

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА»

Факультет комп'ютерних наук і технологій
(повне найменування факультету)

Кафедра «Системний аналіз та обчислювальна математика»
(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавра

(ступінь вищої освіти)

на тему:

«Розробка програмного інтерфейсу сайту громадської організації»
(назва теми)

Виконав: студент 4 курсу, групи КНТ-811СП

Спеціальності 124 – «Системний аналіз»

(код і назва напрямку підготовки, спеціальності)

Освітня програма (спеціалізація)

«Інтелектуальні технології та прийняття рішень
в складних системах»

СОРОЧИНСЬКИЙ Д.С.

(ПРИЗВИЩЕ та ініціали)

Керівник САВРАНСЬКА А.В.

(ПРИЗВИЩЕ та ініціали)

Рецензент ЛОЗОВСЬКА М.О.

(ПРИЗВИЩЕ та ініціали)

м. Запоріжжя

2024 рік

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет «Факультет комп'ютерних наук і технологій»

Кафедра «Системний аналіз та обчислювальна математика»

Ступінь вищої освіти Бакалавр

Спеціальність 124 – «Системний аналіз»

Освітня програма (спеціалізація) «Інтелектуальні технології та прийняття рішень в складних системах»

ЗАТВЕРДЖУЮ

**В.о. завідувача кафедри «Системний аналіз та
обчислювальна математика»**

канд. фіз.-мат. наук, _____

доц. Терещенко Еліна Валентинівна «_____»

_____ 2024 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

СОРОЧИНСЬКОГО Дмитра Сергійовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) «Розробка програмного інтерфейсу сайту громадської організації» керівник проєкту (роботи) к.ф.-м.н., доцент, Савранська Алла Володимирівна,
затверджені наказом закладу вищої освіти від «__» __ 20__ року №__
2. Строк подання студентом проєкту (роботи) 2 червня 2024 року
3. Вихідні дані до проєкту (роботи) VS Code(IDE), React.js, Redux, Axios, Formik, Material-UI, React Router DOM, Figma, Jest,
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) У вступі обґрунтовано актуальність теми, мету, об'єкт, предмет досліджень і завдання. Перший розділ охоплює вибір технологій, архітектуру сайту та інтеграцію компонентів, зокрема використання React. У другому розділі досліджено дизайн сайту та вибір бібліотек для створення інтерфейсів. У третьому розділі розглянуто підтримку та розвиток сайту, включаючи впровадження нових функцій.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів) _____
6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
Теоретична частина	САВРАНСЬКА А.В.	11 січня 2024	
Практична частина	ЄСІНА О. А. Керівник Відокремленого підрозділу "Запорізька асоціація територіальних громад ГО "Солідарна Справа Громад"	19 січня 2024	
Нормоконтроль	ШИРОКОРАД Д.В.		

7. Дата видачі завдання «11» січня 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Приміт ка
1	Отримання завдання	11 січня 2024	
2	Вивчення референсів	25 січня 2024	
3	Обрання технологій	3 лютого 2024	
4	Розробка основного функціоналу	27 березня 2024	
5	Створення дизайну веб-сайту	13 квітня 2024	
6	Інтеграція бази даних	15 квітня 2024	
7	Розробка функціоналу для запитів	29 квітня 2024	
8	Тестування веб-сайту	16 травня 2024	
9	Корегування функціоналу	27 травня 2024	
10	Подання готової роботи	2 червня 2024	

Студент(ка)  Дмитро СОРОЧИНСЬКИЙ.

Керівник проєкту (роботи) _____ Алла САВРАНСЬКА

РЕФЕРАТ

Пояснювальна записка: 87 с., 20 рис., 3 табл., 1 додаток, 22 джерела.

Об'єктом дослідження в роботі є процес розробки програмного інтерфейсу для веб-сайту громадської організації.

Предметом дослідження є технології створення веб-сайту, дизайн користувацького інтерфейсу, методи інтеграції компонентів та підтримка веб-сайту громадської організації.

Мета досліджень: обґрунтування теоретичних і практичних аспектів створення веб-сайту для громадських організацій та розробка практичних рекомендацій щодо його вдосконалення і підтримки.

У *теоретичній частині* проведено огляд останніх досліджень та публікацій з питань розробки веб-сайтів, дизайну користувацького інтерфейсу та інтеграції компонентів. Наведено сутність технологій веб-розробки, описано різні підходи до створення зручних та функціональних веб-сайтів, їх аналіз та ефективність.

У *практичній частині* розроблено програмний інтерфейс для веб-сайту громадської організації. Запропоновано архітектуру сайту, реалізовано ключові функції та інтеграцію компонентів, створено естетично привабливий та зручний для користувачів інтерфейс. В даному розділі розроблено модель підтримки та розвитку веб-сайту, а також план впровадження нових функцій для підвищення ефективності роботи організації.

Наукова новизна полягає у розробці програмного інтерфейсу веб-сайту громадської організації з використанням сучасних веб-технологій та підходів до дизайну користувацького інтерфейсу.

Практична цінність отриманих результатів дипломної роботи полягає у створенні веб-сайту, який дозволяє покращити комунікацію з громадськістю, залучити нових волонтерів та підвищити рівень довіри з боку суспільства.

Ключові слова: ВЕБ-САЙТ, ГРОМАДСЬКА ОРГАНІЗАЦІЯ, REACT, ВЕБ-ТЕХНОЛОГІЇ, ПІДТРИМКА ВЕБ-САЙТУ, ДИЗАЙН ВЕБ-САЙТУ, ІНТЕГРАЦІЯ, РОЗВИТОК ВЕБ-САЙТУ.

ABSTRACT

Explanatory note: 87 p., 20 figures, 3 tables, 1 appendix, 22 sources.

The object of research in this paper is the process of developing a software interface for a public organization's website.

The subject of research is the technology of creating a website, user interface design, methods of integrating components and supporting a public organization's website.

The purpose of the research is to substantiate the theoretical and practical aspects of creating a website for public organizations and to develop practical recommendations for its improvement and support.

In the theoretical part, the article reviews the latest research and publications on website development, user interface design and component integration. The essence of web development technologies is presented, various approaches to creating user-friendly and functional websites, their analysis and effectiveness are described.

In the practical part, a software interface for a public organization's website was developed. The architecture of the site is proposed, key functions and integration of components are implemented, and an aesthetically pleasing and user-friendly interface is created. This section develops a model for maintaining and developing the website, as well as a plan for implementing new functions to improve the efficiency of the organization.

The scientific novelty is the development of a software interface for a public organization's website using modern web technologies and approaches to user interface design.

The practical value of the results of the thesis is the creation of a website that allows to improve communication with the public, attract new volunteers and increase the level of public trust.

Keywords: WEBSITE, PUBLIC ORGANIZATION, REACT, WEB TECHNOLOGIES, WEBSITE SUPPORT, WEBSITE DESIGN, INTEGRATION, WEBSITE DEVELOPMENT.

ЗМІСТ

Завдання.....	2
РЕФЕРАТ.....	4
ABSTRACT	6
ВСТУП.....	9
РОЗДІЛ 1 ТЕХНОЛОГІЧНІ АСПЕКТИ РОЗРОБКИ ВЕБ-САЙТУ...	13
1.1 Вибір та обґрунтування технологій для розробки веб-сайту....	13
1.2 Архітектура та структура веб-сайту.....	20
1.3 Реалізація ключових функцій та інтеграція компонентів.....	23
РОЗДІЛ 2 ДИЗАЙН ВЕБ-САЙТУ.....	27
2.1 Принципи дизайну.....	27
2.2 Контент сайту.....	29
2.3 Адміністративна частина сайту.....	33
РОЗДІЛ 3 ПІДТРИМКА ТА РОЗВИТОК ВЕБ-САЙТУ.....	36
3.1 Підтримка веб-сайту.....	36
3.2 Розвиток веб-сайту.....	39
3.3 Інтеграція з іншими системами та сервісами.....	42
3.4 План розвитку веб-сайту.....	46
ВИСНОВКИ.....	50
ПЕРЕЛІК ПОСИЛАНЬ.....	53
ДОДАТОК А Приклад коду React компоненту.....	55
ДОДАТОК Б Приклад коду Redux стану.....	58

ВСТУП

Актуальність теми. В сучасному світі, де інформаційні технології проникають у всі сфери життя, громадські організації стикаються з необхідністю адаптації до нових умов. Ефективна комунікація з громадськістю, залучення волонтерів та підтримка їхньої активності вимагають наявності сучасних і зручних інструментів. Інтернет став головним засобом отримання інформації та спілкування, тому наявність функціонального веб-сайту для громадської організації є не просто бажаною, а необхідною умовою успішної діяльності.

Завдяки веб-сайту організація може не лише інформувати громадськість про свою діяльність, але й активно залучати нових учасників до своїх проектів. Веб-сайт дозволяє швидко реагувати на поточні події та потреби, надаючи необхідну інформацію в реальному часі. Крім того, зручний і добре структурований сайт сприяє покращенню іміджу організації, підвищує рівень довіри та взаємодії з громадськістю.

Громадські організації відіграють ключову роль у розв'язанні багатьох соціальних проблем, підтримці вразливих верств населення, організації благодійних заходів і впровадженні соціальних змін. Наявність сучасного веб-сайту дозволяє таким організаціям:

- Швидко та ефективно розповсюджувати інформацію про свою діяльність і досягнення.
- Залучати нових волонтерів та партнерів.
- Забезпечувати прозорість фінансової діяльності та звітування перед спонсорами і громадськістю.
- Організовувати та координувати волонтерську діяльність за допомогою онлайн-інструментів.

Веб-сайт також надає можливість громадським організаціям проводити різноманітні онлайн-кампанії, збирати пожертви та надавати консультації. Це значно розширює можливості організації та дозволяє їй бути більш ефективною

у своїй діяльності. Наприклад, завдяки інтерактивним функціям сайту волонтери можуть реєструватися на заходи, отримувати необхідні матеріали та звітувати про виконану роботу.

Конкуренція за увагу користувачів в інтернеті є дуже високою. Громадські організації, які не мають сучасних веб-ресурсів, ризикують залишитися на узбіччі, не отримуючи необхідної підтримки та визнання. Інвестиції в розробку сучасного веб-сайту є стратегічно важливими для підвищення конкурентоспроможності організації на ринку благодійних та соціальних послуг.

Сучасний веб-сайт дозволяє громадській організації вигідно виділятися серед інших, демонструючи свою інноваційність та готовність до змін. Це не лише сприяє залученню нових учасників і спонсорів, але й підвищує рівень довіри з боку громадськості. Завдяки зручному інтерфейсу та інтерактивним функціям, користувачі можуть легко знаходити необхідну інформацію, що підвищує їхню залученість і лояльність.

Багато успішних громадських організацій вже скористалися перевагами сучасних веб-технологій. Наприклад, такі організації як "Червоний Хрест" [1], "UNICEF" [2] та інші міжнародні організації активно використовують веб-сайти для залучення волонтерів, проведення благодійних заходів та збору пожертв. Їхній досвід демонструє, що інвестиції в розробку і підтримку сучасних веб-ресурсів є виправданими і приносять значні результати.

