Державний вищий навчальний заклад

Ужгородський національний університет

Факультет інформаційних технологій

Кафедра програмного забезпечення систем

**ЛАБОРАТОРНА РОБОТА № 3**

**Тема:** Каркас (структура) проєкту

Виконав студент:

ІІІ курсу спеціальності «Інженерія програмного забезпечення»

Нищий Богдан Вячеславович

**Ужгород-2024**

**Some news project**

Для зарахування роботи, студент повинен оформити відповідного взірця звіт, який повинен містити:

1. Детальний опис ідеї з відповідними ілюстраціями (необхідно, щоб зрозуміти структуру папок/файлів)
2. Представлення структури проєкту, директорій...
3. Обгрунтувати обране рішення (по архітектурі)
4. Представити стек проєкту.
5. Детальне обгрунтування у випадку відхилення від запропонованої на заняттях архітектури.
6. Опублікувати всі зміни на GitHub
7. Створити PullRequest (нас додати як переглядачів (reviewers))
8. ВИСНОВКИ

**Хід роботи**

Веб платформа 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.

**Реєстрація та авторизація**

На початку роботи з веб-додатком, користувача зустрічає вікно авторизації, де він може ввести дані для входу в свій акаунт, якщо він вже існує в системі. Дані для авторизації складають електронна пошта, яка використовувалась при реєстрації, та пароль для входу в акаунт. Передбачається перевірка на правильність введення електронної пошти.

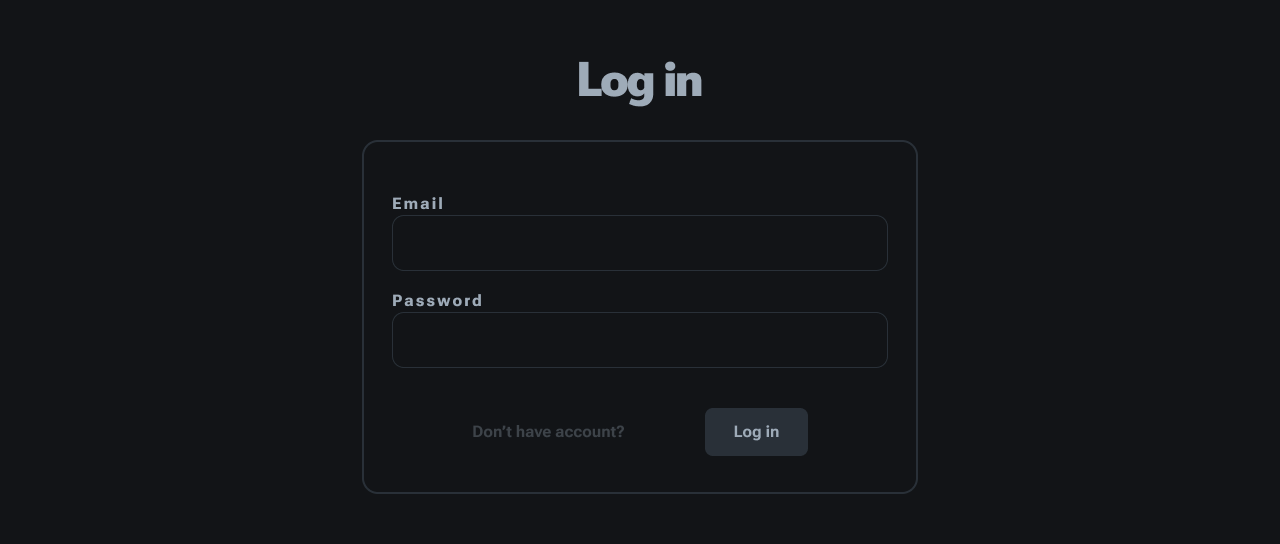


Рис. 1. Логін сторінка.

Не маючи акаунта, користувач може натиснути на кнопку «Don’t have account?» щоб перейти до вікна реєстрації, де додаток запропонує користувачу створити акаунт. В поле nickname користувач має ввести ім’я, яке буде відображатись у його профілі та використовуватись для позначення автора статей. Наступне поле – електронна адреса, що буде використовуватись в акаунті. Два наступні поля призначенні для вписування паролю для входу під час авторизації та підтвердження введеного паролю (відповідно). Якщо ж користувач випадково опинився на сторінці для реєстрації, він може повернутись до авторизації з допомогою кнопки «Already have account?».

