  return (
    <header className={styles.header}>
      <nav className={styles.nav}>
        <span className={styles.brand}> Блог Идеи</span>
        <Link to="/" className={getLinkClass('/')}>
          Главная
        </Link>
        <Link to="/gallery" className={getLinkClass('/gallery')}>
          Галерея идей
        </Link>
        <Link to="/add" className={getLinkClass('/add')}>

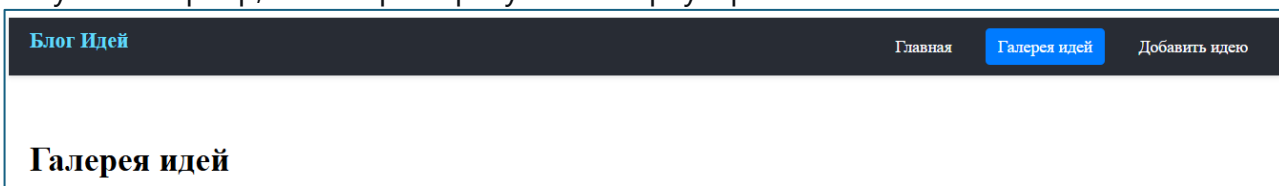
```

```
        Добавить идею
    </Link>
</nav>
</header>
);
}
```

Что изменилось:

- Вместо `style={{ }}` используем `className={styles.класс}`
- Добавлена подсветка активной ссылки
- Добавлен бренд в шапку

Запустите сервер, посмотрите результат в браузере.



Шаг 3: Стили для HomePage (главная страница)

Файл `src/pages/HomePage.module.css`

```
css
/* HomePage.module.css - стили для главной страницы */

/* Основной контейнер страницы */
.container {
  padding: 20px;
  text-align: center;
  max-width: 800px;
  margin: 0 auto;
}

/* Заголовок страницы */
.title {
  font-size: 2.5rem;
  color: #282c34;
  margin-bottom: 20px;
  animation: fadeIn 1s ease-in;
}
```

```
/* Подзаголовок/описание */
.subtitle {
  font-size: 1.2rem;
  color: #555;
  margin-bottom: 40px;
  line-height: 1.6;
}

/* Секция с информацией */
.infoSection {
  background-color: #f8f9fa;
  border-radius: 12px;
  padding: 24px;
  margin-top: 30px;
  text-align: left;
}

/* Заголовок секции */
.sectionTitle {
  color: #282c34;
  margin-bottom: 20px;
  text-align: center;
}

/* Список возможностей */
.featuresList {
  list-style: none;
  padding: 0;
  margin: 0;
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
  gap: 16px;
}

/* Элемент списка */
.featureItem {
  padding: 16px;
  background-color: white;
  border-radius: 8px;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.05);
  font-size: 1.1rem;
  transition: transform 0.2s ease, box-shadow 0.2s ease;
}

/* Эффект при наведении на элемент списка */
```

```

.featureItem:hover {
  transform: translateY(-3px);
  box-shadow: 0 4px 12px rgba(0, 0, 0, 0.1);
}

/* Кнопка призыва к действию */
.ctaButton {
  display: inline-block;
  background-color: #007bff;
  color: white;
  text-decoration: none;
  padding: 12px 24px;
  border-radius: 8px;
  font-size: 1.1rem;
  font-weight: bold;
  margin-top: 30px;
  transition: background-color 0.3s ease, transform 0.2s ease;
}

.ctaButton:hover {
  background-color: #0056b3;
  transform: scale(1.05);
}

/* Анимация появления */
@keyframes fadeIn {
  from {
    opacity: 0;
    transform: translateY(-20px);
  }
  to {
    opacity: 1;
    transform: translateY(0);
  }
}

/* Адаптивность */
@media (max-width: 600px) {
  .title {
    font-size: 1.8rem;
  }

  .subtitle {
    font-size: 1rem;
  }
}

```

```
.featuresList {  
  grid-template-columns: 1fr;  
}  
}
```

Файл `src/pages/HomePage.tsx` (обновленный)

```
tsx  
import { Link } from 'react-router-dom';  
import styles from './HomePage.module.css'; // ← импорт CSS Module  
  
function HomePage() {  
  return (  
    <div className={styles.container}>  
      <h1 className={styles.title}>Добро пожаловать в Блог Идеи!</h1>  
  
      <p className={styles.subtitle}>  
        Это место, где вы можете делиться своими мыслями,  
        находить вдохновение от других и воплощать идеи в жизнь!  
      </p>  
  