Розробка веб-сайту є лише першим кроком на шляху до цифрової трансформації громадської організації. Майбутнє розвиток передбачає постійне оновлення та вдосконалення веб-ресурсу, впровадження нових функцій та технологій. Зокрема, можна розглянути можливість інтеграції з соціальними мережами, впровадження системи електронних підписів для зручності волонтерів, розширення функціональності за рахунок мобільних додатків тощо.

Зв'язок з замовником та отримання технічного завдання. У процесі створення веб-сайту для громадської організації "Запорізька асоціація територіальних громад ГО "Солідарна Справа Громад" важливо забезпечити

тісний зв'язок із замовником та отримати детальне технічне завдання. Це включає:

Первинні консультації: На початковому етапі необхідно провести кілька зустрічей із представниками організації, щоб зрозуміти їхні потреби, цілі та очікування від веб-сайту.

Розробка технічного завдання: На основі отриманих даних розробляється технічне завдання, яке детально описує функціональні вимоги до сайту, структуру, дизайн, інтерактивні елементи та інші важливі аспекти.

Затвердження технічного завдання: Технічне завдання повинно бути узгоджено і затверджено обома сторонами. Це забезпечує чітке розуміння обсягу робіт та уникнення можливих непорозумінь у майбутньому.

Постійний зворотний зв'язок: Під час розробки веб-сайту необхідно підтримувати постійний зв'язок із замовником, регулярно інформуючи його про прогрес робіт і отримуючи зворотний зв'язок. Це дозволяє вносити необхідні корективи на ранніх етапах та досягти бажаного результату.

Таким чином, наявність тісного зв'язку із замовником і чітке технічне завдання є критично важливими елементами успішної розробки веб-сайту, який відповідатиме потребам громадської організації і сприятиме її ефективній діяльності.

Ціль і задачі дослідження Ціль дипломної роботи полягає у розробці ефективного, зручного та адаптивного веб-сайту для громадської організації. Для досягнення цієї мети були поставлені наступні задачі:

- Аналіз існуючих технологій веб-розробки та вибір оптимальних для реалізації проекту.
- Проектування архітектури та структури веб-сайту.
- Реалізація ключових функцій та інтеграція необхідних компонентів.
- Забезпечення привабливого дизайну та зручності використання веб-сайту.

Об'єкт і предмет дослідження Об'єктом дослідження є процес розробки веб-сайту для громадської організації, а предметом — технологічні та

дизайнерські аспекти створення інтерфейсу користувача та адміністрування сайту.

Методи дослідження В роботі застосовувалися методи аналізу, проектування, програмування та тестування. Зокрема, використовувалися такі технології, як React, Redux, Formik, MUI, Axios та React Router DOM для забезпечення високої продуктивності, масштабованості та зручності використання веб-сайту.

Апробація роботи і інформаційна база дослідження Інформаційною базою дослідження слугували наукові статті, інтернет-ресурси та технічна документація щодо використаних технологій. Публікації, підготовлені на основі роботи, висвітлюють ключові аспекти розробки ефективного веб-сайту для громадської організації. Результати дослідження використовувалися під час навчального процесу, що підтверджує їхню актуальність та практичну цінність.

Структура і обсяг дипломної роботи Дипломна робота складається з вступу, трьох розділів, висновків, списку використаних джерел та додатків. Загальний обсяг роботи складає 64 сторінки. У першому розділі розглядаються технологічні аспекти розробки веб-сайту, у другому — дизайн та контент сайту, у третьому — підтримка та розвиток веб-сайту.

В сучасному світі, де інтернет стає головним засобом комунікації, наявність сучасного та функціонального веб-сайту є необхідною умовою для успішної діяльності громадської організації. Веб-сайт дозволяє організації ефективно взаємодіяти з громадськістю, залучати нових учасників та підтримувати активність існуючих волонтерів.

РОЗДІЛ 1 ТЕХНІЧНІ АСПЕКТИ РОЗРОБКИ ВЕБ-САЙТУ

1.1 Вибір та обґрунтування технологій для розробки веб-сайту

У цьому розділі буде розглянуто основні технології, що були обрані для розробки веб-сайту, їхні переваги та обґрунтування вибору. Використання сучасних веб-технологій є ключовим фактором успіху проекту, тому детальний аналіз кожної з них є важливим. Для розробки веб-сайтів у більшості випадків використовують фреймворки [3].

Фреймворк – це платформа для розробки програмного забезпечення, що забезпечує структуру та набір інструментів для швидкого створення додатків. Вони допомагають стандартизувати процес розробки, полегшують підтримку коду та дозволяють швидко впроваджувати нові функції.

На цей час основними фреймворками для створення веб-сайтів є React [4], Vue [5] та Angular [6].

React — це бібліотека для створення користувацьких інтерфейсів, розроблена Facebook. Вона дозволяє створювати швидкі та інтерактивні веб-додатки завдяки компонентному підходу, що дозволяє повторно використовувати код і спрощує процес розробки.

- **Компонентний підхід:** React дозволяє розробникам розбивати користувацький інтерфейс на невеликі, незалежні компоненти, які можна повторно використовувати. Це сприяє зменшенню коду та покращенню його подальшої підтримки.

- **Віртуальний DOM:** використання віртуального DOM дозволяє React ефективно оновлювати лише ті частини сторінки, які змінилися, що підвищує продуктивність додатку.

- **Сильна екосистема:** велика кількість бібліотек та інструментів, розроблених спільнотою, дозволяє швидко додавати нові функції та оптимізувати роботу додатку.

Провівши порівняння React з іншими популярними фреймворками для JavaScript - Vue та Angular було зроблено висновок:

1. **Віртуальний DOM:** React та Vue використовують віртуальний DOM, що полегшує оновлення сторінки та підвищує продуктивність. Angular також має свій механізм для оптимізації оновлень DOM.

2. **Розмір бібліотеки:** Vue та React мають невеликі розміри, що дозволяє швидко завантажувати додаток. Angular, з іншого боку, може мати більший розмір через вбудовані функціональність та модульну структуру.

3. **Компонентна архітектура:** У всіх трьох фреймворках є підтримка компонентів, але підхід до роботи з ними трохи відрізняється. У React та Vue компоненти є більш простими та легкими для розуміння, в той час як Angular має більш розгалужену систему компонентів та директив.

4. **Швидкість розробки:** Vue часто відзначають за швидкість розробки завдяки простоті та зручності використання. React також має велику швидкість розробки, але деякі розробники вважають, що для складних проектів Angular може бути більш підходящим завдяки своїм вбудованим рішенням та стандартам.

5. **Спільнота та екосистема:** React має велику та активну спільноту розробників, що дозволяє швидко знаходити відповіді на питання та розв'язувати проблеми. Vue також має досить велику спільноту, але вона менша в порівнянні з React. Angular має розвинуту екосистему, але його спільнота менша порівняно з React та Vue.

У кожного фреймворка є свої переваги та недоліки, і вибір між ними залежить від потреб вашого проекту та вашого досвіду розробки. React відомий своєю простотою та швидкістю розробки, що робить його вигідним вибором для багатьох розробників.

У якості бібліотеки для керуванням стану додатку можна обирати серед двох лідерів своєї справи це Redux [7] та MobX [8]. Те що у Redux є централізоване зберігання стану дає дуже велику перевагу.

Redux — це бібліотека для керування станом додатків, яка забезпечує єдине джерело правди для всього стану додатку, що робить його передбачуваним і легким для налагодження.

- **Централізоване зберігання стану:** всі дані додатку зберігаються в одному місці, що дозволяє легко відстежувати зміни та керувати станом.
- **Передбачувані зміни стану:** зміни стану відбуваються через функції-редюсери, що робить додаток більш передбачуваним і легким для тестування.
- **Інструменти розробника:** Redux DevTools дозволяє відстежувати всі дії, що призводять до зміни стану, і зворотно відтворювати їх для налагодження.

Компонентні бібліотеки дають можливість використовувати вже деякий реалізований функціонал з гарною стилістикою та правильно налаштованою роботою. Серед світових лідерів є Material-UI [9], Ant Design [10] та безліч інших менш відомих бібліотек. При розробці було обрано саме Material-UI через більшу розповсюдженість в останній час, ця інформація була взята з сайту npmjs [11].

Material-UI — це бібліотека компонентів, яка реалізує принципи матеріального дизайну від Google і дозволяє створювати естетично привабливі та зручні для користувачів інтерфейси.

- **Готові компоненти:** великий набір готових до використання компонентів, таких як кнопки, форми, діалоги тощо, що дозволяє швидко створювати інтерфейс.
- **Адаптивність:** компоненти Material-UI легко адаптуються до різних розмірів екранів, що забезпечує зручне користування на мобільних пристроях.
- **Тематизація:** можливість легко змінювати теми інтерфейсу, що дозволяє підлаштувати вигляд сайту під потреби бренду чи конкретного проекту.

У якості маршрутизатора по сайту великого вибору не існує. Для React існує дуже відома, і мабуть єдина бібліотека — React Router DOM [12].

React Router DOM — це бібліотека для керування маршрутизацією в додатках на основі React, що дозволяє створювати багатосторінкові додатки з динамічним оновленням контенту без перезавантаження сторінки.

- **Динамічні маршрути:** можливість створення маршрутів, що динамічно змінюються в залежності від стану додатку чи користувацьких дій.
- **Простота налаштування:** інтуїтивно зрозумілий синтаксис та гнучкі налаштування дозволяють швидко інтегрувати маршрутизацію у додаток.
- **Підтримка історії браузера:** забезпечує коректну роботу кнопок «Назад» та «Вперед» у браузері, що покращує користувацький досвід.

При обиранні бібліотеки для запитів одразу було обрано Axios [13]. Це дуже відома бібліотека яка має свою заслужену популярність через правильно налаштовану роботу, та досить зрозумілий у використанні функціонал.

Axios — це популярна бібліотека для виконання HTTP-запитів у додатках на основі JavaScript, що дозволяє легко взаємодіяти з API та сервером.

- **Простота використання:** Axios надає зручний синтаксис для виконання HTTP-запитів, таких як GET, POST, PUT, DELETE, що значно спрощує роботу з сервером.
- **Підтримка промісів:** Використовує проміси (Promises), що дозволяє легко обробляти асинхронні операції та уникати так званого «callback hell».
- **Обробка запитів та відповідей:** Axios автоматично обробляє JSON-дані у запитах та відповідях, що спрощує роботу з форматуванням даних.
- **Налаштовувані інтерцептори:** Інтерцептори дозволяють перехоплювати та змінювати запити або відповіді перед їх відправкою чи обробкою, що корисно для додавання токенів автентифікації чи ведення журналу активності.

- **Підтримка скасування запитів:** Axios дозволяє скасовувати запити, що може бути корисним для оптимізації роботи додатку та зменшення навантаження на сервер.
- **Адаптери для різних середовищ:** Підтримує різні адаптери, що дозволяє використовувати Axios як на стороні клієнта (в браузері), так і на стороні сервера (Node.js [14]).
- **Обробка помилок:** Забезпечує зручний механізм для обробки помилок HTTP-запитів, що допомагає швидко виявляти та виправляти проблеми в додатку.

