

РОЗРОБКА МЕТОДУ РЕЄСТРАЦІЇ НА БЛАГОДІЙНИЙ ЗАХІД

Вінницький національний технічний університет

Анотація

У дослідженні розроблено метод реєстрації на благодійний захід через вебресурс. Представлено архітектуру клієнтської та серверної частин застосунку, реалізовану із використанням React, Node.js, MongoDB, Formik, Joi та Nodemailer. Описано алгоритм реєстрації на благодійний захід: від заповнення форми до підтвердження участі та сповіщення адміністрації. Застосовані механізми валідації, авторизації та електронного підтвердження забезпечують зручність, безпеку й продуктивність процесу. Розроблений метод підвищує довіру користувачів, оптимізує адміністрування заходу та відповідає сучасним вимогам цифрової благодійності.

Ключові слова: онлайн-реєстрація, благодійні заходи, вебресурс, автоматизація, React, Node.js, MongoDB, цифровізація, користувацький інтерфейс, безпека даних.

Abstract

The study presents a method for registering for a charitable event through a web-based platform. It outlines the architecture of both the client-side and server-side of the application, implemented using React, Node.js, MongoDB, Formik, Joi, and Nodemailer. The registration algorithm is described in detail – from filling out the form to confirming participation and notifying the administration. Validation, authorization, and email confirmation mechanisms are applied to ensure convenience, security, and efficiency of the process. The developed method enhances user trust, streamlines event administration, and meets modern digital charity standards.

Keywords: online registration, charity events, web platform, automation, React, Node.js, MongoDB, digitalization, user interface, data security.

Вступ

Реєстрація на благодійний захід через вебресурс є зручним, доступним і ефективним способом взаємодії з потенційними учасниками. Такий підхід дає змогу людям подавати заявки у будь-який зручний час, не прив'язуючись до графіка роботи організаторів або необхідності телефонувати. Згідно з дослідженням Eventtia, понад 85% організаторів заходів використовують онлайн-системи реєстрації для спрощення процесу залучення учасників та зниження кількості помилок у заявках [1]. Простота інтерфейсу та чітка структура вебплатформи роблять процес реєстрації інтуїтивно зрозумілим навіть для користувачів з мінімальним досвідом у роботі з цифровими технологіями. Завдяки вбудованій перевірці даних користувач бачить помилки одразу під час введення, що знижує ймовірність некоректної інформації. Після відправлення заявки учасник миттєво отримує підтвердження електронною поштою, що створює відчуття завершеності процесу та впевненості у своїй участі.

Попри це, наразі благодійні проекти часто використовують соціальні мережі, месенджери або сторонні сервіси для реєстрації учасників і координації дій, що не завжди є зручним і системним підходом. Відсутність єдиної цифрової платформи ускладнює організацію взаємодії, підвищує ризик дублювання зусиль та знижує прозорість процесу. Створення єдиного вебресурсу з використанням методу реєстрації на благодійний захід, обміном інформацією та можливістю підтримки ініціатив у режимі реального часу могло б значно покращити ефективність благодійної діяльності та забезпечити більш злагоджену комунікацію між учасниками [2].

З боку організаторів реєстрація онлайн дозволяє уникнути ручного опрацювання заявок, автоматизує перевірку дублювань і забезпечує зручне збереження інформації у базі даних. Це значно полегшує подальшу комунікацію, формування списків учасників та контроль за кількістю зареєстрованих. За даними Event Smart, 30% реєстрацій на заходи відбуваються поза робочими годинами, що підкреслює важливість доступності онлайн-платформ [3]. Крім того, наявність особистого кабінету дозволяє користувачу самостійно відстежувати статус своєї заявки, змінювати контактні дані або переглядати історію участі. Усе це сприяє підвищенню довіри до заходу та створює позитивний досвід взаємодії з платформою. В умовах сучасного цифрового суспільства саме онлайн-

реєстрація забезпечує швидкість, зручність, прозорість і якість процесу залучення людей до благодійної діяльності.

