

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ І ПРИКЛАДНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ДИРКАЧ ХРИСТИНА МИХАЙЛІВНА

Допускається до захисту:
в.о. завідувача кафедри
інформаційних технологій,
д-р. техн. наук, професор
_____Наталія ВЕСЕЛОВСЬКА
«___» _____ 2026 р.

**РОЗРОБКА ВЕБ-СИСТЕМИ ДЛЯ АГРЕГАЦІЇ
ТА АНАЛІЗУ ДАНИХ СОЦІАЛЬНИХ ПРОЄКТІВ**

Спеціальність 122 Комп'ютерні науки

Кваліфікаційна (бакалаврська) робота

Керівник:

Юрій АНТОНОВ, доцент кафедри
інформаційних технологій,
к. фіз.-мат. наук, доцент

Оцінка: _____ / _____ / _____
(бали/за шкалою ЄКТС/за національною шкалою)

Голова ЕК: _____
(підпис)

АНОТАЦІЯ

Диркач Х.М. Розробка вебсистеми для агрегації та аналізу даних соціальних проєктів. Спеціальність 122 «Комп'ютерні науки», Освітня програма «Комп'ютерні технології обробки даних». Донецький національний університет імені Василя Стуса, Вінниця, 2026.

У кваліфікаційній роботі представлено розробку вебсистеми для агрегації та аналізу даних соціальних проєктів у сфері допомоги тваринам. Досліджено сучасні підходи до розробки вебсистем, методи організації та зберігання даних, а також особливості реалізації клієнт-серверної архітектури. Розроблена вебсистема забезпечує перегляд, пошук і фільтрацію соціальних проєктів, управління організаціями та користувачами, збереження обраних ініціатив і відображення аналітичних показників.

Ключові слова: вебсистема, соціальні проєкти, агрегація даних, аналіз даних, React, Node.js, Express, PostgreSQL, Prisma ORM, REST API.

80 с., 15 рис., 2 табл., 55 джерел.

ABSTRACT

Dyrkach K.M. Development of a Web System for Aggregation and Analysis of Social Project Data. Specialty 122 «Computer Science», Educational Programme «Computer technologies for data processing». Vasyl Stus Donetsk National University, Vinnytsia, 2026.

The bachelor's thesis presents the development of a web system for aggregation and analysis of social project data in the field of animal welfare. Modern approaches to web system development, data organization and storage, and client-server architecture implementation are investigated. The developed system provides project browsing, search and filtering, user and organization management, functionality for saving selected projects, and analytical dashboards.

Keywords: web system, social projects, data aggregation, data analysis, React, Node.js, Express, PostgreSQL, Prisma ORM, REST API.

80 p., 15 figs., 2 tables, 55 sources.

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ АГРЕГАЦІЇ ТА АНАЛІЗУ ДАНИХ СОЦІАЛЬНИХ ПРОЄКТІВ.....	6
1.1. Соціальні проєкти як об'єкт інформаційного представлення.....	6
1.2. Особливості збору та агрегації даних соціальних проєктів	9
1.3. Методи аналізу та візуалізації даних у сучасних вебсистемах	13
1.4. Огляд існуючих платформ та інформаційних систем соціальної підтримки	15
1.4.1. GoFundMe	16
1.4.2. GlobalGiving.....	17
1.4.3. dobro.ua.....	18
1.4.4. Petfinder	19
1.4.5. Benevity Causes Portal	21
Порівняльний аналіз існуючих платформ.....	22
1.5. Аналіз технологій та архітектурних підходів до розробки вебсистем соціальної підтримки	23
1.5.1. Клієнт-серверна архітектура вебсистем	24
1.5.2. Сучасні технології розробки клієнтської частини вебсистем.....	25
1.5.3. Сучасні технології розробки серверної частини вебсистем.....	27
1.5.4. Технології зберігання та обробки даних	28
1.5.5. Використання REST API для взаємодії компонентів системи.....	30
Висновок до розділу 1	31
РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБСИСТЕМИ ДЛЯ АГРЕГАЦІЇ ТА АНАЛІЗУ ДАНИХ СОЦІАЛЬНИХ ПРОЄКТІВ	33
2.1. Постановка задачі та вимоги.....	33
2.2. Аналіз вимог до вебсистеми	34
2.2.1. Загальні вимоги	35
2.2.2. Функціональні вимоги.....	36
2.2.3. Нефункціональні вимоги.....	37

2.2.4. Вимоги до даних та інформаційних ресурсів	39
2.3. Проектування архітектури вебсистеми.....	40
2.4. Проектування структури бази даних	42
Висновок до розділу 2	45
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ВЕБСИСТЕМИ ДЛЯ АГРЕГАЦІЇ ТА АНАЛІЗУ ДАНИХ СОЦІАЛЬНИХ ПРОЄКТІВ	47
3.1. Вибір засобів та середовища розробки.....	47
3.2. Реалізація клієнтської частини вебсистеми.....	49
3.3. Реалізація серверної частини вебсистеми	57
3.4. Реалізація структури бази даних	59
3.5. Реалізація функціональних модулів системи.....	62
3.5.1. Модуль роботи з соціальними проєктами.....	62
3.5.2. Модуль управління організаціями	63
3.5.3. Модуль автентифікації та профілю користувача	64
3.5.4. Адміністративний модуль.....	65
3.5.5. Аналітичний модуль.....	66
3.6. Тестування та результати роботи системи.....	67
3.7. Перспективи розвитку вебсистеми	69
Висновок до розділу 3	70
ЗАГАЛЬНІ ВИСНОВКИ	72
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	74

ВСТУП

Актуальність роботи:

У сучасних умовах розвитку цифрових технологій соціальні проєкти дедалі активніше використовують вебплатформи для поширення інформації про власну діяльність, залучення волонтерів та пошуку підтримки. Це особливо актуально для ініціатив у сфері захисту тварин, діяльність яких передбачає обробку значного обсягу інформації про організації, потреби, благодійні збори та соціальні проєкти.

Водночас інформація про такі ініціативи часто розміщується на різних ресурсах і не має єдиної структури представлення. Це ускладнює пошук необхідних відомостей, аналіз поточного стану проєктів та взаємодію користувачів із соціальними організаціями.

Одним із перспективних напрямів вирішення зазначеної проблеми є створення вебсистем, що забезпечують централізоване зберігання, структурування та аналіз даних соціальних проєктів. Такі системи дозволяють підвищити доступність інформації, спростити її обробку та забезпечити отримання узагальнених аналітичних показників.

У зв'язку з цим актуальним є завдання розробки вебсистеми для агрегації та аналізу даних соціальних проєктів, орієнтованої на підтримку ініціатив у сфері захисту тварин.

Мета дослідження: Розробка вебсистеми для агрегації та аналізу даних соціальних проєктів у сфері захисту тварин, яка забезпечує централізоване зберігання, пошук, фільтрацію та аналіз інформації про соціальні ініціативи.

Завдання дослідження:

- Проаналізувати сучасні підходи до організації інформаційних систем для підтримки соціальних проєктів;
- Дослідити існуючі платформи та вебсистеми соціальної підтримки;
- Визначити функціональні та нефункціональні вимоги до вебсистеми;
- Спроектувати архітектуру вебсистеми та структуру бази даних;
- Реалізувати клієнтську та серверну частини системи;

- Реалізувати механізми управління користувачами, організаціями та соціальними проєктами;
- Реалізувати функції пошуку, фільтрації та взаємодії користувачів із соціальними проєктами;
- Реалізувати інструменти аналізу та візуалізації даних соціальних проєктів;
- Провести тестування вебсистеми та проаналізувати результати її роботи.

Об'єкт дослідження: Процес агрегації, зберігання, обробки та аналізу даних соціальних проєктів у веборієнтованих інформаційних системах.

Предмет дослідження: Методи, інструменти та технології проєктування і реалізації вебсистем для агрегації та аналізу даних соціальних проєктів.

Методи дослідження: У роботі використовуються методи аналізу предметної області, методи проєктування інформаційних систем, принципи нормалізації реляційних баз даних, а також методи моделювання архітектури програмного забезпечення.

Практичне значення дослідження: Практичне значення роботи полягає в розробці вебсистеми, яка забезпечує зручне управління даними соціальних проєктів, підтримує пошук і фільтрацію інформації, надає адміністративні інструменти та дозволяє отримувати узагальнені аналітичні показники щодо діяльності соціальних ініціатив. Розроблене рішення може бути використане громадськими організаціями, благодійними фондами, волонтерськими об'єднаннями та притулками для тварин як інструмент представлення, моніторингу та аналізу власної діяльності.

Апробація результатів дослідження: Результати кваліфікаційної роботи самостійно отримані автором і в пресі не публікувалися.

Структура кваліфікаційної роботи: Бакалаврська робота складається із вступу, трьох розділів та висновків та списку використаних джерел.

Робота містить 80 сторінок, 15 рисунків, 2 таблиці та список літератури з 55 джерел.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ АГРЕГАЦІЇ ТА АНАЛІЗУ ДАНИХ СОЦІАЛЬНИХ ПРОЄКТІВ

1.1. Соціальні проєкти як об'єкт інформаційного представлення

У сучасному суспільстві соціальні проєкти виступають важливим інструментом вирішення суспільно значущих проблем та реалізації громадських ініціатив [1]. Їхня діяльність охоплює різні сфери, зокрема освіту, охорону здоров'я, екологію, підтримку вразливих категорій населення, благодійність та захист тварин. Ефективність таких проєктів значною мірою залежить від доступності інформації про їхню діяльність, потреби та результати роботи.

Соціальний проєкт можна визначити як комплекс взаємопов'язаних заходів, спрямованих на досягнення конкретної суспільно корисної мети протягом визначеного проміжку часу та за наявності певних ресурсних обмежень [2]. На відміну від комерційних проєктів, основною метою соціальних ініціатив є створення суспільної цінності та вирішення актуальних соціальних проблем [3].

Для соціальних проєктів характерними є такі особливості:

- Орієнтація на суспільну користь та соціальний ефект;
- Взаємодія з різними групами зацікавлених сторін;
- Залучення волонтерських, благодійних та громадських ресурсів;
- Потреба у відкритому інформуванні громадськості про результати діяльності;
- Залежність від своєчасного поширення актуальної інформації.

У сфері допомоги тваринам соціальні проєкти можуть реалізовуватися у формі діяльності притулків, програм адопції, благодійних зборів, волонтерських кампаній, заходів із лікування та реабілітації тварин, інформаційно-просвітницьких ініціатив та інших форм громадської активності [4]. Такі проєкти потребують постійної взаємодії з потенційними волонтерами, благодійниками та представниками громадськості, що зумовлює необхідність ефективного управління інформацією.

Соціальний проєкт проходить кілька основних етапів життєвого циклу, серед яких можна виділити ініціацію, планування, реалізацію, моніторинг та завершення [5]. На кожному з цих етапів формується та накопичується значний обсяг даних, які потребують структурованого зберігання та подальшого аналізу. Основні етапи життєвого циклу соціального проєкту представлено на рис. 1.1.

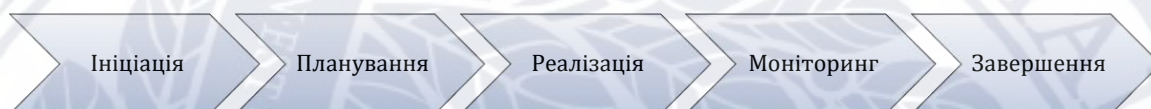


Рисунок 1.1 – Основні етапи життєвого циклу соціального проєкту

Важливою особливістю соціальних проєктів є необхідність постійної комунікації із зацікавленими сторонами. До них належать волонтери, благодійники, представники громадських організацій, органи місцевого самоврядування та інші учасники соціальної діяльності. Для забезпечення ефективної взаємодії між цими групами необхідним є своєчасне поширення актуальної інформації про діяльність проєкту, його потреби, результати та можливості долучення до реалізації ініціативи.

У процесі розвитку цифрових технологій значна частина соціальної активності перемістилася до інформаційного простору [6]. Сьогодні соціальні проєкти активно використовують вебсайти, соціальні мережі, платформи благодійних зборів та інші цифрові канали комунікації для поширення інформації та залучення підтримки [7]. Завдяки цьому суттєво розширюється аудиторія проєктів і підвищується доступність інформації для потенційних учасників.

Разом із перевагами цифровізації виникають і нові виклики. Зростання кількості інформаційних ресурсів призводить до накопичення значних обсягів

даних, які відрізняються структурою, рівнем деталізації та актуальністю. Частина інформації може дублюватися, втрачати актуальність або бути представлена у різних форматах. За таких умов виникає необхідність використання спеціалізованих інформаційних систем, що забезпечують впорядкування даних та підтримують їх подальшу обробку.

Для ефективного представлення соціальних проєктів інформація повинна бути структурованою, зрозумілою та доступною для різних категорій користувачів. Важливими характеристиками інформаційного представлення є повнота даних, достовірність, актуальність та можливість швидкого пошуку необхідних відомостей. Саме ці характеристики визначають якість інформаційної підтримки соціальних ініціатив та впливають на ефективність взаємодії між організаціями й користувачами.

Крім представлення інформації, важливу роль відіграє можливість її аналізу [8]. Узагальнення даних щодо категорій проєктів, регіонів реалізації, потреб організацій та активності користувачів дозволяє отримувати додаткову інформацію про стан соціального сектору та сприяє прийняттю більш обґрунтованих управлінських рішень. Саме тому сучасні вебсистеми все частіше поєднують функції зберігання інформації із засобами її аналітичного опрацювання.

Основні види інформації, що супроводжують соціальні проєкти, включають відомості про організацію-виконавця, опис проєкту, категорію діяльності, поточний статус реалізації, потреби проєкту, контактні дані, місце реалізації та результати діяльності. Крім того, важливими є дані про залучені ресурси, активність користувачів та взаємодію із зацікавленими сторонами.

Особливістю сучасних соціальних проєктів є значна кількість інформації, розміщеної на різних інформаційних ресурсах [9]. Дані можуть бути опубліковані на офіційних вебсайтах організацій, сторінках у соціальних мережах, платформах благодійних зборів та інших цифрових сервісах. Така фрагментація інформаційного простору створює низку проблем, пов'язаних із пошуком, оновленням та аналізом даних.

Відсутність єдиного середовища представлення інформації ускладнює отримання актуальних відомостей про діяльність соціальних ініціатив, знижує ефективність пошуку проєктів та обмежує можливості проведення комплексного аналізу. Як наслідок, виникає потреба у створенні спеціалізованих інформаційних систем, здатних забезпечити централізоване накопичення, структурування та аналіз даних соціальних проєктів.

Отже, соціальні проєкти можна розглядати як складний об'єкт інформаційного представлення, для ефективної роботи з яким необхідне використання сучасних вебтехнологій, засобів управління даними та інструментів аналітичної обробки інформації.

Аналіз особливостей соціальних проєктів показує, що ефективність їх діяльності значною мірою залежить від доступності, повноти та актуальності інформації. Водночас дані про соціальні ініціативи часто розміщуються у різних інформаційних джерелах, мають неоднорідну структуру та різні рівні деталізації. За таких умов виникає потреба у використанні спеціалізованих інформаційних систем, здатних забезпечити збір, структурування та подальшу обробку інформації [10]. Тому важливим завданням є дослідження особливостей збору та агрегації даних соціальних проєктів, що розглядаються у наступному підрозділі.

1.2. Особливості збору та агрегації даних соціальних проєктів

Ефективне функціонування соціальних проєктів значною мірою залежить від якості інформації, яка використовується для підтримки їх діяльності. Для прийняття обґрунтованих рішень, планування заходів, залучення ресурсів та взаємодії з користувачами необхідним є отримання актуальних і достовірних даних про соціальні ініціативи, організації та їх потреби.

Першим етапом роботи з інформацією є збір даних. Під збором даних розуміють процес отримання відомостей із різних джерел з метою їх подальшого зберігання, обробки та використання. У сфері соціальних проєктів джерелами даних можуть виступати вебсайти організацій, соціальні мережі, благодійні

платформи, інформаційні портали, відкриті набори даних та внутрішні інформаційні системи [11].

Після збору інформації важливим етапом є її агрегація. Агрегація даних являє собою процес об'єднання інформації з різних джерел у єдину структуру, придатну для подальшого аналізу та використання [12]. Основною метою агрегації є формування цілісного інформаційного середовища, яке забезпечує швидкий доступ до необхідних відомостей та спрощує їх обробку.

Для соціальних проєктів особливу цінність становлять дані про організації, напрями діяльності, категорії проєктів, місце реалізації, контактну інформацію, актуальні потреби, статуси виконання та результати діяльності. Сукупність таких даних дозволяє формувати повне уявлення про стан окремих ініціатив та здійснювати їх подальший аналіз.

Однією з особливостей соціальних проєктів є велика кількість джерел, з яких формується інформація про їхню діяльність. Дані можуть надходити з офіційних вебсайтів організацій, сторінок у соціальних мережах, платформ для збору благодійних внесків, інформаційних порталів, відкритих державних реєстрів та внутрішніх інформаційних систем. Кожне з цих джерел містить власний набір відомостей, які відрізняються структурою, форматом представлення та рівнем деталізації. Основні джерела даних соціальних проєктів наведено на рис. 1.2.

Як видно з рис. 1.2, інформація про соціальні проєкти формується з багатьох незалежних джерел. Це створює додаткові ускладнення під час її збору та обробки, оскільки відомості можуть дублюватися, суперечити одна одній або оновлюватися з різною швидкістю. Основні проблеми, які виникають під час збору та агрегації даних соціальних проєктів, наведено на рис. 1.3.