Для створення форм на веб-сайті існує багато бібліотек серед яких є 2 лідери це – Formik [15] та React-hook-form [16]. Оскільки на сайті немає дуже складних форм, одразу було обрано Formik, так як React-hook-form більше використовують саме для складних форм на декілька етапів.

Formik — це бібліотека для керування формами в додатках на основі React, що спрощує процес створення, валідації та обробки форм.

- **Простота використання:** Formik надає зручний та інтуїтивно зрозумілий API, що дозволяє легко створювати та керувати формами у React-додатках.
- **Управління станом форми:** Formik автоматично керує станом форми, зберігаючи значення полів та статуси валідації, що значно спрощує роботу розробників.
- **Валідація:** Підтримує різні методи валідації, включаючи синхронну та асинхронну валідацію, що дозволяє легко перевіряти дані на коректність перед їх відправкою.

Вибір React, Redux, Material-UI та React Router DOM для розробки веб-сайту громадської організації обумовлений їхньою здатністю забезпечити високу продуктивність, зручність користування та ефективність розробки. Ці технології дозволяють створити сучасний, функціональний та естетично привабливий веб-сайт, що відповідає вимогам сучасних користувачів та сприяє

досягненню цілей організації. Додатково було встановлено декілька бібліотек для спрощення розробки веб-сайту, всі використані бібліотеки та пакети можна побачити на Рисунку 1.1.

```
"dependencies": {  
  "@emotion/react": "^11.11.4",  
  "@emotion/styled": "^11.11.5",  
  "@mui/icons-material": "^5.15.15",  
  "@mui/material": "^5.15.15",  
  "@reduxjs/toolkit": "^2.2.3",  
  "@swiper/core": "^0.0.21",  
  "@testing-library/jest-dom": "^5.17.0",  
  "@testing-library/react": "^13.4.0",  
  "@testing-library/user-event": "^13.5.0",  
  "@types/jest": "^27.5.2",  
  "@types/node": "^16.18.89",  
  "@types/react": "^18.2.66",  
  "@types/react-dom": "^18.2.22",  
  "axios": "^1.6.8",  
  "formik": "^2.4.6",  
  "moment": "^2.30.1",  
  "react": "^18.2.0",  
  "react-dom": "^18.2.0",  
  "react-redux": "^9.1.0",  
  "react-router-dom": "^6.22.3",  
  "react-scripts": "5.0.1",  
  "redux-saga": "^1.3.0",  
  "typescript": "^4.9.5",  
  "web-vitals": "^2.1.4"  
}
```

Рисунок 1.1 – Список всіх технологій для розробки веб-сайту

Крім того, ці інструменти дозволяють значно скоротити час розробки завдяки своїм готовим рішенням та бібліотекам компонентів. Це особливо важливо для громадської організації, яка прагне швидко реагувати на зміни та актуальні потреби своєї аудиторії. Завдяки використанню цих технологій, веб-сайт може бути легко масштабованим та адаптованим до зростаючих вимог.

Важливо також відзначити, що вибір саме цих технологій забезпечує високу підтримуваність коду та можливість його подальшого розвитку. Це означає, що в майбутньому додавання нових функцій або внесення змін буде здійснюватися швидко та без значних витрат ресурсів. Усі ці фактори роблять вибір React, Redux, Material-UI та React Router DOM оптимальним для створення веб-сайту громадської організації, що відповідає найвищим стандартам якості та функціональності.

1.2. Архітектура та структура веб-сайту

У цьому розділі буде розглянуто архітектурні рішення та структура веб-сайту, що розробляється. Від правильного вибору архітектури залежить зручність підтримки коду, продуктивність додатку та можливість його масштабування у майбутньому.

Компонентна архітектура

Основою архітектури веб-сайту є компонентний підхід (Рис. 1.2), що використовується у React. Кожен компонент відповідає за конкретну частину інтерфейсу і може бути повторно використаний у різних частинах додатку.

- **Контейнерні та презентаційні компоненти:** розподіл компонентів на контейнерні (які відповідають за бізнес-логіку та взаємодію з Redux) і презентаційні (які відповідають за відображення даних) сприяє чистоті коду та полегшує його тестування.
- **Компоненти вищого порядку (НОС):** використання компонентів вищого порядку для повторного використання логіки між різними компонентами, що дозволяє зменшити дублювання коду.

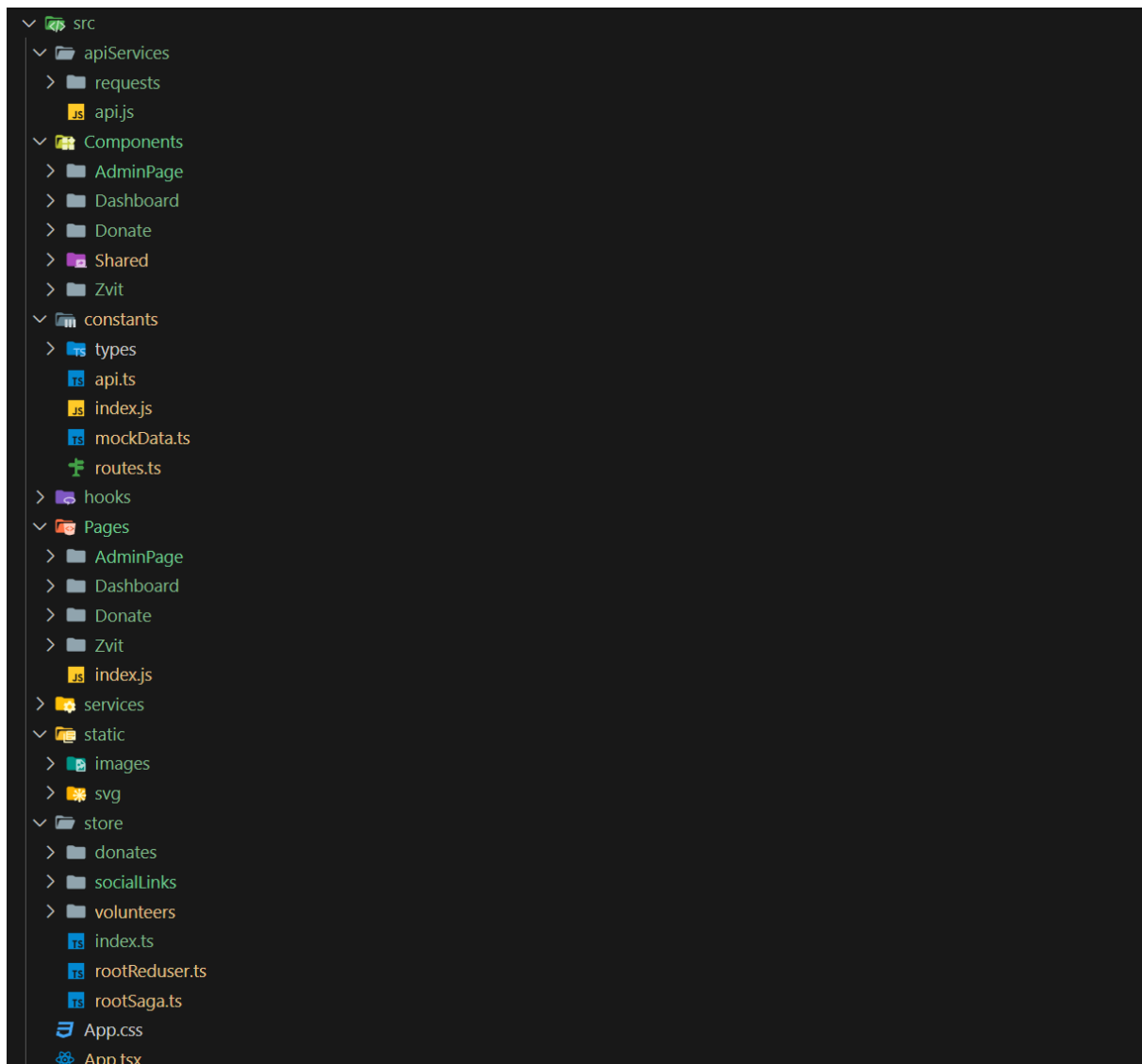


Рисунок 1.2 – Вигляд компонентної архітектури

Дерево компонентів

Дерево компонентів визначає, як компоненти взаємодіють один з одним і як дані передаються між ними. Грамотно побудоване дерево компонентів сприяє легкому управлінню станом та зменшує кількість непотрібних перерендерів.

- **Глибина дерева компонентів:** варто уникати надто глибоких дерев компонентів, оскільки це може призвести до складнощів в управлінні станом та продуктивністю.
- **Розподіл відповідальності:** кожен компонент повинен мати чітко визначену відповідальність, що спрощує його тестування та підтримку.

Структура файлів і папок

Правильна організація файлів і папок проекту є важливою для зручної навігації по коду та його підтримки.

- **Розподіл за функціональними модулями:** структурування файлів за функціональними модулями (наприклад, компоненти, редюсери, дії) сприяє логічній організації проекту.
- **Дотримання стандартів:** використання загальноприйнятих стандартів іменування файлів та папок сприяє зручності роботи в команді та спрощує інтеграцію нових розробників.

Управління станом

- Використання Redux для управління станом додатку дозволяє централізовано зберігати всі дані та забезпечувати їх узгодженість.
- **Структура стану:** структура стану повинна бути зрозумілою та легкою для навігації, з чітким розподілом даних за відповідними розділами.
- **Модульний підхід:** розподіл стану за модулями (редюсерами) дозволяє легко керувати окремими частинами стану та спрощує їх тестування

Правильна архітектура та структура веб-сайту є ключовими для його успішної розробки та підтримки. Використання компонентного підходу (Рис. 1.3), грамотне управління станом та організація файлів сприяють зручності роботи з кодом та забезпечують можливість подальшого масштабування додатку.



Рисунок 1.3 – Вигляд стану Redux.

1.3. Реалізація ключових функцій та інтеграція компонентів

У цьому розділі розглянуто процес реалізації ключових функцій веб-сайту та інтеграції різних компонентів. Детальний опис кожного етапу розробки допоможе зрозуміти, як були реалізовані основні функціональні можливості сайту.

Реалізація користувацьких сторінок

Однією з важливих частин веб-сайту є реалізація користувацьких сторінок, таких як головна сторінка, сторінка про організацію, новини, заходи тощо.

- **Головна сторінка:** повинна містити основну інформацію про організацію, останні новини та події, заклики до дії (СТА) для залучення волонтерів.
- **Сторінка новин та подій:** актуальні новини та анонси майбутніх заходів з можливістю підписки на розсилку. Як приклад реалізації добавлено код

у Додатку А. В ньому відображено компонент, який відмальовує блок однієї новини, кількість таких блоків на сторінці залежить від кількості новин у базі даних.

- **Сторінка про організацію:** детальна інформація про місію, бачення та цілі організації, її історію та досягнення. Як приклад реалізації додано код у Додатку В. В ньому відображено редюсер який дозволяє з будь якого місця додатку зробити запит, та отримати актуальний список волонтерів збережений у пам'яті веб-сайту.

Інтерактивні елементи

Важливою частиною сучасного веб-сайту є інтерактивні елементи, що підвищують залученість користувачів.

