Державний вищий навчальний заклад

Ужгородський національний університет

Факультет інформаційних технологій

Кафедра програмного забезпечення систем

**ЛАБОРАТОРНА РОБОТА № 5**

**Тема:** Реалізація фронт-енд частини. Ч2 – маршрутизація

Виконав студент:

ІІІ курсу спеціальності «Інженерія програмного забезпечення»

Нищий Богдан Вячеславович

**Ужгород-2024**

**Some news project**

1. Підключити react-router або @tanstack/react-router

2. Реалізувати маршрутизацію у прєкті (користувач повинен мати можливість навігації між сторінками)

3. Створити пул-реквест

4. Оформити відповідного взірця звіт

**Хід роботи**

Веб платформа some news для перегляду новин на різні теми, розробку своїх новин і статей, поширення цих статей для інших користувачів.

Кожен користувач може вподобати статті для подальшого прочитання, а також зарепостити їх для інших користувачів, якщо новина справила враження.

Новини на різні теми будуть генеруватись з допомогою news.api, а також будуть публікуватись користувачами в будь-який момент.



Рис. 1. Підключена залежність для react-router.

Для організації раутінгу було обрано BrowserRouter.



Рис. 2. Створений BrowserRouter для маршрутизації між відповідними сторінками проєкту.

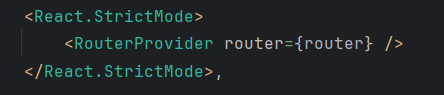


Рис. 3. Підключення router до react проєкту.

За що відповідають маршрути проєкту:

«/login» та «/register» ведуть на сторінки авторизації та реєстрації відповідно.

«/» та «/home» ведуть на сторінку Home.

Далі «/sport», «/tech», «/science», «/community» ведуть на відповідні їм сторінки зі статтями (Sport.jsx, Technology.jsx, Science.jsx, Community.jsx).

«/profile» приведе користувача в його профіль (Profile.jsx).

Важливо зауважити, що навігація по сторінкам можлива лише у випадку авторизації користувача.

«/user-articles» – маршрут до сторінки статей користувача (UserArticles.jsx).

«/new-article» маршрут до сторінки створення користувацької статті (NewArticle.jsx).

«/user-full-article» маршрут до сторінки, яка відповідає за повний вигляд конкретної користувацької статті (UserFullArticle.jsx).

Кожна відповідна сторінка має свій набір CSS стилів, а також використовує глобальні стилі.

Для навігації користувача між сторінками використовується тег <Link>:

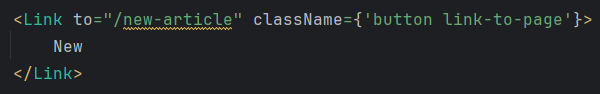


Рис. 4. Використання тега <Link> для переходу на сторінку створення нової статті користувача.

Атрибут **to** працює за принципом **href** для тегу <a>.

