



XIV МІЖНАРОДНА НАУКОВО-ПРАКТИЧНА КОНФЕРЕНЦІЯ

**МАТЕМАТИКА.
ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ.
ОСВІТА**

**ЛУЦЬК - СВІТЯЗЬ
13 - 15 червня 2025 р.**

**ТЕЗИ ДОПОВІДЕЙ
(друкуються в авторській редакції)**

Балич Павло, Кісь Ярослав Система підтримки та прийняття рішень логістичних операцій постачання.....	67
Бондарчук Андрій, Полоневич Ольга, Ткаленко Оксана, Лисенко Микола, Білавка Володимир Технологічна конвергенція децентралізованого штучного інтелекту речей.....	69
Васьків Роман, Веретеннікова Наталія, Пасічник Володимир Інформаційні інструменти підвищення ефективності проєктних процесів для розподілених ІТ-команд.....	71
Веремеєнко Андрій, Кунанець Наталія, Пасічник Володимир Темпоральні онтології в інтелектуальних системах ситуаційної обізнаності.....	74
Волощук Олена, Ніщененко Дмитро Проблематика класифікації типів шумів для адаптивного очищення різнорідних сенсорних даних.....	76
Гавор Артур, Нестеренко Катерина Інтелектуальний аналіз даних у блокчейн орієнтованих базах даних з використанням машинного навчання	79
Гангало Ігор, Читулян Вадим Самовідновлювані тестові середовища з використанням технологій цифрових двійників.....	82
Герцюк Микола Аналіз набору технологій для розробки рішення для оцінки токсичного впливу хімічних речовин.....	84
Домніков Ілля, Філімонова Тетяна Розробка вебзастосунку для вимірювання фізичних об'єктів у реальному масштабі на екрані за допомогою еталонного калібрування.....	86
Дубич Павло, Книш Юрій Створення інтерактивного підручника з вивчення мови програмування C++.....	88
Дубич Павло, Сметюх Богдан, Пастернак Вікторія Роль інформаційних технологій у формуванні навичок майбутнього (SOFT і HARD SKILLS).....	90
Євсєєв Сергій, Король Ольга Цифрові технології в освіті та науці.....	92
Жебка Сергій, Сініцин Ігор, Жебка Вікторія Метод оптимізації розміру блоку з урахуванням енергоспоживання в PROOF-OF-WORK мережах.....	94
Жуковський Максим, Юнчик Валентина Створення інтерактивного інтерфейсу онлайн-курсів з використанням REACT.JS.....	96
Завацький Владислав, Сторчак Каміла, Зінченко Ольга, Лашевська Наталія, Кузьміч Ірина Ефективна структура передачі даних для щільних мереж інтернету речей.....	99
Замрій Ірина, Нестеренко Катерина, Задонцев Юрій, Глушкова Оксана, Шахматов Іван Методика підвищення функціональної стійкості інформаційної системи через виявлення вторгнень та реконфігурації мережі.....	101

Результати моделювання

Розмір блоку B (МБ)	Кількість транзакцій	Енергія $E(B)$ (Вт·год)	Затримка $\tau(B)$ (с)	Функція втрат $L(B)$
0,5	1024	0,47	0,25	0,47025
1,0	2048	1,00	0,5	1,0005
1,5	3072	1,63	0,75	1,63075
2,0	4096	2,46	1,0	2,461

Результати моделювання показують, що енерговитрати при збільшенні розміру блоку зростають надлінійно, а ефективність транзакцій на одиницю енергії зменшується. Оптимізація повинна враховувати не лише пропускну здатність, а й енергетичну стійкість мережі. Запропонований метод дає змогу адаптивно обирати розмір блоку, що мінімізує сумарні втрати в умовах високої складності та обмеженого енергобалансу. Він може бути інтегрований у консенсус-алгоритм як механізм «зеленого майнінгу» з регулятором, залежним від енергетичної політики.

Список використаних джерел:

1. Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System [Електронний ресурс]. – 2008. Режим доступу: <https://bitcoin.org/bitcoin.pdf>
2. Mora C. et al. Bitcoin emissions alone could push global warming above 2°C // Nature Climate Change. – 2018. – Vol. 8. – P. 931–933.
3. Zhang H., Li Y. Energy-aware blockchain: from consensus mechanisms to practical applications // IEEE Communications Surveys and Tutorials. – 2020. – Vol. 22(4). – P. 2586–2625.
4. Соколюк В. А., Корнута О. І. Енергоефективність у блокчейн-системах: проблеми та перспективи // Інформаційні технології і комп'ютерна інженерія. – 2021. – № 2. – С. 59–65.