      <div className={styles.infoSection}>  
        <h3 className={styles.sectionTitle}>Что вы можете делать на сайте?</h3>  
        <ul className={styles.featuresList}>  
          <li className={styles.featureItem}> Опубликовать свои идеи</li>  
          <li className={styles.featureItem}> Смотреть идеи других пользователей</li>  
          <li className={styles.featureItem}> Редактировать свои идеи</li>  
          <li className={styles.featureItem}> Удалять идеи</li>  
        </ul>  
      </div>  
  
      <Link to="/gallery" className={styles.ctaButton}>  
        Начать просмотр идей →  
      </Link>  
    </div>  
  );  
}
```

```
export default HomePage;
```

Запустите сервер, посмотрите результат в браузере.

Шаг 4: Стили для GalleryPage (галерея идей)

Файл `src/pages/GalleryPage.module.css`

css

```
/* GalleryPage.module.css - стили для страницы галереи */
```

```
/* Основной контейнер */
```

```
.container {  
  padding: 20px;  
  max-width: 1400px;  
  margin: 0 auto;  
}
```

```
/* Заголовок страницы */
```

```
.title {  
  text-align: center;  
  font-size: 2rem;  
  color: #282c34;  
  margin-bottom: 30px;  
}
```

```
/* Индикатор загрузки */
```

```
.loading {  
  text-align: center;  
  font-size: 1.2rem;  
  color: #666;  
  padding: 40px;  
}
```

```
/* Сообщение об ошибке */
```

```
.error {  
  text-align: center;  
  color: #dc3545;  
  background-color: #f8d7da;  
  padding: 12px;  
  border-radius: 8px;  
  margin: 20px auto;  
  max-width: 500px;  
}
```

```
/* Пустое состояние (нет идей) */
```

```
.empty {  
  text-align: center;  
  font-size: 1.2rem;  
  color: #666;  
  padding: 40px;  
}
```

```

}

/* Сетка карточек */
.grid {
  display: flex;
  flex-wrap: wrap;
  gap: 24px;
  justify-content: center;
}

/* Карточка идеи */
.card {
  border: 1px solid #ddd;
  border-radius: 12px;
  padding: 20px;
  width: 300px;
  box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);
  transition: transform 0.2s ease, box-shadow 0.2s ease;
  background-color: white;
}

.card:hover {
  transform: translateY(-5px);
  box-shadow: 0 8px 20px rgba(0, 0, 0, 0.15);
}

/* Название идеи */
.cardTitle {
  margin: 0 0 12px 0;
  font-size: 1.3rem;
  color: #282c34;
  border-bottom: 2px solid #007bff;
  padding-bottom: 8px;
}

/* Информация об авторе */
.cardAuthor {
  margin: 0 0 8px 0;
  color: #555;
  font-size: 0.9rem;
}

.cardAuthor strong {
  color: #282c34;
}

```

```
/* Описание идеи */
.cardDescription {
  margin: 0 0 20px 0;
  color: #666;
  line-height: 1.5;
  font-size: 0.95rem;
}

/* Контейнер для кнопок */
.cardActions {
  display: flex;
  gap: 10px;
  margin-top: 16px;
}

/* Кнопка редактирования */
.editButton {
  padding: 8px 16px;
  cursor: pointer;
  background-color: #007bff;
  color: white;
  border: none;
  border-radius: 6px;
  font-size: 0.9rem;
  transition: background-color 0.2s ease;
  text-decoration: none;
  display: inline-block;
  text-align: center;
}

.editButton:hover {
  background-color: #0056b3;
}

/* Кнопка удаления */
.deleteButton {
  padding: 8px 16px;
  cursor: pointer;
  background-color: #dc3545;
  color: white;
  border: none;
  border-radius: 6px;
  font-size: 0.9rem;
  transition: background-color 0.2s ease;
}
```

```

}

.deleteButton:hover {
  background-color: #c82333;
}

/* Адаптивность */
@media (max-width: 700px) {
  .card {
    width: calc(100% - 40px);
  }

  .grid {
    justify-content: stretch;
  }
}

```

Файл `src/pages/GalleryPage.tsx` (обновленный)