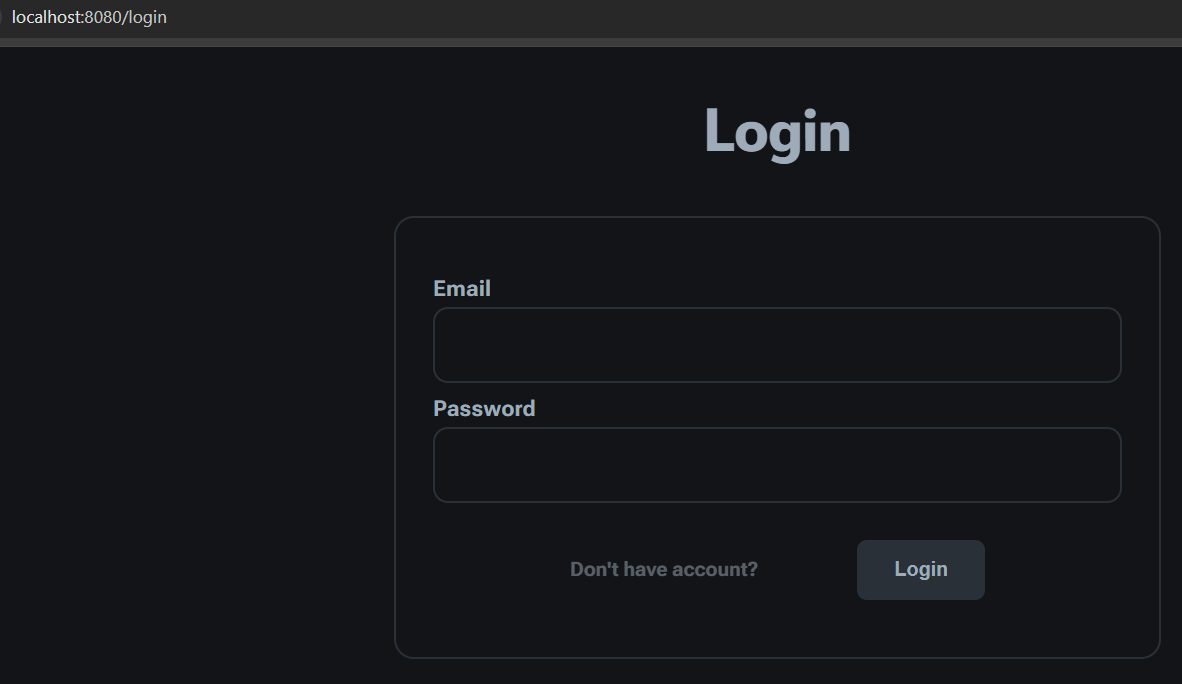


Рис. 5. Сторінка, що зустрічає користувача перед авторизацією.

Прикладом використання тега <Link> та маршрутизації між сторінками є сторінка авторизації, з якої користувач може потрапити на сторінку реєстрації простим натисканням кнопки-посилання «Don’t have account?». Фактично це і є <Link>, що при натисканні переносить сесію користувача на сторінку «/register».

