

СТВОРЕННЯ ІНТЕРАКТИВНОГО ІНТЕРФЕЙСУ ОНЛАЙН-КУРСІВ З ВИКОРИСТАННЯМ REACT.JS

Максим Жуковський

CREATING AN INTERACTIVE INTERFACE FOR ONLINE COURSES USING REACT.JS

Maksym Zhukovskiy

Волинський національний університет імені Лесі Українки, просп. Волі, 13, Луцьк, 43025,
Україна

Abstract. *This study presents the development of an interactive online course interface using React.js with a component-based architecture built on functional components and hooks. State management is implemented via the Context API, providing centralized data handling and simplified code maintenance. React Router with lazy loading is employed for efficient routing and performance optimization. The responsive design is achieved through CSS Flexbox and Grid layouts to ensure proper display across various screen sizes. Interactive features include progress indicators, quiz-based assessments, and results visualization, while data exchange with the backend is handled via a RESTful JSON API.*

В епоху цифрової трансформації онлайн-освіта набуває все більшої популярності, ставлячи високі вимоги до якості та зручності користувацьких інтерфейсів освітніх платформ. Ефективний, інтуїтивно зрозумілий та динамічний інтерфейс є ключовим фактором залучення користувачів, полегшення процесу навчання та забезпечення позитивного досвіду взаємодії з навчальним контентом. Сучасні вебтехнології, зокрема бібліотека React.js, надають потужні інструменти для створення складних односторінкових застосунків (SPA), які забезпечують швидке оновлення даних без перезавантаження сторінки, компонентний підхід до розробки та широкі можливості для кастомізації. Важливість розробки користувацького інтерфейсу, який є не лише функціональним, але й естетично привабливим та адаптивним до різних пристроїв, неможливо переоцінити в контексті конкуренції освітніх онлайн-платформ.

Метою дослідження є розробка інтерактивного та функціонального користувацького інтерфейсу (frontend) для вебзастосунку онлайн-курсів "SkillUp", використовуючи бібліотеку React.js, з акцентом на зручність використання, компонентну архітектуру та ефективну взаємодію з серверною частиною.

У рамках дослідження було створено клієнтську частину вебзастосунку "SkillUp", яка забезпечує користувачам повний цикл взаємодії з платформою: від реєстрації та перегляду курсів до їх створення, редагування та проходження інтерактивних тестів.

Архітектурною основою frontend-частини стала бібліотека React.js, обрана за її компонентний підхід, використання віртуального DOM для оптимізації рендерингу та велику екосистему. Для управління глобальним станом застосунку, зокрема станом аутентифікації користувача, використано React Context API (AuthProvider), що дозволило уникнути "прокидання" пропсів через багато рівнів компонентів та забезпечити чисту й зрозумілу логіку доступу до даних користувача (AuthContext, RequireAuth).

Клієнтська частина вебзастосунку "SkillUp" побудована на React.js із компонентним підходом та віртуальним DOM для швидкого рендерингу. Глобальний стан, зокрема аутентифікація, керується через React Context API (AuthProvider), що усуває прокидання пропсів і спрощує доступ до даних користувача (AuthContext, RequireAuth). Навігація реалізована за допомогою react-router-dom: у App.tsx описані публічні маршрути (головна,

категорії, перегляд курсу, вхід/реєстрація) та захищені шляхи для створення і редагування курсів, секцій, тестів і особистого кабінету. RequireAuth гарантує, що неавторизовані користувачі потрапляють на сторінку входу, а Outlet підтримує вкладені роутинги.

Інтерфейс стилізовано через Material-UI з кастомною темою (theme.tsx), де прописано кольори, типографіку, тіні та responsiveFontSizes. MUI-компоненти використовуються для форм (CourseCreatePage, AccountSetting), навігаційної панелі (Navbar) та карток контенту, а для сповіщень під'єднано react-toastify. Валідація форм (реєстрація, вхід, створення/редагування курсів) побудована на базі Formik + Yup, що забезпечує миттєвий зворотний зв'язок при введенні. Для завантаження файлів (титульне зображення курсу, аватар користувача) застосовується FormData.

Для взаємодії з backend на Laravel використовується axios: централізований apiClient автоматично додає токен авторизації з localStorage до заголовків. Типи даних узгоджені через model.tsx. Сторінки курсоутворення дозволяють обирати категорію, завантажувати обкладинку та задавати опис; редагування курсів і секцій реалізовано у відповідних компонентах (EditorCourse, SectionCreate, EditSection).

Окремої уваги потребує модуль тестування (QuizTestPage.tsx): інтерактивний інтерфейс із кількома станами (завантаження, старт, процес, завершення), таймером (загальний або на питання) та типами запитань – одиничний і множинний вибір (RadioGroup, Checkbox) і зіставлення з drag-and-drop (react-beautiful-dnd). Модуль динамічно відображає питання й зображення, збирає відповіді, відправляє їх на сервер для оцінки та показує результати.

Проект організовано на основі чистої компонентної ієрархії: від загальних (Navbar, RequireAuth) до вузькоспеціалізованих (TestComponent, StartScreen, GameOverScreen). Завдяки React.js, MUI та допоміжним бібліотекам створено зручний, адаптивний і сучасний інтерфейс, готовий до подальшого розширення функціоналу.

Відтак реалізація клієнтської частини “SkillUp” на основі React.js із компонентною архітектурою, React Context API для аутентифікації та гнучким роутингом довела високу продуктивність і зручність подальшого розширення платформи. Адаптивний дизайн на Material-UI із власною темою забезпечив єдину візуальну стилістику, а інтеграція Formik із Yup дала змогу миттєво виявляти помилки при роботі з формами. Асинхронні запити через axios та обробка файлів відпрацьовують без збоїв, а унікальний тестовий модуль із drag-and-drop, таймерами й підтримкою різних типів питань створює справжній інтерактивний досвід. Отримані результати свідчать про стійкість обраного рішення та відкривають широкі перспективи для подальшого масштабування й впровадження нових можливостей у навчальну екосистему.

Бібліографія

1. React – A JavaScript library for building user interfaces [Електронний ресурс] // React. – <https://react.dev/>
2. MUI: The React component library you always wanted [Електронний ресурс] // Material-UI. – <https://mui.com/>
3. React Router: Declarative routing for React.js [Електронний ресурс] // React Router. – <https://reactrouter.com/>
4. Formik – Build forms in React, without the tears [Електронний ресурс] - Formik - <https://formik.org/>
5. Axios – Promise based HTTP client for the browser and node.js [Електронний ресурс] // Axios. – <https://axios-http.com/>
6. Dan-IT. Що таке React.js і для чого він потрібен? [Електронний ресурс] // Dan-IT. – <https://dan-it.com.ua/uk/blog/chto-takoe-react-js-i-dlja-chego-on-nuzhen/>
7. Mozilla Developer Network. Web technology for developers [Електронний ресурс] // Mozilla Developer Network. – <https://developer.mozilla.org/en-US/docs/Web>