Як показано на рис. 1.3, найбільший вплив на якість інформації мають розподіленість джерел даних, дублювання відомостей, різноманітність форматів представлення інформації, а також проблема підтримки її актуальності [13]. Сукупність цих факторів ускладнює формування єдиного інформаційного середовища та знижує ефективність подальшого аналізу даних.

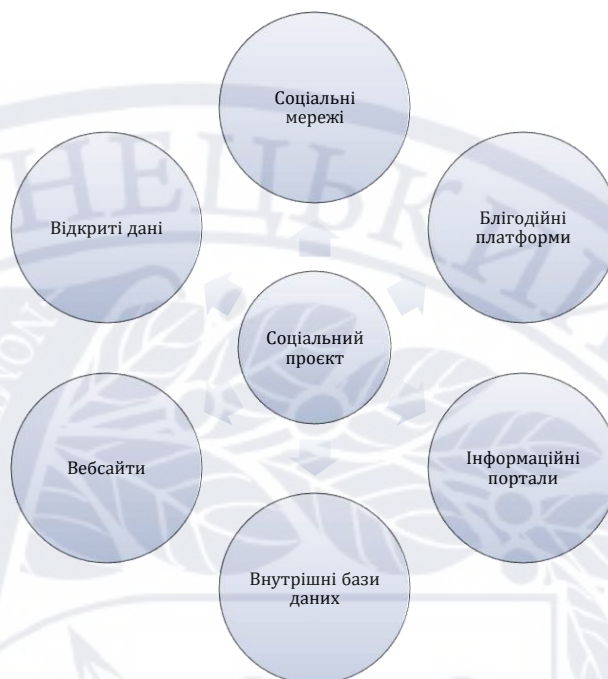


Рисунок 1.2 – Основні джерела даних соціальних проєктів



Рисунок 1.3 – Основні проблеми збору та агрегації даних соціальних проєктів

Однією з найбільш поширених проблем є розподіленість інформації. Відомості про один і той самий соціальний проєкт можуть одночасно

публікуватися на вебсайті організації, у соціальних мережах та на спеціалізованих благодійних платформах. У результаті користувачеві доводиться здійснювати пошук інформації в кількох джерелах, що ускладнює отримання повної картини щодо діяльності проєкту.

Іншою важливою проблемою є дублювання даних. Однакові відомості можуть багаторазово повторюватися на різних ресурсах або публікуватися різними організаціями. Це створює ризик виникнення надлишкової інформації та ускладнює визначення актуальної версії даних [14].

Суттєвим фактором також є проблема актуальності інформації. Соціальні проєкти характеризуються високою динамічністю: змінюються потреби організацій, статуси зборів, контактні дані та інші параметри діяльності. Інформація, яка була актуальною на момент публікації, з часом може втратити свою цінність для користувачів. Тому інформаційні системи повинні забезпечувати можливість оперативного оновлення даних.

Додаткові складності виникають через різноманітність форматів представлення інформації. Дані можуть бути представлені у вигляді текстових описів, таблиць, документів, графічних матеріалів або публікацій у соціальних мережах. Для подальшого використання такої інформації необхідне її попереднє структурування та приведення до єдиного формату.

Якість даних безпосередньо впливає на результати подальшого аналізу. Наявність пропущених значень, застарілих записів або суперечливих відомостей може призводити до викривлення результатів обробки інформації та ускладнювати прийняття управлінських рішень [15].

У сучасних інформаційних системах для вирішення зазначених проблем використовуються підходи агрегації, консолідації та інтеграції даних. Агрегація передбачає об'єднання відомостей із різних джерел у межах єдиного інформаційного середовища. Консолідація спрямована на накопичення інформації в централізованому сховищі даних, тоді як інтеграція забезпечує узгодження структури даних та встановлення взаємозв'язків між окремими

інформаційними об'єктами. Сукупне використання цих підходів дозволяє підвищити повноту, достовірність та доступність інформації [16].

Важливою перевагою агрегації даних є можливість формування єдиного інформаційного простору для користувачів системи. Завдяки цьому відомості про соціальні проєкти стають більш доступними для пошуку, порівняння та подальшого аналізу. Крім того, централізоване представлення даних створює передумови для формування статистичних показників та виявлення закономірностей у діяльності соціальних ініціатив.

Тому, збір та агрегація даних є важливими складовими функціонування сучасних вебсистем підтримки соціальних проєктів. Вони забезпечують формування єдиного інформаційного середовища, необхідного для ефективної роботи користувачів та подальшого аналітичного опрацювання інформації. Проте для отримання практично корисних результатів недостатньо лише накопичувати дані. Не менш важливим є їх аналіз та візуалізація, що розглядаються у наступному підрозділі.

1.3. Методи аналізу та візуалізації даних у сучасних вебсистемах

Після накопичення та агрегації даних важливим етапом роботи інформаційної системи є їх подальший аналіз [17]. Саме аналіз дозволяє перетворити сукупність розрізнених відомостей на корисну інформацію, яка може бути використана для оцінки поточного стану об'єкта дослідження, виявлення закономірностей та підтримки процесу прийняття рішень.

У сучасних вебсистемах аналіз даних є одним із ключових інструментів підвищення ефективності роботи користувачів із великими обсягами інформації. Використання аналітичних методів дозволяє не лише накопичувати дані, а й отримувати на їх основі нові знання, необхідні для планування діяльності, контролю виконання завдань та оцінювання результативності окремих процесів.

Залежно від поставлених цілей можуть застосовуватися різні підходи до аналізу даних. Найбільш поширеним є описовий аналіз, який спрямований на узагальнення наявної інформації та формування статистичних показників. Такий

підхід дозволяє визначати кількість об'єктів певної категорії, оцінювати їх розподіл за окремими характеристиками та отримувати загальне уявлення про досліджувану предметну область [18].

У сфері соціальних проєктів описовий аналіз може використовуватися для визначення кількості проєктів за категоріями, аналізу їх територіального розподілу, оцінки активності організацій, а також виявлення найбільш поширених потреб та напрямів діяльності. Отримані результати можуть використовуватися як користувачами системи, так і представниками організацій для прийняття управлінських рішень.

Крім описового аналізу, сучасні інформаційні системи підтримують діагностичний аналіз, метою якого є пошук причин виникнення певних явищ або змін показників. Використання такого підходу дозволяє досліджувати взаємозв'язки між різними характеристиками даних та виявляти фактори, що впливають на діяльність організацій або окремих соціальних ініціатив.

Важливу роль у роботі вебсистем відіграє візуалізація даних. Навіть якісно підготовлена аналітична інформація може бути складною для сприйняття, якщо вона представлена виключно у вигляді таблиць або текстових описів. Візуалізація дозволяє подати результати аналізу у більш зрозумілій та наочній формі, що сприяє швидшому сприйняттю інформації та полегшує її інтерпретацію [19].

У сучасних вебсистемах для візуалізації даних широко використовуються графіки, діаграми, індикатори, інформаційні панелі та інтерактивні елементи інтерфейсу. Вибір конкретного способу представлення залежить від характеру даних та цілей аналізу. Наприклад, для порівняння кількісних показників часто застосовуються стовпчикові діаграми, для відображення структури даних – кругові діаграми, а для аналізу змін у часі – лінійні графіки [20].

Особливого поширення набули аналітичні панелі (dashboard), які поєднують декілька інструментів візуалізації в межах одного інформаційного простору. Dashboard забезпечує швидкий доступ до ключових показників

системи та дозволяє користувачеві отримувати узагальнену інформацію без необхідності детального перегляду великих масивів даних [21].

Для вебсистем, що працюють із соціальними проєктами, використання dashboard-панелей є особливо актуальним. Вони дозволяють відображати загальну кількість проєктів, статистику за категоріями, дані щодо активності організацій, інформацію про поточні потреби та інші показники, які характеризують стан соціальних ініціатив. Завдяки цьому користувач отримує можливість швидко оцінити ситуацію та перейти до детальнішого аналізу окремих даних.

Важливою тенденцією розвитку сучасних вебсистем є інтерактивність візуалізацій. Користувачі можуть змінювати параметри відображення інформації, застосовувати фільтри, виконувати сортування та деталізацію даних без необхідності повторного завантаження сторінки [22]. Такий підхід підвищує зручність роботи із системою та сприяє більш ефективному використанню аналітичних можливостей.

Аналіз та візуалізація даних є важливими складовими сучасних вебсистем. Вони забезпечують перетворення накопичених даних на зрозумілу та корисну інформацію, підтримують процес прийняття рішень і сприяють підвищенню ефективності роботи користувачів із великими обсягами даних. Для реалізації зазначених можливостей використовуються різноманітні програмні рішення та вебплатформи, огляд яких наведено у наступному підрозділі.

1.4. Огляд існуючих платформ та інформаційних систем соціальної підтримки

Цифровізація суспільних процесів сприяла активному розвитку інформаційних систем, призначених для підтримки соціальних ініціатив, благодійних організацій та волонтерських проєктів [23]. Сучасні вебплатформи надають користувачам можливість отримувати інформацію про соціальні проєкти, здійснювати фінансову підтримку ініціатив, знаходити необхідні ресурси та взаємодіяти з організаціями через єдиний інформаційний простір.

Розвиток вебтехнологій дозволив значно спростити доступ до інформації про благодійні та соціальні проекти. Якщо раніше пошук відомостей про діяльність організацій здійснювався переважно через окремі вебсайти або соціальні мережі, то сьогодні значна частина інформації концентрується на спеціалізованих платформах, які забезпечують централізоване представлення даних, пошук проектів, збір пожертв та підтримку взаємодії між учасниками соціальних ініціатив.

Існуючі платформи суттєво відрізняються за функціональними можливостями, цільовою аудиторією та напрямками діяльності. Одні системи орієнтовані переважно на організацію благодійних зборів, інші забезпечують представлення діяльності некомерційних організацій або підтримують процеси адопції тварин. Разом із тим більшість таких рішень спрямовані на вирішення окремих завдань і не завжди забезпечують комплексний підхід до агрегації, аналізу та візуалізації даних соціальних проектів.

Для визначення функціональних можливостей сучасних інформаційних систем та виявлення напрямів їх подальшого розвитку доцільно провести аналіз найбільш поширених платформ соціальної підтримки. Результати такого аналізу дозволяють визначити сильні та слабкі сторони існуючих рішень, а також обґрунтувати доцільність розробки власної вебсистеми для агрегації та аналізу даних соціальних проектів.

1.4.1. GoFundMe

GoFundMe є однією з найбільших міжнародних платформ для організації благодійних зборів та краудфандингових кампаній. Сервіс був створений у 2010 році та сьогодні використовується мільйонами користувачів у різних країнах світу для збору коштів на соціальні, медичні, освітні та благодійні потреби [24].

Основною функцією платформи є створення кампаній зі збору коштів, які можуть бути підтримані іншими користувачами шляхом здійснення фінансових внесків. Для кожного проекту формується окрема сторінка з описом ініціативи,

ціллю збору, інформацією про автора кампанії та поточними результатами фінансування.

Платформа надає користувачам можливість здійснювати пошук проєктів за категоріями, переглядати популярні кампанії та отримувати інформацію про хід збору коштів у режимі реального часу. Важливою перевагою GoFundMe є простота використання та висока доступність для широкого кола користувачів.

Разом із тим функціональність платформи орієнтована переважно на фінансову підтримку ініціатив. Інструменти аналізу даних представлені обмежено, а основна увага приділяється процесу збору коштів та взаємодії між авторами кампаній і благодійниками. Крім того, система не забезпечує централізованого управління соціальними проєктами та не підтримує розвинений аналітичний функціонал для оцінювання діяльності організацій.

GoFundMe є ефективним інструментом для організації благодійних зборів та залучення фінансової підтримки. Проте платформа не призначена для комплексної агрегації, аналізу та візуалізації даних соціальних проєктів, що обмежує можливості її використання як повноцінної інформаційної системи соціальної підтримки.

1.4.2. GlobalGiving

GlobalGiving є міжнародною благодійною платформою, яка об'єднує некомерційні організації, соціальні проєкти та благодійників з різних країн світу [25]. Основною метою системи є підтримка соціальних ініціатив шляхом надання інструментів для пошуку проєктів, залучення фінансування та представлення результатів діяльності організацій.

Платформа забезпечує доступ до великої кількості соціальних проєктів, що охоплюють різні сфери суспільної діяльності, зокрема освіту, охорону здоров'я, захист довкілля, гуманітарну допомогу та підтримку тварин. Для кожної ініціативи формується окрема сторінка, яка містить опис проєкту, інформацію про організацію, поточний стан реалізації та можливість здійснення благодійного внеску.

Однією з переваг GlobalGiving є високий рівень структурованості інформації. Користувачі можуть виконувати пошук проєктів за категоріями, країнами та напрямками діяльності, що спрощує навігацію серед великої кількості соціальних ініціатив. Крім того, платформа забезпечує прозорість діяльності організацій завдяки публікації звітів, оновлень та результатів реалізації проєктів.

Важливою особливістю системи є підтримка довгострокової взаємодії між організаціями та благодійниками. Користувачі можуть відстежувати розвиток проєктів, отримувати інформацію про їх результати та брати участь у підтримці ініціатив протягом тривалого часу.

Разом із тим функціональність GlobalGiving орієнтована насамперед на представлення соціальних проєктів та залучення фінансування. Можливості аналітичної обробки інформації є обмеженими, а основна увага приділяється інформаційній підтримці проєктів та взаємодії з донорами. Платформа також не забезпечує комплексних інструментів для аналізу накопичених даних або формування розширених аналітичних звітів.

GlobalGiving є розвиненою платформою для представлення соціальних ініціатив та організації благодійної підтримки на міжнародному рівні. Водночас система зосереджена переважно на фінансуванні та інформаційному супроводі проєктів, тоді як функції агрегації та аналізу даних реалізовані обмежено.

1.4.3. dobro.ua

dobro.ua є однією з найбільших українських платформ благодійності, що забезпечує взаємодію між благодійними організаціями, соціальними проєктами та громадянами, які бажають надати допомогу [26]. Платформа функціонує як централізований інформаційний ресурс, де користувачі можуть ознайомитися з актуальними соціальними ініціативами та підтримати їх шляхом здійснення благодійних внесків.

На платформі представлено проєкти різних напрямів діяльності, серед яких допомога дітям, підтримка медичних закладів, гуманітарні ініціативи, соціальна допомога населенню та проєкти, пов'язані із захистом тварин. Для кожної

ініціативи створюється окрема сторінка з детальним описом проєкту, інформацією про організацію, метою збору коштів та поточними результатами кампанії.

Однією з переваг платформи є високий рівень прозорості. Користувачі мають можливість переглядати інформацію про використання зібраних коштів, звітність організацій та результати реалізованих проєктів. Такий підхід сприяє підвищенню довіри до благодійних ініціатив та забезпечує відкритість діяльності організацій.

Система підтримує пошук та тематичне групування проєктів, що дозволяє користувачам швидко знаходити ініціативи відповідно до власних інтересів. Крім того, платформа забезпечує зручний механізм здійснення благодійних внесків та підтримує регулярну комунікацію між організаціями та благодійниками.

Разом із тим функціональність dobro.ua орієнтована переважно на організацію благодійної підтримки та збір коштів. Незважаючи на наявність структурованого каталогу проєктів, можливості аналітичної обробки інформації залишаються обмеженими. Користувачі отримують доступ до інформації про окремі проєкти, проте інструменти аналізу даних, формування статистичних показників та візуалізації інформації представлені в обмеженому обсязі.

Платформа dobro.ua є успішним прикладом української інформаційної системи соціальної підтримки, яка забезпечує ефективну взаємодію між організаціями та благодійниками. Водночас її основна увага зосереджена на підтримці благодійних кампаній, тоді як функції агрегації та аналітичної обробки даних реалізовані частково.

1.4.4. Petfinder

Petfinder є однією з найбільших міжнародних онлайн-платформ, призначених для пошуку домашніх тварин та підтримки процесу адопції [27]. Система об'єднує притулки, рятувальні організації та потенційних власників

тварин, забезпечуючи централізований доступ до інформації про тварин, які потребують нового дому.

Основною функцією платформи є формування каталогу тварин із можливістю пошуку та фільтрації за різними параметрами. Користувачі можуть здійснювати пошук за видом тварини, породою, віком, місцем перебування та іншими характеристиками. Для кожної тварини створюється окрема інформаційна сторінка, яка містить фотографії, опис, інформацію про стан здоров'я та контактні дані організації, що відповідає за процес адопції.

Однією з ключових переваг Petfinder є зручна організація інформації та високий рівень деталізації даних. Платформа забезпечує швидкий доступ до актуальної інформації про тварин та дозволяє ефективно взаємодіяти між притулками й потенційними власниками. Наявність розвинених механізмів пошуку та фільтрації значно спрощує процес вибору тварини та підвищує ефективність роботи системи.

Важливою особливістю платформи є централізоване представлення інформації від великої кількості організацій. Завдяки цьому користувачі можуть отримувати доступ до даних із різних притулків через єдиний інтерфейс, що фактично реалізує один із принципів агрегації інформації у сфері допомоги тваринам.

Разом із тим функціональність Petfinder зосереджена переважно на процесі адопції тварин. Платформа не підтримує комплексне управління соціальними проєктами, не надає розвинених інструментів аналізу накопичених даних та не орієнтована на формування аналітичних показників діяльності організацій. Крім того, система не охоплює широкий спектр соціальних ініціатив поза сферою адопції та утримання тварин.