```

tsx
import { useState, useEffect } from 'react';
import { Link } from 'react-router-dom';
import type { Idea } from '../types';
import styles from './GalleryPage.module.css'; // ← импорт CSS Module

function GalleryPage() {
  const [ideas, setIdeas] = useState<Idea[]>([]);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState('');

  const loadIdeas = async () => {
    try {
      setLoading(true);
      const response = await fetch('http://localhost:3001/ideas');

      if (!response.ok) {
        throw new Error(`Ошибка HTTP: ${response.status}`);
      }

      const data = await response.json();
      setIdeas(data);
    } catch (err) {
      setError('Не удалось загрузить идеи. Убедитесь, что сервер запущен.');
```

```

};

const deleteIdea = async (id: number) => {
  if (!confirm('Вы уверены, что хотите удалить эту идею?')) {
    return;
  }

  try {
    const response = await fetch(`http://localhost:3001/ideas/${id}`, {
      method: 'DELETE',
    });

    if (!response.ok) {
      throw new Error('Ошибка при удалении');
    }

    setIdeas(prevIdeas => prevIdeas.filter(idea => idea.id !== id));
  } catch (err) {
    alert('Не удалось удалить идею');
    console.error(err);
  }
};

useEffect(() => {
  loadIdeas();
}, []);

if (loading) return <div className={styles.loading}>Загрузка идей...</div>;
if (error) return <div className={styles.error}>{error}</div>;

return (
  <div className={styles.container}>
    <h1 className={styles.title}>Галерея идей</h1>

    {ideas.length === 0 ? (
      <div className={styles.empty}>
        <p>Пока нет идей. Будьте первым, кто добавит идею!</p>
        <Link to="/add" className={styles.editButton}>Добавить первую идею</Link>
      </div>
    ) : (
      <div className={styles.grid}>
        {ideas.map((idea) => (
          <div key={idea.id} className={styles.card}>
            <h3 className={styles.cardTitle}>{idea.name}</h3>
            <p className={styles.cardAuthor}>

```

```

    <strong>Автор:</strong> {idea.nick}
  </p>
  <p className={styles.cardDescription}>{idea.description}</p>
  <div className={styles.cardActions}>
    <Link to={`/edit/${idea.id}`} className={styles.editButton}>
      Редактировать
    </Link>
    <button
      onClick={() => deleteIdea(idea.id)}
      className={styles.deleteButton}
    >
      Удалить
    </button>
  </div>
</div>
  )})
</div>
})
</div>
);
}

export default GalleryPage;

```

Шаг 5: Стили для AddIdeaPage (форма добавления)

Файл `src/pages/AddIdeaPage.module.css`

```

css
/* AddIdeaPage.module.css - стили для формы добавления идеи */

/* Основной контейнер */
.container {
  max-width: 600px;
  margin: 40px auto;
  padding: 30px;
  background-color: white;
  border-radius: 16px;
  box-shadow: 0 4px 20px rgba(0, 0, 0, 0.1);
}

/* Заголовок страницы */
.title {

```

```

text-align: center;
font-size: 1.8rem;
color: #282c34;
margin-bottom: 30px;
}

/* Сообщение об ошибке */
.error {
background-color: #f8d7da;
color: #721c24;
padding: 12px 16px;
border-radius: 8px;
margin-bottom: 20px;
border-left: 4px solid #dc3545;
}

/* Группа полей формы */
.formGroup {
margin-bottom: 20px;
}

/* Метка (Label) */
.label {
display: block;
margin-bottom: 8px;
font-weight: bold;
color: #333;
}

/* Поле ввода */
.input {
width: 100%;
padding: 12px;
border: 1px solid #ccc;
border-radius: 8px;
font-size: 16px;
transition: border-color 0.2s ease, box-shadow 0.2s ease;
box-sizing: border-box;
}

.input:focus {
outline: none;
border-color: #007bff;
box-shadow: 0 0 0 3px rgba(0, 123, 255, 0.1);
}

```

```

.input:disabled {
  background-color: #e9ecef;
  cursor: not-allowed;
}

/* Текстовое поле (textarea) */
.textarea {
  width: 100%;
  padding: 12px;
  border: 1px solid #ccc;
  border-radius: 8px;
  font-size: 16px;
  font-family: inherit;
  transition: border-color 0.2s ease, box-shadow 0.2s ease;
  box-sizing: border-box;
  resize: vertical;
}

.textarea:focus {
  outline: none;
  border-color: #007bff;
  box-shadow: 0 0 0 3px rgba(0, 123, 255, 0.1);
}

.textarea:disabled {
  background-color: #e9ecef;
  cursor: not-allowed;
}

/* Кнопка отправки */
.submitButton {
  width: 100%;
  background-color: #28a745;
  color: white;
  border: none;
  padding: 12px 24px;
  border-radius: 8px;
  font-size: 18px;
  font-weight: bold;
  cursor: pointer;
  transition: background-color 0.2s ease, transform 0.2s ease;
  margin-top: 10px;
}

```