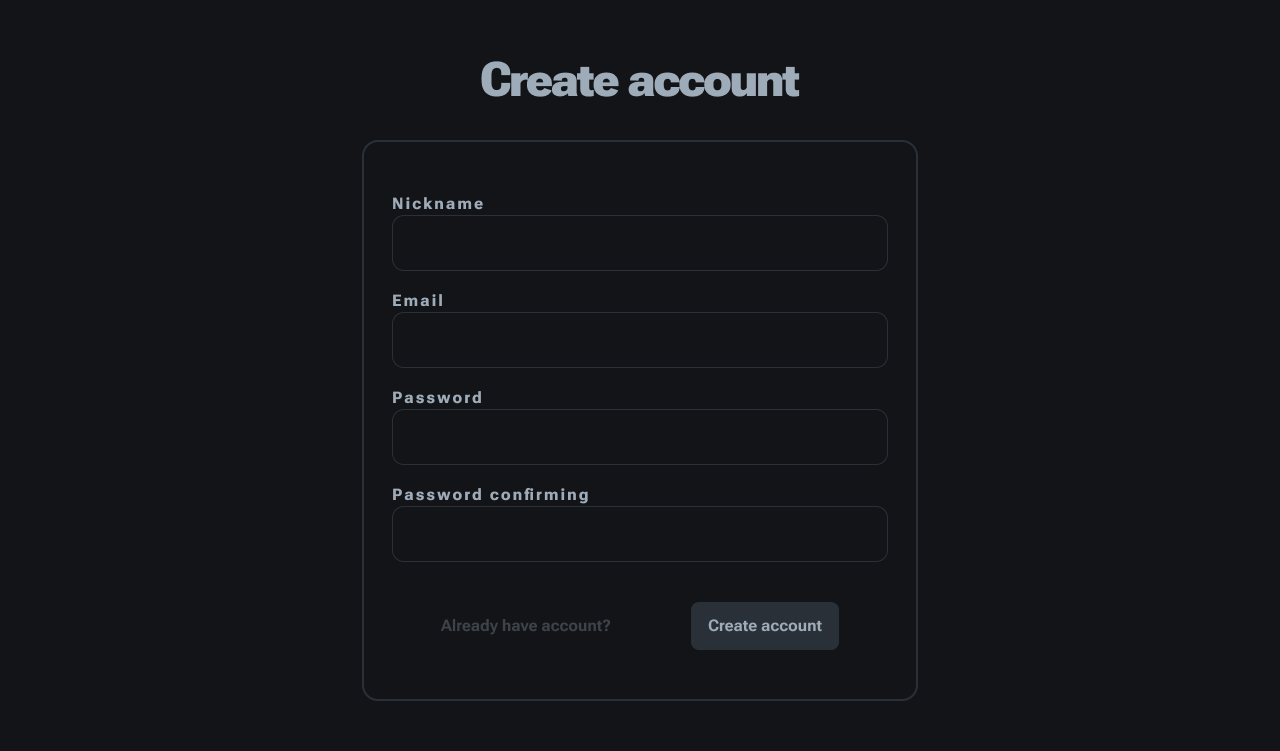


Рис. 2. Сторінка реєстрації.

**Сесія користувача**

Після авторизації чи реєстрації користувач опиняється на головній сторінці, де відображені загальні новини та статті з різних тем. Користувач може відкрити будь-яку та прочитати її докладніше.

У верхній частині будь-якої сторінки додатку є навігація. Тут з лівого краю назва проекту, а праворуч посилання на інші сторінки додатку, в яких знаходяться тематичні новини та статті. Крайнє посилання веде на сторінку репостів спільноти. Там користувач зможе знайти найпопулярніші статті, які зарепостили багато інших користувачів.

Крайню праву частину панелі навігації займає посилання на профіль користувача. Це посилання представлене фотографією, яку користувач поставив як фото профілю.