Petfinder є успішним прикладом спеціалізованої інформаційної системи у сфері допомоги тваринам, яка забезпечує ефективну агрегацію інформації про тварин та підтримує процес їх адопції. Водночас платформа орієнтована на вирішення вузького кола завдань і не забезпечує комплексного аналізу соціальних проєктів та діяльності організацій.

1.4.5. Benevity Causes Portal

Benevity Causes Portal є спеціалізованою платформою для взаємодії між неприбутковими організаціями, благодійними фондами та корпоративними донорами [28]. Система призначена для цифрового представлення діяльності організацій, спрощення процесу залучення фінансування та підвищення прозорості благодійної діяльності.

Платформа надає організаціям можливість створювати власні профілі, публікувати інформацію про діяльність, представляти соціальні ініціативи та взаємодіяти з потенційними донорами. Користувачі можуть здійснювати пошук організацій, ознайомлюватися з напрямками їх діяльності та отримувати інформацію про реалізовані проекти.

Однією з ключових переваг Benevity є орієнтація на централізоване управління інформацією про неприбуткові організації. Система забезпечує структуроване представлення даних, підтримує перевірку організацій та сприяє формуванню єдиного інформаційного середовища для благодійної діяльності. Такий підхід дозволяє підвищити достовірність інформації та спрощує процес пошуку партнерів для реалізації соціальних ініціатив.

Важливою особливістю платформи є підтримка взаємодії між організаціями та корпоративним сектором. Завдяки цьому Benevity використовується не лише як каталог благодійних організацій, а й як інструмент координації соціальної відповідальності бізнесу та підтримки благодійних проектів.

Разом із тим основна увага системи зосереджена на представленні організацій та забезпеченні процесів благодійної підтримки. Можливості аналізу накопичених даних та візуалізації статистичних показників реалізовані обмежено. Крім того, платформа не орієнтована на підтримку спеціалізованих напрямів діяльності, таких як допомога тваринам, а також не забезпечує комплексної аналітичної обробки інформації про соціальні проекти.

Benevity Causes Portal є сучасною інформаційною системою для підтримки діяльності неприбуткових організацій та взаємодії з донорами. Водночас

функціональність платформи зосереджена переважно на управлінні інформацією про організації та залученні фінансування, тоді як можливості агрегації та аналізу даних соціальних проєктів залишаються обмеженими.

Порівняльний аналіз існуючих платформ

Для наочного порівняння функціональних можливостей розглянутих платформ сформовано порівняльну таблицю 1.1 [29].

Таблиця 1.1 – Порівняння функціональних можливостей платформ соціальної підтримки

Функціональна можливість	GoFund Me	GlobalGiving	dobro.ua	Petfinder	Benevity	SocialData Hub
Каталог проєктів	+	+	+	+	-	+
Пошук інформації	+	+	+	+	+	+
Фільтрація даних	±	+	+	+	±	+
Профілі організацій	-	+	+	+	+	+
Підтримка благодійних зборів	+	+	+	-	+	-
Підтримка проєктів допомоги тваринам	-	±	±	+	-	+
Агрегація даних із різних організацій	-	±	±	+	+	+
Збереження обраних проєктів	-	-	-	-	-	+
Історія взаємодії користувача	-	-	-	-	-	+
Адміністрування проєктів	-	±	±	-	+	+
Адміністрування організацій	-	±	±	-	+	+
Аналітичний модуль	-	-	-	-	-	+
Візуалізація статистичних показників	-	-	-	-	-	+

Умовні позначення:

- «+» – функція реалізована;
- «±» – функція реалізована частково;
- «-» – функція відсутня.

Як видно з таблиці 1.1, більшість розглянутих платформ орієнтовані на вирішення окремих завдань соціальної підтримки. GoFundMe, GlobalGiving та dobro.ua зосереджені переважно на благодійних зборах та представленні соціальних ініціатив. Petfinder спеціалізується на підтримці процесу адопції тварин, тоді як Benevity орієнтована на взаємодію між неприбутковими організаціями та донорами.

Водночас жодна з розглянутих систем не поєднує можливості агрегації даних, централізованого управління соціальними проектами, персоналізованої взаємодії користувачів та аналітичної обробки інформації в межах єдиної вебплатформи. Саме усунення зазначених обмежень стало одним із ключових чинників розробки вебсистеми SocialData Hub.

1.5. Аналіз технологій та архітектурних підходів до розробки вебсистем соціальної підтримки

Сучасні вебсистеми є складними програмними комплексами, які забезпечують збір, зберігання, обробку та представлення інформації користувачам через мережу Інтернет. Для забезпечення ефективної роботи таких систем використовуються різноманітні архітектурні підходи, технології розробки програмного забезпечення та системи керування базами даних.

Вибір архітектури та технологічного стеку є одним із ключових етапів створення інформаційної системи, оскільки саме від цих рішень залежать продуктивність, масштабованість, надійність та зручність подальшого супроводу програмного продукту. Крім того, правильно обрані технології дозволяють спростити процес розробки, забезпечити ефективну взаємодію між компонентами системи та створити якісний користувацький інтерфейс.

У більшості сучасних вебзастосунків використовується клієнт-серверний підхід, який передбачає розподіл функціональності між окремими компонентами системи. Такий підхід дозволяє відокремити інтерфейс користувача від бізнес-логіки та механізмів зберігання даних, що позитивно впливає на гнучкість архітектури та спрощує подальший розвиток програмного забезпечення.

Для реалізації вебсистем підтримки соціальних проєктів широко використовуються сучасні JavaScript-технології, серверні платформи, реляційні системи керування базами даних та програмні інтерфейси взаємодії між компонентами системи. Аналіз таких технологій дозволяє обґрунтувати вибір інструментів, які доцільно використовувати під час розробки вебсистеми для агрегації та аналізу даних соціальних проєктів.

1.5.1. Клієнт-серверна архітектура вебсистем

Одним із найбільш поширених архітектурних підходів до створення сучасних вебзастосунків є клієнт-серверна архітектура. Її основна ідея полягає у розподілі функціональності між двома основними компонентами системи: клієнтом та сервером [30]. Такий підхід забезпечує ефективну організацію обробки даних, спрощує підтримку програмного забезпечення та дозволяє створювати масштабовані інформаційні системи.

Клієнтська частина системи відповідає за взаємодію з користувачем та відображення інформації через графічний інтерфейс. Саме клієнт забезпечує введення даних, надсилання запитів до сервера та відображення отриманих результатів. У сучасних вебсистемах роль клієнта зазвичай виконує вебзастосунок, який працює у браузері користувача.

Серверна частина системи забезпечує обробку запитів, виконання бізнес-логіки та взаємодію з базою даних. Сервер отримує запити від клієнта, виконує необхідні операції над даними та повертає результати у зручному для подальшого використання форматі. Такий розподіл відповідальності дозволяє відокремити логіку обробки даних від інтерфейсу користувача та спрощує подальшу модернізацію системи.

У клієнт-серверній архітектурі обмін інформацією між компонентами здійснюється за допомогою мережевих протоколів. Користувач взаємодіє з клієнтським застосунком, який формує запит до сервера. Сервер обробляє отримані дані, за необхідності звертається до бази даних та повертає результат клієнту. Завдяки цьому забезпечується централізоване зберігання інформації та єдині правила її обробки для всіх користувачів системи.

Використання клієнт-серверної архітектури має низку переваг. До них належать централізоване управління даними, можливість незалежного розвитку клієнтської та серверної частин системи, підвищення рівня безпеки, а також спрощення процесу супроводу програмного забезпечення. Крім того, такий підхід сприяє кращій масштабованості системи та дозволяє ефективно працювати з великою кількістю користувачів.

Для вебсистеми агрегації та аналізу даних соціальних проєктів використання клієнт-серверної архітектури є доцільним, оскільки забезпечує розмежування функцій між компонентами системи. Клієнтська частина відповідає за взаємодію користувача з платформою та візуалізацію інформації, тоді як серверна частина виконує обробку даних, реалізує механізми авторизації, адміністрування та забезпечує доступ до бази даних.

Клієнт-серверна архітектура є ефективним підходом до побудови сучасних вебсистем, оскільки забезпечує гнучкість, масштабованість та зручність подальшого розвитку програмного забезпечення. Саме тому вона широко використовується при створенні інформаційних систем різного призначення, зокрема платформ для агрегації та аналізу даних.

1.5.2. Сучасні технології розробки клієнтської частини вебсистем

Клієнтська частина вебсистеми забезпечує взаємодію користувача із програмним продуктом та відповідає за відображення інформації, обробку дій користувача і формування запитів до серверної частини. Від якості реалізації клієнтського інтерфейсу значною мірою залежить зручність використання системи, швидкість роботи та загальний користувацький досвід.

Однією з найпоширеніших технологій для розробки сучасних вебінтерфейсів є бібліотека React. Вона була створена компанією Meta та використовується для побудови динамічних односторінкових вебзастосунків. Основною особливістю React є компонентний підхід, відповідно до якого інтерфейс формується з незалежних багаторазово використовуваних компонентів. Така організація коду спрощує розробку, тестування та подальший супровід програмного забезпечення [31].

Важливою перевагою React є використання віртуальної DOM-моделі, яка дозволяє оновлювати лише ті елементи сторінки, що змінилися [32]. Завдяки цьому зменшується кількість операцій із документом сторінки та підвищується продуктивність вебзастосунку.

Для підвищення надійності програмного коду в сучасній веброзробці широко використовується мова TypeScript. Вона є розширенням JavaScript та додає механізми статичної типізації. Використання TypeScript дозволяє виявляти значну кількість помилок ще на етапі розробки, підвищує читабельність коду та спрощує підтримку великих програмних проєктів.

Під час створення сучасних клієнтських застосунків важливу роль також відіграють інструменти автоматизації процесу розробки. Одним із таких інструментів є Vite — сучасне середовище для створення вебзастосунків, яке забезпечує швидкий запуск проєкту, ефективне складання програмного коду та підтримку сучасних можливостей JavaScript і TypeScript. Використання Vite дозволяє прискорити процес розробки та підвищити продуктивність процесу розробки.

Поєднання React, TypeScript та Vite дозволяє створювати сучасні вебінтерфейси, які характеризуються високою швидкістю, зручністю супроводу та можливістю подальшого масштабування. Саме тому зазначені технології широко використовуються під час розробки сучасних інформаційних систем та вебплатформ.

Використання сучасних клієнтських технологій є важливою умовою створення ефективних вебсистем. React забезпечує побудову інтерактивного

інтерфейсу користувача, TypeScript підвищує надійність програмного коду, а Vite спрощує та прискорює процес розробки застосунку.

1.5.3. Сучасні технології розробки серверної частини вебсистем

Серверна частина вебсистеми відповідає за реалізацію бізнес-логіки, обробку запитів користувачів, взаємодію з базою даних та забезпечення безпечного доступу до інформації. Саме сервер виконує основні операції з обробки даних та координує взаємодію між різними компонентами системи.

Однією з найпоширеніших платформ для розробки серверних застосунків є Node.js. Дана технологія являє собою середовище виконання JavaScript поза межами браузера, що дозволяє використовувати єдину мову програмування як для клієнтської, так і для серверної частин вебсистеми. Такий підхід спрощує процес розробки та зменшує складність підтримки програмного забезпечення.

Важливою особливістю Node.js є використання подієво-орієнтованої моделі та неблокуючих операцій введення-виведення. Завдяки цьому сервер здатний ефективно обробляти велику кількість одночасних запитів без значного зниження продуктивності. Така архітектура особливо добре підходить для сучасних вебзастосунків, які активно взаємодіють із базами даних та зовнішніми сервісами [33].

Для спрощення розробки серверної логіки широко використовується фреймворк Express. Він надає набір інструментів для створення вебсерверів, налаштування маршрутів, обробки HTTP-запитів та реалізації програмних інтерфейсів взаємодії. Використання Express дозволяє структурувати код застосунку та спрощує реалізацію серверної архітектури.

Однією з переваг Express є його модульність. Розробник може розділяти функціональність на окремі контролери, маршрути та проміжні обробники, що підвищує читабельність коду та спрощує його подальший супровід. Крім того, фреймворк підтримує інтеграцію з різними бібліотеками та інструментами, що розширює його функціональні можливості.

Поєднання Node.js та Express широко використовується під час створення REST API, серверів авторизації, адміністративних панелей та інших вебсистем, що потребують швидкої обробки даних та підтримки великої кількості користувачів. Завдяки своїй продуктивності, гнучкості та активній спільноті розробників зазначені технології стали стандартом для багатьох сучасних вебпроектів.

Використання Node.js та Express дозволяє створювати ефективні серверні застосунки, які забезпечують обробку даних, реалізацію бізнес-логіки та взаємодію з іншими компонентами системи. Саме тому ці технології широко застосовуються під час розробки сучасних інформаційних систем та вебплатформ.

1.5.4. Технології зберігання та обробки даних

Ефективне функціонування сучасних вебсистем неможливе без надійних засобів зберігання та обробки інформації. Бази даних забезпечують централізоване зберігання даних, підтримують їх цілісність та дозволяють виконувати швидкий доступ до необхідної інформації. Вибір системи керування базами даних суттєво впливає на продуктивність, масштабованість та надійність програмного забезпечення.

Для зберігання структурованої інформації у вебсистемах широко використовуються реляційні системи керування базами даних. Їх основною особливістю є представлення даних у вигляді взаємопов'язаних таблиць, між якими встановлюються логічні зв'язки. Такий підхід дозволяє уникати дублювання інформації, забезпечувати цілісність даних та спрощувати їх подальшу обробку.

Однією з найбільш поширених реляційних систем керування базами даних є PostgreSQL. Вона належить до відкритих СУБД та використовується для створення інформаційних систем різного рівня складності. PostgreSQL підтримує механізми транзакцій, індексацію, зовнішні ключі, обмеження

цілісності та широкий набір засобів оптимізації запитів, що робить її придатною для використання у сучасних вебзастосунках [34].

Важливою перевагою PostgreSQL є підтримка складних структур даних та можливість поєднання класичної реляційної моделі із напівструктурованими даними за допомогою JSON-полів. Такий підхід забезпечує гнучкість проєктування бази даних та дозволяє ефективно працювати як зі структурованою, так і з частково структурованою інформацією.

Під час проєктування реляційних баз даних важливу роль відіграють принципи нормалізації. Нормалізація являє собою процес організації структури даних, спрямований на усунення дублювання інформації та запобігання аномаліям вставки, оновлення і видалення даних. Найбільш поширеними є перша, друга та третя нормальні форми, які забезпечують логічну узгодженість структури бази даних та підвищують ефективність її використання.

Для спрощення взаємодії програмного коду з базою даних у сучасній веброзробці широко застосовуються ORM-технології (Object-Relational Mapping). ORM забезпечує автоматичне перетворення об'єктів програмного коду у записи бази даних та навпаки, що дозволяє зменшити кількість низькорівневих SQL-запитів і підвищити продуктивність розробки.

Однією з популярних ORM-технологій для екосистеми Node.js є Prisma ORM [35]. Вона надає інструменти для проєктування структури бази даних, автоматичного створення міграцій та формування типізованих запитів до СУБД. Використання Prisma дозволяє забезпечити узгодженість між структурою бази даних та програмним кодом, а також спрощує супровід і подальший розвиток інформаційної системи.

Поєднання PostgreSQL та Prisma ORM дозволяє створювати надійні та масштабовані рішення для зберігання даних. PostgreSQL забезпечує ефективне управління інформацією та підтримку цілісності даних, тоді як Prisma спрощує процес розробки та взаємодії серверної частини застосунку з базою даних.

Використання сучасних технологій зберігання та обробки даних є важливою складовою створення вебсистем. Застосування PostgreSQL та Prisma

ORM дозволяє забезпечити надійне зберігання інформації, підтримку нормалізованої структури бази даних та ефективну взаємодію між програмним кодом і СУБД.

1.5.5. Використання REST API для взаємодії компонентів системи

У сучасних вебсистемах важливу роль відіграють механізми взаємодії між клієнтською та серверною частинами застосунку. Для забезпечення обміну даними між компонентами системи широко використовуються програмні інтерфейси прикладного програмування (Application Programming Interface, API). Одним із найпоширеніших підходів до побудови таких інтерфейсів є архітектурний стиль REST [36].

REST (Representational State Transfer) являє собою набір принципів організації взаємодії між компонентами розподілених інформаційних систем. Основною ідеєю REST є представлення даних у вигляді ресурсів, доступ до яких здійснюється через стандартизовані HTTP-запити. Такий підхід забезпечує простоту реалізації, універсальність та сумісність із різними програмними платформами.

У REST API кожен ресурс має власну адресу (URL), а взаємодія з ним здійснюється за допомогою стандартних HTTP-методів [37]. Найчастіше використовуються методи GET для отримання даних, POST для створення нових записів, PUT або PATCH для оновлення інформації та DELETE для видалення даних. Використання єдиних принципів взаємодії спрощує розробку та підтримку програмних систем.

Однією з переваг REST API є незалежність клієнтської та серверної частин застосунку. Клієнт взаємодіє із сервером виключно через визначені програмні інтерфейси, не маючи доступу до внутрішньої логіки роботи системи. Це дозволяє розробляти та вдосконалювати окремі компоненти незалежно один від одного, що позитивно впливає на масштабованість та супровід програмного забезпечення.