```
.submitButton:hover:not(:disabled) {
  background-color: #218838;
  transform: scale(1.02);
}
```

```
.submitButton:disabled {
  background-color: #6c757d;
  cursor: not-allowed;
  opacity: 0.7;
}
```

```
/* Адаптивность */
```

```
@media (max-width: 700px) {
  .container {
    margin: 20px;
    padding: 20px;
  }
}
```

```
.title {
  font-size: 1.5rem;
}
}
```

Файл `src/pages/AddIdeaPage.tsx` (обновленный)

tsx

```
import { useState } from 'react';
import { useNavigate } from 'react-router-dom';
import styles from './AddIdeaPage.module.css'; // ← импорт CSS Module
```

```
function AddIdeaPage() {
  const [nick, setNick] = useState('');
  const [name, setName] = useState('');
  const [description, setDescription] = useState('');
  const [isSubmitting, setIsSubmitting] = useState(false);
  const [error, setError] = useState('');
  const navigate = useNavigate();

  const handleSubmit = async (e: React.FormEvent) => {
    e.preventDefault();
    setIsSubmitting(true);
    setError('');

    try {
      const response = await fetch('http://localhost:3001/ideas', {
        method: 'POST',
        headers: {
```

```

    'Content-Type': 'application/json',
  },
  body: JSON.stringify({
    nick,
    name,
    description,
  }),
});

if (!response.ok) {
  const errorData = await response.json();
  throw new Error(errorData.message || 'Ошибка при добавлении идеи');
}

navigate('/gallery');
} catch (err) {
  setError(err instanceof Error ? err.message : 'Произошла ошибка');
} finally {
  setIsSubmitting(false);
}
};

return (
  <div className={styles.container}>
    <h1 className={styles.title}>Добавить новую идею</h1>

    {error && (
      <div className={styles.error}>
        {error}
      </div>
    )}

    <form onSubmit={handleSubmit}>
      <div className={styles.formGroup}>
        <label className={styles.label}>Ваш никнейм:</label>
        <input
          type="text"
          value={nick}
          onChange={(e) => setNick(e.target.value)}
          required
          disabled={isSubmitting}
          className={styles.input}
          placeholder="Введите ваш никнейм"
        />
      </div>
    </form>
  </div>
);

```

```

<div className={styles.formGroup}>
  <label className={styles.label}>Название идеи:</label>
  <input
    type="text"
    value={name}
    onChange={(e) => setName(e.target.value)}
    required
    disabled={isSubmitting}
    className={styles.input}
    placeholder="Краткое название вашей идеи"
  />
</div>

<div className={styles.formGroup}>
  <label className={styles.label}>Описание идеи:</label>
  <textarea
    value={description}
    onChange={(e) => setDescription(e.target.value)}
    required
    rows={5}
    disabled={isSubmitting}
    className={styles.textarea}
    placeholder="Подробно опишите вашу идею..."
  />
</div>

<button
  type="submit"
  disabled={isSubmitting}
  className={styles.submitButton}
  >
  {isSubmitting ? 'Отправка...' : 'Опубликовать идею'}
</button>
</form>
</div>
);
}

export default AddIdeaPage;

```

Шаг 6: Стили для EditIdeaPage (форма редактирования)

Файл `src/pages/EditIdeaPage.module.css`

css

```
/* EditIdeaPage.module.css - стили для формы редактирования */
```

```
/* Основной контейнер */
```

```
.container {  
  max-width: 600px;  
  margin: 40px auto;  
  padding: 30px;  
  background-color: white;  
  border-radius: 16px;  
  box-shadow: 0 4px 20px rgba(0, 0, 0, 0.1);  
}
```

```
/* Заголовок страницы */
```

```
.title {  
  text-align: center;  
  font-size: 1.8rem;  
  color: #282c34;  
  margin-bottom: 10px;  
}
```

```
/* Подзаголовок с ID идеи */
```

```
.subtitle {  
  text-align: center;  
  color: #6c757d;  
  margin-bottom: 30px;  
  font-size: 0.9rem;  
}
```

```
/* Индикатор загрузки */
```

```
.loading {  
  text-align: center;  
  font-size: 1.2rem;  
  color: #666;  
  padding: 50px;  
}
```

