

У підсумку реалізовано прототип автономної системи фільтрації, яка дає змогу збирати та аналізувати дані з відкритих джерел, гнучко керувати категоріями загроз, а також автоматично оновлювати списки блокування для використання на комп'ютерах і маршрутизаторах.

Список використаних джерел

1. Kosinski M. What is ransomware? *International Business Machines*. URL: <https://www.ibm.com/think/topics/ransomware>
2. What is a Botnet? *CrowdStrike*. URL: <https://www.crowdstrike.com/.../botnets/>
3. Антонов Ю. С., Римар П. В., Антонова О. Г. Проблема DoS/DDoS атак навчальних ресурсів здобувачами. *Сучасний захист інформації*. 2019. № 4(40). С. 52–62. DOI: 10.31673/2409-7292.2019.045262.
4. Документація Pi-Hole. URL: <https://github.com/pi-hole/docs>
5. Dnsmasq. *ArchWiki*. URL: <https://wiki.archlinux.org/title/Dnsmasq>

УДК 004.738.1

*Підруцький Д. А., здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Січко Т. В., канд. техн. наук, доцент, доцент
кафедри інформаційних технологій*

РОЗРОБКА ВЕБСАЙТА ДЛЯ ПРОДАЖУ ВІЙСЬКОВОГО СПОРЯДЖЕННЯ

Донецький національний університет імені Василя Стуса, м. Вінниця

У контексті триваючої війни особливої важливості набуває забезпечення військових якісним спорядженням. Традиційні канали закупівлі не завжди здатні оперативно задовольнити попит, тому виникає потреба у створенні спеціалізованих онлайн-ресурсів для продажу військового обладнання.

Актуальність посилюється потребою у безпечному та масштабованому середовищі для обробки великого обсягу критичних даних.

Наразі, активно досліджуються питання побудови вебплатформ на основі технологій, як от MongoDB, Express, React. Особлива увага приділяється захисту персональних даних JWT, bcrypt, структурі REST API, та інтеграції платіжних систем, як от Stripe, у системи електронної комерції.

Метою роботи є розробка повнофункціонального вебсайта військового спорядження з акцентом на забезпеченні безпеки користувачів, масштабованості та зручного користувацького інтерфейсу.

Постановка задачі:

- створити backend-сервер на Node.js з використанням Express;
- спроектувати та реалізувати гнучку структуру бази даних на MongoDB;
- побудувати frontend-інтерфейс на основі React із використанням Tailwind CSS, Axios, React Router DOM і React Toastify;
- забезпечити безпеку автентифікації, збереження даних і реалізувати платіжну систему Stripe;

– організувати взаємодію між клієнтом і сервером за принципами REST API.

У проєкті реалізовано збереження та обробку даних на основі нереляційної бази даних MongoDB, що забезпечує гнучке й масштабоване зберігання інформації про:

- користувачів, включно з ролями, контактами, хешованими паролями;
- товари, їх опис, категорії, фото, ціну, наявність;
- замовлення, а саме: деталі кошика, способи доставки, статус, дата створення;
- адміністративні дії, як-от: створення / редагування / видалення товарів, зміна статусу замовлень.

Усі дані зберігаються у вигляді взаємопов'язаних колекцій, що дає змогу ефективно здійснювати запити, оновлення, фільтрацію та сортування. Для взаємодії з базою даних використано Mongoose – обгортку для MongoDB, що дає змогу будувати схеми з валідацією, відношеннями та методами для логіки на рівні моделей [1].

Розроблена структура бази даних забезпечує:

- швидкий доступ до інформації навіть за великої кількості записів;
- безпечне збереження конфіденційних даних через bcrypt та JWT;
- можливість масштабування та подальшого розвитку системи, наприклад, додавання складів, історії замовлень, обліку постачання тощо.

Фронтенд-частина проєкту реалізована з використанням бібліотеки **React**, яка забезпечує компонентну структуру інтерфейсу, що дає змогу створювати зручний, динамічний та масштабований вебдодаток. Для стилізації застосовано **Tailwind CSS**, що допоможе швидко будувати адаптивний інтерфейс без надлишкового CSS-коду. Передача даних між клієнтом і сервером реалізована за допомогою **Axios**, що забезпечує зручну роботу з API-запитами [2]. Для навігації між сторінками без перезавантаження використовується **React Router DOM**, а для зворотного зв'язку з користувачем – **React Toastify**, що дає змогу виводити сповіщення про помилки, підтвердження дій чи інші події в зручному форматі.