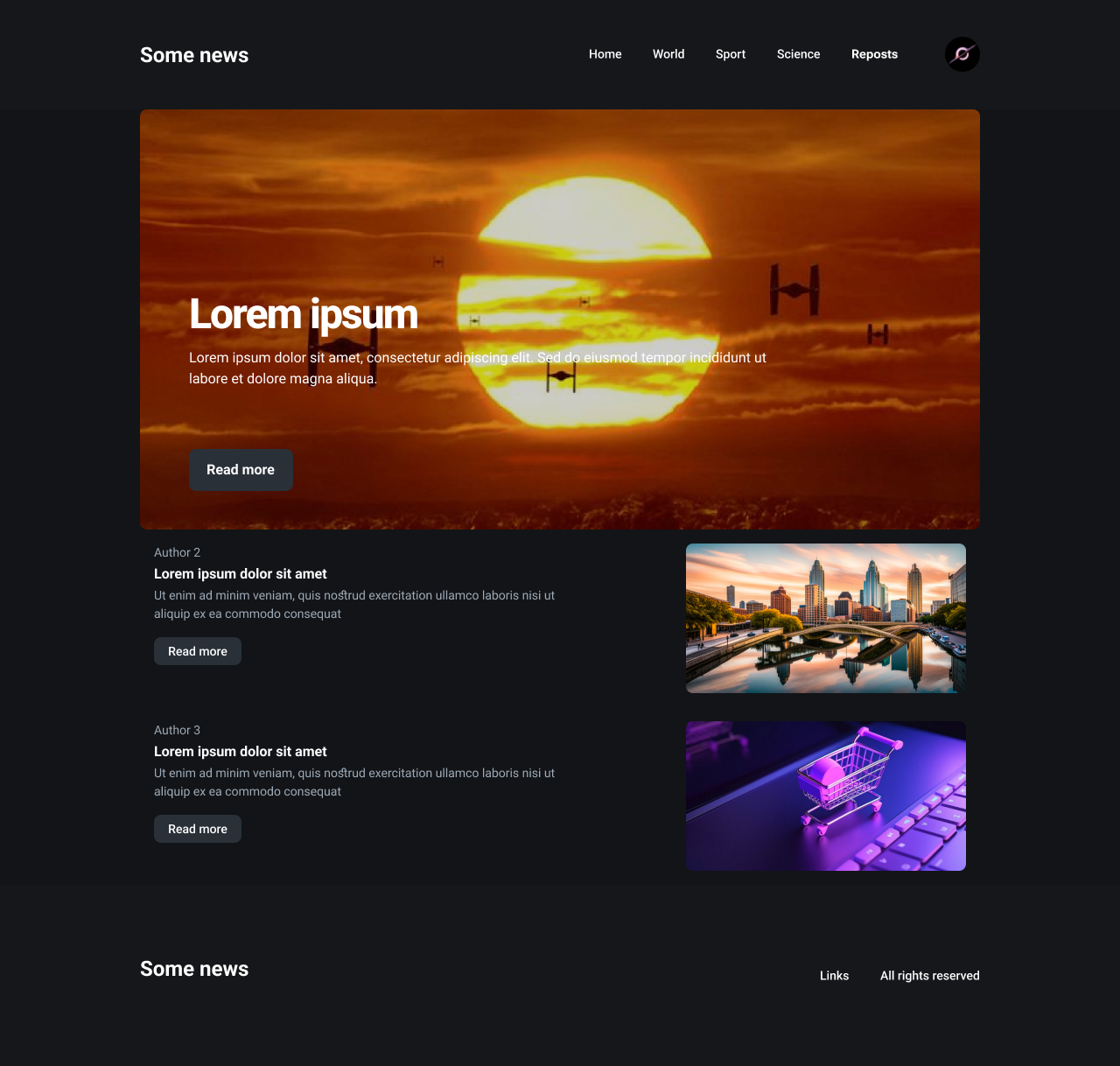


Рис. 3. Головна сторінка веб-додатку.

Припустимо користувачу закортіло переглянути одну з новин, які запропонував додаток. Він вибрав її використовуючи кнопку «Read more».

Перед ним з’явилась сторінка конкретної новини, яку він тепер може читати, репостити, або лайкати, якщо вона йому сподобалась. Він може дізнатись, скільки репостів цієї новини було зроблено, поглянувши лівіше від кнопки «Repost», а також знизу тексту статті поглянути, хто є автором.

Варто зазначити, що і тут користувача супроводжують хедер і футер.