- **Форми зворотного зв'язку:** інтеграція форм зворотного зв'язку, що дозволяють користувачам легко зв'язатися з організацією, залишити відгуки чи задати питання.

- **Форма реєстрації волонтерів:** зручна форма для реєстрації нових волонтерів з можливістю вибору напрямків діяльності та часу участі. (Рис. 1.4)

Додавання Волонтерів

Ім'я

Введіть ім'я

Прізвище

Введіть прізвище

Посилання на соціальні мережі

Посилання на профіль

Upload Photo

or drop file

ВІДПРАВИТИ

Рисунок 1.4 – Форма додавання нових волонтерів

- **Інтерактивна карта заходів:** відображення на карті місць проведення заходів з можливістю перегляду деталей та реєстрації на участь.

Інтеграція з соціальними мережами

- Інтеграція веб-сайту з соціальними мережами дозволяє розширити аудиторію та підвищити впізнаваність організації.
- **Кнопки соціальних мереж:** додавання кнопок для швидкого поширення інформації про заходи та новини у соціальних мережах.
- **Віджети соціальних мереж:** інтеграція віджетів, що відображають стрічку новин з офіційних сторінок організації у соціальних мережах.

Функції для адміністраторів

Для зручного управління контентом на сайті необхідно реалізувати функції для адміністраторів.

- **Панель управління:** розробка панелі управління, що дозволяє адміністраторам додавати, редагувати та видаляти новини, заходи та інші матеріали.
- **Управління користувачами:** можливість керування користувачами сайту, надання прав доступу, перегляд активності.

Тестування та оптимізація

Для забезпечення високої якості веб-сайту важливо провести всебічне тестування та оптимізацію.

- **Тестування користувацького інтерфейсу:** перевірка роботи всіх функцій сайту на різних пристроях та браузерах.
- **Оптимізація продуктивності:** використання сучасних інструментів для оптимізації завантаження сторінок та швидкості роботи сайту.
- **Безпека:** забезпечення захисту даних користувачів та захист сайту від можливих атак.

Реалізація ключових функцій та інтеграція компонентів є важливим етапом розробки веб-сайту. Детальне планування та виконання кожного з етапів забезпечують створення функціонального, зручного та безпечного веб-ресурсу, що відповідає потребам громадської організації та її користувачів.

РОЗДІЛ 2 Дизайн веб-сайту

2.1 Принципи дизайну

Принципи дизайну відіграють ключову роль у створенні привабливого та зручного для користувачів веб-сайту. Дизайн передчасно було створено у застосунку **Figma** [17]. При розробці веб-сайту для громадської організації ми керувалися наступними принципами:

Мінімалізм

Мінімалістичний дизайн дозволяє сконцентрувати увагу користувачів на основних елементах та інформації. Використання чистих ліній, простих форм і обмеженої кольорової палітри допомагає уникнути зайвої візуальної метушні, що сприяє легшому сприйняттю контенту.

Кольорова палітра

При виборі кольорової палітри ми зупинилися на спокійних та гармонійних кольорах, які створюють позитивний емоційний настрій. Основною думкою було те, що потрібно використовувати кольорову палітру під основні кольори логотипу компанії. Основний колір - синій, що асоціюється з надійністю та стабільністю, додаткові кольори — білий, жовтий та сірий, що забезпечують контраст та чіткість.

Юзабіліті

Зручність використання (юзабіліті) — один з головних пріоритетів у дизайні нашого веб-сайту. Інтуїтивно зрозуміла навігація, великі кнопки для взаємодії та чітка структура сторінок забезпечують легкий доступ до всієї необхідної інформації.

Адаптивний дизайн

Сайт розроблений з використанням принципів адаптивного дизайну, що забезпечує коректне відображення та функціонування на різних пристроях: комп'ютерах, планшетах та смартфонах. Це дозволяє користувачам зручно користуватися сайтом незалежно від їхнього місця знаходження та типу

пристрою. Дизайн налаштовано для коректного відображення від 900px* та до 2560px.

***PX** - піксель це найдрібніша одиниця цифрового зображення в растровій графіці.

Типографіка

Для текстових елементів було обрано шрифт, який забезпечує високу читабельність. Всі заголовки, підзаголовки та основний текст мають чітку ієрархію, що сприяє легкому орієнтуванню на сторінці.

Консистентність

Консистентність у дизайні означає використання однакових стилів, шрифтів та кольорів на всіх сторінках сайту. Це створює відчуття цілісності та професійності, що підвищує довіру користувачів до веб-сайту.

Використання зображень та графіки

Зображення та графічні елементи використовуються для ілюстрації діяльності організації, підвищення візуальної привабливості та підтримки контенту. Всі зображення мають високу якість та відповідають загальній стилістиці сайту.

Виходячи з усіх вищезгаданих принципів, при створенні дизайну ми користувалися здобутим до цього досвідом та сучасними тенденціями в веб-дизайні. Це дозволило нам розробити веб-сайт, який не тільки відповідає функціональним вимогам, але й виглядає сучасно та привабливо для користувачів. Дотримання принципів мінімалізму, використання гармонійної кольорової палітри, забезпечення високого рівня юзабіліті, адаптивний дизайн, чітка типографіка та консистентність у всіх елементах дизайну сприяють створенню цілісного та зручного для користувачів продукту. Ми впевнені, що такий підхід сприятиме ефективному досягненню цілей громадської організації та забезпечить позитивний досвід взаємодії з сайтом для всіх користувачів.

2.2 Управління контентом сайту та підтримка

Веб-сайт громадської організації забезпечує високий рівень підтримки для відвідувачів та користувачів. При розробці ми працювали над тим, щоб зробити досвід використання сайту максимально зручним та ефективним.

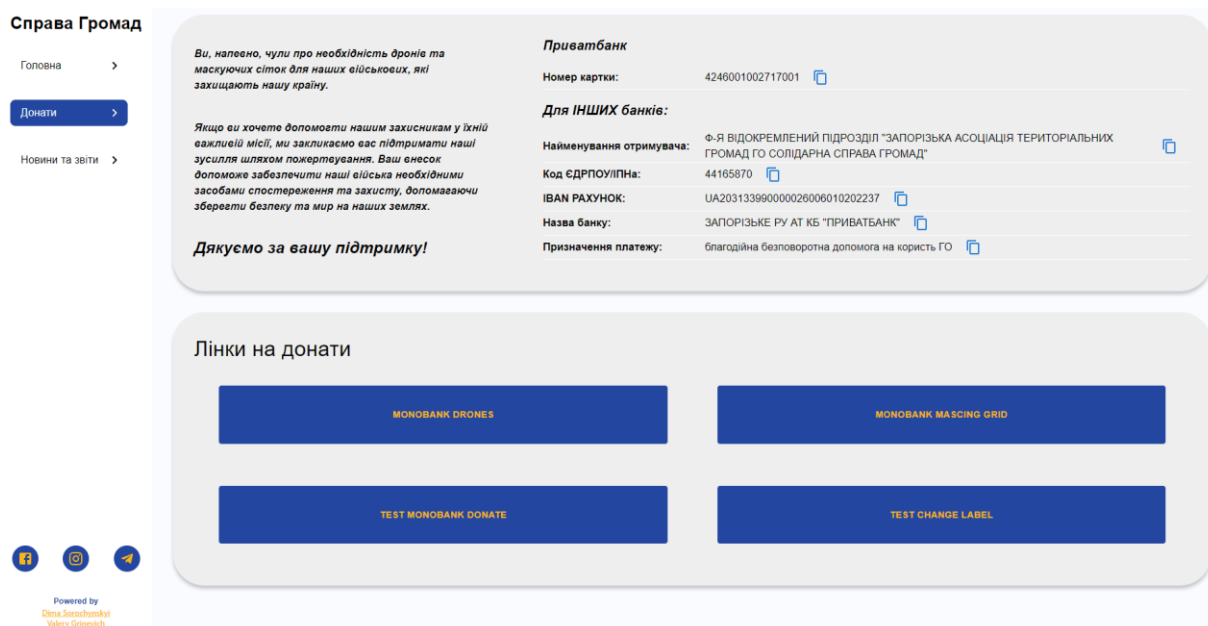


Рисунок 2.1 – Сторінка донатів з відображенням бокової панелі

Для оптимізації простору та підвищення зручності використання, сайт буде розділений на чотири основні сторінки:

- На сторінці **"Про нас"** коротко, але інформативно представлена інформація про мету та діяльність громадської організації. Відвідувачі сайту можуть швидко ознайомитись з основними цілями та завданнями організації, її історією та досягненнями. Також на сторінці реалізовано відображення кожного члену громадської організації та можливе посилання для зв'язку з ним. Це все допоможе стати трішки ближче до команди.
- Сторінка **"Донатити"** містить зручні кнопки для пожертвувань, які спрямовані на різні платіжні системи або банки. Кожна кнопка має чітке підпис, що дозволяє користувачам легко вибрати необхідний спосіб пожертвування.

Також на цій сторінці подані реквізити для переказу коштів, що забезпечує можливість пожертвування тим, хто віддає перевагу цьому способу перед онлайн-платежами. Коротке повідомлення з подякою за пожертвування створює теплу атмосферу та підтримує зв'язок із відвідувачами.

- Сторінка "**Новини та звіти**" регулярно оновлюється і містить найсвіжіші новини та звіти про діяльність організації. Інформація на цій сторінці дозволяє відвідувачам бути в курсі останніх подій та досягнень громадської організації. Як приклад реалізації добавлено код у **Додатку А**

- Сторінка "**Admin Page**" призначена для адміністратора сайту та містить інструменти для редагування контенту. Після входу в систему адміністратор може редагувати та оновлювати новини та звіти, додавати або видаляти їх. Також на цій сторінці адміністратор може редагувати кнопки для пожертвувань, змінюючи їх вміст або додаючи нові, а також оновлювати контактну інформацію для зв'язку з організацією.

Для забезпечення зручного спілкування з користувачами веб-сайту передбачено кілька засобів зв'язку:

- **Електронна пошта:** Для письмового зв'язку з користувачами передбачена адреса, де оперативно відповідають на запитання та пропозиції.

- **Телефонний номер:** Для термінового зв'язку встановлений номер, де готові надати консультації та відповісти на питання.

- **Telegram:** Спілкування з користувачами також можливе через Telegram-канал, до якого можна приєднатися за посиланням на веб-сайті.

- **Instagram:** Для активного спілкування та інформування користувачів про події та новини організації, створено Instagram-акаунт, до якого можна підписатися за посиланням на веб-сайті.

- **Facebook:** Facebook-акаунт є ще одним способом підтримки зв'язку з аудиторією. Тут публікуються новини, анонси та інша важлива інформація, а також Facebook-акаунт використовується для взаємодії з підписниками через коментарі та повідомлення.

- Instagram, Telegram та Facebook допомагають підтримувати активний зв'язок з аудиторією. Instagram дозволяє показати роботу у привабливому вигляді та привернути увагу нових користувачів.

Моніторинг та підтримка сайту є важливими аспектами забезпечення його стабільного функціонування та високої якості обслуговування користувачів. У цьому розділі розглянемо ключові методи та інструменти, які використовуються для відстеження роботи сайту, виявлення потенційних проблем та їх оперативного вирішення. Це включає використання систем моніторингу, аналізу даних, резервного копіювання інформації та безпеки сайту та всього додатку в цілому.