Бекенд-частина створена на основі **Node.js** з використанням фреймворка **Express**, що дає змогу реалізувати гнучке управління серверною логікою, обробку HTTP-запитів, керування сесіями, авторизацію через JWT та взаємодію з базою даних. У систему інтегровано платіжну платформу **Stripe**, яка дає змогу здійснювати реальні онлайн-платежі з дотриманням високих стандартів безпеки [3]. Загальна архітектура побудована за принципами REST, що дає змогу ефективно обмінюватися даними між клієнтською та серверною частинами, забезпечуючи стабільну роботу інтернет-магазину та його готовність до подальшого розширення функціоналу.

Отже, ключовим складником сайту є правильно спроектована та реалізована база даних, що є основою для надійного функціонування онлайн-магазину. Вона дає змогу ефективно зберігати, обробляти та захищати інформацію в умовах динамічного середовища та високих вимог до безпеки.

Список використаних джерел

1. Шаров С. В., Петровський В. В. Огляд нереляційних баз даних. *Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення*: тези доп. всеукр. наук. інтернет-конф. (м. Тернопіль, 26–27 жовт.) / Тернопіль: Тернопільський національний економічний університет, 2015. С. 16–18. URL: https://eprints.mdpu.org.ua/id/eprint/602/1/%D1%88%D0%B0%D1%80%D0%BE%D0%B2_%D0%BF%D0%B5%D1%82%D1%80%D0%BE%D0%B2%D1%81%D0%BA%D0%B8%D0%B9.pdf
2. Здор О. В. Розробка веб-додатку SPA з використанням фреймворку React: пояснювальна записка до кваліфікаційної роботи здобувача вищої освіти на другому (магістерському) рівні, спеціальність 172 Телекомунікації та радіотехніка / М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. Харків, 2023. 69 с. URL: <https://openarchive.nure.ua/entities/publication/91e0cfc4-f711-441e-a638-bbdcfff965e3> (дата звернення: 21.05.2025).
3. REST API CRUD Operations Using ExpressJS. *geeksforgeeks.org*. 19.02.2025. URL: <https://www.geeksforgeeks.org/rest-api-using-the-express-to-perform-crud-create-read-update-delete/> (дата звернення: 21.05.2025).
4. Степанюк О. С., Січко Т. В. Особливості використання реляційних та нереляційних баз даних в Big Data. *Комп'ютерні технології обробки даних*: матеріали всеукр. наук.-практ. конф. (м. Вінниця, 2020). С. 103–106. URL: <https://jktod.donnu.edu.ua/article/view/10450>

УДК 004.01

*Сіклічук А. С., здобувачка вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,
Фриз І. В., канд. фіз-мат. наук, старший
викладач кафедри інформаційних технологій*

РОЗРОБКА CRM-СИСТЕМИ ДЛЯ ТАНЦЮВАЛЬНОЇ СТУДІЇ

Донецький національний університет імені Василя Стуса, м. Вінниця

У сучасному цифровому середовищі автоматизація процесів управління клієнтськими базами є необхідністю для більшості бізнесів, зокрема й у сфері послуг. Танцювальні студії, як частина цієї сфери, також потребують ефективних рішень для організації своєї роботи: обліку клієнтів, тренувань, абонементів, відвідуваності та розкладу. Використання CRM-систем дає змогу не лише зменшити адміністративне навантаження, а й покращити якість взаємодії з клієнтами та знизити ризики людських помилок.

З огляду на ці потреби в роботі розглянуто підходи до створення CRM-системи для танцювальної студії, яка відповідає основним вимогам до такого типу програмного забезпечення: зручність користування, гнучкість, масштабованість і можливість подальшого розширення. Як оптимальне рішення обрано десктопну клієнт-серверну архітектуру, що поєднує незалежність від браузера з можливістю централізованого зберігання даних [1].

Для реалізації системи доцільно використати такі інструменти і технології:

1. ASP.NET Core Web API для побудови серверної частини;
2. WPF (Windows Presentation Foundation) для створення десктопного клієнта [2];
3. MySQL як СУБД [3];