```
/* Сообщение об ошибке */
```

```
.error {  
  background-color: #f8d7da;  
  color: #721c24;  
  padding: 12px 16px;  
  border-radius: 8px;  
}
```

```
margin-bottom: 20px;
border-left: 4px solid #dc3545;
text-align: center;
}

/* Группа полей формы */
.formGroup {
margin-bottom: 20px;
}

/* Метка (Label) */
.label {
display: block;
margin-bottom: 8px;
font-weight: bold;
color: #333;
}

/* Поле ввода */
.input {
width: 100%;
padding: 12px;
border: 1px solid #ccc;
border-radius: 8px;
font-size: 16px;
transition: border-color 0.2s ease, box-shadow 0.2s ease;
box-sizing: border-box;
}

.input:focus {
outline: none;
border-color: #ffc107;
box-shadow: 0 0 0 3px rgba(255, 193, 7, 0.1);
}

.input:disabled {
background-color: #e9ecef;
cursor: not-allowed;
}

/* Текстовое поле */
.textarea {
width: 100%;
padding: 12px;
border: 1px solid #ccc;
```

```

border-radius: 8px;
font-size: 16px;
font-family: inherit;
transition: border-color 0.2s ease, box-shadow 0.2s ease;
box-sizing: border-box;
resize: vertical;
}

.textarea:focus {
  outline: none;
  border-color: #ffc107;
  box-shadow: 0 0 0 3px rgba(255, 193, 7, 0.1);
}

.textarea:disabled {
  background-color: #e9ecef;
  cursor: not-allowed;
}

/* Контейнер для кнопок */
.actions {
  display: flex;
  gap: 12px;
  margin-top: 20px;
}

/* Кнопка сохранения */
.saveButton {
  flex: 1;
  background-color: #ffc107;
  color: #282c34;
  border: none;
  padding: 12px 24px;
  border-radius: 8px;
  font-size: 16px;
  font-weight: bold;
  cursor: pointer;
  transition: background-color 0.2s ease, transform 0.2s ease;
}

.saveButton:hover:not(:disabled) {
  background-color: #e0a800;
  transform: scale(1.02);
}

```

```

.saveButton:disabled {
  background-color: #e9ecef;
  color: #6c757d;
  cursor: not-allowed;
}

/* Кнопка отмены */
.cancelButton {
  flex: 1;
  background-color: #6c757d;
  color: white;
  border: none;
  padding: 12px 24px;
  border-radius: 8px;
  font-size: 16px;
  font-weight: bold;
  cursor: pointer;
  transition: background-color 0.2s ease;
}

.cancelButton:hover {
  background-color: #5a6268;
}

/* Адаптивность */
@media (max-width: 700px) {
  .container {
    margin: 20px;
    padding: 20px;
  }

  .title {
    font-size: 1.5rem;
  }

  .actions {
    flex-direction: column;
  }
}

```

Файл `src/pages/EditIdeaPage.tsx` (обновленный)

```

tsx
import { useState, useEffect } from 'react';
import { useParams, useNavigate } from 'react-router-dom';
import styles from './EditIdeaPage.module.css'; // ← импорт CSS Module

```

```

function EditIdeaPage() {
  const { id } = useParams<{ id: string }>();
  const navigate = useNavigate();
  const [nick, setNick] = useState('');
  const [name, setName] = useState('');
  const [description, setDescription] = useState('');
  const [loading, setLoading] = useState(true);
  const [isSubmitting, setIsSubmitting] = useState(false);
  const [error, setError] = useState('');

  useEffect(() => {
    const loadIdea = async () => {
      try {
        setLoading(true);
        const response = await fetch(`http://localhost:3001/ideas/${id}`);

        if (!response.ok) {
          throw new Error('Идея не найдена');
        }

        const data = await response.json();
        setNick(data.nick);
        setName(data.name);
        setDescription(data.description);
      } catch (err) {
        setError(err instanceof Error ? err.message : 'Ошибка загрузки');
      } finally {
        setLoading(false);
      }
    };

    if (id) {
      loadIdea();
    }
  }, [id]);

  const handleSubmit = async (e: React.FormEvent) => {
    e.preventDefault();
    setIsSubmitting(true);
    setError('');

    try {
      const response = await fetch(`http://localhost:3001/ideas/${id}`, {
        method: 'PUT',
        headers: {

```

