Лекция 12
Расширенные возможности React Router (продолжение)

Изучите возможности использования строк запроса для обмена и управления состоянием приложения через URL-адреса, работайте с различными типами данных в параметрах запроса и используйте React Router для навигации и эффектов в зависимости от изменений URL-адреса.

1. Финальное приложение

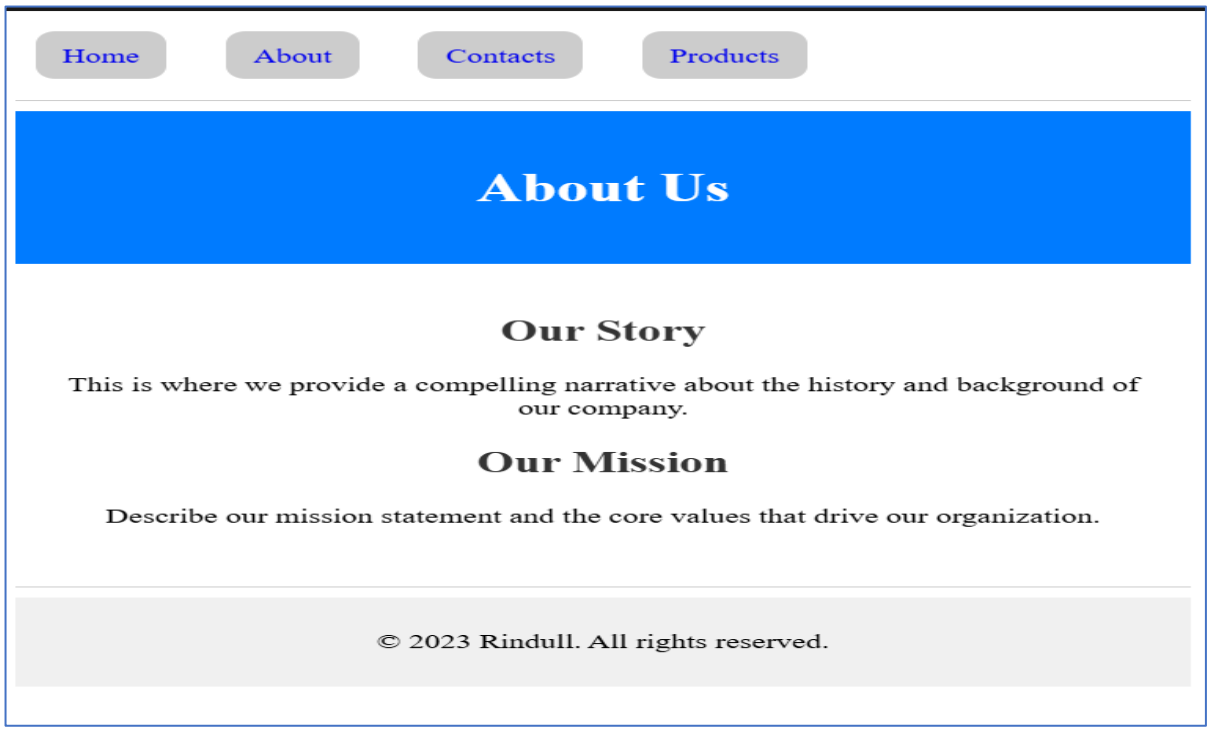
Мы успешно интегрировали функциональность параметров запроса в наше приложение, обеспечив динамическое взаимодействие и возможность обмена данными через URL. Давайте подведем итоги:

– **Четыре страницы**: Теперь наше приложение имеет четыре страницы, каждая из которых выполняет свою отдельную функцию. От главной страницы до страницы «Продукты» каждый компонент играет роль в формировании пользовательского опыта;

– **Параметры запроса**: Изучив каждую страницу, вы можете увидеть, как параметры запроса влияют на контент и функциональность, особенно на странице «Товары», где представлена динамическая фильтрация на основе параметров запроса;

– **Взаимодействие с пользователем**: Изучите, как изменения в строке запроса приводят к обновлениям в приложении в режиме реального времени, улучшая пользовательский опыт;

– **Практические занятия**: Примените полученные знания в своих проектах, экспериментируя с параметрами запросов для создания привлекательных веб-приложений, основанных на данных.



Скачайте и откройте приложение в среде Visual Studio Code (по ссылке https://codesandbox.io/p/sandbox/7894f928-2969-4f12-8184-f463ebee27c9-dcydq9?from-embed)

2. Настройка компонента React

Давайте вспомним основные концепции, которые вы изучили:

– **Базовая маршрутизация приложений**: React Router позволяет определять маршруты в вашем приложении. Вы используете <BrowserRouter> компонент для обертывания вашего приложения, а <Route> компонент — для указания путей и соответствующих компонентов, которые будут отображаться, когда эти пути соответствуют текущему URL-адресу;

– **Ленивая загрузка**: Вы уже изучили концепцию ленивой загрузки — метода асинхронной загрузки компонентов или маршрутов только при необходимости. С помощью этого метода React.lazy() вы можете динамически импортировать компоненты, повышая производительность и минимизируя время первоначальной загрузки;

– **Навигация**: React Router предоставляет различные компоненты навигации, включая <Link> и `<NavLink>. Эти компоненты позволяют создавать ссылки на различные маршруты внутри вашего приложения. Когда пользователи нажимают на эти ссылки, React Router обрабатывает обновления URL-адресов и отображает соответствующие компоненты.

Освоив эти базовые понятия, вы заложили прочный фундамент для разработки динамичных и удобных для навигации приложений React. Но не останавливайтесь на достигнутом! Официальная документация React Router поможет вам разобраться с любыми дополнительными понятиями.