УДК 004.42

Є. А. Суслова
О. В. Войцеховська

ІНТЕГРАЦІЯ ЧАТУ В ІНТЕРНЕТ-МАГАЗИН З ВИКОРИСТАННЯМ ПРОТОКОЛУ WEBSOCKET

Вінницький національний технічний університет

Анотація.

Робота присвячена інтеграції чату в інтернет-магазин з використанням протоколу WebSocket, що дозволяє клієнтам звертатися за допомогою в режимі реального часу, отримувати консультацію та рішення для своїх запитів. Проаналізовано та вибрано сучасні технології для забезпечення якісної та оперативної роботи чату, такі як фреймворк Express.js та бібліотека Socket.io.

Ключові слова: інтернет-магазин, протокол WebSocket, інтеграція чату, Express.js, Socket.io.

Abstract.

The work is devoted to the integration of chat in the online store using the WebSocket protocol, which allows customers to ask for help in real time, receive advice and solutions to their requests. Modern technologies have been analyzed and selected to ensure high-quality and efficient chat operation, such as the Express.js framework and the Socket.io library.

Keywords: online-store, WebSocket protocol, chat integration, Express.js, Socket.io.

Вступ

На сьогоднішній день, електронна комерція займає все більш значуще місце на ринку. Велика кількість інтернет-магазинів дає можливість купувати товари та послуги з будь-якої точки світу, не виходячи з дому, що приваблює потенційних клієнтів. Зі зростанням конкуренції на ринку електронної комерції стає все складніше утримати клієнтів на сайті, тому необхідно забезпечувати найвищий рівень сервісу для їх залучення та утримання. У цьому контексті, внутрішній чат у інтернет-магазині є важливим елементом для комунікації, підтримки та збереження клієнтів, які шукають швидку та ефективну допомогу при здійсненні онлайн-покупок.

Результати дослідження

Основне призначення інтеграції чату в інтернет-магазин – забезпечення можливості для клієнта швидко отримати консультацію під час здійснення онлайн-покупок, що допоможе залишити клієнта з позитивними враженнями від використання сервісу, отримати від нього позитивні відгуки, і, як наслідок, збільшити кількість клієнтів в інтернет-магазині.

При цьому важливо використовувати технологічно продуктивні та функціональні інструменти для забезпечення якісної та оперативної роботи чату. Тому, при розробці внутрішнього чату було вирішено використати протокол веб-комунікації WebSocket, який забезпечує спосіб обміну даними між браузером і сервером через постійне з’єднання. Дані можна передавати в обох напрямках у вигляді “пакетів”, без розриву з’єднання та додаткових HTTP-запитів. WebSocket відмінно підходить для сервісів, що потребують безперервного обміну даними, наприклад чат, онлайн-ігри, системи торгівлі в реальному часі тощо[1]. Для реалізації роботи протоколу WebSocket використано фреймворк Express.js та бібліотеку Socket.io.

Socket.io — JavaScript-бібліотека для двостороннього обміну даними між веб-клієнтами та серверами в реальному часі. Одна її складова відповідає за роботу у браузері, а інша, з таким самим API, на сервері. В результаті користувачі веб-додатку можуть обмінюватися повідомленнями в чаті, разом редагувати файли, стежити за показниками, що змінюються онлайн, тощо. Socket.io однаково швидко і безпечно працює в будь-якому браузері, незалежно від операційної системи чи пристрою [2].

Популярним способом демонстрації двостороннього зв’язку, який надає Socket.io, є базова програма для чату. Коли сервер отримує нове повідомлення, сокет надсилає його клієнту та сповіщає його, обходячи потребу надсилати запити між клієнтом і сервером [3]. Приклад демонстрації двостороннього зв’язку, який надає Socket.io, подано на рисунку 1.

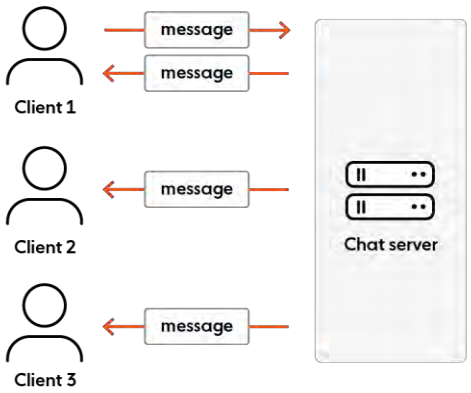

Рисунок 1 — Демонстрація двостороннього зв’язку, який надає Socket.io,

Socket.io добре працює з Node.js, середовищем виконання JavaScript на стороні сервера. Node.js дозволяє запускати код JavaScript на сервері, створювати веб-додатки та використовувати одну мову як на стороні клієнта, так і на стороні сервера.

На серверній стороні було створено сервер WebSocket з використанням фреймворка Express.js. Для налаштування WebSocket використано метод io() бібліотеки Socket.io та прив'язано його до існуючого сервера [4]. Код для ініціалізації socket-сервера показано на рисунку 2. Ініціалізація socket-сервера відбувається кожен раз, як сервер вебдодатку перезапускається.

```
const server = require('http').Server(app);
this.io = new Server(server, {
    cors: { origin: "*" }
})
server.listen(8080);
```

Рисунок 2 — Програмний код для ініціалізації сокет-сервера

На стороні клієнта для з’єднання з сервером потрібно використовувати метод .connect об’єкта io, в якому як параметр вказати домен: const socket = io.connect('http://localhost:8080');.

У даному контексті адміністратор інтернет-магазину має привілейований статус, що дозволяє йому отримувати повний доступ до всіх груп користувачів. Цей доступ дозволяє адміністратору взаємодіяти з будь-яким користувачем незалежно від його групи. З іншого боку, звичайним користувачам надається обмежений доступ до однієї спеціальної групи, де присутній адміністратор. Такий підхід забезпечує конфіденційність та безпеку взаємодії звичайних користувачів. Ця реалізація гарантує гнучкість та ефективність взаємодії між користувачами та адміністратором, що є головною метою розробки внутрішнього чату.

Висновки

Отже, використання відкритих технологій, таких як Express.js та Socket.io, дозволяє ефективно розробляти та підтримувати чат-систему, забезпечуючи її безперебійну роботу та високу швидкість роботи. Крім того, внутрішній чат у інтернет-магазині, що використовує технології Express.js та Socket.io, може стати ключовим фактором у забезпеченні конкурентної переваги на ринку електронної комерції та підвищенні задоволеності клієнтів від використання сервісу та знизити відсоток відмов від покупок.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. WebSocket [Електронний ресурс]. – Режим доступу: https://uk.javascript.info/websocket.
2. Technologies Socket.IO [Електронний ресурс]. – Режим доступу: https://brander.ua/technologies/socketio.

3. Socket.IO: What it is, when to use it, and how to get started [Електронний ресурс]. – Режим доступу: https://ably.com/topic/socketio
4. Socket.IO documentation. Bidirectional and low-latency communication for every platform [Електронний ресурс]. – Режим доступу: https://socket.io/.

Суслова Єва Андріївна – студент групи 2КІ-19б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця

Войцеховська Олена Валеріївна – кандидат технічних наук, доцент кафедри обчислювальної техніки, Вінницький національний технічний університет, Вінниця

Suslova Yeva A. **—** student of group 2KI-19B, Faculty of Information Technology and Computer Engineering, Vinnytsia National Technical University, Vinnytsia

Voytsekhovska Olena V. **—** PhD, Assistant Professor of the Chair of Computer Techniques, Vinnytsia National Technical University, Vinnytsia