У більшості сучасних вебзастосунків передача даних між компонентами здійснюється у форматі JSON. Такий формат є компактним, зручним для обробки та підтримується практично всіма мовами програмування і програмними платформами. Використання JSON спрощує обмін інформацією між клієнтом та сервером і забезпечує високу швидкість взаємодії компонентів системи.

Для вебсистеми агрегації та аналізу даних соціальних проєктів використання REST API є доцільним, оскільки дозволяє організувати обмін інформацією між клієнтською частиною, серверною логікою та базою даних. Через REST API реалізуються операції отримання переліку проєктів, перегляду детальної інформації, керування організаціями, авторизації користувачів, адміністрування даних та формування аналітичної інформації.

REST API є одним із найбільш поширених підходів до організації взаємодії між компонентами сучасних вебсистем. Його використання забезпечує гнучкість архітектури, незалежність окремих модулів системи та ефективний обмін даними між клієнтською і серверною частинами застосунку.

Висновок до розділу 1

У межах аналізу існуючих платформ соціальної підтримки було встановлено, що сучасні інформаційні системи успішно вирішують окремі завдання, пов'язані зі збором благодійних внесків, представленням діяльності організацій та підтримкою процесів адопції тварин. Водночас жодна з розглянутих платформ не поєднує функції агрегації даних, управління соціальними проєктами, персоналізованої взаємодії користувачів та аналітичної обробки інформації в межах єдиного програмного рішення.

Також було досліджено сучасні технології та архітектурні підходи до створення вебсистем. Аналіз клієнт-серверної архітектури, технологій React, TypeScript, Node.js, Express, PostgreSQL, Prisma ORM та REST API показав їх придатність для побудови масштабованих інформаційних систем із розподіленою обробкою даних та сучасним користувацьким інтерфейсом.

Отримані результати дозволили сформувати теоретичне та технологічне підґрунтя для подальшого проектування вебсистеми агрегації та аналізу даних соціальних проєктів. На основі проведеного аналізу у наступному розділі буде визначено вимоги до системи, розроблено її архітектуру та спроектовано структуру бази даних.



РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБСИСТЕМИ ДЛЯ АГРЕГАЦІЇ ТА АНАЛІЗУ ДАНИХ СОЦІАЛЬНИХ ПРОЄКТІВ

2.1. Постановка задачі та вимоги

У результаті проведеного аналізу предметної області, існуючих платформ соціальної підтримки та сучасних технологій веброзробки було встановлено необхідність створення вебсистеми, яка забезпечуватиме централізовану агрегацію, зберігання, пошук та аналіз даних соціальних проєктів.

Сучасні соціальні ініціативи функціонують у середовищі, де інформація про проєкти, організації та потреби часто розміщується на різних вебресурсах і представлена у неоднорідному вигляді. Це ускладнює доступ користувачів до актуальних даних, знижує ефективність пошуку необхідної інформації та обмежує можливості її подальшого аналізу.

Особливо актуальною дана проблема є для соціальних проєктів у сфері допомоги тваринам, діяльність яких пов'язана з роботою притулків, волонтерських організацій, благодійних фондів та окремих ініціативних груп. Відсутність єдиного інформаційного середовища ускладнює представлення таких проєктів, взаємодію між організаціями та потенційними користувачами, а також отримання узагальненої інформації щодо їх діяльності.

Для вирішення зазначених проблем у межах даної роботи поставлено задачу розробки вебсистеми для агрегації та аналізу даних соціальних проєктів [38]. Система повинна забезпечувати централізоване зберігання інформації про соціальні ініціативи, підтримувати механізми пошуку та фільтрації, надавати інструменти адміністрування даних і забезпечувати можливість отримання узагальнених аналітичних показників.

Основною метою розробки є створення інформаційної платформи, яка об'єднуватиме дані про соціальні проєкти та організації в межах єдиного інформаційного середовища, забезпечуватиме зручний доступ до інформації для користувачів та підтримуватиме базові засоби аналізу діяльності соціальних ініціатив.

Розроблювана вебсистема орієнтована на підтримку двох основних категорій користувачів: відвідувачів платформи та адміністраторів. Користувачі повинні мати можливість переглядати інформацію про соціальні проєкти, виконувати пошук і фільтрацію даних, зберігати цікаві ініціативи та отримувати доступ до узагальненої інформації про діяльність організацій. Адміністратори, у свою чергу, повинні забезпечувати наповнення та актуалізацію інформаційних ресурсів системи, керувати проєктами, організаціями та обліковими записами користувачів.

Для досягнення поставленої мети необхідно визначити вимоги до майбутньої вебсистеми, які охоплюють функціональні можливості, характеристики якості програмного забезпечення та особливості організації даних. Аналіз зазначених вимог наведено у наступному підрозділі.

2.2. Аналіз вимог до вебсистеми

Перед початком проєктування вебсистеми необхідно визначити вимоги до її функціонування, структури та обробки даних. Формалізація вимог дозволяє встановити межі майбутньої системи, визначити її основні можливості та забезпечити відповідність розробленого програмного забезпечення поставленим цілям [39].

Вимоги до вебсистеми формуються на основі проведеного аналізу предметної області, дослідження існуючих платформ соціальної підтримки та особливостей роботи користувачів із соціальними проєктами. Вони визначають функціональні можливості системи, характеристики якості програмного забезпечення, вимоги до організації даних та особливості взаємодії між компонентами системи.

У межах даної роботи вимоги до вебсистеми поділено на загальні, функціональні та нефункціональні вимоги, а також вимоги до даних та інформаційних ресурсів. Такий підхід дозволяє комплексно описати майбутню систему та створити основу для подальшого проєктування її архітектури й структури бази даних.

2.2.1. Загальні вимоги

Вебсистема для агрегації та аналізу даних соціальних проєктів повинна забезпечувати єдине інформаційне середовище для зберігання, обробки та представлення інформації про соціальні ініціативи у сфері допомоги тваринам. Система має підтримувати роботу з даними про проєкти, організації та користувачів, а також надавати інструменти для пошуку, аналізу та адміністрування інформації.

Розроблювана система повинна функціонувати як вебзастосунок із доступом через сучасні веббраузери без необхідності встановлення додаткового програмного забезпечення на пристрої користувача. Взаємодія між клієнтською та серверною частинами має здійснюватися через мережу Інтернет із використанням стандартизованих механізмів обміну даними.

Система повинна підтримувати багатокористувацький режим роботи та забезпечувати одночасний доступ до інформаційних ресурсів для різних категорій користувачів. При цьому права доступу мають визначатися відповідно до ролі користувача в системі.

У межах вебсистеми необхідно забезпечити підтримку двох основних ролей: користувача та адміністратора. Користувачі повинні мати можливість переглядати інформацію про соціальні проєкти, використовувати засоби пошуку та фільтрації, зберігати цікаві ініціативи та переглядати історію власної активності. Адміністратори повинні мати розширені права доступу, які дозволяють керувати користувачами, організаціями та соціальними проєктами.

Вебсистема повинна забезпечувати централізоване зберігання інформації та підтримувати актуальність даних шляхом їх своєчасного оновлення. Особлива увага має приділятися зручності використання системи, зрозумілій навігації та швидкому доступу до необхідної інформації.

Загальні вимоги визначають основні принципи функціонування вебсистеми та формують основу для подальшого визначення її функціональних і нефункціональних характеристик.

2.2.2. Функціональні вимоги

Функціональні вимоги визначають перелік можливостей та операцій, які повинна забезпечувати вебсистема для досягнення поставлених цілей. Вони описують поведінку системи під час взаємодії з користувачами та визначають основні бізнес-процеси, що реалізуються в межах програмного забезпечення.

Вебсистема для агрегації та аналізу даних соціальних проєктів повинна забезпечувати виконання таких функцій:

- реєстрацію нових користувачів у системі;
- авторизацію та автентифікацію користувачів;
- розмежування прав доступу відповідно до ролі користувача;
- перегляд переліку соціальних проєктів;
- перегляд детальної інформації про окремий соціальний проєкт;
- перегляд інформації про організації, пов'язані із соціальними проєктами;
- пошук соціальних проєктів за ключовими словами;
- фільтрацію проєктів за категорією, місцем реалізації та статусом;
- збереження проєктів до персонального списку користувача;
- перегляд історії взаємодії користувача із системою;
- редагування персональних даних користувача;
- формування та відображення статистичної інформації щодо соціальних проєктів;
- відображення аналітичних показників у вигляді інформаційних блоків та графічних елементів.

Для забезпечення адміністрування системи необхідно реалізувати окремий адміністративний модуль, який повинен підтримувати:

- створення нових соціальних проєктів;
- редагування інформації про соціальні проєкти;
- видалення соціальних проєктів;
- створення нових організацій;
- редагування інформації про організації;

- видалення організацій;
- перегляд інформації про користувачів;
- редагування облікових записів користувачів;
- видалення користувачів;
- перегляд статистичних показників роботи системи.

Функціональні можливості системи повинні забезпечувати повний цикл роботи з інформацією: від її внесення та зберігання до пошуку, аналізу та представлення користувачам. Реалізація зазначених функцій дозволить створити єдине інформаційне середовище для підтримки соціальних проєктів та забезпечить ефективну взаємодію між користувачами, організаціями та адміністраторами системи.

Для узагальнення функціональних можливостей вебсистеми доцільно побудувати діаграму варіантів використання (Use Case Diagram), яка відображає взаємодію основних категорій користувачів із функціями системи [40].

2.2.3. Нефункціональні вимоги

Нефункціональні вимоги визначають характеристики якості програмного забезпечення та встановлюють обмеження щодо функціонування вебсистеми [41]. На відміну від функціональних вимог, вони описують не перелік операцій, які виконує система, а умови, за яких ці операції повинні виконуватися. Дотримання нефункціональних вимог забезпечує стабільність, безпеку, зручність використання та можливість подальшого розвитку програмного продукту.

Розроблювана вебсистема повинна відповідати таким нефункціональним вимогам:

Продуктивність. Система повинна забезпечувати швидке завантаження сторінок, обробку запитів користувачів та відображення результатів пошуку і фільтрації. Час відгуку основних функцій не повинен створювати відчутних затримок під час роботи користувача із платформою.

Надійність. Вебсистема повинна забезпечувати стабільну роботу та

коректну обробку помилок. У випадку виникнення непередбачених ситуацій система повинна запобігати втраті даних та інформувати користувача про наявність проблеми.

Безпека. Необхідно забезпечити захист персональних даних користувачів та контроль доступу до функціональних можливостей системи. Доступ до адміністративного функціоналу повинен бути обмежений відповідними правами доступу. Передача даних між клієнтською та серверною частинами повинна здійснюватися із використанням безпечних механізмів автентифікації та авторизації.

Масштабованість. Архітектура системи повинна забезпечувати можливість подальшого розширення функціональності без необхідності суттєвої зміни існуючих компонентів. У майбутньому система повинна підтримувати інтеграцію додаткових модулів та сервісів.

Зручність використання. Інтерфейс користувача повинен бути інтуїтивно зрозумілим та забезпечувати швидкий доступ до основних функцій системи. Навігація між сторінками повинна бути логічною та послідовною, а основні дії користувача — виконуватися за мінімальну кількість кроків.

Адаптивність інтерфейсу. Вебсистема повинна коректно відображатися на пристроях із різними розмірами екранів, включаючи персональні комп'ютери, планшети та мобільні пристрої. Адаптивний дизайн забезпечує зручність використання системи незалежно від типу пристрою.

Підтримуваність. Програмний код системи повинен мати зрозумілу структуру та бути придатним для подальшого супроводу й модернізації. Використання сучасних технологій розробки та модульної архітектури повинно спрощувати внесення змін і розширення функціональності.

Сумісність. Система повинна підтримувати роботу в сучасних веббраузерах та не вимагати встановлення спеціального програмного забезпечення на стороні користувача.

Дотримання зазначених нефункціональних вимог дозволить забезпечити стабільну роботу вебсистеми, підвищити якість взаємодії користувачів із

платформою та створити основу для її подальшого розвитку і масштабування.

2.2.4. Вимоги до даних та інформаційних ресурсів

Інформаційне забезпечення є однією з ключових складових будь-якої вебсистеми, оскільки саме дані виступають основою для реалізації функціональних можливостей програмного забезпечення. Від структури, якості та організації даних залежить ефективність роботи системи, швидкість обробки інформації та можливість подальшого аналізу діяльності соціальних проєктів.

Розроблювана вебсистема повинна забезпечувати накопичення, зберігання, оновлення та обробку інформації про соціальні проєкти, організації та користувачів. Дані мають зберігатися у структурованому вигляді та підтримувати можливість швидкого пошуку, фільтрації та формування аналітичних показників.

Основними інформаційними ресурсами системи є відомості про соціальні проєкти. Для кожного проєкту повинна зберігатися така інформація:

- назва проєкту;
- короткий опис;
- повний опис;
- категорія проєкту;
- місце реалізації;
- статус проєкту;
- контактна інформація;
- дата створення та оновлення запису;
- інформація про організацію, відповідальну за реалізацію проєкту.

Важливою складовою системи є дані про організації, які здійснюють соціальну діяльність. Для кожної організації повинні зберігатися:

- назва організації;
- опис діяльності;
- контактні дані;
- місто розташування;

- посилання на офіційні інформаційні ресурси;
- дата створення та оновлення запису.

Окрему категорію інформаційних ресурсів становлять дані користувачів системи. Для забезпечення авторизації та персоналізованої взаємодії необхідно зберігати:

- ім'я користувача;
- адресу електронної пошти;
- пароль у зашифрованому вигляді;
- роль користувача;
- дату реєстрації;
- службову інформацію щодо активності користувача.

Для реалізації персоналізованих функцій система повинна підтримувати зберігання інформації про взаємодію користувачів із соціальними проєктами. До таких даних належать:

- збережені проєкти;
- історія перегляду проєктів;
- дата та час взаємодії користувача із системою.

Інформаційні ресурси вебсистеми повинні відповідати вимогам актуальності, достовірності, цілісності та узгодженості. Для запобігання дублюванню даних необхідно використовувати механізми нормалізації бази даних, а також забезпечувати контроль зв'язків між інформаційними об'єктами системи.

Визначення вимог до даних та інформаційних ресурсів дозволяє сформувати основу для подальшого проєктування структури бази даних, встановлення зв'язків між сутностями та побудови архітектури зберігання інформації у вебсистемі.

2.3. Проєктування архітектури вебсистеми

Після визначення функціональних та нефункціональних вимог до вебсистеми наступним етапом є проєктування її архітектури. Архітектура

програмного забезпечення визначає структуру системи, взаємозв'язки між її компонентами та принципи організації обробки даних [42]. Грамотно спроектована архітектура забезпечує масштабованість, підтримуваність та можливість подальшого розвитку програмного продукту.

Під час проектування вебсистеми для агрегації та аналізу даних соціальних проєктів було обрано клієнт-серверний підхід, який є одним із найбільш поширених рішень для сучасних вебзастосунків. Така архітектура передбачає розподіл системи на окремі компоненти, кожен із яких виконує визначені функції та взаємодіє з іншими компонентами через стандартизовані інтерфейси.

У межах розроблюваної системи можна виділити три основні рівні [43]:

- клієнтський рівень;
- серверний рівень;
- рівень зберігання даних.

Клієнтський рівень забезпечує взаємодію користувача із системою через вебінтерфейс. Основними завданнями цього рівня є відображення інформації, отримання даних від користувача, формування запитів до серверної частини та візуалізація результатів обробки даних.

Серверний рівень реалізує бізнес-логіку системи. Він відповідає за обробку запитів користувачів, виконання операцій над даними, перевірку прав доступу, взаємодію з базою даних та формування відповідей для клієнтської частини. Саме на серверному рівні реалізуються основні механізми управління соціальними проєктами, організаціями та користувачами.

Рівень зберігання даних забезпечує довготривале зберігання інформації та підтримує цілісність даних. У базі даних зберігаються відомості про соціальні проєкти, організації, користувачів, історію взаємодії та інші інформаційні ресурси, необхідні для функціонування системи.

Під час взаємодії користувача із системою запити надходять від клієнтської частини до серверного застосунку через REST API. Після обробки запиту сервер звертається до бази даних, отримує необхідну інформацію та повертає результат клієнту у форматі JSON. Такий підхід забезпечує

незалежність окремих компонентів системи та спрощує їх подальшу модернізацію.

Використання багаторівневої архітектури дозволяє розмежувати відповідальність між компонентами системи, спростити супровід програмного забезпечення та забезпечити можливість масштабування окремих модулів без впливу на роботу всієї системи.

Обрана архітектура вебсистеми відповідає визначеним вимогам, забезпечує гнучкість розробки та створює основу для реалізації функціональних можливостей, передбачених у межах проєкту.

2.4. Проєктування структури бази даних

Структура бази даних є однією з ключових складових вебсистеми, оскільки забезпечує організоване зберігання інформації та підтримує взаємодію між усіма компонентами програмного забезпечення. Якість проєктування бази даних безпосередньо впливає на продуктивність системи, цілісність даних та можливість подальшого розширення функціональності.

Під час проєктування вебсистеми для агрегації та аналізу даних соціальних проєктів структура бази даних формувалася на основі функціональних вимог та особливостей предметної області. Основною метою проєктування було створення моделі даних, яка забезпечувала б ефективне зберігання інформації про користувачів, соціальні проєкти, організації та взаємодію користувачів із системою.

Для реалізації системи було обрано реляційну модель даних, що дозволяє організувати інформацію у вигляді взаємопов'язаних таблиць та забезпечити контроль цілісності даних за допомогою первинних і зовнішніх ключів [44].