Основна частина

Метод реєстрації забезпечує збір і збереження інформації про учасників, підтвердження їхньої участі, а також інформування адміністрації про нові заявки. Це дозволяє ефективно планувати захід, контролювати кількість учасників та підтримувати зворотній зв'язок із зареєстрованими користувачами.

Реєстрація на захід також важлива для побудови бази даних активних учасників, що може бути корисно для подальших комунікацій, анонсів та статистичного аналізу.

Метод реєстрації на захід включає такий функціонал:

1. Авторизований користувач переходить на сторінку заходу.
2. Користувач заповнює форму реєстрації, у якій вводить ім'я, прізвище та email.
3. Після заповнення форми користувач натискає кнопку «Зареєструватись».
4. Користувач бачить індикатор завантаження (loader).
5. Запускається функція 'handleSubmitRegistration', яка відповідає за обробку даних форми.
6. Функція 'validateRegistrationData' перевіряє коректність введених даних на клієнті.
7. Після валідації викликається 'sendRegistrationRequest', яка відправляє POST-запит на серверний API '/api/registrations'.
8. На сервері функція 'createRegistration' приймає запит, виконує валідацію через 'validateRegistrationSchema' (з бібліотеки Joi).
9. Серверна функція 'checkDuplicateRegistration' перевіряє, чи не існує вже реєстрації користувача на цей захід.
10. Якщо унікальна, серверна функція 'saveRegistrationToDB' зберігає заявку у колекцію MongoDB 'registrations'.
11. Функція 'generateConfirmationToken' створює унікальний токен для підтвердження.
12. Функція 'sendConfirmationEmail' відправляє користувачу підтвердження через email за допомогою 'nodemailer'.
13. Функція 'notifyAdminNewRegistration' надсилає повідомлення адміністратору про нову заявку.
14. Сервер повертає результат у форматі JSON з підтвердженням успішної реєстрації.
15. На клієнті викликається 'handleRegistrationResponse', який знімає індикатор завантаження та виводить повідомлення користувачу.
16. Дані заявки оновлюються в особистому кабінеті через API-метод 'getUserRegistrations'.

Технічна реалізація методу реєстрації на захід передбачає комплексну взаємодію між клієнтською та серверною частинами застосунку, а також базою даних і механізмами безпеки. На клієнтському боці реалізовано React-компонент 'RegistrationForm' у поєднанні з кастомним хуком 'useRegistration', які відповідають за інтерфейс форми та логіку її обробки. Для побудови форми використано бібліотеку 'Formik', а для валідації введених даних – схеми 'Yup', що дозволяють перевіряти коректність полів до надсилання запиту.

React – це популярна JavaScript-бібліотека для створення користувацьких інтерфейсів (UI), розроблена компанією Facebook. Вона дозволяє створювати багатократно використовувані компоненти, які динамічно оновлюють вміст веб-сторінок без повного перезавантаження, забезпечуючи швидку та інтерактивну роботу додатків. React працює на основі віртуального DOM, що підвищує продуктивність при зміні даних [4].

На стороні сервера використовується стек 'Node.js' та 'Express'. Створений API-ендпоінт 'POST /api/registrations' обробляє запити на реєстрацію. Дані надсилаються до колекції 'registrations' у базі даних MongoDB, де зберігаються такі поля, як 'userId', 'eventId', 'name', 'surname', 'email', 'status' та 'confirmationToken'. Ці дані використовуються як для підтвердження участі, так і для подальшої аналітики.

MongoDB – це документоорієнтована система керування базами даних з відкритим вихідним кодом, яка зберігає дані у форматі BSON (розширення JSON). Вона належить до категорії NoSQL-баз даних, що означає, що дані організовані не в таблицях, як у реляційних СКБД, а у вигляді колекцій документів. MongoDB забезпечує високу продуктивність, масштабованість і гнучкість у роботі з великими обсягами нереляційних даних [5].