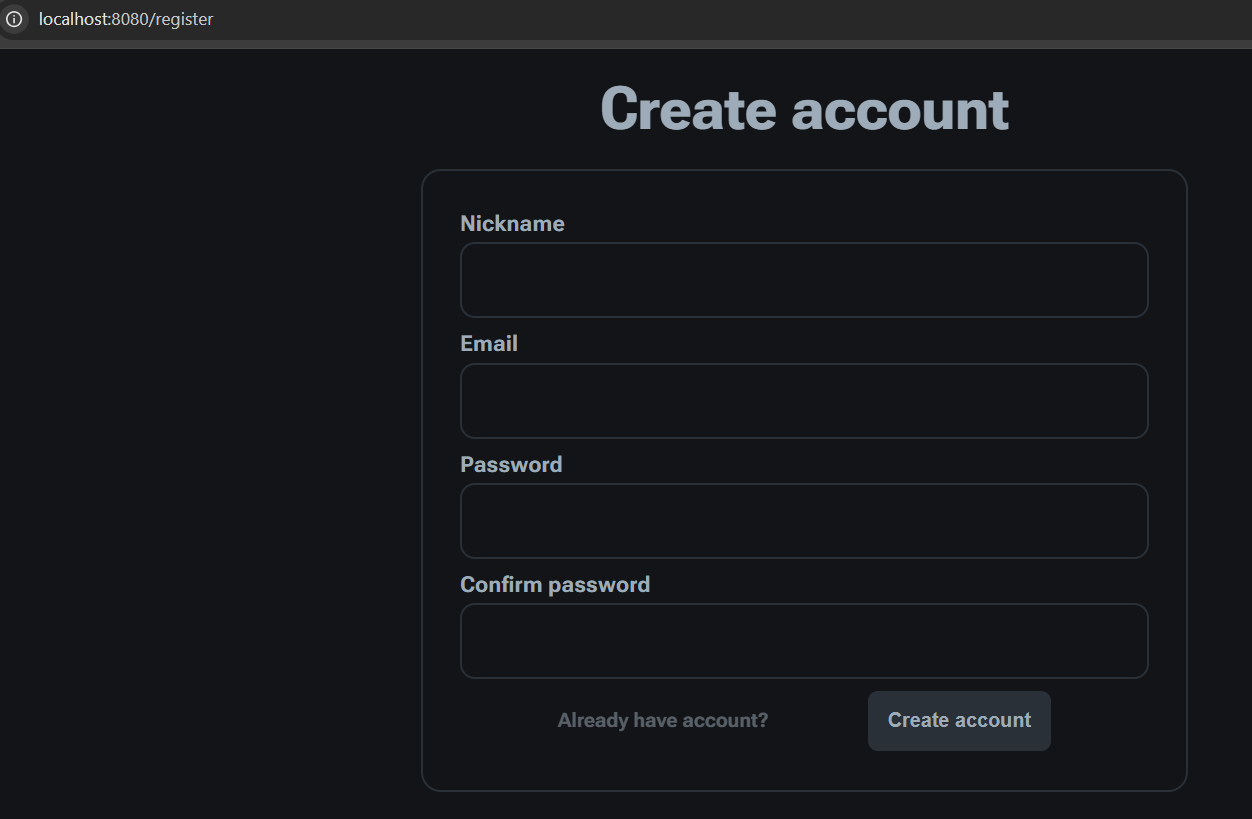


Рис. 6. Результат натискання кнопки переходу на сторінку реєстрації.

Тут, кнопка «Already have account?» працює так само, як попередня, лиш вона перенаправляє користувача на сторінку авторизації.

Повернувшись на сторінку авторизації, користувач може увійти в свій акаунт.

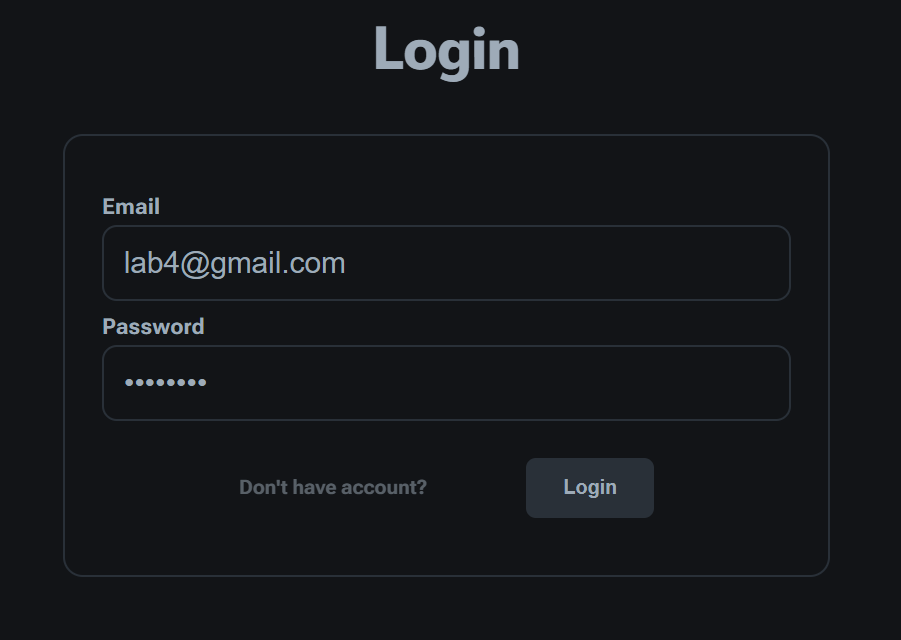


Рис. 7. Процес авторизації.

Якщо користувач ввів дані для авторизації вірно, процес авторизації вважається успішним і користувача перенаправляє на сторінку «/home».