У процесі аналізу предметної області було визначено такі основні інформаційні сутності:

- користувач;
- соціальний проєкт;
- організація;

- категорія проєкту;
- статус проєкту;
- потреба;
- варіант підтримки;
- адреса;
- географічна одиниця;
- медіаресурс;
- збережений проєкт;
- переглянутий проєкт.

Кожна із зазначених сутностей відображає окремий об'єкт предметної області та містить набір атрибутів, необхідних для реалізації функціональних можливостей вебсистеми.

Для забезпечення цілісності інформації між сутностями встановлюються логічні зв'язки. Соціальний проєкт належить певній організації, користувач може взаємодіяти з багатьма проєктами, а інформація про такі взаємодії повинна зберігатися окремо для подальшого аналізу активності користувачів.

У результаті аналізу предметної області було сформовано набір взаємопов'язаних сутностей, які забезпечують зберігання інформації про соціальні проєкти, організації, користувачів та їх взаємодію із системою.

Центральною сутністю бази даних є сутність «Проект», яка містить основну інформацію про соціальну ініціативу. Кожен проєкт пов'язаний із конкретною організацією, категорією діяльності та поточним статусом реалізації. Такий підхід дозволяє забезпечити структуроване представлення інформації та спростити виконання пошуку і фільтрації даних.

Для зберігання інформації про організації використовується окрема сутність «Організація». Розділення даних про проєкти та організації дозволяє уникнути дублювання інформації та забезпечує можливість пов'язування декількох проєктів з однією організацією.

Особливістю структури бази даних є виділення окремих сутностей для зберігання географічної інформації. Дані про країни, регіони, міста та адреси

представлені окремими таблицями, що дозволяє підтримувати ієрархічну структуру місцезнаходжень та забезпечує відповідність принципам нормалізації.

Для реалізації класифікації проєктів використовуються довідникові сутності «Категорія проєкту» та «Статус проєкту». Застосування довідників спрощує адміністрування системи та забезпечує узгодженість інформації під час створення нових записів.

Окрему групу сутностей становлять об'єкти, що забезпечують підтримку потреб проєктів та способів їх підтримки. Для цього використовуються таблиці потреб, варіантів підтримки та додаткових описів способів допомоги. Такий підхід дозволяє гнучко описувати потреби соціальних ініціатив та надавати користувачам детальну інформацію щодо можливих способів участі.

Для реалізації персоналізованої взаємодії користувачів із системою використовуються сутності збережених та переглянутих проєктів. Вони дозволяють формувати історію активності користувачів, підтримувати персональні добірки проєктів та збирати статистичні дані щодо популярності соціальних ініціатив.

Під час проєктування структури бази даних особлива увага приділялася принципам нормалізації [45]. Використання нормалізованої моделі дозволяє зменшити дублювання інформації, спростити оновлення даних та забезпечити узгодженість інформаційних ресурсів системи.

Перша нормальна форма забезпечується використанням атомарних значень у полях таблиць та відсутністю повторюваних груп атрибутів. У структурі бази даних кожна таблиця має первинний ключ, а атрибути зберігають окремі неподільні значення. Наприклад, інформація про адресу не зберігається одним текстовим полем у таблиці організацій або проєктів, а винесена до окремої сутності адреси.

Друга нормальна форма забезпечується винесенням логічно незалежних даних до окремих таблиць. Зокрема, інформація про організації, категорії проєктів, статуси, потреби та медіаресурси зберігається окремо від основної

таблиці проєктів. Це дозволяє уникнути повторного збереження однакових даних у різних записах та спрощує їх оновлення.

Третя нормальна форма досягається усуненням транзитивних залежностей між атрибутами. Наприклад, географічна інформація представлена окремими сутностями країни, регіону, міста та адреси. Завдяки цьому назви країн, регіонів і міст не дублюються безпосередньо в таблицях проєктів або організацій, а зберігаються у відповідних довідникових таблицях.

Окремі зв'язки типу «багато до багатьох» реалізовано через проміжні таблиці. Наприклад, зв'язок між соціальними проєктами та їх потребами реалізовано за допомогою таблиці зв'язку, що дозволяє одному проєкту мати декілька потреб, а одну й ту саму потребу використовувати в різних проєктах.

Для реалізації зв'язків між сутностями використовуються зовнішні ключі, які забезпечують контроль цілісності даних та підтримують узгодженість інформації між таблицями. Такий підхід дозволяє ефективно реалізувати функції пошуку, фільтрації, збереження проєктів та формування статистичної інформації [46].

Для реалізації зв'язків між сутностями використовуються зовнішні ключі, які забезпечують контроль цілісності даних та підтримують узгодженість інформації між таблицями. Такий підхід дозволяє ефективно реалізувати функції пошуку, фільтрації, збереження проєктів та формування статистичної інформації.

Спроектована структура бази даних забезпечує надійне зберігання інформації, підтримує реалізацію функціональних вимог вебсистеми та створює основу для подальшого розвитку програмного забезпечення.

Висновок до розділу 2

У другому розділі було виконано проєктування вебсистеми для агрегації та аналізу даних соціальних проєктів на основі результатів дослідження предметної області та вимог до майбутнього програмного забезпечення.

У процесі роботи сформульовано основну задачу розробки та визначено цілі створення системи. Встановлено, що вебсистема повинна забезпечувати централізоване представлення інформації про соціальні проєкти, підтримувати пошук і фільтрацію даних, надавати засоби персоналізованої взаємодії користувачів та забезпечувати можливість отримання аналітичної інформації щодо діяльності соціальних ініціатив.

Було виконано аналіз вимог до системи, у межах якого визначено загальні, функціональні та нефункціональні вимоги. Зокрема, встановлено перелік основних функцій користувача та адміністратора, визначено вимоги до продуктивності, безпеки, масштабованості та зручності використання вебсистеми. Також сформовано вимоги до інформаційних ресурсів та структури даних, необхідних для реалізації функціональних можливостей програмного забезпечення.

На основі визначених вимог спроектовано архітектуру вебсистеми із використанням клієнт-серверного підходу. Обрана архітектура забезпечує розподіл відповідальності між компонентами системи, підтримує ефективну взаємодію між клієнтською та серверною частинами та створює умови для подальшого розвитку програмного продукту.

Окрему увагу приділено проєктуванню структури бази даних. Було визначено основні сутності предметної області, встановлено зв'язки між ними та сформовано концептуальну модель даних. Під час проєктування структури бази даних застосовано принципи нормалізації, що дозволило мінімізувати дублювання інформації та забезпечити цілісність даних.

Отримані результати проєктування створюють необхідне підґрунтя для практичної реалізації вебсистеми [47]. На основі розробленої архітектури та структури бази даних у наступному розділі буде розглянуто процес реалізації програмного забезпечення, особливості розробки окремих функціональних модулів та результати роботи створеної системи.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ВЕБСИСТЕМИ ДЛЯ АГРЕГАЦІЇ ТА АНАЛІЗУ ДАНИХ СОЦІАЛЬНИХ ПРОЄКТІВ

3.1. Вибір засобів та середовища розробки

Практична реалізація вебсистеми SocialData Hub виконувалася із використанням сучасного стеку вебтехнологій, який забезпечує створення масштабованих клієнт-серверних застосунків та відповідає вимогам, сформульованим у другому розділі роботи.

Для розробки клієнтської частини системи використано бібліотеку React та мову програмування TypeScript [48]. Використання React дозволяє будувати інтерфейс за компонентним принципом, що спрощує повторне використання елементів та підтримку програмного коду. Застосування TypeScript забезпечує статичну типізацію даних і сприяє підвищенню надійності програмного забезпечення.

Серверна частина реалізована на платформі Node.js із використанням фреймворку Express. Таке рішення дозволяє створити REST API для обміну даними між клієнтською частиною та базою даних, а також реалізувати бізнес-логіку системи, механізми авторизації та адміністративний функціонал.

Для зберігання інформації про користувачів, організації та соціальні проєкти використано систему керування базами даних PostgreSQL. Взаємодія між серверною частиною та базою даних реалізована за допомогою Prisma ORM, що забезпечує зручну роботу з даними та підтримку міграцій структури бази даних.

Обмін даними між компонентами системи здійснюється через REST API із використанням формату JSON. Такий підхід забезпечує незалежність клієнтської та серверної частин вебсистеми та спрощує подальший розвиток програмного продукту.

Розробка програмного забезпечення виконувалася у середовищі Visual Studio Code. Для керування версіями програмного коду використовувалася система Git із розміщенням репозиторію на платформі GitLab [49]. Такий підхід

забезпечив контроль змін програмного коду та спростив процес розробки окремих компонентів системи.

Структуру реалізованого проєкту наведено на рисунку 3.1.

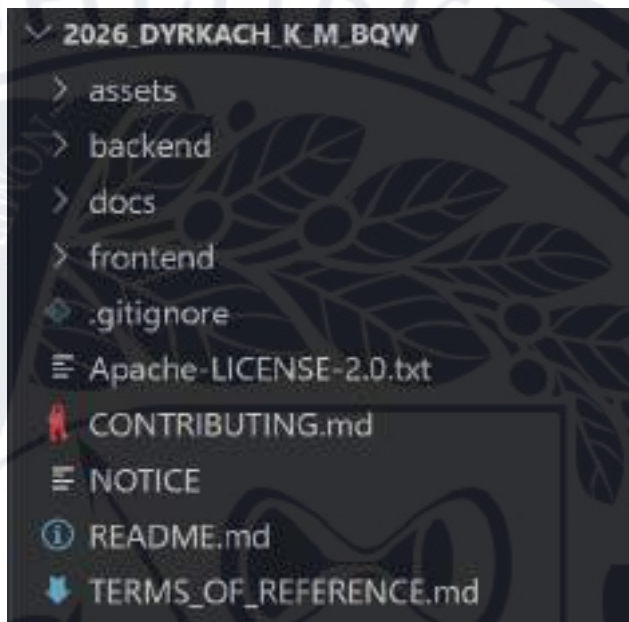


Рисунок 3.1 – Структура проєкту вебсистеми SocialData Hub у середовищі Visual Studio Code

Під час розробки використовувався підхід модульної організації програмного коду. Кожен функціональний модуль системи реалізовувався незалежно та містив власні компоненти інтерфейсу, маршрути, сервіси доступу до даних та засоби обробки запитів. Такий підхід спрощує подальшу підтримку та розширення функціональності вебсистеми.

Реалізація вебсистеми виконувалася із розділенням клієнтської та серверної частин на окремі програмні модулі. Клієнтська частина розташована у каталозі frontend та містить сторінки, компоненти інтерфейсу, маршрутизацію та засоби взаємодії з REST API. Серверна частина розташована у каталозі backend та містить контролери, маршрути, бізнес-логіку системи та механізми взаємодії з базою даних. Документація проєкту та результати проєктування зберігаються у каталозі docs.

Обрані засоби розробки дозволили реалізувати вебсистему відповідно до поставлених функціональних вимог та забезпечили можливість її подальшого розвитку і масштабування.

3.2. Реалізація клієнтської частини вебсистеми

Клієнтська частина вебсистеми SocialData Hub реалізована із використанням бібліотеки React та мови TypeScript. Основним завданням клієнтської частини є забезпечення взаємодії користувачів із системою, перегляд інформації про соціальні проєкти, організації та потреби, а також виконання операцій авторизації та адміністрування.

Клієнтська частина реалізована відповідно до компонентного підходу. Інтерфейс вебсистеми складається з набору незалежних компонентів, які можуть повторно використовуватися на різних сторінках застосунку. До таких компонентів належать навігаційне меню, картки проєктів, інформаційні блоки статистики, форми введення даних та елементи фільтрації.

Компонентна структура дозволила зменшити дублювання програмного коду, спростити підтримку інтерфейсу та забезпечити можливість подальшого розширення функціональності системи.

Окрім логічної структури сторінок, під час розробки було сформовано модульну структуру клієнтської частини застосунку. Вона передбачає розділення програмного коду на окремі каталоги відповідно до їх призначення. Такий підхід дозволяє впорядкувати структуру проєкту, спростити супровід програмного забезпечення та забезпечити можливість подальшого розширення функціональності системи.

У каталозі pages реалізовано сторінки вебсистеми, які відповідають за відображення основних розділів застосунку. Каталог components містить багаторазово використовувані елементи інтерфейсу, services забезпечує взаємодію з REST API, а каталог utils містить допоміжні функції та засоби обробки даних. Основними точками входу клієнтської частини є файли main.tsx

та App.tsx, які забезпечують ініціалізацію застосунку та налаштування маршрутизації.

Структуру клієнтської частини вебсистеми наведено на рисунку 3.2.

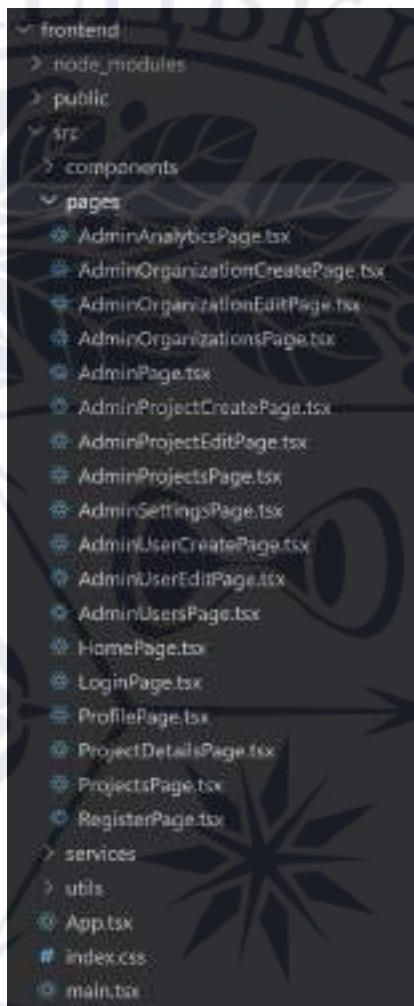


Рисунок 3.2 – Структура клієнтської частини вебсистеми SocialData Hub

Під час розробки інтерфейсу особлива увага приділялася простоті навігації, адаптивності та доступності. Інтерфейс побудований за компонентним підходом, що дозволяє повторно використовувати окремі елементи сторінок та спрощує подальшу підтримку системи.

Для реалізації навігації між сторінками використано бібліотеку React Router. У системі реалізовано окремі маршрути для користувацької та адміністративної частин вебзастосунку. Користувацька частина включає сторінки головної сторінки, каталогу проєктів, перегляду деталей проєкту,

реєстрації, авторизації та профілю користувача. Адміністративна частина містить сторінки управління проектами, організаціями, користувачами, налаштуваннями системи та перегляду аналітичної інформації.

Структуру сторінок вебсистеми SocialData Hub наведено на рисунку 3.3.

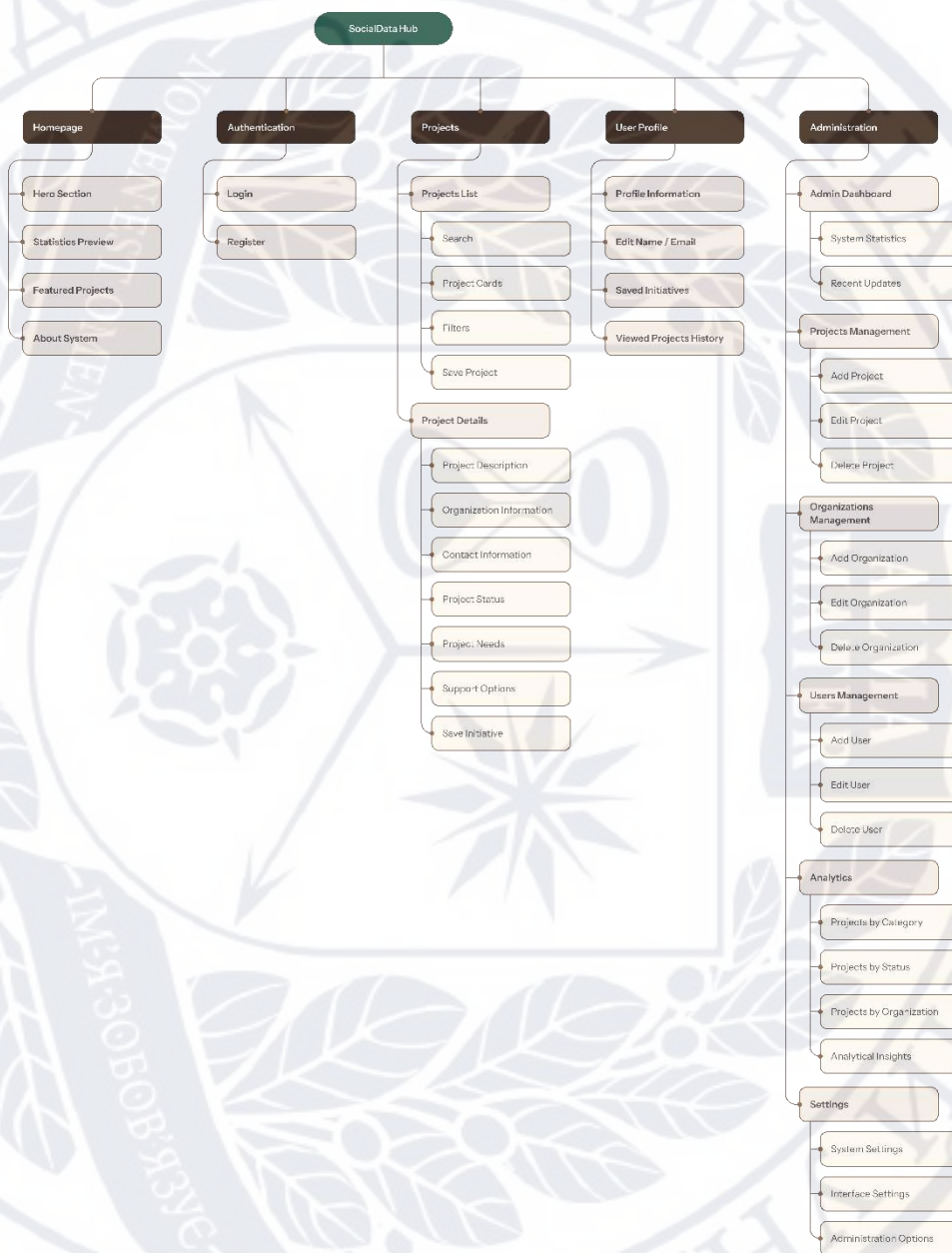


Рисунок 3.3 – Карта сторінок вебсистеми SocialData Hub

Головною сторінкою вебсистеми є сторінка представлення платформи, яка містить короткий опис призначення системи, статистичну інформацію, перелік актуальних потреб та інформацію про принцип роботи платформи.