Моніторинг веб-сайту є критичним елементом підтримки, який дозволяє виявити проблеми на ранніх етапах та оперативно їх вирішити. Основні інструменти моніторингу включають:

Google Analytics: Це один з найпопулярніших інструментів для моніторингу веб-сайтів. Він дозволяє відстежувати трафік, джерела відвідувань, поведінку користувачів на сайті, а також визначати ефективність маркетингових кампаній. Google Analytics забезпечує широкий спектр звітів, які допомагають зрозуміти, як користувачі взаємодіють із сайтом.

Pingdom: Це інструмент для моніторингу часу завантаження та доступності сайту. Pingdom дозволяє відстежувати, як швидко завантажуються сайт у різних регіонах світу, і надсилає сповіщення у разі виникнення проблем з доступністю.

New Relic: Цей інструмент дозволяє відстежувати продуктивність веб-додатків та серверів у реальному часі. New Relic надає детальні звіти про час завантаження сторінок, запити до баз даних та інші важливі параметри продуктивності.

Аналіз даних є важливим етапом у забезпеченні стабільної роботи сайту. Основні кроки включають:

Аналіз відвідуваності: Вивчення звітів про відвідуваність сайту дозволяє визначити, які сторінки є найбільш популярними, звідки приходять користувачі

та як вони взаємодіють із контентом. Це допомагає виявити сильні та слабкі сторони сайту та прийняти обґрунтовані рішення щодо його оптимізації.

Оптимізація часу завантаження: Швидкість завантаження сторінок є одним з ключових факторів користувацького досвіду. Інструменти, такі як Google PageSpeed Insights [18] та GTmetrix [19], допомагають визначити, які елементи сайту впливають на час завантаження, і надають рекомендації щодо їх оптимізації.

Тестування навантаження: Використання інструментів для тестування навантаження, таких як Apache JMeter [20] або LoadRunner, дозволяє визначити, як сайт працює під високими навантаженнями. Це допомагає підготувати сайт до пікових періодів відвідуваності та забезпечити його стабільну роботу.

Резервне копіювання є невід'ємною частиною підтримки сайту. Регулярне створення резервних копій дозволяє відновити сайт у разі технічних проблем або втрати даних. Основні аспекти резервного копіювання включають:

Регулярні резервні копії: Необхідно створювати резервні копії бази даних та файлів сайту на регулярній основі. Це можна здійснювати автоматично за допомогою спеціальних інструментів або скриптів.

Зберігання резервних копій: Резервні копії повинні зберігатися на зовнішніх серверах або у хмарних сховищах для забезпечення їхньої безпеки. Важливо також зберігати кілька версій резервних копій на випадок виникнення проблем із останньою версією.

Тестування відновлення: Регулярне тестування процесу відновлення з резервних копій допомагає переконатися, що у разі потреби сайт можна швидко відновити без втрати даних.

Безпека сайту є критичним аспектом його підтримки. Основні кроки для забезпечення безпеки включають:

Використання HTTPS: Використання протоколу HTTPS забезпечує захищене з'єднання між сервером та користувачем, що запобігає перехопленню даних.

Оновлення програмного забезпечення: Регулярне оновлення CMS, плагінів та інших компонентів сайту допомагає захистити його від відомих вразливостей.

Захист від атак: Використання фаєрволів, антивірусного програмного забезпечення та інших інструментів для захисту сайту від атак, таких як DDoS, SQL-ін'єкції та XSS.

Регулярні перевірки безпеки: Проведення регулярних перевірок безпеки, таких як сканування на вразливості та перевірка логів, допомагає виявити потенційні проблеми та оперативно їх усунути.

Моніторинг та підтримка сайту є важливими складовими забезпечення його стабільної роботи та високої якості обслуговування користувачів. Використання сучасних інструментів моніторингу, аналізу даних, резервного копіювання та безпеки допомагає забезпечити надійність та ефективність роботи сайту. Регулярне тестування та оптимізація сайту дозволяють забезпечити високий рівень користувацького досвіду та задоволення потреб користувачів.

2.3 Адміністративна частина сайту

Адміністративна частина веб-сайту нашої громадської організації є важливим елементом, який дозволяє ефективно керувати контентом, забезпечуючи зручність та оперативність в оновленні інформації. Цей підрозділ сайту надає широкі можливості для адміністраторів, які залогінені в систему. Якщо користувач не залогінений, при спробі перейти до адміністративної частини він побачить форму логіну.

Авторизація користувачів

Авторизація є ключовим елементом безпеки адміністративної частини сайту. Користувачі, які бажають отримати доступ до розширених можливостей управління контентом, спершу повинні увійти до системи за допомогою форми

логіну. Форма логіну є простою та інтуїтивно зрозумілою, що дозволяє користувачам швидко отримати доступ до необхідних інструментів. Після успішного входу в систему, користувачі можуть використовувати всі можливості адміністративної частини сайту.

Можливості адміністративної частини

Після авторизації, користувачі отримують доступ до широкого спектру функцій, які дозволяють ефективно керувати контентом сайту. Основні можливості адміністративної частини включають:

1. Управління пожертвами (донатами)

Адміністратори мають можливість переглядати, редагувати та додавати нові пожертви. На панелі управління донатами відображається список усіх наявних пожертв, з коротким описом кожної з них. Кожна пожертва може бути відредагована: змінити назву, оновити посилання або змінити лейбл пожертви. Це забезпечує гнучкість у керуванні фінансовими ресурсами та дозволяє швидко реагувати на зміни в потребах організації.

2. Додавання новин

Форма для додавання новин дозволяє адміністраторам оперативно оновлювати інформацію на сайті. Користувач може додати до чотирьох фотографій та текст самої новини, що дозволяє створювати візуально привабливий та інформативний контент. Крім того, адміністратори мають можливість видаляти старі новини, забезпечуючи актуальність та своєчасність інформації.

3. Звіти про виконану роботу

Схоже до форми новин, форма для додавання звітів про виконану роботу дозволяє додавати до чотирьох фотографій та текстовий опис звіту. Адміністратори можуть додавати нові звіти, а також видаляти старі, забезпечуючи прозорість діяльності організації та інформування спільноти про досягнення та виконані завдання.

4. Управління даними про волонтерів

Адміністративна частина сайту також включає форму для додавання, редагування та видалення волонтерів. Адміністратори можуть вводити повне ім'я волонтера, його посаду, додавати посилання на соціальні мережі та завантажувати фотографію. Це дозволяє зручно керувати інформацією про команду, забезпечуючи точність та актуальність даних.

Розширені можливості адміністраторів

Адміністративна частина сайту надає розширені можливості для адміністраторів, що дозволяє їм ефективно управляти всіма аспектами контенту сайту. Кожен розділ має інтуїтивно зрозумілий інтерфейс, що забезпечує легкість використання та швидке виконання необхідних дій. Важливою особливістю адміністративної частини є можливість попереднього перегляду змін перед їх опублікуванням, що дозволяє уникнути помилок та забезпечити високу якість контенту.

Безпека та конфіденційність

Безпека адміністративної частини є пріоритетом у розробці нашого веб-сайту. Всі дії адміністратора логуються, що дозволяє відстежувати зміни та забезпечувати контроль доступу. Крім того, всі дані передаються через захищений з'єднання (SSL), що забезпечує конфіденційність та захист інформації від несанкціонованого доступу.

Зручність та доступність

Ми прагнули зробити адміністративну частину сайту максимально зручною та доступною для користувачів з різним рівнем технічної підготовки. Інтуїтивно зрозумілий інтерфейс, детальні інструкції та підказки роблять роботу з адміністративною частиною легкою та ефективною. Крім того, ми передбачили можливість доступу до адміністративної частини з мобільних пристроїв, що дозволяє адміністраторам керувати контентом у будь-який час та з будь-якого місця.

Адміністративна частина веб-сайту нашої громадської організації є важливим інструментом для забезпечення ефективного управління контентом.

Широкий спектр функцій, інтуїтивно зрозумілий інтерфейс та висока безпека роблять її незамінним помічником для адміністраторів. Можливість керування пожертвами, додавання новин, звітів та управління даними про волонтерів забезпечує оперативність та актуальність інформації на сайті, що сприяє досягненню цілей організації та підтримці високого рівня взаємодії з аудиторією. Завдяки зручності та доступності адміністративної частини, ми можемо швидко реагувати на зміни та забезпечувати ефективне управління ресурсами, що є важливим фактором успіху нашої організації.

РОЗДІЛ 3 Підтримка та розвиток веб-сайту

3.1 Підтримка веб-сайту

Підтримка веб-сайту є важливою складовою забезпечення його стабільної та безперебійної роботи. До основних аспектів підтримки належать регулярні оновлення контенту та програмного забезпечення, виправлення помилок, моніторинг роботи сайту та забезпечення безпеки.

Оновлення контенту

Оновлення контенту включає в себе додавання нових статей, новин, звітів та іншої інформації, яка є важливою для користувачів. Регулярне оновлення контенту дозволяє підтримувати актуальність інформації на сайті та залучати нових відвідувачів.

- **Планування оновлень:** важливо мати чіткий план оновлень, який включає дати публікації нових матеріалів та відповідальних осіб.
- **Автоматизація процесу:** використання систем управління контентом (CMS) дозволяє автоматизувати процес оновлення контенту, що значно спрощує завдання для адміністраторів сайту.
- **Моніторинг актуальності:** регулярний перегляд та оновлення існуючих матеріалів для забезпечення їх відповідності актуальним даним та вимогам.

Оновлення програмного забезпечення

Своєчасне оновлення програмного забезпечення є ключовим аспектом забезпечення безпеки та стабільності роботи веб-сайту. Це включає оновлення версій CMS, плагінів, бібліотек та інших компонентів.

- **План оновлень:** створення графіку регулярних оновлень програмного забезпечення.
- **Безпечне оновлення:** забезпечення безпечного процесу оновлення, включаючи резервне копіювання даних перед оновленням.

- **Тестування після оновлення:** перевірка функціональності сайту після кожного оновлення для виявлення можливих проблем.

Виправлення помилок

Виявлення та виправлення помилок є невід'ємною частиною підтримки веб-сайту. Це включає як технічні помилки, так і логічні недоліки в контенті.

- **Система відстеження помилок:** використання інструментів для відстеження та управління помилками.
- **Пріоритезація помилок:** класифікація помилок за рівнем критичності та впливу на користувачів.
- **Процес виправлення:** розробка та впровадження чіткого процесу виправлення помилок, включаючи тестування виправлень перед їх впровадженням на основний сайт.

Моніторинг роботи сайту

Моніторинг роботи сайту включає відстеження його доступності, продуктивності та поведінки користувачів. Це дозволяє виявляти проблеми на ранніх стадіях та оперативно реагувати на них.

Інструменти аналітики

Використання інструментів аналітики, таких як Google Analytics, для збору та аналізу даних про відвідувачів сайту, їх поведінку та джерела трафіку.

- **Налаштування аналітики:** встановлення та налаштування аналітичних інструментів.
- **Аналіз даних:** регулярний аналіз зібраних даних для виявлення тенденцій та проблемних зон.
- **Звітування:** створення звітів на основі аналітичних даних для подальшого використання у прийнятті рішень.