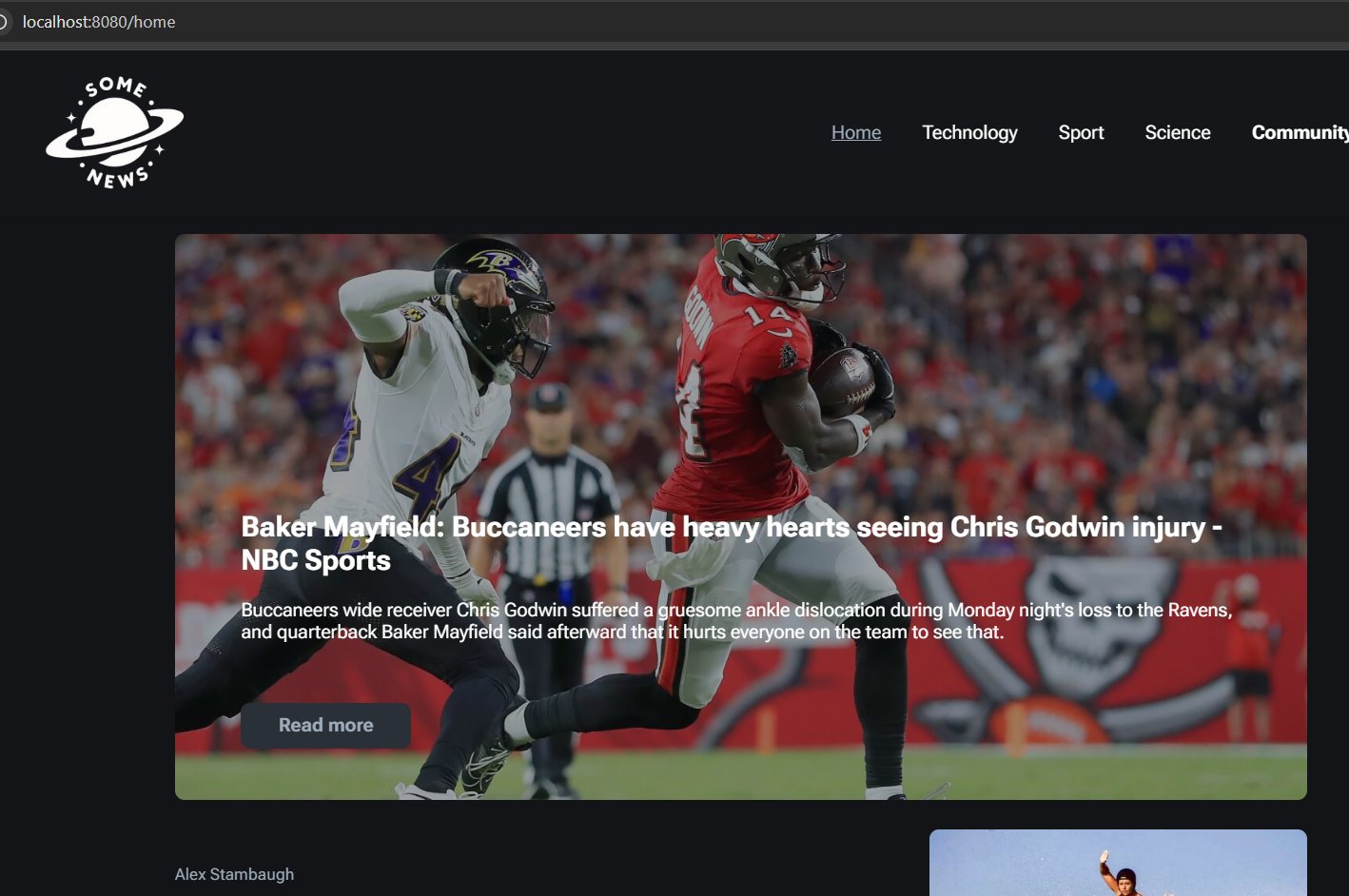


Рис. 8. Результат успішної авторизації.

Висновок: Було представлено маршрутизацію проєкту some news, виконану з допомогою react-router, визначено, які маршрути відповідають за конкретні сторінки проєкту, та розказано, як відбувається перехід між сторінками з допомогою тегу <Link>, що є частиною залежності react-router-dom.