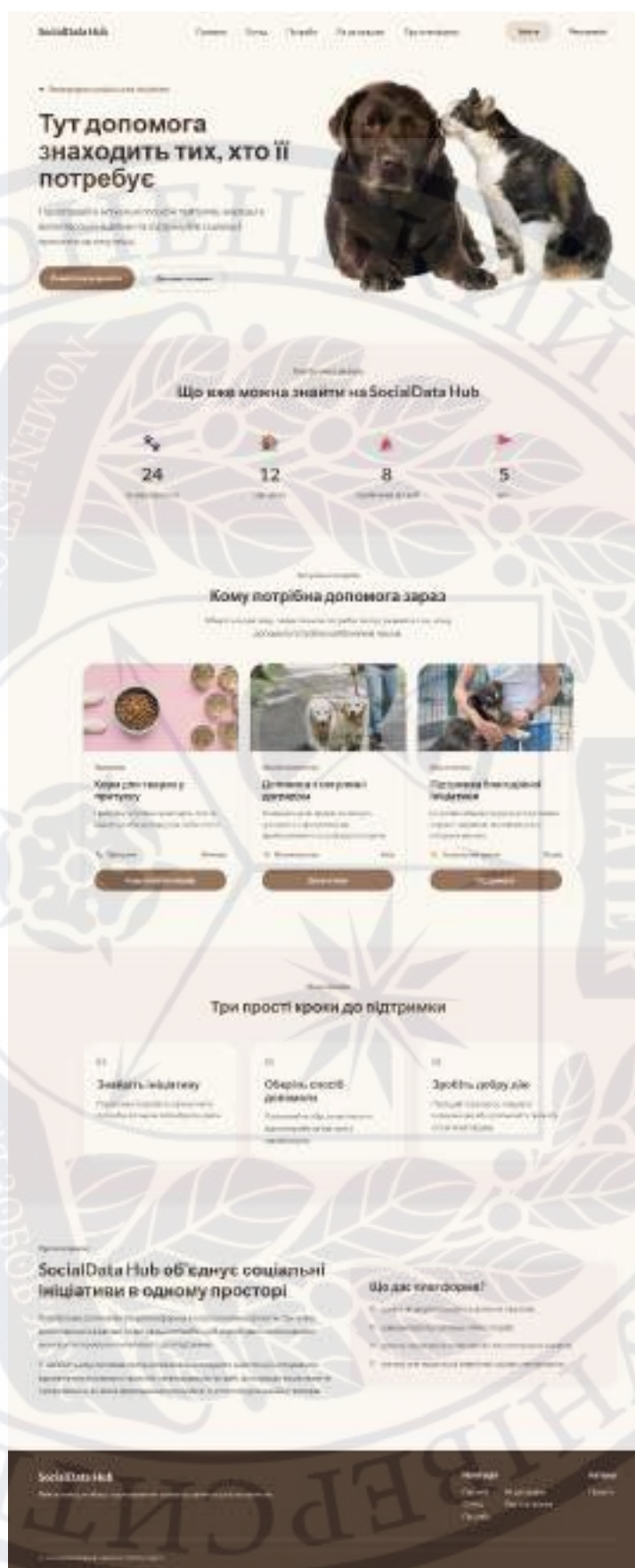


Рисунок 3.4 – Головна сторінка вебсистеми SocialData Hub

Для пошуку соціальних ініціатив реалізовано сторінку каталогу проєктів. Користувач може переглядати перелік доступних проєктів, здійснювати пошук за назвою або організацією, а також використовувати фільтрацію за статусом проєкту.

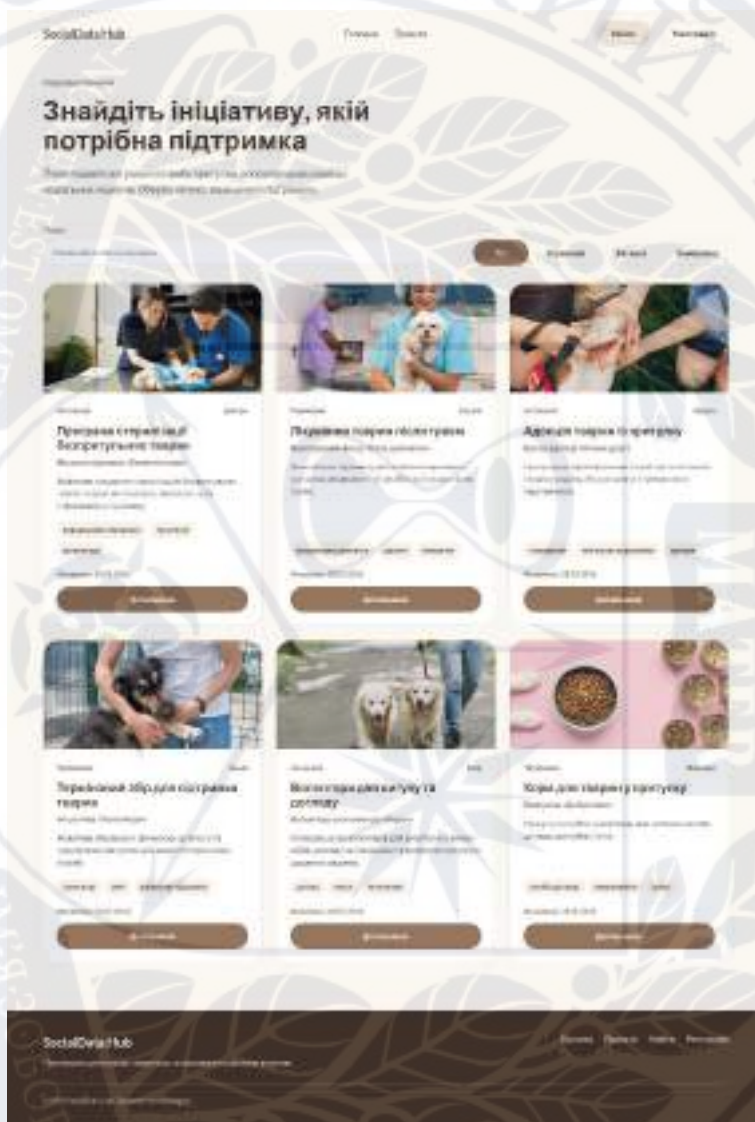


Рисунок 3.5 – Сторінка перегляду соціальних проєктів

Для кожного соціального проєкту реалізовано окрему сторінку детального перегляду. На цій сторінці відображається розширена інформація про проєкт, організацію, місце реалізації, категорії потреб та доступні способи підтримки.



Рисунок 3.6 – Сторінка детального перегляду соціального проєкту

Для роботи з обліковими записами користувачів у вебсистемі реалізовано сторінки реєстрації та авторизації. Вони забезпечують створення нового облікового запису, перевірку введених даних та подальший доступ до функціональних можливостей системи відповідно до ролі користувача.

Після авторизації користувач отримує доступ до сторінки профілю, де може переглядати персональну інформацію та взаємодіяти із збереженими ініціативами. Реалізація профілю дозволяє персоналізувати роботу із платформою та забезпечує зручний доступ до основних дій користувача.

Для адміністрування вебсистеми реалізовано окремий набір сторінок адміністративної панелі. Адміністратор може здійснювати управління соціальними проєктами, організаціями та користувачами системи, переглядати

аналітичну інформацію, а також виконувати налаштування окремих параметрів платформи. Наявність окремого адміністративного інтерфейсу забезпечує централізоване керування даними та спрощує супровід вебсистеми.

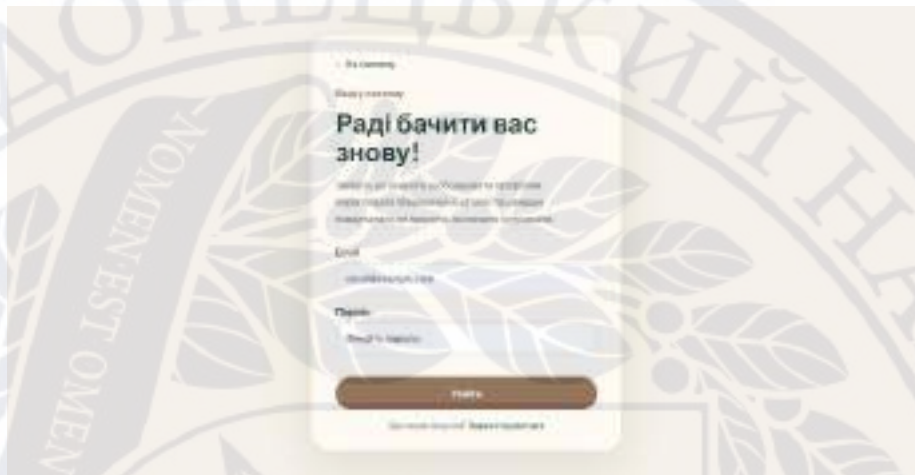


Рисунок 3.7 – Сторінка авторизації користувача

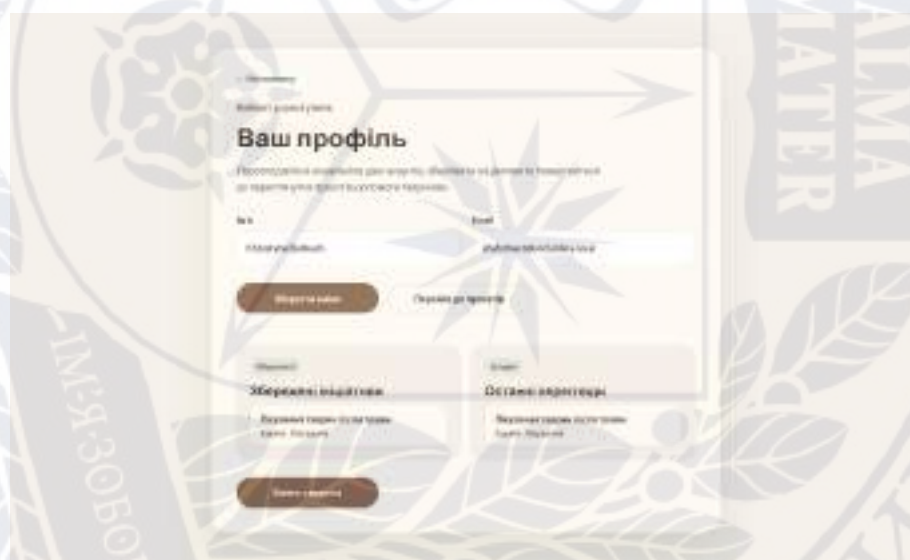


Рисунок 3.8 – Сторінка профілю користувача

Під час розробки інтерфейсу також реалізовано адаптивний дизайн, який забезпечує коректне відображення вебсистеми на мобільних пристроях, планшетах та персональних комп'ютерах.

Приклад адаптації сторінки детального перегляду проєкту для мобільних пристроїв наведено на рисунку 3.10.

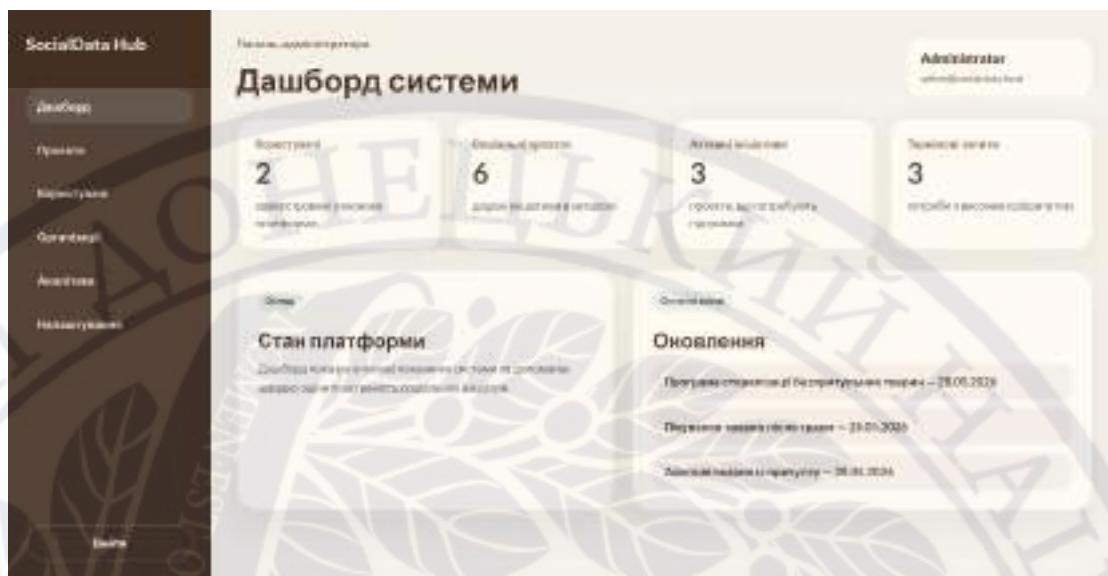


Рисунок 3.9 – Адмін-сторінка

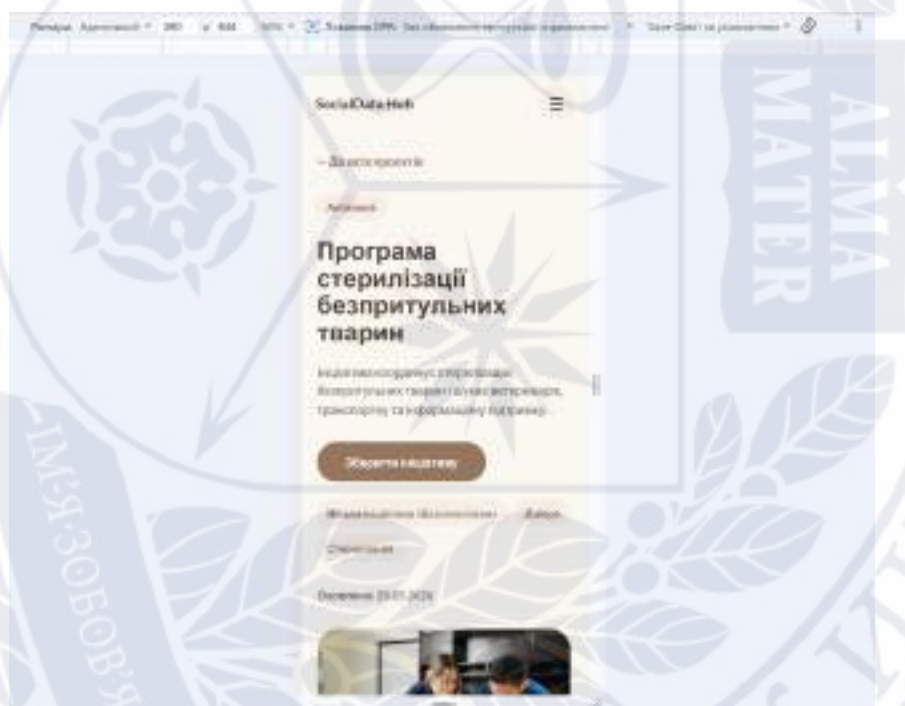


Рисунок 3.10 – Адаптивне відображення сторінки соціального проєкту на мобільному пристрої

Таким чином, реалізована клієнтська частина вебсистеми забезпечує зручну взаємодію користувачів із платформою, підтримує адаптивне

відображення інтерфейсу та надає доступ до основного функціоналу системи незалежно від типу пристрою.

3.3. Реалізація серверної частини вебсистеми

Серверна частина вебсистеми SocialData Hub реалізована із використанням платформи Node.js та фреймворку Express.js. Основним призначенням серверної частини є обробка запитів клієнтського застосунку, виконання бізнес-логіки системи, взаємодія з базою даних та забезпечення безпечного доступу до інформаційних ресурсів.

Архітектура серверної частини побудована відповідно до принципу розділення відповідальності між окремими компонентами. Для обробки HTTP-запитів використовуються маршрути (Routes), які передають керування відповідним контролерам (Controllers). Контролери виконують перевірку отриманих даних та взаємодіють із сервісами й моделями даних.

Структуру серверної частини вебсистеми наведено на рисунку 3.11.