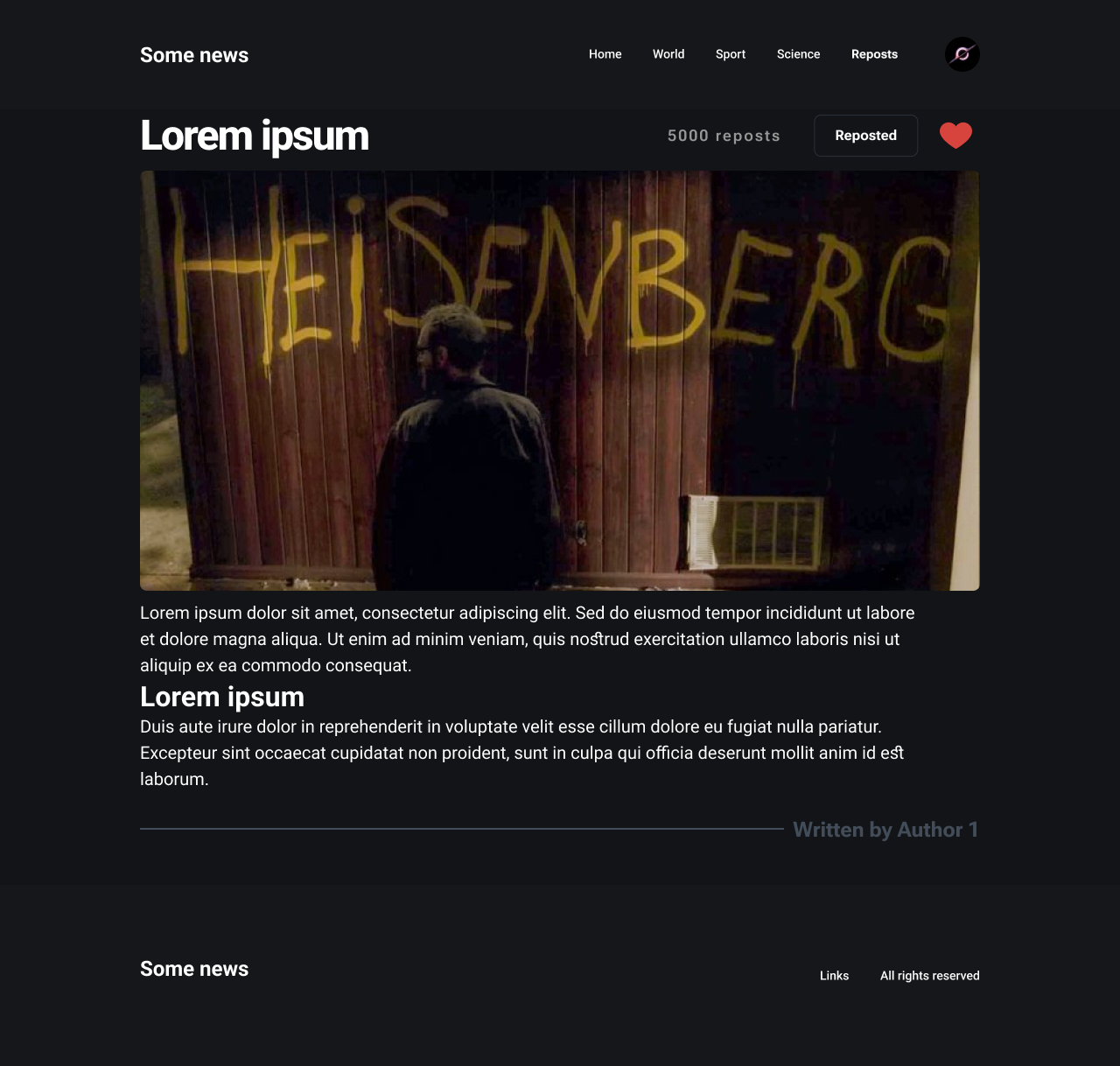


Рис. 4. Сторінка конкретної новини.

**Профіль, репости, лайки, новини користувача**

Після перегляду новини, користувач вирішив перейти в свій профіль. Він натискає на своє фото профілю у панелі навігації та його перекидує на сторінку профілю.

Тут користувач може помітити своє фото профілю, а поруч з ним nickname, який він вписував під час реєстрації. Нижче є три кнопки.

1. «Reposts» перекидує на сторінку новин, репост яких зробив користувач. Тут він може переглянути їх знову;
2. «Likes» перекидує на сторінку новин, які користувач вподобав. Зовні вона не відрізняється від сторінки «Reposts», проте різниця в тому, що інші користувачі не можуть бачити лайки одне одного.
3. «News» перекидує на сторінку новин, автором яких є сам користувач. Тут, з допомогою кнопки «New» він може створити нову статтю для публікації. Для створення такої статті користувачу необхідно написати текст, який задовільняє правило про мінімальну та максимальну кількість символів. Така стаття може супроводжуватись фото.

