

УДК 004.7

*Мінайленко В.Ю., здобувач,  
Єфремов Ю.М., к.т.н., доцент  
Державний університет «Житомирська політехніка»*

## **ПРОЄКТУВАННЯ КЛІЄНТСЬКОЇ ЧАСТИНИ ВЕБПЛАТФОРМИ ДЛЯ ТОРГІВЛІ ГОДИННИКАМИ НА ОСНОВІ БІБЛІОТЕКИ REACT**

Успіх будь-якого сучасного сайту насамперед залежить від того, наскільки зручно ним користуватися. Для спеціалізованих торгових платформ це питання є критично важливим. Кожен відвідувач має швидко знаходити потрібні речі в каталозі та легко керувати своїми оголошеннями. Також дуже важливо, щоб користувачі могли без жодних затримок стежити за змінами цін під час активних аукціонів. Особливу увагу було приділено вибору технологій для розробки зовнішньої частини проєкту. Це безпосередньо впливає на те, наскільки швидко працюватиме інтерфейс та наскільки стабільно він буде виконувати свої функції. Правильно підібрані інструменти допомагають нам зробити сайт не лише швидким для користувачів, а й зручним для подальшого оновлення та підтримки.

Зовнішній інтерфейс було побудовано на основі бібліотеки React. Головна перевага такого підходу полягає в тому, що весь інтерфейс можна розділити на окремі та незалежні частини. Це можуть бути картки товарів, форми для виставлення нових лотів, блоки зі ставками аукціону або зручні фільтри для пошуку в каталозі. Кожна така частина працює самостійно та містить у собі власну логіку та зовнішній вигляд. Це значно спрощує перевірку роботи всього сайту та дозволяє нам легко додавати нові функції в майбутньому. Такий підхід робить додаток дуже гнучким, адже ми можемо повторно використовувати готові елементи на різних сторінках. Це економить час при розробці та допомагає підтримувати єдиний стиль усього проєкту.

Управління станом застосунку реалізовано за допомогою хуків React (useState, useEffect, useContext). Для даних, що потребують спільного доступу з різних частин інтерфейсу - зокрема інформації про авторизованого користувача та вмісту кошика було застосовано Context API. Такий підхід дозволяє уникнути надмірної передачі пропсів між рівнями компонентів і зберігає код читабельним.

Для переходу між сторінками нашого застосунку ми обрали вбудовану систему маршрутизації від Next.js. Такий підхід значно відрізняється від класичних односторінкових сайтів. Кожна сторінка завантажується окремо, що дозволяє пошуковим системам краще

бачити контент сайту. Захищені маршрути (особистий кабінет, форма виставлення лота) доступні лише автентифікованим користувачам перевірка здійснюється на рівні компонента-обгортки перед рендерингом цільової сторінки.

Для забезпечення точного відліку часу та стабільної роботи інтерфейсу було розроблено спеціальні програмні компоненти. Вони автоматично оновлюють інформацію на екрані через чітко визначені проміжки часу. Також передбачено механізм зупинки цих процесів, коли користувач переходить у інший розділ сайту. Це допомагає не перевантажувати пам'ять пристрою та робить роботу всього сервісу плавною і надійною. Такий підхід гарантує, що кожен колекціонер завжди бачить актуальну інформацію про свої лоти та угоди.

Адаптивний дизайн платформи забезпечено за допомогою CSS-модулів та медіазапитів, що гарантує коректне відображення на пристроях з різними розмірами екранів. Каталог товарів реалізовано у вигляді сітки карток з динамічною зміною кількості стовпців залежно від ширини вікна перегляду. Форми введення даних валідуються на стороні клієнта перед відправкою запиту до серверу.

Взаємодія з серверною частиною здійснюється через REST API з використанням бібліотеки Axios. Запити до захищених ендпоінтів автоматично доповнюються JWT-токеном через axios-інтерцептор, що централізує логіку авторизації та виключає необхідність її дублювання в кожному сервісному методі. Обробка помилок мережевих запитів реалізована глобально на рівні інтерцептора відповідей.

Застосування бібліотеки React у поєднанні з описаними підходами до організації стану та маршрутизації забезпечує високу швидкість відклику інтерфейсу та зручність для кінцевого користувача. Подальший розвиток клієнтської частини передбачає впровадження серверного рендерингу (Next.js) для покращення SEO-показників платформи.

#### **Список використаних джерел:**

1. Meta Open Source. React Documentation: Tutorial: Tic-Tac-Toe [Електронний ресурс]. - Режим доступу: <https://react.dev/learn/tutorial-tic-tac-toe> (дата звернення: 20.03.2026).
2. Dodds K. C. Epic React [Електронний ресурс]. - Режим доступу: <https://epicreact.dev> (дата звернення: 20.03.2026).
3. Wieruch R. The Road to React [Електронний ресурс]. - Режим доступу: <https://www.roadtoreact.com> (дата звернення: 18.03.2026).
4. Fielding R. T. Architectural Styles and the Design of Network-based Software Architectures: PhD thesis [Електронний ресурс]. - Режим доступу: <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm> (дата звернення: 20.03.2026).