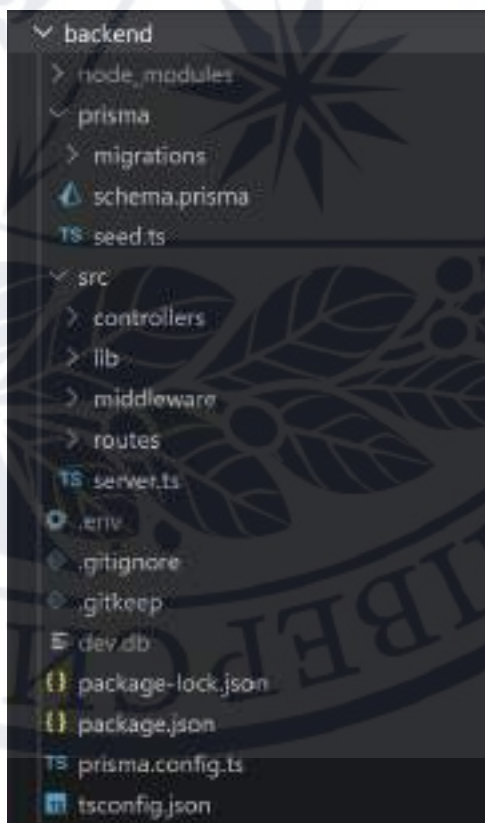


Рисунок 3.11 – Структура серверної частини вебсистеми

Основною точкою входу серверного застосунку є файл `server.ts`, який виконує ініціалізацію вебсервера, підключення необхідних програмних компонентів та реєстрацію маршрутів API.

Для реалізації механізмів перевірки доступу та попередньої обробки запитів використано каталог `middleware`. Проміжне програмне забезпечення забезпечує перевірку прав доступу користувачів, обробку службових запитів та підготовку даних до подальшого опрацювання контролерами.

Для взаємодії з базою даних використано ORM Prisma. Структура даних описується у файлі `schema.prisma`, який містить моделі основних сутностей вебсистеми. Управління структурою бази даних здійснюється за допомогою механізму міграцій, що дозволяє контролювати зміни схеми даних під час розробки та супроводу програмного забезпечення.

Початкове наповнення бази даних демонстраційними даними реалізовано за допомогою сценарію `seed.ts`, що забезпечує швидке розгортання системи для тестування та демонстрації її функціональних можливостей.

Для обміну даними між клієнтською та серверною частинами використано архітектурний підхід REST API. Передача інформації здійснюється у форматі JSON, що забезпечує просту інтеграцію між компонентами системи та підтримується більшістю сучасних вебтехнологій.

У серверній частині реалізовано набір програмних інтерфейсів для роботи з проєктами, організаціями, користувачами та аналітичними даними. Кожен інтерфейс забезпечує виконання операцій створення, перегляду, редагування та видалення інформації відповідно до прав доступу користувача.

Для захисту персональних даних та розмежування доступу до функціональних можливостей системи реалізовано механізм автентифікації та авторизації користувачів [50]. Після успішного входу користувач отримує токен доступу, який використовується під час подальших запитів до серверної частини системи.

Під час обробки запитів сервер також виконує перевірку коректності вхідних даних та формує стандартизовані відповіді для клієнтської частини. Це дозволяє забезпечити стабільну роботу вебсистеми та зменшити ймовірність виникнення помилок під час взаємодії користувачів із платформою.

Реалізована серверна частина забезпечує обробку бізнес-логіки вебсистеми, взаємодію з базою даних, керування доступом користувачів та підтримку основних функціональних можливостей платформи SocialData Hub.

3.4. Реалізація структури бази даних

Для зберігання та обробки інформації у вебсистемі SocialData Hub використано реляційну базу даних PostgreSQL. Проектування структури даних виконувалося з урахуванням функціональних вимог системи та необхідності забезпечення цілісності зв'язків між окремими сутностями.

Для роботи з базою даних використано ORM Prisma, яка забезпечує опис структури даних засобами мови TypeScript, автоматичне формування SQL-запитів та підтримку механізму міграцій. Схему бази даних вебсистеми наведено на рисунку 3.12.

Основною сутністю системи є таблиця користувачів (User), яка містить інформацію про зареєстрованих користувачів платформи. Для кожного користувача зберігаються ім'я, адреса електронної пошти, пароль у зашифрованому вигляді, роль користувача та службові поля дати створення і оновлення запису.

Соціальні ініціативи зберігаються у таблиці Project. Для кожного проєкту визначаються назва, опис, контактна інформація, організація-власник, категорія, статус, місце реалізації, способи підтримки та перелік актуальних потреб. Кожен проєкт пов'язаний із конкретною організацією, категорією та статусом через зовнішні ключі.

Інформація про організації, що реалізують соціальні проєкти, зберігається у таблиці Organization. Для організацій передбачено зберігання назви, опису діяльності, вебсайту та адреси розташування.

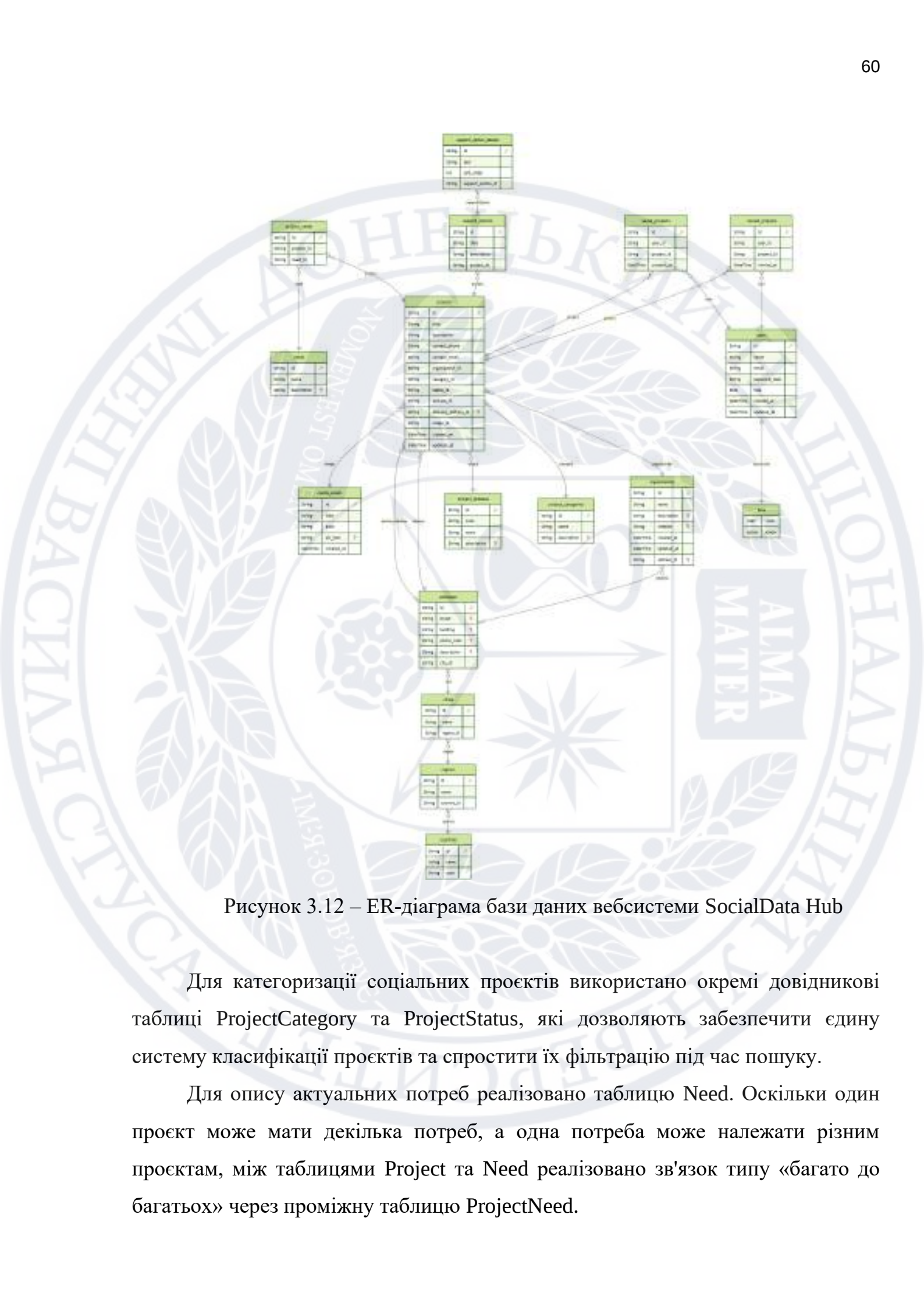


Рисунок 3.12 – ER-діаграма бази даних вебсистеми SocialData Hub

Для категоризації соціальних проєктів використано окремі довідникові таблиці `ProjectCategory` та `ProjectStatus`, які дозволяють забезпечити єдину систему класифікації проєктів та спростити їх фільтрацію під час пошуку.

Для опису актуальних потреб реалізовано таблицю Need. Оскільки один проєкт може мати декілька потреб, а одна потреба може належати різним проєктам, між таблицями Project та Need реалізовано зв'язок типу «багато до багатьох» через проміжну таблицю ProjectNeed.

Для підтримки різних способів допомоги проєктам реалізовано таблиці SupportOption та SupportOptionDetail. Така структура дозволяє зберігати декілька варіантів підтримки для одного проєкту та деталізувати кожен із них окремими інструкціями або описами.

У системі також реалізовано механізми персоналізації взаємодії користувачів із платформою. Для цього використовуються таблиці SavedProject та ViewedProject, які забезпечують збереження обраних проєктів користувачами та фіксацію історії їх перегляду.

Для зберігання географічної інформації реалізовано ієрархічну структуру таблиць Country, Region, City та Address. Такий підхід забезпечує нормалізацію даних та дозволяє уникнути дублювання інформації про місце розташування організацій і соціальних проєктів.

Реалізація зв'язків між сутностями:

У структурі бази даних реалізовано зв'язки типів «один до багатьох» та «багато до багатьох».

Зв'язок «один до багатьох» використовується між сутностями Organization та Project, оскільки одна організація може реалізовувати декілька соціальних проєктів.

Для реалізації зв'язків «багато до багатьох» використано проміжні таблиці. Зокрема, зв'язок між сутностями Project та Need реалізовано через таблицю ProjectNeed. Такий підхід забезпечує гнучке керування потребами проєктів та відповідає принципам нормалізації реляційних баз даних.

Усі зв'язки реалізовано за допомогою зовнішніх ключів, що дозволяє підтримувати цілісність даних та запобігати виникненню некоректних записів у базі даних.

Реалізована структура бази даних забезпечує надійне зберігання інформації про користувачів, організації, соціальні проєкти та їх потреби, підтримує механізми персоналізації взаємодії з платформою та створює основу для подальшого розвитку аналітичних можливостей вебсистеми.

3.5. Реалізація функціональних модулів системи

Під час розробки вебсистеми SocialData Hub було реалізовано набір функціональних модулів, які забезпечують збір, структурування, перегляд та аналіз інформації про соціальні проєкти. Архітектура системи побудована за модульним принципом, що дозволяє ізолювати окремі функції платформи, спростити супровід програмного забезпечення та забезпечити можливість подальшого розширення функціональності [51].

Оснoву системи становить каталог соціальних проєктів, який надає користувачам доступ до актуальних ініціатив, інформації про потреби організацій та способів надання допомоги. Для адміністрування контенту реалізовано окрему адміністративну панель, яка забезпечує керування проєктами, організаціями та користувачами. Крім цього, система містить аналітичний модуль для узагальнення зібраних даних та модуль профілю користувача для персоналізації роботи з платформою.

У межах реалізації вебсистеми були розроблені такі функціональні модулі:

- модуль роботи з соціальними проєктами;
- модуль управління організаціями;
- модуль управління користувачами;
- аналітичний модуль;
- модуль автентифікації та профілю користувача.

Кожен із зазначених модулів реалізує окремий набір функцій та взаємодіє з централізованою базою даних через програмний інтерфейс прикладного рівня (API), забезпечуючи цілісність і актуальність інформації в системі.

3.5.1. Модуль роботи з соціальними проєктами

Модуль роботи з соціальними проєктами є основним функціональним компонентом вебсистеми SocialData Hub, оскільки саме через нього здійснюється взаємодія користувачів із соціальними ініціативами, розміщеними на платформі. Його призначення полягає у зберіганні, відображенні, пошуку та адмініструванні інформації про проєкти, що потребують підтримки.

У межах реалізованого модуля користувач може переглядати каталог соціальних проєктів, отримувати детальну інформацію про кожну ініціативу, ознайомлюватися з актуальними потребами організацій та способами надання допомоги. Для спрощення пошуку необхідної інформації реалізовано механізми фільтрації проєктів за статусом («Активний», «Терміновий», «Завершений»), а також пошук за назвою проєкту, організації або місцем розташування.

На серверному рівні модуль реалізовано із використанням архітектури REST API. Для роботи з даними створено окремі маршрути, які забезпечують отримання списку проєктів, перегляд детальної інформації, створення нових записів, редагування наявних даних та видалення проєктів. Взаємодія з базою даних здійснюється за допомогою ORM Prisma, що забезпечує зручний доступ до сутностей та підтримує цілісність даних.

Структура проєкту включає основну інформацію про ініціативу, категорію, статус, організацію-власницю, контактні дані, перелік актуальних потреб та додаткову інформацію щодо способів підтримки. Такий підхід дозволяє забезпечити повноцінне представлення соціальної ініціативи та надати користувачеві всю необхідну інформацію для прийняття рішення щодо допомоги.

Реалізований модуль виступає центральною частиною вебсистеми та забезпечує основу для роботи інших функціональних компонентів платформи.

3.5.2. Модуль управління організаціями

Модуль управління організаціями призначений для зберігання та адміністрування інформації про установи, що реалізують соціальні проєкти в межах платформи SocialData Hub. Використання окремої сутності «Організація» дозволяє забезпечити структуроване представлення даних, уникнути дублювання інформації та підтримувати зв'язок між організаціями та соціальними ініціативами.

У межах реалізованого модуля забезпечується створення, перегляд, редагування та видалення інформації про організації. Для кожної організації в

системі зберігаються назва, опис діяльності, контактні дані, вебсайт та інформація про місцезнаходження. Також передбачено можливість прив'язки декількох соціальних проєктів до однієї організації, що відповідає особливостям діяльності благодійних фондів, притулків для тварин та громадських об'єднань.

Функціональність модуля реалізована через окремий набір REST API маршрутів, які забезпечують взаємодію клієнтської частини із сервером та виконання операцій над даними організацій. Для роботи з інформацією використовується ORM Prisma, що дозволяє підтримувати цілісність зв'язків між організаціями, проєктами та адресними даними.

Особливістю реалізації є використання централізованого зберігання інформації про організації. Такий підхід спрощує адміністрування системи, забезпечує актуальність даних та дозволяє повторно використовувати інформацію про організацію для різних соціальних проєктів без необхідності дублювання записів.

Реалізований модуль забезпечує основу для структурованого представлення соціальних ініціатив на платформі та підтримує взаємозв'язок між проєктами й організаціями, що їх реалізують.

3.5.3. Модуль автентифікації та профілю користувача

Для забезпечення персоналізованої взаємодії користувачів із вебсистемою SocialData Hub реалізовано модуль автентифікації та профілю користувача. Його основним призначенням є реєстрація нових користувачів, автентифікація під час входу до системи, керування персональними даними та забезпечення доступу до функцій платформи відповідно до ролі користувача.

Модуль передбачає можливість створення нового облікового запису за допомогою форми реєстрації, у якій користувач вказує основні дані для подальшої роботи із системою. Після успішної реєстрації користувач може виконати вхід до системи через форму авторизації, використовуючи власні облікові дані.

Для захисту інформаційних ресурсів реалізовано механізм автентифікації на основі токенів доступу [52]. Після успішної перевірки облікових даних сервер формує токен, який використовується для підтвердження особи користувача під час подальших запитів до API. Такий підхід дозволяє забезпечити безпечну взаємодію між клієнтською та серверною частинами системи.

Після входу до системи користувач отримує доступ до персонального профілю. У профілі відображається основна інформація про обліковий запис, а також реалізовано можливість перегляду збережених соціальних проєктів. Наявність такого функціоналу дозволяє користувачам формувати власні добірки ініціатив та швидко повертатися до проєктів, які викликали зацікавленість.

Для розмежування доступу до функціональних можливостей системи використовується рольова модель. У межах реалізованої вебсистеми передбачено ролі звичайного користувача та адміністратора. Адміністратори мають доступ до функцій керування контентом, організаціями та користувачами, тоді як звичайні користувачі працюють лише з функціоналом перегляду та взаємодії із соціальними проєктами.

Реалізація модуля автентифікації та профілю користувача забезпечує безпечний доступ до ресурсів вебсистеми, підтримує персоналізацію взаємодії з платформою та створює основу для реалізації механізмів керування правами доступу.

3.5.4. Адміністративний модуль

Для забезпечення актуальності інформації та централізованого керування даними у вебсистемі SocialData Hub реалізовано адміністративний модуль. Його основним призначенням є надання адміністраторам інструментів для керування соціальними проєктами, організаціями, користувачами та довідковою інформацією системи.

Доступ до адміністративного модуля мають лише користувачі з відповідною роллю. Після успішної автентифікації адміністратор отримує

доступ до спеціалізованої панелі керування, яка об'єднує основні засоби адміністрування платформи в єдиному інтерфейсі.

У межах адміністративного модуля реалізовано функції створення, редагування та видалення соціальних проєктів. Адміністратор може керувати інформацією про потреби проєктів, змінювати їх статус, оновлювати контактні дані та підтримувати актуальність інформації, що відображається користувачам платформи.