СТВОРЕННЯ ІНТЕРАКТИВНОГО ІНТЕРФЕЙСУ ОНЛАЙН-КУРСІВ З ВИКОРИСТАННЯМ REACT.JS

Максим ЖУКОВСЬКИЙ, здобувач вищої освіти

Науковий керівник: **Валентина ЮНЧИК**, доктор філософії, доцент

Волинський національний університет імені Лесі Українки

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 інтерактивний та функціональний користувацький інтерфейс (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).

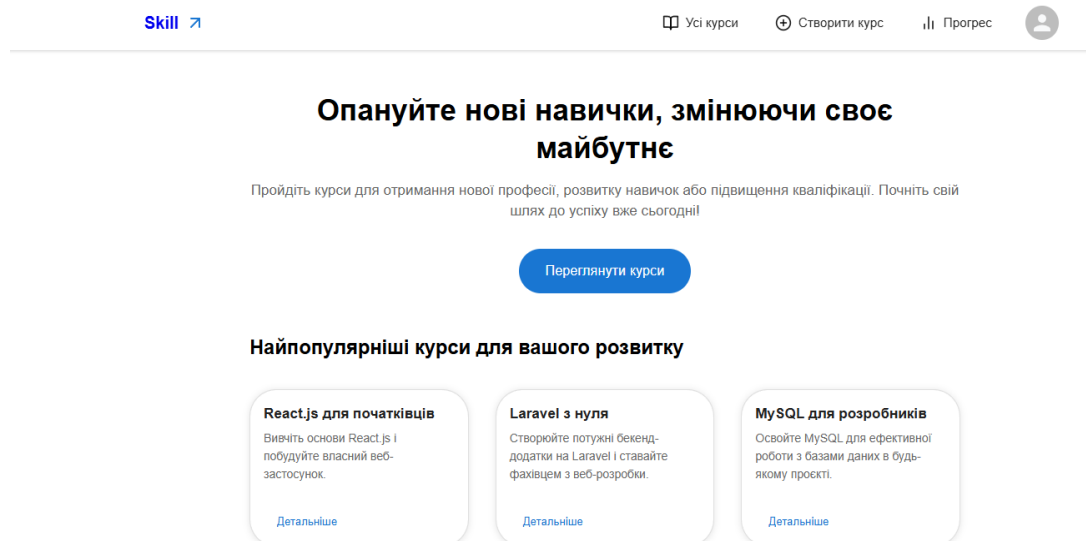


Рисунок 1 – Інтерфейс головної сторінки

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) – Рис. 1, 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.

Список використаних джерел:

1. React – A JavaScript library for building user interfaces [Електронний ресурс]. Режим доступу: <https://react.dev/>
2. MUI: The React component library you always wanted [Електронний ресурс] // Material-UI. Режим доступу: <https://mui.com/>
3. React Router: Declarative routing for React.js [Електронний ресурс] // React Router. Режим доступу: <https://reactrouter.com/>
4. Formik – Build forms in React, without the tears [Електронний ресурс] – Formik. Режим доступу: <https://formik.org/>
5. Axios – Promise based HTTP client for the browser and node.js [Електронний ресурс] // Axios. Режим доступу: <https://axios-http.com/>
6. Dan-IT. Що таке React.js і для чого він потрібен? [Електронний ресурс] // Dan-IT. Режим доступу: <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. Режим доступу: <https://developer.mozilla.org/en-US/docs/Web>.

ЕФЕКТИВНА СТРУКТУРА ПЕРЕДАЧІ ДАНИХ ДЛЯ ЩІЛЬНИХ МЕРЕЖ ІНТЕРНЕТУ РЕЧЕЙ

Владислав ЗАВАЦЬКИЙ, аспірант

Каміла СТОРЧАК, доктор технічних наук, професор

Ольга ЗІНЧЕНКО, доктор технічних наук, доцент

Наталія ЛАЩЕВСЬКА, кандидат технічних наук, доцент

*Ірина КУЗЬМІЧ, заступник директора Національно-наукового
інституту інформаційних технологій*

Державний університет інформаційно-комунікаційних технологій

Постановка задачі. Сучасні мережі Інтернету речей (IoT) стикаються з проблемою ефективного передавання великої кількості даних від численних пристроїв, особливо в умовах обмеженого спектра, високої щільності пристроїв та необхідності швидкого й надійного обміну інформацією [1]. Для розв'язування цієї проблеми потрібні нові структури передачі даних, які б використовували просторові, часові та частотні ресурси більш ефективно.

Мета дослідження. Розробити та проаналізувати ефективну структуру передачі даних для IoT-мереж, яка базується на рознесенні у просторі, часі та частоті, що дозволяє одночасну передачу від багатьох пристроїв без взаємних завад та з мінімальними затримками.

Результати дослідження. Запропонована структура на Рис. 1 дозволяє розділити передачу даних на окремі блоки за часом, частотою та простором, що дає змогу значно зменшити перешкоди між пристроями. У цій структурі кожен пристрій передає дані у певний момент часу, на конкретній частоті,