Перевірки працездатності

Регулярні перевірки працездатності сайту дозволяють виявляти та усувати проблеми, пов'язані з його функціональністю.

- **Автоматизовані перевірки:** використання інструментів для автоматизованого моніторингу працездатності сайту.
- **Регулярні тести:** проведення регулярних тестів працездатності всіх ключових функцій сайту.
- **Оперативне реагування:** розробка плану дій на випадок виявлення проблем з працездатністю.

Забезпечення безпеки

Безпека веб-сайту є пріоритетом у процесі його підтримки. Це включає захист від атак, резервне копіювання даних та забезпечення конфіденційності користувачів.

Захист від атак

Впровадження заходів захисту від хакерських атак та шкідливого програмного забезпечення.

- **Фаєрволи та антивіруси:** встановлення та налаштування фаєрволів, антивірусного програмного забезпечення та інших захисних засобів.
- **Регулярні перевірки безпеки:** проведення регулярних перевірок безпеки для виявлення вразливостей.
- **Навчання персоналу:** навчання адміністраторів та користувачів правилам безпеки.

Резервні копії

Регулярне створення резервних копій даних дозволяє забезпечити їх збереження у разі виникнення проблем.

- **Графік резервування:** створення графіку регулярного резервування даних.
- **Зберігання копій:** зберігання резервних копій у безпечних місцях.
- **Відновлення даних:** розробка та тестування процедур відновлення даних з резервних копій.

Двофакторна аутентифікація

Впровадження двофакторної аутентифікації для підвищення безпеки доступу до адміністративної частини сайту.

- **Налаштування аутентифікації:** встановлення та налаштування системи двофакторної аутентифікації.
- **Навчання користувачів:** навчання адміністраторів та користувачів використанню двофакторної аутентифікації.
- **Моніторинг безпеки:** регулярний моніторинг використання двофакторної аутентифікації для виявлення можливих проблем.

3.2 Розвиток веб-сайту

Розвиток веб-сайту передбачає постійне вдосконалення його функціональності та відповідність потребам користувачів. Використовуючи сучасні інструменти, такі як React, Redux, Formik, MUI, Axios та React Router DOM, ми можемо ефективно реалізувати ці кроки. Один із важливих аспектів розвитку сайту – це зворотний зв'язок від користувачів. Регулярне проведення опитувань серед користувачів дозволяє виявити їх потреби та побажання щодо функціональності та контенту сайту. Опитування можуть бути організовані за допомогою онлайн-сервісів, таких як Google Forms. Зібрані дані аналізуються для визначення основних напрямків розвитку сайту. На основі цього аналізу впроваджуються відповідні зміни. Наприклад, якщо більшість користувачів висловлюють бажання бачити більше мультимедійного контенту, то на сайті можуть бути додані нові розділи з відео та фото галереями.

Збір та аналіз коментарів та відгуків користувачів допомагає виявити проблемні зони та потенційні напрями розвитку. Коментарі та відгуки можуть бути зібрані з різних джерел, включаючи сайт та соціальні мережі. Аналіз зібраних відгуків дозволяє виявити тенденції та проблемні зони, що потребують

коригування. На основі цього аналізу коригується стратегія розвитку сайту. Наприклад, якщо користувачі скаржаться на складність навігації, можна вдосконалити структуру меню або додати функціонал пошуку по сайту.

Аналіз тенденцій у сфері веб-розробки також відіграє важливу роль у розвитку сайту. Відстеження нових технологій та трендів допомагає залишатися актуальними та конкурентоспроможними. Регулярний моніторинг нових технологій дозволяє оцінити можливість їх впровадження на сайті. Планування впровадження нових технологій та інструментів для покращення функціональності сайту є важливим етапом у цьому процесі. Наприклад, впровадження технології Progressive Web Apps (PWA) може значно покращити швидкість завантаження сайту та забезпечити його роботу навіть без підключення до інтернету.

Впровадження нових функцій та можливостей дозволяє покращити користувацький досвід та задовольнити потреби користувачів. Ідентифікація потреб користувачів на основі аналізу зворотного зв'язку та поведінки на сайті дозволяє розробити нові функції, які відповідають їхнім очікуванням. Нові функції розробляються та тестуються перед їх впровадженням. Оцінка ефективності впроваджених функцій допомагає визначити, наскільки вони відповідають потребам користувачів та коригувати стратегію розвитку сайту. Наприклад, можна додати функціонал особистого кабінету, де користувачі зможуть відстежувати свою активність на сайті, зберігати уподобання та налаштування.

Для реалізації цих кроків використовуються такі технології, як React для створення інтерактивних інтерфейсів, Redux для управління станом додатка, Formik для зручної роботи з формами, MUI для створення стильних та адаптивних компонентів, Axios для обміну даними з сервером, та React Router DOM для керування маршрутизацією в додатку. Використання цих інструментів дозволяє ефективно впроваджувати зміни та вдосконалювати функціональність сайту відповідно до потреб користувачів.

У майбутньому, для подальшого розвитку сайту можуть бути впроваджені наступні покращення. Покращення дизайну та користувацького інтерфейсу: За допомогою MUI можна буде впроваджувати нові теми та адаптивні компоненти, що покращать загальний вигляд сайту. Це може включати зміну кольорових схем, оновлення типографії та адаптивність для нових розмірів екранів. Додавання нових функціональних можливостей: Наприклад, інтеграція чат-ботів для автоматичного обслуговування користувачів, розширення можливостей особистого кабінету користувача (збереження історії пожертв, відстеження активності), впровадження функціоналу для проведення онлайн-заходів чи вебінарів.

Інтеграція з додатковими сервісами та платформами: Це може включати додавання нових платіжних систем для прийому пожертв, інтеграцію з CRM-системами для кращого управління взаємодією з волонтерами та спонсорами, або інтеграцію з новими соціальними мережами для ширшого охоплення аудиторії. Оптимізація продуктивності та швидкості завантаження: Використання інструментів, таких як Webpack [21] та серверного рендерингу (Server-Side Rendering) з React, дозволить оптимізувати швидкість завантаження сайту та покращити користувацький досвід, особливо для користувачів з повільним інтернет-з'єднанням. Підвищення рівня безпеки: Впровадження двофакторної аутентифікації для користувачів, покращення механізмів захисту від атак типу XSS та CSRF, регулярні аудити безпеки коду та інші заходи, що забезпечать захист персональних даних користувачів та фінансових транзакцій. Покращення аналітики та звітності: Інтеграція з потужними аналітичними інструментами, такими як Google Analytics або Piwik, дозволить більш детально аналізувати поведінку користувачів на сайті, визначати успішність маркетингових кампаній та виявляти можливості для подальшого покращення функціональності сайту.

Завдяки реалізації цих кроків та впровадженню нових технологій, наш веб-сайт зможе постійно вдосконалюватися та відповідати вимогам сучасних користувачів, забезпечуючи їм зручність, безпеку та задоволення від

користування. Наприклад, можливе впровадження системи рекомендацій контенту на основі штучного інтелекту, що дозволить користувачам отримувати персоналізовані рекомендації щодо новин, статей чи заходів, які можуть їх зацікавити. Також можна розширити функціонал для мобільних пристроїв, розробивши спеціальні мобільні додатки або адаптовану версію сайту для смартфонів та планшетів.

3.3 Інтеграція з іншими системами та сервісами

Інтеграція з іншими системами дозволяє підвищити функціональність веб-сайту та забезпечити зручність для користувачів. Впровадження інтеграцій може включати різні аспекти, такі як інтеграція з соціальними мережами, платіжними системами, системами управління проектами та волонтерами. Це дозволяє розширити можливості сайту та забезпечити його ефективне функціонування. Розглянемо кожен з цих аспектів детальніше.

Інтеграція з соціальними мережами

Інтеграція з соціальними мережами має на меті підвищення видимості та залученості користувачів. Це може включати впровадження кнопок для поширення контенту та використання API соціальних мереж для автоматичного публікування новин та оновлень.

Кнопки для поширення контенту

Вбудовування кнопок для швидкого поширення контенту в соціальних мережах дозволяє підвищити видимість та залученість користувачів. Це важливий елемент, що сприяє активному залученню аудиторії та розширенню охоплення.

- **Розміщення кнопок:** Визначення оптимальних місць для розміщення кнопок на сайті є критичним. Це можуть бути сторінки з новинами,

блогом або товарами. Розміщення кнопок у видимих та зручних місцях сприяє більш активному їх використанню користувачами.

- **Аналіз ефективності:** Аналіз ефективності використання кнопок для поширення контенту включає відстеження кількості поширень, кліків та залученості користувачів. Це дозволяє оцінити, наскільки ефективно кнопки сприяють досягненню поставлених цілей.
- **Коригування стратегії:** Коригування стратегії використання кнопок на основі аналізу ефективності допомагає оптимізувати їх використання. Наприклад, зміна розміщення кнопок або їх дизайну може підвищити їх ефективність.

Інтеграція з API соціальних мереж

Використання API соціальних мереж для автоматичного публікування новин та оновлень дозволяє зекономити час та забезпечити оперативність.

- **Вибір платформ:** Вибір платформ для інтеграції на основі потреб користувачів та популярності мереж є важливим кроком. Наприклад, інтеграція з Facebook, Twitter та Instagram може бути пріоритетною, якщо ці платформи популярні серед цільової аудиторії.
- **Налаштування API:** Налаштування та тестування API для забезпечення стабільної роботи включає створення та налаштування ключів доступу, перевірку функціональності та виявлення можливих проблем. Регулярне тестування та моніторинг допомагають забезпечити безперебійну роботу інтеграції.
- **Моніторинг інтеграції:** Регулярний моніторинг інтеграції для виявлення та виправлення можливих проблем включає відстеження помилок, перевірку логів та аналіз показників ефективності. Це дозволяє швидко реагувати на зміни та забезпечувати стабільну роботу інтеграції.

Інтеграція з платіжними системами

Інтеграція з платіжними системами дозволяє забезпечити можливість здійснювати онлайн-пожертви та інші фінансові транзакції, що сприяє залученню додаткових фінансових ресурсів.

Онлайн-пожертви

Впровадження можливості здійснювати онлайн-пожертви через популярні платіжні системи дозволяє залучати додаткові фінансові ресурси.

- **Вибір платіжних систем:** Вибір найбільш підходящих платіжних систем для інтеграції включає аналіз популярності та зручності використання різних платіжних систем. Наприклад, інтеграція з PayPal, Stripe або місцевими платіжними системами може бути пріоритетною.

- **Гарантування безпеки:** Впровадження заходів для забезпечення безпеки фінансових транзакцій є ключовим аспектом. Це може включати використання SSL сертифікатів, шифрування даних та багатофакторну аутентифікацію для захисту користувачів.

- **Аналіз ефективності:** Аналіз ефективності інтеграції платіжних систем включає відстеження кількості транзакцій, середньої суми пожертв та задоволеності користувачів. Це допомагає оцінити, наскільки ефективно платіжні системи сприяють досягненню фінансових цілей.

Гарантування безпеки транзакцій

Гарантування безпеки транзакцій є важливим аспектом інтеграції платіжних систем.