Для реалізації підтвердження реєстрації електронною поштою інтегровано бібліотеку 'nodemailer'. Вона використовується у функції 'sendConfirmationEmail', яка відправляє лист із посиланням на

підтвердження участі, що містить унікальний токен. Генерація токена здійснюється через функцію 'generateConfirmationToken' з використанням UUID, що гарантує унікальність і безпечність.

Важливим компонентом є валідація даних. На клієнтському рівні валідацію виконує функція 'validateRegistrationData', а на сервері – 'validateRegistrationSchema'. Обидва рівні забезпечують перевірку коректності та повноти введеної інформації. Для обробки помилок використовується механізм 'handleErrors', який формує відповідні HTTP-відповіді (наприклад, 400 при некоректних даних або 409 при конфліктах).

Особливу увагу приділено безпеці. Використовується middleware 'authMiddleware' для перевірки автентичності користувача, а також реалізовано захист від CSRF-атак, що гарантує надійність процесу реєстрації навіть у випадках активної взаємодії з публічними подіями.

Цілісна технічна реалізація забезпечує гнучкий, захищений та масштабований механізм реєстрації, що відповідає потребам сучасного вебзастосунку та гарантує зручність користувачам. Описаний метод забезпечує повний цикл реєстрації з контролем коректності даних, збереженням у базі, підтвердженням для користувача та інформуванням адміністрації.

Щоб краще зрозуміти принципи функціонування вебреєстрації на благодійний захід і всі етапи, які проходить заявка користувача – від заповнення форми до підтвердження участі – було розроблено детальний алгоритм роботи цього процесу, що зображений на рисунку 1.

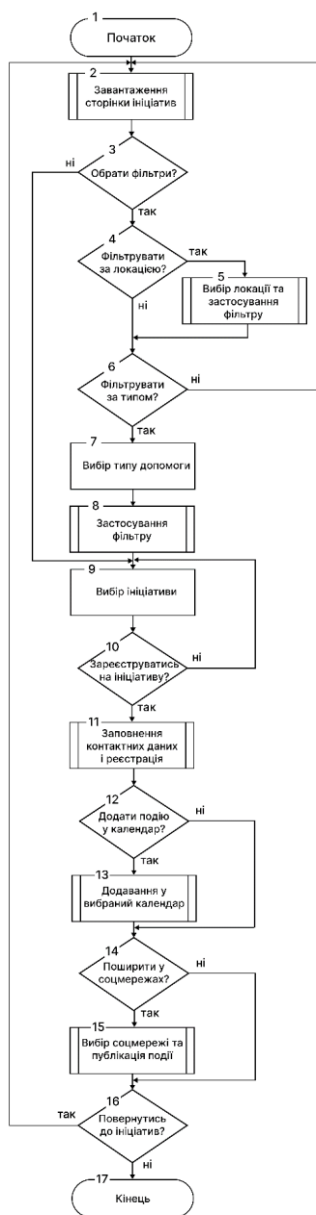


Рисунок 1 – Алгоритм реєстрації на благодійний захід

Алгоритм вибору та участі у благодійній ініціативі описує послідовність дій користувача на платформі, що дає змогу брати участь у благодійних ініціативах, застосовувати фільтри, реєструватися та взаємодіяти з подіями. Цей алгоритм є важливим, оскільки забезпечує логічну, зручну та ефективну модель взаємодії з платформою, підвищує залученість користувачів та сприяє поширенню інформації про ініціативи. Початок процесу визначається завантаженням сторінки ініціатив, що є стартовою точкою взаємодії користувача з системою.

Після завантаження користувачу пропонується обрати фільтри, які допоможуть звузати перелік ініціатив. Якщо користувач вирішує скористатися фільтрами, йому пропонується спочатку фільтрувати ініціативи за локацією, що дозволяє відобразити лише події, актуальні для конкретного регіону. У разі підтвердження вибору, застосовується відповідний фільтр. Також користувач має можливість відфільтрувати за типом допомоги. Якщо користувач обирає цю опцію, він має можливість вибрати конкретний тип благодійної активності, наприклад, притулок для тварин, ЗСУ, діти-сироти тощо. Після цього застосовується відповідний фільтр і відображає перелік заходів обраного типу.

