

РОЗРОБКА TELEGRAM-БОТА І ЗАСТОСУНКУ НА ТЕХНОЛОГІЧНОМУ СТЕЦІ MERN ДЛЯ КЕРУВАННЯ ПІДПИСКАМИ ТА КОРИСТУВАЧАМИ ІНФОРМАЦІЙНИХ РЕСУРСІВ

Данило Санько

DEVELOPMENT OF A TELEGRAM BOT AND APPLICATION ON THE MERN
TECHNOLOGY STACK FOR MANAGING SUBSCRIPTIONS AND USERS OF
INFORMATION RESOURCES

Danylo Sanko

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

Abstract. *This study presents the development of a subscription management system for Telegram-based information resources. The system consists of two integrated software components: a Telegram bot developed using Node.js and MongoDB, responsible for handling user subscription, payments and automated notifications and a web application built on the MERN stack (MongoDB, Express.js, React, Node.js), providing administrators with tools for user management, subscription status monitoring and system oversight. The solution ensures automation of subscription workflows, improves user experience and provides a scalable architecture for managing paid access to Telegram channels and groups*

В умовах цифровізації дедалі більше інформаційних ресурсів мігрують у месенджери, зокрема Telegram, який забезпечує зручну платформу для розповсюдження контенту, спілкування з аудиторією та монетизації доступу через підписку. Це створює попит на автоматизовані інструменти, що дозволяють ефективно керувати підписками, контролювати оплату та забезпечувати зручну адміністративну взаємодію. У межах даної роботи було реалізовано два програмні продукти: Telegram-бот для управління підписками та вебзастосунок для адміністрування користувачів, побудований на базі сучасного MERN-стеку (MongoDB, Express.js, React, Node.js).

Telegram-бот реалізований із використанням Node.js, що забезпечує високу продуктивність, асинхронну обробку запитів і надійне масштабування при взаємодії з великою кількістю користувачів. Бот автоматично додає нових підписників до приватних каналів або груп Telegram, перевіряє факт оплати (наприклад, у мережі USDT), надсилає повідомлення про наближення завершення підписки та видаляє користувачів із ресурсу після закінчення оплаченого періоду. Однією з ключових переваг використання MERN-стеку є єдність мови програмування — JavaScript — як на клієнтському, так і на серверному боці, що значно спрощує розробку, обслуговування та масштабування проєкту. Це дозволяє зменшити кількість конфліктів при передачі даних між частинами системи, пришвидшити розробку нових функцій та залучати розробників без необхідності глибокого занурення в різні технології. MongoDB, як документно-орієнтована база, ідеально підходить для зберігання гнучких структур даних, що змінюються в залежності від бізнес-логіки підписок.

Адміністративна частина представлена як вебзастосунок, створений із використанням MERN-стеку. Node.js та Express.js відповідають за серверну логіку та REST API, MongoDB виступає в ролі гнучкої документно-орієнтованої бази даних, а React забезпечує динамічний, швидкий і масштабований інтерфейс користувача. Компонентна архітектура React дозволяє легко підтримувати, розширювати й модифікувати адміністративну панель. Для стилізації був

використаний фреймворк Tailwind CSS, який дозволив швидко створювати адаптивний, сучасний дизайн завдяки утилітарному підходу до стилізації. Комбінація цих технологій дала змогу створити інтуїтивно зрозумілий і приємний для ока дизайн, який підвищив зручність користування для адміністраторів системи.

Серед основних можливостей вебзастосунку — перегляд усіх підключених користувачів, контроль статусу підписки, інформація про дату завершення доступу, швидкий пошук користувачів, зміна доступів та ручне додавання або видалення користувачів у разі потреби. Уся інформація синхронізується з Telegram-ботом, що забезпечує цілісність і актуальність даних.

Особливістю реалізації є зручна панель керування на основі бібліотеки AdminJS, яка дає змогу адміністратору змінювати структуру бази, додавати ресурси, налаштовувати тарифи, переглядати журнал дій і контролювати динаміку підписок без необхідності звернення до бази даних або редагування коду вручну.

Архітектура системи побудована на принципах REST, що забезпечує стандартизовану взаємодію між клієнтською та серверною частинами, а також дозволяє у майбутньому інтегрувати мобільний додаток або додаткові зовнішні сервіси.

Таким чином, створене рішення дозволяє автоматизувати весь життєвий цикл підписки в Telegram-ресурсах — від моменту оплати до закінчення доступу, забезпечуючи при цьому адміністратору інструменти для повного контролю та гнучкого керування користувачами.

Бібліографія

1. Index | Node.js v23.3.0 Documentation. *Node.js – Run JavaScript Everywhere*. URL: <https://nodejs.org/en/docs/> (date of access: 16.05.2025).
2. Team M. D. MongoDB Documentation. *MongoDB: The Developer Data Platform | MongoDB*. URL: <https://www.mongodb.com/docs/> (date of access: 16.05.2025).
3. telegraf.js - v4.16.3. *telegraf.js - v4.16.3*. URL: <https://telegraf.js.org/> (date of access: 16.05.2025).
4. Telegram APIs. *Telegram APIs*. URL: <https://core.telegram.org/> (date of access: 16.05.2025).
5. Express - Node.js web application framework. *Express - Node.js web application framework*. URL: <https://expressjs.com/> (date of access: 16.05.2025).