- **Використання захисних засобів:** Впровадження засобів захисту, таких як SSL сертифікати та шифрування даних, забезпечує захист інформації користувачів під час проведення транзакцій.

- **Моніторинг транзакцій:** Регулярний моніторинг транзакцій для виявлення підозрілих активностей включає аналіз логів, виявлення аномалій та відстеження незвичайної активності. Це допомагає запобігати шахрайству та забезпечувати безпеку користувачів.

- **Навчання користувачів:** Навчання користувачів безпечному проведенню транзакцій включає створення інструкцій, рекомендацій та проведення тренінгів. Це підвищує обізнаність користувачів та допомагає запобігати можливим загрозам.

Інтеграція з системами управління проектами та волонтерами

Інтеграція з системами управління проектами та волонтерами дозволяє підвищити ефективність роботи організації та покращити координацію дій.

Системи управління завданнями

Використання систем для управління завданнями та проектами організації дозволяє підвищити ефективність роботи.

- **Вибір систем:** Вибір найбільш підходящих систем для інтеграції включає аналіз популярності, зручності використання та функціональності різних систем. Наприклад, інтеграція з Asana, Trello або Jira може бути пріоритетною.
- **Налаштування інтеграції:** Налаштування та тестування інтеграції для забезпечення стабільної роботи включає налаштування API, створення робочих просторів та налаштування доступу. Регулярне тестування допомагає виявити та виправити можливі проблеми.
- **Моніторинг ефективності:** Регулярний моніторинг ефективності інтеграції включає відстеження виконання завдань, аналіз продуктивності та оцінку задоволеності користувачів. Це дозволяє вчасно коригувати стратегію та забезпечувати ефективність роботи.

Платформи для волонтерів

Інтеграція з платформами для залучення та координації роботи волонтерів дозволяє значно покращити організацію роботи та ефективність діяльності. Вибір найбільш підходящих платформ включає ретельний аналіз популярності та функціональності різних варіантів. Наприклад, платформи такі як VolunteerMatch та Golden можуть бути пріоритетними завдяки своїм широким можливостям та зручності використання.

Процес налаштування інтеграції передбачає створення профілів волонтерів, налаштування доступу та комунікаційних каналів, що забезпечує стабільну роботу системи. Регулярне тестування допомагає виявляти можливі проблеми та вчасно їх усувати, забезпечуючи надійність інтеграції.

Ефективність інтеграції з платформами для волонтерів постійно моніториться шляхом аналізу задіяності волонтерів, оцінки їх задоволеності та аналізу виконання завдань. Це дозволяє своєчасно вносити необхідні корективи та забезпечувати ефективну роботу волонтерів, що є критично важливим для успішної реалізації проектів громадської організації.

Інтеграція з іншими системами та сервісами є важливим аспектом розвитку веб-сайту. Вона дозволяє підвищити функціональність, зручність та ефективність роботи сайту. Впровадження інтеграцій потребує ретельного планування, налаштування та регулярного моніторингу для забезпечення стабільної роботи та досягнення поставлених цілей.

3.4 План розвитку веб-сайту

План розвитку веб-сайту є основним стратегічним документом, який визначає напрями, завдання та необхідні ресурси для досягнення поставлених цілей. Він включає в себе стратегічне планування, аналіз цільової аудиторії, ресурсне планування та оцінку ефективності. Розглянемо кожен із цих аспектів детальніше.

Стратегічне планування

Першим кроком у стратегічному плануванні є визначення основних напрямків розвитку веб-сайту на найближчі роки. Це дозволяє чітко планувати ресурси та заходи, необхідні для досягнення цілей.

Аналіз поточного стану веб-сайту є важливою частиною стратегічного планування. Він включає оцінку існуючих функціональних можливостей, контенту, дизайну та технічної інфраструктури. Аналіз допомагає виявити сильні та слабкі сторони сайту, а також можливості для покращення. Зібрані дані слугують основою для визначення напрямків розвитку.

Наступним кроком є визначення пріоритетних напрямків розвитку на основі потреб цільової аудиторії та ринкових тенденцій. Це може включати розширення функціональності сайту, покращення користувацького досвіду, оптимізацію для мобільних пристроїв та впровадження нових технологій. Пріоритети мають бути чітко визначені та узгоджені з бізнес-цілями організації.

Планування заходів для реалізації стратегічних напрямків розвитку є важливою частиною процесу. Це включає розробку детального плану дій з конкретними термінами виконання та відповідальними особами. План має враховувати всі аспекти розвитку, включаючи технічні, фінансові та організаційні питання. Регулярний моніторинг виконання плану дозволяє своєчасно вносити корективи та забезпечувати ефективність процесу.

Цільова аудиторія

Аналіз та визначення цільової аудиторії є ключовими елементами плану розвитку веб-сайту. Це дозволяє більш ефективно планувати заходи, спрямовані на залучення та утримання користувачів.

Першим кроком є аналіз цільової аудиторії, який включає визначення її потреб, поведінки, демографічних характеристик та інтересів. Це можна зробити за допомогою опитувань, анкетування, аналізу статистики відвідувань та соціальних мереж. Зібрані дані допомагають створити портрет цільової аудиторії, що є основою для подальших дій.

Сегментація аудиторії дозволяє більш точно визначити потреби різних груп користувачів та розробити персоналізовані стратегії для кожної з них. Наприклад, можна виділити сегменти за віком, географічним розташуванням, інтересами та поведінковими факторами. Це допомагає створити більш релевантний контент та послуги, що задовольняють потреби кожного сегменту.

Врахування потреб цільової аудиторії при плануванні розвитку веб-сайту є ключовим аспектом. Це включає адаптацію контенту, функціональності та дизайну відповідно до очікувань користувачів. Наприклад, молодша аудиторія може вимагати інтерактивних елементів та активного використання соціальних мереж, тоді як старша аудиторія може цінувати простоту та зручність навігації.

Регулярний зворотний зв'язок від користувачів допомагає постійно вдосконалювати сайт та забезпечувати високу задоволеність аудиторії.

Ресурсне планування

Ресурсне планування є невід'ємною частиною плану розвитку веб-сайту. Воно включає визначення необхідних людських та фінансових ресурсів для реалізації запланованих заходів.

Оцінка ресурсів починається з визначення обсягу робіт та необхідних навичок для їх виконання. Це включає розрахунок кількості персоналу, необхідного для розробки, підтримки та розвитку сайту, а також визначення необхідного рівня кваліфікації. Наприклад, для впровадження нових функцій можуть знадобитися розробники з досвідом роботи з певними технологіями, а для створення контенту – копірайтери та редактори.

Пошук додаткових джерел фінансування є важливим елементом ресурсного планування. Це може включати залучення інвесторів, грантів, спонсорів або проведення краудфандингових кампаній. Важливо мати чіткий план фінансування, який враховує всі можливі джерела та обсяги фінансування, необхідні для реалізації плану розвитку.

Планування витрат включає розробку бюджету для реалізації запланованих заходів. Бюджет має враховувати всі витрати, пов'язані з розробкою, підтримкою та розвитком сайту, включаючи заробітну плату персоналу, витрати на обладнання та програмне забезпечення, маркетингові витрати та інші необхідні ресурси. Регулярний перегляд та коригування бюджету дозволяють забезпечити ефективне використання фінансових ресурсів та своєчасне реагування на зміни у потребах та умовах.

Оцінка ефективності

Регулярний аналіз досягнутих результатів дозволяє коригувати план розвитку веб-сайту та забезпечити його ефективність. Це включає визначення ключових метрик для оцінки ефективності реалізації плану, проведення регулярного аналізу та коригування плану на основі отриманих даних.

Метрики оцінки можуть включати показники відвідуваності сайту, конверсії, задоволеності користувачів, ефективності маркетингових кампаній та інші показники, які відображають досягнення поставлених цілей. Наприклад, зростання кількості відвідувачів та підвищення рівня конверсії можуть свідчити про успішність реалізації запланованих заходів.

Регулярний аналіз досягнутих результатів на основі визначених метрик дозволяє виявити сильні та слабкі сторони реалізованих заходів. Це може включати аналіз статистики відвідувань, зворотного зв'язку від користувачів, ефективності маркетингових кампаній та інших показників. Зібрані дані дозволяють зробити висновки про ефективність реалізованих заходів та визначити напрями для подальшого вдосконалення.

Коригування плану розвитку на основі результатів аналізу є важливим елементом забезпечення його ефективності. Це може включати внесення змін до стратегії, коригування пріоритетів, перерозподіл ресурсів та впровадження нових заходів для досягнення поставлених цілей. Регулярне коригування плану дозволяє забезпечити його відповідність актуальним потребам та умовам, що сприяє досягненню високих результатів у розвитку веб-сайту.

План розвитку веб-сайту є комплексним документом, який визначає напрями, завдання та ресурси для досягнення стратегічних цілей. Стратегічне планування, аналіз цільової аудиторії, ресурсне планування та оцінка ефективності є ключовими аспектами цього процесу. Впровадження комплексного підходу до планування та розвитку веб-сайту дозволяє забезпечити його стабільну роботу, високу якість контенту та послуг, а також задоволення потреб цільової аудиторії. Регулярний моніторинг та коригування плану дозволяють своєчасно реагувати на зміни та забезпечувати ефективність реалізації запланованих заходів.

ВИСНОВКИ

Під час розробки веб-сайту для громадської організації досягнуті важливі результати, що відповідають вимогам сучасних стандартів веб-розробки та потребам користувачів. Використання технологій React, Redux, Formik, MUI, Axios та React Router DOM дозволило створити інтерфейс, який є зручним, швидким і адаптивним до різних пристроїв.

Технологічні досягнення

Розробка веб-сайту була побудована на основі сучасних технологій, які забезпечують високу продуктивність і масштабованість. Використання React дозволило створити динамічні компоненти, які легко змінюються та оновлюються без перезавантаження сторінки. Redux забезпечив ефективне управління станом додатку, що значно спростило процес обробки даних і покращило користувацький досвід.

Formik спростив роботу з формами, надаючи інструменти для валідації даних та обробки помилок, що є важливим для забезпечення високої якості введених даних. MUI забезпечив зручність створення стильних і функціональних інтерфейсів, а також адаптивність додатку до різних екранів та пристроїв. Використання Axios дозволило ефективно здійснювати запити до серверу та обробляти відповіді, що підвищило швидкість і надійність роботи додатку. React Router DOM забезпечив зручність навігації по сайту, дозволяючи створювати багатосторінкові додатки з простим і зрозумілим маршрутизатором.

Дизайн та користувацький досвід

Одним із ключових аспектів розробки веб-сайту було забезпечення привабливого дизайну та зручності використання. Завдяки MUI ми змогли створити естетично привабливий інтерфейс, який відповідає сучасним тенденціям дизайну. Дизайн був ретельно продуманий, щоб забезпечити легкий доступ до основних функцій та інформації, зокрема для тих, хто вперше відвідує сайт.

Важливим елементом було також впровадження адаптивного дизайну, який дозволяє користувачам зручно використовувати сайт на різних пристроях, включаючи мобільні телефони та планшети. Це значно підвищує доступність сайту та задовольняє потреби ширшої аудиторії.