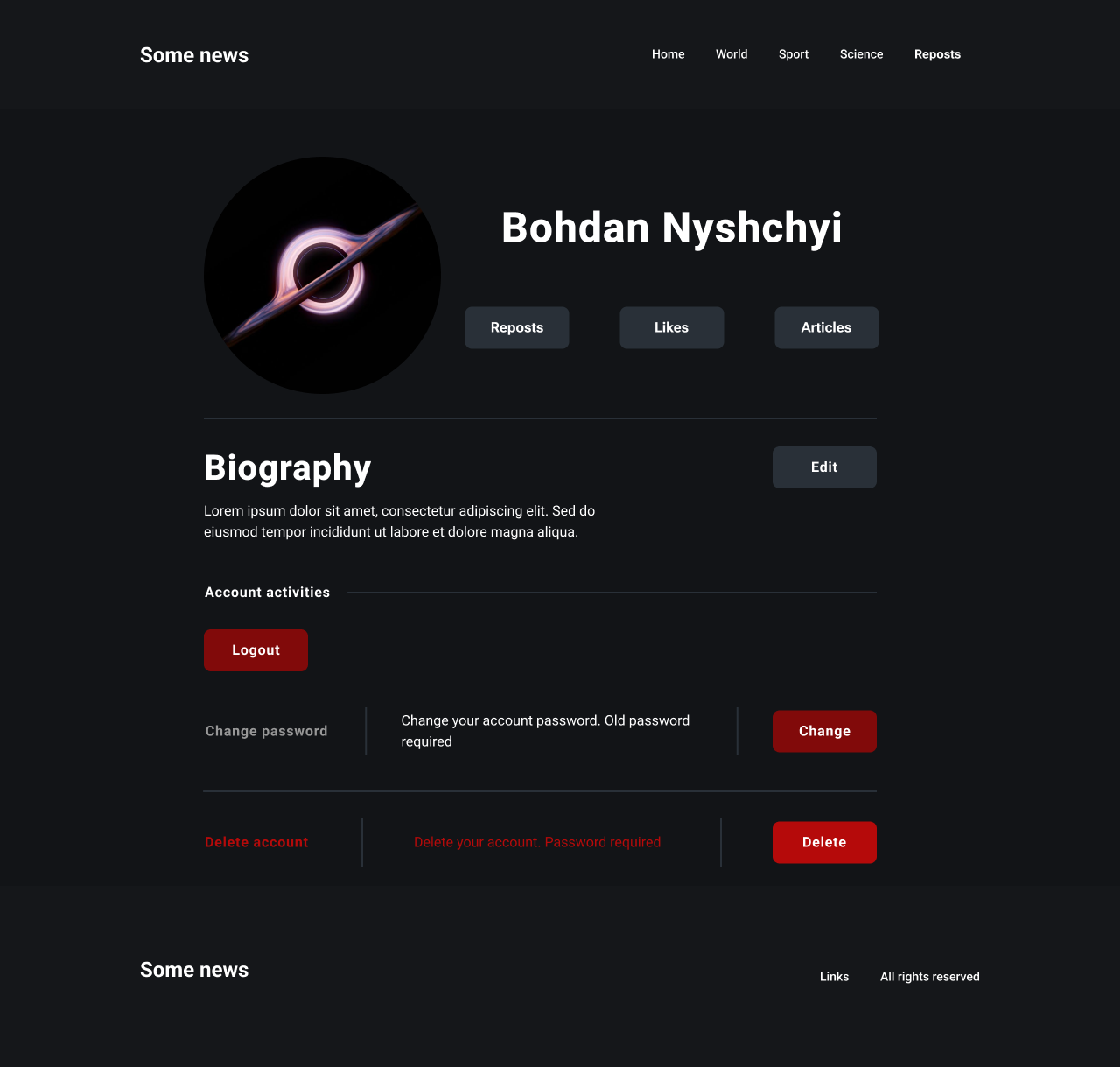


Рис. 5. Сторінка профілю.

Далі внизу знаходиться місце для опису акаунта чи біографії користувача. Його можна відредагувати в будь який момент, натиснувши кнопку «Edit».

Остання частина сторінки профілю представляє собою керування акаунтом. Користувач може вийти з акаунта з допомогою кнопки «Logout».

Далі, для зміни паролю користувач може натиснути кнопку «Change».

Зрештою, для видалення акаунта користувач може натиснути кнопку «Delete». Звісно додаток запитає підтвердження такої дії.