```

    'Content-Type': 'application/json',
  },
  body: JSON.stringify({
    nick,
    name,
    description,
  }),
});

if (!response.ok) {
  const errorData = await response.json();
  throw new Error(errorData.message || 'Ошибка при обновлении');
}

navigate('/gallery');
} catch (err) {
  setError(err instanceof Error ? err.message : 'Произошла ошибка');
} finally {
  setIsSubmitting(false);
}
};

if (loading) return <div className={styles.loading}>Загрузка данных идеи...</div>;
if (error) return <div className={styles.error}>{error}</div>;

return (
  <div className={styles.container}>
    <h1 className={styles.title}>Редактирование идеи</h1>
    <div className={styles.subtitle}>ID: {id}</div>

    <form onSubmit={handleSubmit}>
      <div className={styles.formGroup}>
        <label className={styles.label}>Никнейм:</label>
        <input
          type="text"
          value={nick}
          onChange={(e) => setNick(e.target.value)}
          required
          disabled={isSubmitting}
          className={styles.input}
        />
      </div>

      <div className={styles.formGroup}>
        <label className={styles.label}>Название:</label>

```

```

<input
  type="text"
  value={name}
  onChange={(e) => setName(e.target.value)}
  required
  disabled={isSubmitting}
  className={styles.input}
/>
</div>

<div className={styles.formGroup}>
  <label className={styles.label}>Описание:</label>
  <textarea
    value={description}
    onChange={(e) => setDescription(e.target.value)}
    required
    rows={5}
    disabled={isSubmitting}
    className={styles.textarea}
  />
</div>

<div className={styles.actions}>
  <button
    type="submit"
    disabled={isSubmitting}
    className={styles.saveButton}
  >
    {isSubmitting ? 'Сохранение...' : '📄 Сохранить изменения'}
  </button>

  <button
    type="button"
    onClick={() => navigate('/gallery')}
    className={styles.cancelButton}
  >
    Отмена
  </button>
</div>
</form>
</div>
);
}

```

```
export default EditIdeaPage;
```

Шаг 7: Проверка работы

Действия для проверки:

1. Запустите бэкенд:

```
powershell
cd blog-ideas-backend
pnpm run dev
```

2. Запустите фронтенд (в новом терминале):

```
powershell
cd blog-ideas
pnpm run dev
```

3. Откройте браузер: <http://localhost:5173>

4. Проверьте визуально:

- Главная страница - приветствие с анимацией, карточки возможностей
- Шапка - навигация с подсветкой активной ссылки
- Галерея - карточки идей с эффектами при наведении
- Формы - красивые поля ввода с анимацией фокуса

5. Проверьте функциональность:

- Создание новой идеи
- Редактирование идеи
- Удаление идеи
- Навигация между страницами

Что должно получиться:

Страница	Внешний вид
Главная	Центрированный контент, анимированный заголовок, список возможностей в виде карточек, кнопка призыва к действию
Галерея	Сетка карточек, карточки увеличиваются при наведении, кнопки действий стилизованы
Добавление	Карточка формы с тенью, поля с анимацией фокуса, широкая кнопка отправки
Редактирование	Аналогично добавлению, но с желтой темой кнопки сохранения
Шапка	Темный фон, ссылки подсвечиваются при наведении, активная ссылка синяя

Итоговая структура проекта

```
text
blog-ideas/
├─ src/
│  ├─ components/
│  │  └─ Header.tsx
│  │  └─ Header.module.css      ← стили шапки
│  └─ pages/
│     ├─ HomePage.tsx
│     ├─ HomePage.module.css    ← стили главной
│     ├─ GalleryPage.tsx
│     ├─ GalleryPage.module.css ← стили галереи
│     ├─ AddIdeaPage.tsx
│     ├─ AddIdeaPage.module.css ← стили формы добавления
│     └─ EditIdeaPage.tsx
│        └─ EditIdeaPage.module.css ← стили формы редактирования
├─ App.tsx
├─ main.tsx
├─ types.ts
├─ index.html
├─ package.json
└─ tsconfig.json
```

Преимущества проделанной работы

1. **Чистота кода** - JSX-файлы содержат только разметку и логику
2. **Переиспользование стилей** - CSS-классы можно использовать в других компонентах
3. **Изоляция стилей** - CSS Modules гарантирует, что стили не конфликтуют
4. **Удобство поддержки** - дизайнер может править CSS, не трогая JSX
5. **Адаптивность** - добавлены медиа-запросы для мобильных устройств
6. **Анимации** - добавлены эффекты наведения и появления

РЕЗУЛЬТАТ: Все стили успешно вынесены в отдельные CSS-файлы с использованием CSS Modules. Приложение выглядит современно и профессионально!