Підтримка та розвиток

Одним з основних завдань під час розробки сайту було забезпечення його подальшої підтримки та розвитку. Регулярний зворотний зв'язок від користувачів є критично важливим для визначення напрямків вдосконалення сайту. Ми впровадили механізми для збору та аналізу відгуків користувачів, що дозволяє швидко реагувати на їхні потреби та покращувати функціональність сайту.

Аналіз нових технологій та тенденцій у веб-розробці також є важливим елементом нашої стратегії розвитку. Ми постійно моніторим нові інструменти та технології, щоб забезпечити актуальність нашого сайту та підвищити його конкурентоспроможність.

Інтеграція з іншими системами та сервісами

Інтеграція з іншими системами та сервісами дозволяє значно розширити функціональність сайту та підвищити його зручність для користувачів. Впровадження кнопок для поширення контенту в соціальних мережах, інтеграція з API соціальних мереж для автоматичного публікування новин, а також інтеграція з платіжними системами для прийому пожертв є важливими кроками у цьому напрямку.

Забезпечення безпеки транзакцій та використання захисних засобів, таких як SSL сертифікати та шифрування даних, є ключовими аспектами для захисту даних користувачів та фінансових операцій.

Перспективи розвитку

У майбутньому ми плануємо впровадити нові функціональні можливості, які покращать користувацький досвід та задовільнять потреби наших користувачів. До таких функцій можуть належати інтеграція чат-ботів для автоматичного обслуговування користувачів, розширення можливостей

особистого кабінету користувача, впровадження функціоналу для проведення онлайн-заходів та вебінарів.

Планується також інтеграція з додатковими сервісами та платформами, включаючи нові платіжні системи та CRM-системи для кращого управління взаємодією з волонтерами та спонсорами. Оптимізація продуктивності та швидкості завантаження сайту, підвищення рівня безпеки та покращення аналітики та звітності також є важливими напрямками розвитку.

Впровадження веб-сайту для громадської організації стало важливим кроком у покращенні її діяльності та взаємодії з громадськістю. Завдяки використанню сучасних технологій, продуманому дизайну та постійній підтримці ми створили платформу, яка відповідає потребам користувачів та забезпечує ефективну роботу організації. У майбутньому ми плануємо продовжувати вдосконалення сайту, впроваджуючи нові функції та технології для задоволення зростаючих потреб наших користувачів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Червоний Хрест - благодійна організація [Електронний ресурс]. — Режим доступу: <https://redcross.org.ua>.
2. UNICEF - благодійна організація [Електронний ресурс]. — Режим доступу: <https://www.unicef.org/ukraine/>.
3. Wikipedia - Енциклопедія та визначення слова фреймворк [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/Програмний_каркас.
4. React.js - JavaScript бібліотека [Електронний ресурс]. — Режим доступу: <https://react.dev>.
5. Vue.js - JavaScript Framework для розробки інтерфейсів веб-сайтів [Електронний ресурс]. — Режим доступу: <https://vuejs.org>.
6. Angular - JavaScript Framework для розробки інтерфейсів веб-сайтів [Електронний ресурс]. — Режим доступу: <https://angular.dev>.
7. Redux.js - JavaScript бібліотека [Електронний ресурс]. — Режим доступу: <https://redux.js.org>.
8. MobX.js - JavaScript бібліотека [Електронний ресурс]. — Режим доступу: <https://mobx.js.org>.
9. Material UI - бібліотека React компонентів з відкритим кодом [Електронний ресурс]. — Режим доступу: <https://mui.com>.
10. Ant Design - бібліотека компонентів з відкритим кодом [Електронний ресурс]. — Режим доступу: <https://ant.design>.
11. NpmJS - Сайт для пошуку бібліотек для JavaScript [Електронний ресурс]. — Режим доступу: <https://www.npmjs.com/>.
12. React Router DOM – npm пакет для динамічної маршрутизації та приватних посилань у веб-додатку [Електронний ресурс]. — Режим доступу: <https://reactrouter.com/en/main>.

- 13.Axios - бібліотека для HTTP запитів [Електронний ресурс]. — Режим доступу: <https://axios-http.com>.
- 14.Node.js - середовище виконання JavaScript на сервері [Електронний ресурс]. — Режим доступу: <https://nodejs.org>.
- 15.Formik - бібліотека для створення форм [Електронний ресурс]. — Режим доступу: <https://formik.org>.
- 16.React-hook-form - бібліотека для створення форм [Електронний ресурс]. — Режим доступу: <https://www.react-hook-form.com>.
- 17.Figma - онлайн сервіс для розробки інтерфейсів [Електронний ресурс]. — Режим доступу: <https://www.figma.com>.
- 18.Google PageSpeed Insights - додаток для перевірки швидкості роботи веб-сайтів [Електронний ресурс]. — Режим доступу: <https://pagespeed.web.dev>.
- 19.GTmetrix - додаток для аналізу швидкості завантаження сайту [Електронний ресурс]. — Режим доступу: <https://gtmetrix.com>.
- 20.Apache JMeter - інструмент для проведення навантажувального тестування, розробляється Apache Software Foundation [Електронний ресурс]. — Режим доступу: <https://jmeter.apache.org>.
- 21.Webpack - модульний збирач JavaScript [Електронний ресурс]. — Режим доступу: <https://webpack.js.org>.
- 22.Babel - JavaScript компілятор [Електронний ресурс]. — Режим доступу: <https://babeljs.io>.

ДОДАТОК А

Приклад коду React компоненту

```
import React from 'react';
import { Box, ImageList, ImageListItem, Typography } from '@mui/material';
import { LinkButton } from '../Shared';

interface IProps {
  item: {
    label: string;
    description: string;
    photos: any[];
    link?: string;
    created_at: string;
  };
}

const styles = {
  container: {
    width: '100%',
    padding: '1rem',
    marginBottom: '2rem',
    backgroundColor: '#FFF',
    borderRadius: '3rem',
    boxShadow: 'rgba(0, 0, 0, 0.15) 0px 5px 5px, rgba(0, 0, 0, 0.05) 0px 5px 5px',
  },
  imageList: {
    display: 'flex',
    flexWrap: 'wrap',
    justifyContent: 'space-between',
    maxHeight: '300px',
    overflow: 'hidden'
  },
  imageListItem: {
    width: 'calc(24% - 8px)',
    height: '300px',
    borderRadius: '3rem',
    margin: '4px',
    display: 'flex',
    justifyContent: 'center',
    alignItems: 'center',
    overflow: 'hidden'
  },
  image: {
    width: '100%',
```

```

    height: '290px',
    borderRadius: '3rem',
  }
};

const NewsBlock: React.FC<IProps> = props => {
  const { item } = props;

  return (
    <Box sx={styles.container}>
      <Box sx={{ padding: '2rem' }}>
        <Box sx={{ display: 'flex', justifyContent: 'space-between' }}>
          <Typography sx={{ fontSize: '1.5rem', fontWeight: '600'
}}>{item.label}</Typography>
          <Typography sx={{ fontSize: '0.75rem', fontWeight: '600', color:
'grey' }}>
            {item.created_at}
          </Typography>
        </Box>
        <Typography>{item.description}</Typography>
        <ImageList sx={styles.imageList} variant='standard' cols={4}>
          {item.photos.map((photo, index) => (
            <ImageListItem key={index} sx={styles.imageListItem}>
              <img src={photo} alt={` ${index}`} style={styles.image} />
            </ImageListItem>
          ))}
        </ImageList>
        {item.link && <LinkButton label='Детальніше' link={item.link} />}
      </Box>
    </Box>
  );
};

export default NewsBlock;

```

Справа Громад

Головна >

Донати >

Новини та звіти >

Створення дронів

Раді повідомити, що завдяки вашій підтримці ми змогли зібрати та відправити 20 FPV-дронів для виконання важливих завдань на передовій. Ці дрони допоможуть нашим захисникам ефективніше виконувати розвідувальні операції, виявляти ворожі позиції та забезпечувати безпеку наших воїнів. Кожен дрон – це результат спільних зусиль і небайдужості наших волонтерів, донорів та усіх, хто долучився до нашої ініціативи. Завдяки вашій фінансовій підтримці та матеріалам, ми змогли оснастити дрони всім необхідним для виконання їхніх завдань. Дякуємо всім, хто допомагає нам у цій важливій справі. Разом ми наближаємо перемогу та робимо все можливе для захисту нашої країни. Ваш внесок – це велика допомога для наших захисників. Слава Україні!

ЗРОБИТИ ДОНАТ



Маскувальні сітки

Ми почали плести нові маскувальні сітки для наших захисників, і нам потрібна ваша допомога! Кожна маскувальна сітка – це маленький вклад в безпеку наших воїнів, і ви можете стати частиною цієї важливої справи. Нам потрібні матеріали для плетіння: старі речі кольору хаки, зеленого, сірого, коричневого та інших відтінків, які допоможуть створити надійний камуфляж. Якщо у вас є непотрібні футболки, сорочки, штани або інші тканини відповідних кольорів – приносьте їх до нас. Також будемо раді будь-якій фінансовій допомозі для придбання необхідних матеріалів. Разом ми зможемо зробити більше! Допоможемо нашим захисникам залишатися непомітними та в безпеці. Долучайтесь до нашої ініціативи та робіть свій внесок у спільну перемогу. Дякуємо за вашу підтримку!

ЗРОБИТИ ДОНАТ



Powered by
Olena Kucherenko
Volodymyr Gerasymchuk

Вигляд компоненту на сторінці

Створення дронів

Раді повідомити, що завдяки вашій підтримці ми змогли зібрати та відправити 20 FPV-дронів для виконання важливих завдань на передовій. Ці дрони допоможуть нашим захисникам ефективніше виконувати розвідувальні операції, виявляти ворожі позиції та забезпечувати безпеку наших воїнів. Кожен дрон – це результат спільних зусиль і небайдужості наших волонтерів, донорів та усіх, хто долучився до нашої ініціативи. Завдяки вашій фінансовій підтримці та матеріалам, ми змогли оснастити дрони всім необхідним для виконання їхніх завдань. Дякуємо всім, хто допомагає нам у цій важливій справі. Разом ми наближаємо перемогу та робимо все можливе для захисту нашої країни. Ваш внесок – це велика допомога для наших захисників. Слава Україні!

ЗРОБИТИ ДОНАТ



Вигляд компоненту

ДОДАТОК Б

Приклад коду Redux стану

```
import { PayloadAction, createSlice } from '@reduxjs/toolkit';
import { IVolunteer, InitialState } from '../types';

const initialState: InitialState = {
  volunteers: [],
  volunteersLoading: false,
};

export const volunteers = createSlice({
  name: 'volunteers',
  initialState,
  reducers: {
    startLoading: (state): void => {
      state.volunteersLoading = true;
    },
    saveVolunteersFailure: (state): void => {
      state.volunteersLoading = false;
    },
    saveVolunteersSuccess: (state, action: PayloadAction<IVolunteer[]>): void => {
      state.volunteers = action.payload;
      state.volunteersLoading = false;
    }
  }
});

export const { startLoading, saveVolunteersFailure, saveVolunteersSuccess } =
volunteers.actions;

export default volunteers.reducer;
```