На основі застосованих критеріїв користувач переглядає список ініціатив і обирає одну з них для участі. Після вибору система запитує, чи хоче користувач зареєструватися на подію. У разі згоди він заповнює контактні дані (ім'я, прізвище та електронну адресу) та завершує реєстрацію, підтверджуючи дію кнопкою.

Після реєстрації користувачеві пропонується додати подію у свій календар. Якщо він погоджується, система додає подію у вибраний календар, що дозволяє йому краще організувати час і не пропустити подію, також ініціатива додатково буде відображатись в особистому кабінеті користувача.

Далі користувачу надається можливість поширити інформацію про ініціативу в соціальних мережах. Якщо він погоджується, обирається відповідна мережа користувачем, і система автоматично публікує подію. На завершальному етапі система запитує, чи бажає користувач повернутися до списку ініціатив для перегляду інших. Якщо так – процес повторюється знову, якщо ні – алгоритм завершується. Це дозволяє створити зручну, повторювану і повністю контрольовану логіку взаємодії з ініціативами.

Висновки

Отже, розробка методу реєстрації на благодійний захід з використанням React, Node.js, MongoDB, Formik, Joi та Nodemailer є не лише технічно доцільним рішенням, а й важливим кроком у напрямку цифровізації взаємодії між організаторами та учасниками. Завдяки зручному інтерфейсу, можливості подати заявку у будь-який час, миттєвому підтвердженню та доступу до інформації в особистому кабінеті, користувач отримує комфортний і зрозумілий процес реєстрації. Водночас для адміністраторів це – автоматизована система обробки заявок, що значно полегшує організаційну роботу, мінімізує помилки та сприяє ефективному плануванню заходу.

Розроблений алгоритм реєстрації на благодійний захід дозволяє чітко структурувати усі етапи взаємодії – від вибору події до підтвердження участі. Такий підхід забезпечує прозорість, безпечність і масштабованість процесу, що відповідає сучасним вимогам до цифрових сервісів у сфері громадської та благодійної діяльності.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. How to Easily Set Up Online Registration For Your Events [Електронний ресурс] – Режим доступу до ресурсу: <https://www.eventtia.com/en/blog/set-up-hassle-free-online-registration-experience> (дата звернення: 02.06.2025).
2. Багнюк О.В., Романюк О.В. «Розробка програмного забезпечення для планування та координації благодійних ініціатив» / Матеріали LIV Всеукраїнської науково-технічної конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ–2025): збірник доповідей [Електронний ресурс] – Режим доступу до ресурсу: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2025/paper/view/24024/19897>
3. Define Online Event Registration [Електронний ресурс] – Режим доступу до ресурсу: <https://eventsmart.com/searches/online-event-registration/> (дата звернення: 03.06.2025).
4. React. JavaScript-бібліотека для створення користувацьких інтерфейсів [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.legacy.reactjs.org/> (дата звернення: 05.03.2025).
5. What Is MongoDB? An Expert Guide [Електронний ресурс] – Режим доступу до ресурсу: <https://www.oracle.com/ua/database/mongodb/> (дата звернення: 04.06.2025).

Багнюк Ольга Віталіївна – студент групи ЗПІ-21б, факультет інформаційних технологій та комп’ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: olabagnuk94@gmail.com.

Романюк Оксана Володимирівна – кандидат технічних наук, доцент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: romaniukoksana@gmail.com.

Olha Bahniuk – student of group ЗПІ-21b, Faculty for Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: olabagnuk94@gmail.com.

Oksana Romaniuk – Ph.D., Associate Professor of Software Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: romaniukoksana@gmail.com.