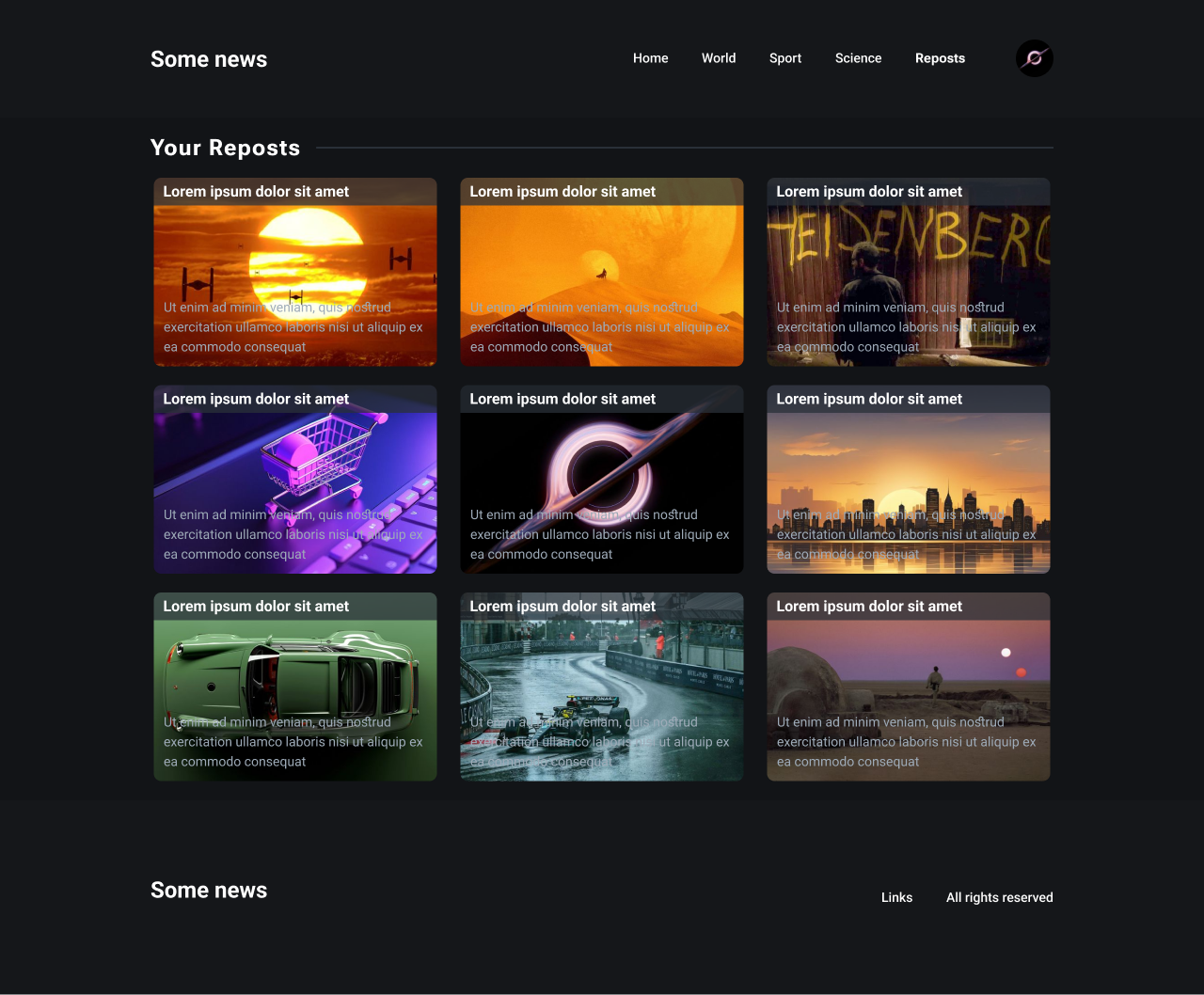


Рис. 6. Сторінка репостів користувача.

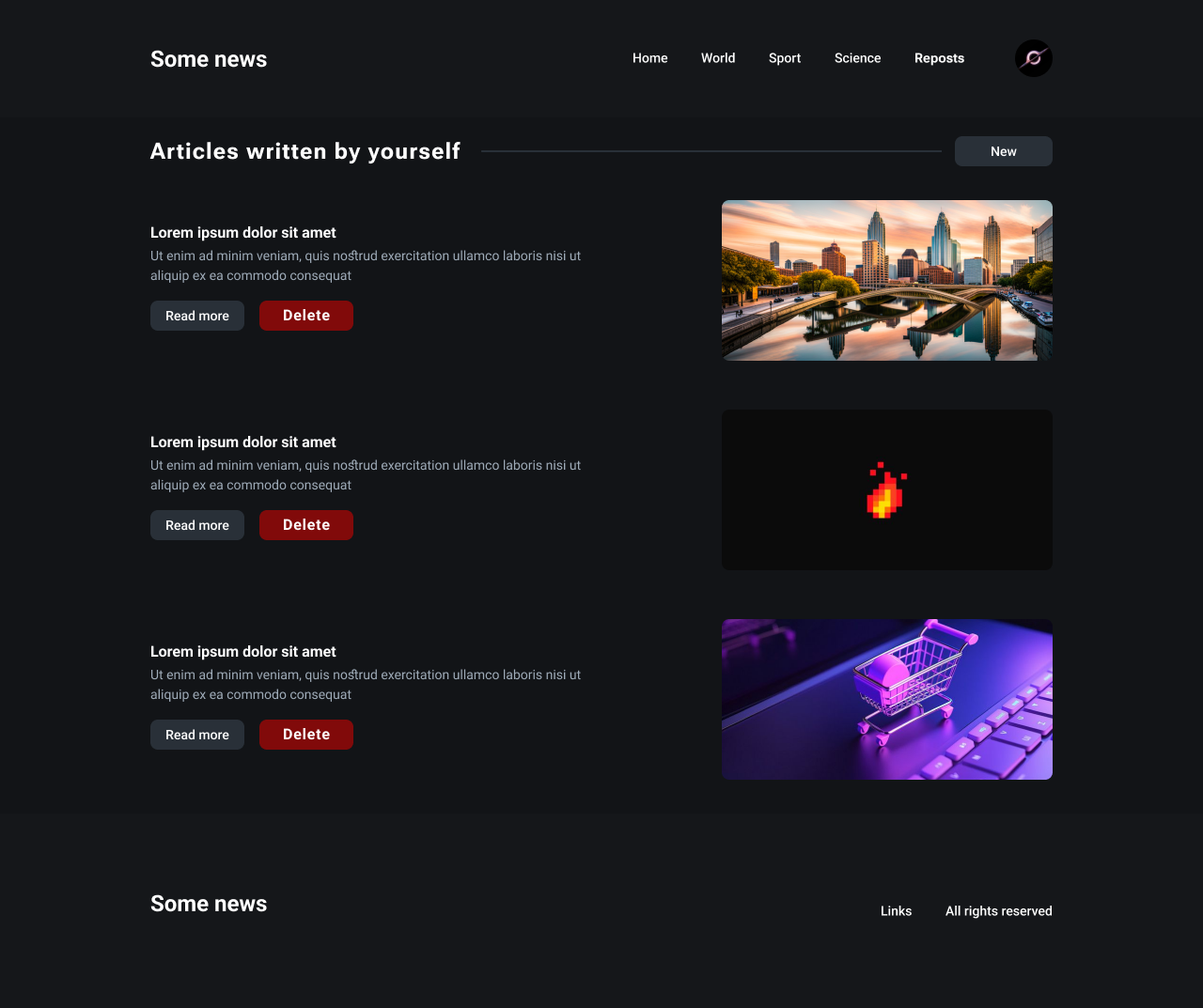


Рис. 7. Сторінка новин, автором яких є користувач.

Примітка: редагувати опубліковані новини неможливо, проте їх можна видалити.

Користувач зможе сам писати статтю з допомогою текстового редактора. В статтю можна додати фотографії, цитати та частини коду, які будуть відповідно виводитись у режимі перегляду статті.

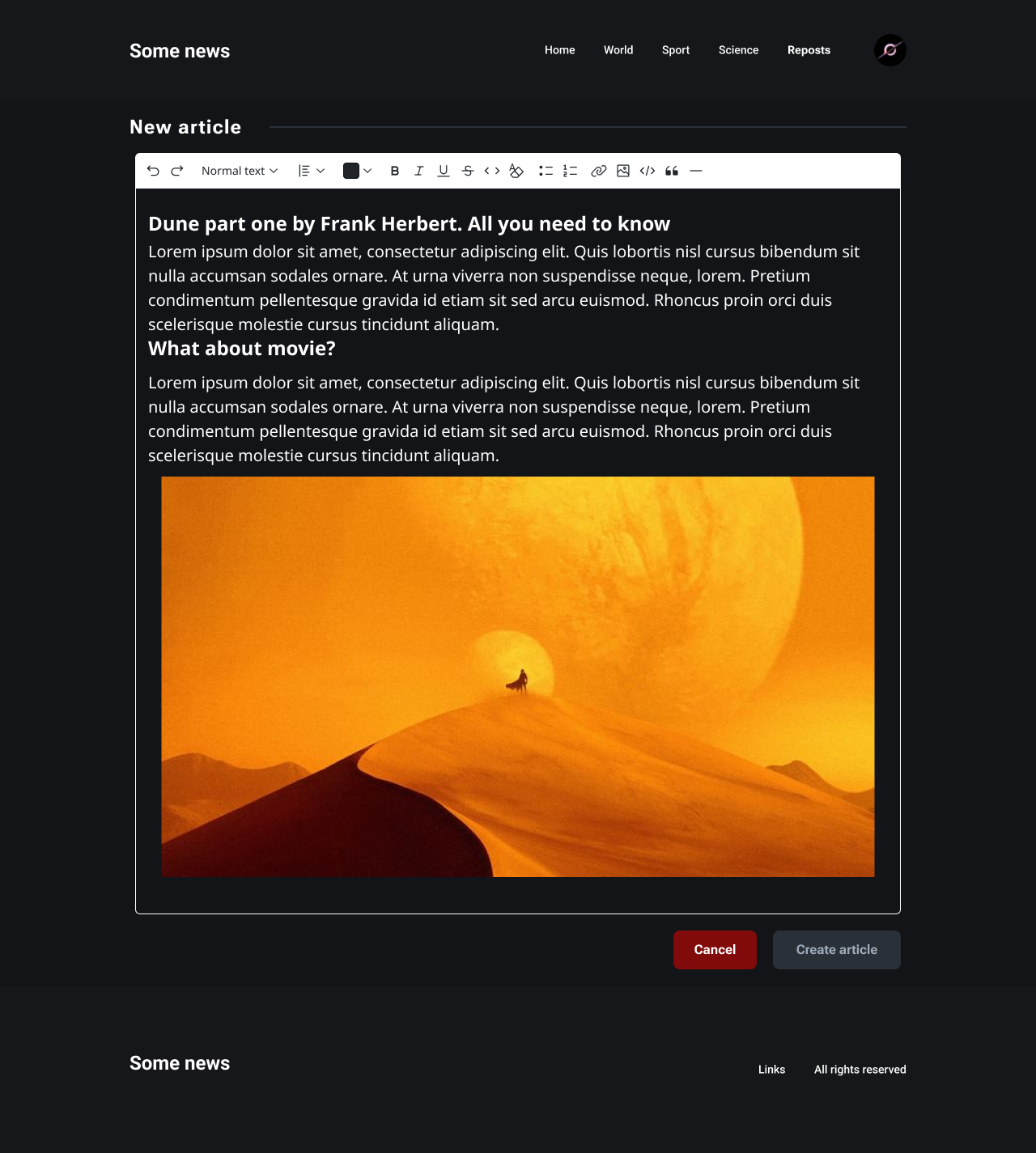


Рис. 8. Сторінка створення нової статті.

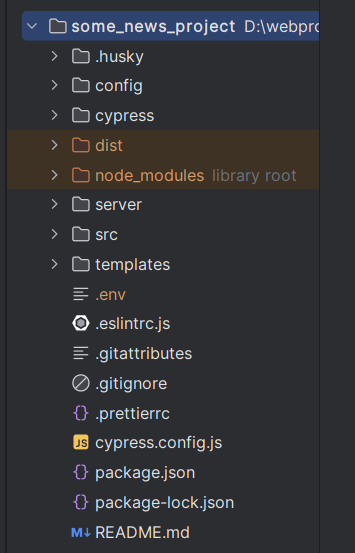


Рис. 9. Загальний вигляд папки проєкту.

Тут:

.husky – директорія для створення git hooks, а точніше для виконання деяких дій перед комітом змін до репозиторію.

config – директорія, що містить файли конфігурації бази даних, webpack та webpack.cypress.

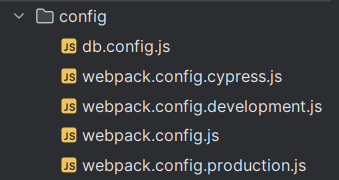


Рис. 10. Вміст директорії config.

cypress – директорія, згенерована при завантажені та кофігурації інструмента cypress для створення тестів.

dist та node\_modules – папки, що генеруються залежно від написаного коду та залежностей.

server – директорія, що містить файл сервера проєкту, та папку routes, створену для опрацювання запитів, що надходять до сервера, та видачу необхідних даних.

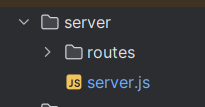


Рис. 11. Вміст папки server.

src – директорія, що містить усі головні директорії та файли проєкту, написані розробником. Тут зібрані директорії з матеріалами проєкту (файли стилів, фото) (assets), React компонентами проєкту (components), сторінками проєкту (pages), моделями проєкту (models), що включають такі моделі як User та Article. Тут знаходиться файл index.js, що з’єднує всі ці директорії в один додаток, а також його стилі, що застосовуються загально до всіх сторінок.

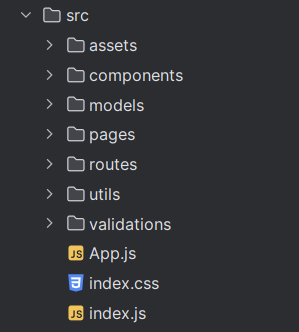


Рис. 12. Вміст директорії src.

templates – директорія, що містить html файл проєкту. Фактично, це той файл, з яким працює index.js, щоб виводити вміст додатку.

.env файл містить чутливі дані, такі як рядок з’єднання з базою даних, api ключі тощо. Цей файл не додається в репозиторій проєкту.

.eslintrc.js файл конфігурації eslint – інструмента для виправлення проблемних частин коду, та його аналізу.

.gitattributes файл атрибутів та налаштувань для git.

package.json файл, що містить завантажені залежності проєкту та основну інформацію про нього.

package-lock.json – файл для збереження версія залежностей та пришвидшення їх завантаження.

README.md – файл, що був доданий до репозиторію під час виконання попередньої лабораторної роботи. Він містить опис ідеї проєкту та дизайн сторінок.

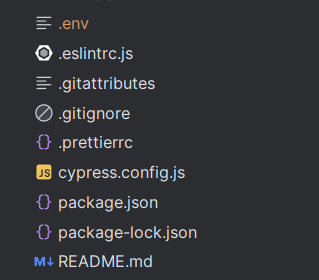


Рис. 13. Інші файли проєкту.

Фронт-енд частина додатку реалізовується з допомогою React. Стилі проєкту пишуться на CSS без використання додаткових бібліотек та інструментів.

Бек-енд частина виконується з допомогою платформи node.js та express.js, що є каркасом для створення серверної частини веб-застосунків.

Для зв’язку між фронт-ендом та бек-ендом було обрано axios – бібліотека, що дає можливість робити запити на сервер з клієнтської частини.

База даних проєкту – MongoDB. Це NoSQL документо-орієнтована база даних

Для отримання готових статей, новин було обрано news.api – інструмент для швидкого отримання новин з допомогою api.

Для перевірки коректності коду та виявлення його проблем було обрано інструмент eslint.

Для покращення читабельності коду було обрано інструмент prettier.

Відповідно було налаштовано ці два інструменти для використання їх в проєкті.

Для виконання певних дій перед комітом (наприклад аналіз коду завдяки eslint та покращення читабельності завдяки prettier) було застосовано git-hooks скрипти, що виконуються перед якимись діями (у цьому випадку eslint та prettier виконують свої задачі на аналіз та виправлення читабельності коду перед виконанням коміта).

Бібліотека husky використовується в проєкті для полегшення налаштування git-hooks та покращення зручності створення цих скриптів.



Рис. 14. Додавання залежностей для husky, eslint, prettier.

Для створення component та end-2-end тестів проєкту було обрано інструмент cypress, який дає змогу зручно писати та використовувати тести. Було створено декілька тестів для перевірки компонент та їх функціоналу.

Webpack використовується для пакування проєкту, задля його коректного запуску в браузері.

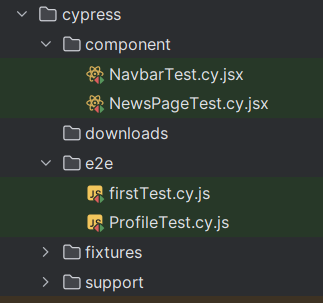
****

Рис. 15. Створені component та end-to-end тести, з допомогою cypress.

Висновок: Було представлено структуру проєкту some news. Було показано директорії проєкту, розписано основну інформацію про них, аргументовано їхню важливість. Продемонстровано дизайн та опис ідеї проєкту. Для розробки був обраний набір бібліотек та інструментів, в тому числі: React, node.js, express.js, eslint, cypress та інші.