Окремий функціональний блок модуля призначений для управління організаціями. Адміністратор має можливість створювати нові записи про організації, редагувати їхні дані та встановлювати зв'язки між організаціями та соціальними проєктами. Такий підхід забезпечує централізоване зберігання інформації та спрощує супровід системи.

Також реалізовано засоби управління користувачами. Адміністратор може переглядати список зареєстрованих користувачів, отримувати інформацію про їхні ролі та виконувати необхідні адміністративні операції відповідно до встановлених правил доступу.

Робота адміністративного модуля базується на взаємодії клієнтської частини із серверним API та базою даних. Для виконання операцій використовуються захищені маршрути, доступ до яких обмежується механізмами авторизації. Це дозволяє запобігати несанкціонованому доступу до адміністративних функцій та забезпечує безпеку інформаційних ресурсів системи.

Адміністративний модуль забезпечує централізоване керування даними вебсистеми, підтримує актуальність інформації та створює необхідні умови для стабільного функціонування платформи SocialData Hub.

3.5.5. Аналітичний модуль

Одним із ключових завдань вебсистеми SocialData Hub є не лише зберігання інформації про соціальні проєкти, а й забезпечення можливості її узагальнення та аналізу. Для реалізації цієї функціональності у складі системи

розроблено аналітичний модуль, який забезпечує формування статистичної інформації на основі накопичених даних.

Основним призначенням модуля є отримання узагальнених показників щодо діяльності соціальних проєктів та організацій, представлених на платформі. Аналітична інформація використовується для оцінювання наповнення системи, аналізу структури соціальних ініціатив та підтримки прийняття рішень під час адміністрування інформаційних ресурсів.

Реалізований модуль виконує автоматичний збір статистичних показників із бази даних та формує інформаційні блоки для подальшого відображення у вебінтерфейсі. Зокрема, здійснюється підрахунок загальної кількості соціальних проєктів, організацій, категорій та зареєстрованих користувачів. Додатково формується інформація щодо розподілу проєктів за категоріями діяльності та статусами реалізації.

Для отримання статистичних даних використовується серверна частина системи, яка виконує необхідні запити до бази даних PostgreSQL та повертає узагальнені результати через REST API. Після цього отримані дані передаються до клієнтської частини та відображаються у вигляді інформаційних карток і статистичних показників.

Використання аналітичного модуля дозволяє швидко отримувати актуальну інформацію про стан наповнення платформи, оцінювати активність соціальних ініціатив та виявляти найбільш представлені напрями діяльності. Це підвищує інформативність вебсистеми та розширює можливості використання накопичених даних.

Реалізований аналітичний модуль забезпечує базові засоби аналізу та візуалізації інформації, що відповідає поставленій меті розробки вебсистеми для агрегації та аналізу даних соціальних проєктів.

3.6. Тестування та результати роботи системи

Після завершення реалізації вебсистеми SocialData Hub було проведено тестування основних функціональних модулів. Метою тестування була

перевірка коректності роботи клієнтської та серверної частин системи, оцінка стабільності взаємодії між компонентами та підтвердження відповідності реалізованого функціоналу встановленим вимогам [53].

У процесі тестування перевірялися функції реєстрації та авторизації користувачів, робота каталогу соціальних проєктів, механізми пошуку та фільтрації, операції створення, редагування та видалення даних, робота адміністративної панелі, а також адаптивність інтерфейсу на різних типах пристроїв.

Таблиця 3.1 – Результати функціонального тестування вебсистеми

№	Перевірка	Очікуваний результат	Результат
1	Реєстрація користувача	Створення нового облікового запису	Успішно
2	Авторизація користувача	Вхід до системи з валідними даними	Успішно
3	Пошук проєктів	Відображення релевантних результатів	Успішно
4	Фільтрація проєктів	Коректне відображення записів відповідно до фільтра	Успішно
5	Перегляд сторінки проєкту	Відображення повної інформації	Успішно
6	Створення проєкту	Додавання нового запису до бази даних	Успішно
7	Редагування проєкту	Збереження внесених змін	Успішно
8	Видалення проєкту	Коректне видалення запису	Успішно
9	Створення організації	Додавання організації до системи	Успішно
10	Управління користувачами	Виконання адміністративних операцій	Успішно
11	Відображення статистики	Коректний розрахунок показників	Успішно
12	Адаптація для мобільних пристроїв	Коректне відображення інтерфейсу	Успішно

Під час проведення тестування критичних помилок, які б унеможливлювали використання вебсистеми, виявлено не було. Усі основні функціональні модулі успішно виконали покладені на них завдання та продемонстрували коректну взаємодію між клієнтською частиною, серверним API та базою даних [54].

Особлива увага приділялася перевірці адаптивності інтерфейсу. Тестування виконувалося для настільних комп'ютерів, планшетів та мобільних пристроїв різних розмірів екранів. Результати тестування підтвердили коректне

відображення інтерфейсу та збереження функціональності системи незалежно від типу пристрою.

За результатами тестування підтверджено працездатність реалізованої вебсистеми та відповідність функціональних можливостей вимогам, визначеним на етапі проектування.

3.7. Перспективи розвитку вебсистеми

Розроблена вебсистема SocialData Hub реалізує основний набір функцій для агрегації, зберігання та аналізу даних соціальних проєктів. Водночас архітектура системи та використані технології створюють можливості для її подальшого розвитку, розширення функціональності та підвищення ефективності використання.

Одним із перспективних напрямів розвитку є впровадження рекомендаційного механізму для персоналізації взаємодії користувачів із платформою [55]. Такий механізм може формувати індивідуальні добірки соціальних проєктів на основі історії переглядів, збережених ініціатив, категорій інтересів користувача та його попередньої активності. Це дозволить спростити пошук релевантних проєктів та підвищити залученість користувачів до соціальних ініціатив.

Іншим напрямом розвитку може стати розширення аналітичного модуля. Зокрема, доцільним є впровадження інтерактивних інформаційних панелей із графіками та діаграмами, які дозволять аналізувати динаміку розвитку соціальних проєктів, структуру потреб організацій та активність користувачів у різних категоріях діяльності.

Перспективним також є впровадження автоматизованої агрегації даних із зовнішніх джерел. Використання API соціальних платформ, благодійних сервісів та відкритих інформаційних ресурсів дозволить автоматично оновлювати інформацію про проєкти та зменшити обсяг ручного адміністрування системи.

Для підвищення зручності використання платформи можливим є розширення функціоналу профілю користувача. Зокрема, можуть бути

реалізовані механізми сповіщень про оновлення обраних проєктів, відстеження історії підтримки соціальних ініціатив та налаштування персональних рекомендацій.

Окремим напрямом розвитку може стати створення мобільного застосунку на основі наявної вебсистеми. Використання сучасних кросплатформних технологій дозволить забезпечити доступ до функціоналу платформи з мобільних пристроїв та підвищити доступність соціальних проєктів для ширшого кола користувачів.

Запропонована вебсистема має значний потенціал для подальшого розвитку та може бути розширена новими функціональними можливостями, спрямованими на підвищення ефективності взаємодії користувачів із соціальними проєктами та покращення процесів аналізу й обробки інформації.

Висновок до розділу 3

У третьому розділі бакалаврської роботи було розглянуто процес реалізації вебсистеми SocialData Hub для агрегації та аналізу даних соціальних проєктів. На основі сформованих вимог та розроблених проєктних рішень виконано програмну реалізацію клієнтської та серверної частин системи, а також створено структуру бази даних для зберігання інформаційних ресурсів платформи.

Для розробки вебсистеми використано сучасний стек технологій, до складу якого увійшли React, TypeScript, Node.js, Express, PostgreSQL та Prisma ORM. Застосування зазначених технологій дозволило реалізувати масштабовану архітектуру та забезпечити ефективну взаємодію між окремими компонентами програмного забезпечення.

У межах роботи було реалізовано основні функціональні модулі системи, зокрема модуль роботи із соціальними проєктами, модуль управління організаціями, модуль автентифікації та профілю користувача, адміністративний модуль та аналітичний модуль. Реалізований функціонал забезпечує зберігання, пошук, фільтрацію, адміністрування та аналіз інформації про соціальні ініціативи.

Проведене тестування підтвердило коректність роботи реалізованих компонентів, стабільність взаємодії між клієнтською та серверною частинами системи, а також працездатність вебсистеми на різних типах пристроїв. Результати тестування засвідчили відповідність реалізованого програмного забезпечення поставленим функціональним і нефункціональним вимогам.

Отримані результати свідчать про успішне досягнення мети роботи та створення працездатної вебсистеми для агрегації та аналізу даних соціальних проєктів, яка може використовуватися як основа для подальшого розвитку та розширення функціональних можливостей платформи.

ВИСНОВКИ

У бакалаврській роботі розглянуто актуальну задачу розробки вебсистеми для агрегації та аналізу даних соціальних проєктів. Актуальність дослідження зумовлена зростанням кількості соціальних ініціатив, діяльність яких пов'язана з волонтерською допомогою, благодійністю та підтримкою тварин, а також відсутністю єдиного інформаційного середовища для централізованого представлення та аналізу таких проєктів.

У процесі виконання роботи було проведено аналіз предметної області, досліджено особливості функціонування соціальних проєктів, діяльності благодійних організацій та існуючих платформ соціальної підтримки. Виконаний аналіз дозволив визначити основні проблеми організації та представлення інформації, а також сформулювати вимоги до майбутньої вебсистеми.

На основі результатів дослідження було виконано проєктування вебсистеми SocialData Hub. Визначено функціональні та нефункціональні вимоги, розроблено архітектуру програмного забезпечення, структуру бази даних та модель взаємодії між основними компонентами системи. Для реалізації програмного продукту обрано сучасний стек вебтехнологій, до складу якого увійшли React, TypeScript, Node.js, Express, PostgreSQL та Prisma ORM.

У межах роботи реалізовано вебсистему, що забезпечує централізоване зберігання, пошук, фільтрацію та аналіз інформації про соціальні проєкти. Створено функціональні модулі роботи із соціальними проєктами, організаціями, користувачами та адміністративними ресурсами системи. Реалізовано механізми авторизації, управління даними та формування аналітичної інформації на основі накопичених відомостей.

Проведене тестування підтвердило коректність роботи розробленого програмного забезпечення, працездатність основних функціональних модулів та стабільну взаємодію між клієнтською частиною, серверним API та базою даних. Також підтверджено коректне відображення інтерфейсу на персональних комп'ютерах, планшетах та мобільних пристроях.

У результаті виконання бакалаврської роботи досягнуто поставленої мети та створено працездатну вебсистему для агрегації та аналізу даних соціальних проєктів. Розроблена система може використовуватися як інформаційна платформа для представлення соціальних ініціатив, а також слугувати основою для подальшого розвитку аналітичних інструментів, рекомендаційних механізмів та засобів автоматизованої агрегації даних із зовнішніх джерел.



СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Організація Об'єднаних Націй в Україні. Спільний аналіз країни для України (Common Country Analysis 2023). Офіційний звіт. URL: <https://ukraine.un.org/uk/node/256565> (дата звернення: 28.05.2026).
2. Безпалько О. В. Соціальне проектування : навчальний посібник. Київ : Київський університет імені Бориса Грінченка, 2010. 127 с. URL: https://elibrary.kubg.edu.ua/id/eprint/2615/1/O_Bezpalko_SP_IL_KUBG.pdf (дата звернення: 28.05.2026).
3. Anheier H. K., Toepler S. Nonprofit Organizations: Theory, Management, Policy. 3rd ed. London: Routledge, 2023. 582 p. URL: <https://www.routledge.com/Nonprofit-Organizations-Theory-Management-Policy/Anheier-Toepler/p/book/9781138625495> (дата звернення: 28.05.2026).
4. Офіційний сайт гуманістичного руху UAnimals: порятунок та захист тварин в Україні. URL: <https://uanimals.org> (дата звернення: 28.05.2026).
5. Project Management Institute. A Guide to the Project Management Body of Knowledge (PMBOK Guide). 7th ed. Newtown Square: Project Management Institute, 2021. 250 p.
6. United Nations. E-Government Survey 2024: Accelerating Digital Transformation for Sustainable Development. New York : United Nations, 2024. URL: <https://desapublications.un.org/publications/un-e-government-survey-2024> (дата звернення: 28.05.2026).
7. Castells M. The Networks of Outrage and Hope: Social Movements in the Internet Age. Cambridge: Polity Press, 2015. 328 p.
8. Turban E., Pollard C., Wood G. Information Technology for Management: Driving Digital Transformation to Increase Local and Global Performance, Growth and Sustainability. 12th ed. Hoboken: John Wiley & Sons, 2021. 640 p. URL: <https://www.wiley.com/en-us/shop/general-introductory-computer-science/information-technology-for-management-driving-digital->

- [transformation-to-increase-local-and-global-performance-growth-and-sustainability-12th-edition-p-9781119702900](#) (дата звернення: 28.05.2026).
9. Пунченко О. П., Лазаревич А. А. Інформатизація як засіб репрезентації інформаційних ресурсів суспільства. Гуманітарний вісник Запорізької державної інженерної академії. 2015. Вип. 63. С. 138–147. URL: <http://vestnikzgia.com.ua/article/view/57498> (дата звернення: 01.06.2026).
 10. Ситнік Б. Т. Основи інформаційних систем і технологій: навчальний посібник. Харків : УкрДУЗТ, 2019. 184 с. URL: <https://files.znu.edu.ua/files/Bibliobooks/Inshi78/0058425.pdf> (дата звернення: 01.06.2026).
 11. Геряк Ю., Берко А. Система критеріїв оцінки якості даних у розподілених інформаційних системах. *Information Systems and Networks*. 2024. Issue 16. Р. 193–204. URL: https://science.lpnu.ua/sites/default/files/journal-paper/2024/nov/36467/241164maket4-193-204_0.pdf (дата звернення: 01.06.2026).
 12. Doan A., Halevy A., Ives Z. Principles of Data Integration. Waltham : Morgan Kaufmann, 2012. 520 p.
 13. Шаховська Н. Б. Організація просторів даних у складних інформаційних системах : автореф. дис. ... доктора техн. наук : 05.13.06 / Нац. ун-т «Львівська політехніка». Львів, 2012. 39 с. URL: <https://ena.lpnu.ua/items/c27ad993-d343-4a98-8f37-57403f543ff2> (дата звернення: 03.06.2026).
 14. Kimball R., Ross M. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling. 3rd ed. Indianapolis : John Wiley & Sons, 2013. 576 p.
 15. Batini C., Scannapieco M. Data and Information Quality: Dimensions, Principles and Techniques. Cham : Springer, 2016. 431 p. URL: <https://link.springer.com/book/10.1007/978-3-319-24106-7> (дата звернення: 01.06.2026).
 16. Шаховська Н. Б., Пасічник В. В. Сховища та простори даних : монографія. Львів : Видавництво Національного університету "Львівська політехніка",

2009. 244 с.
- 17.Provost F., Fawcett T. Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking. Sebastopol : O'Reilly Media, 2013. 414 p.
- 18.Єріна А. М., Пальян З. О. Статистика : підручник. Київ: КНЕУ, 2010. 352 с.
- 19.Few S. Information Dashboard Design: Displaying Data for At-a-Glance Monitoring. 2nd ed. Burlingame : Analytics Press, 2013. 246 p.
- 20.Cairo A. The Functional Art: An Introduction to Information Graphics and Visualization. Berkeley : New Riders, 2013. 384 p. URL: <https://www.peachpit.com/store/functional-art-introduction-to-information-graphics-9780133041187> (дата звернення: 01.06.2026).
- 21.Козак В. І., Готович В. А. Дослідження варіантів проектування інтерфейсу користувача в інформаційних інтерактивних аналітичних панелях. Актуальні задачі сучасних технологій : матеріали XII Міжн. наук.-практ. конф. молодих учених та студентів (Тернопіль, 6–7 грудня 2023 р.). Тернопіль: ТНТУ, 2023.
- 22.Shneiderman B., Plaisant C., Cohen M., Jacobs S. Designing the User Interface: Strategies for Effective Human-Computer Interaction. 6th ed. Boston : Pearson, 2021. 622 p.
- 23.Smith A. Digital Charity and Crowdfunding: The Evolution of Online Giving Platforms. *International Journal of Nonprofit and Voluntary Sector Marketing*. 2023. Vol. 28, No. 1. e1762.
- 24.GoFundMe: Online Fundraising Platform. Official Website. URL: <https://www.gofundme.com> (дата звернення: 02.06.2026).
- 25.GlobalGiving: Non-profit Crowdfunding Platform. Annual Review and Reports. URL: <https://www.globalgiving.org> (дата звернення: 02.06.2026).
- 26.Благодійна платформа dobro.ua. Звіти про діяльність фонду. URL: <https://dobro.ua> (дата звернення: 02.06.2026).

27. Petfinder: Adopt a Pet, Search for Dogs or Cats. Animal Welfare Network. URL: <https://www.petfinder.com> (дата звернення: 02.06.2026).
28. Benevity Causes Portal: Corporate Purpose & Goodness Platform. URL: <https://benevity.com/nonprofits> (дата звернення: 02.06.2026).
29. Ахновська І., Левченко О. Розвиток краудфандингу для фінансування малого бізнесу в умовах інформаційної економіки. Підприємництво та інновації. 2021. Вип. 16. С. 76–82. DOI: <https://doi.org/10.37320/2415-3583/16.13> (дата звернення: 02.06.2026).
30. Гейко І. В. Дослідження принципів проектування вебзастосунків на прикладі розробки електронної дошки для дистанційного навчання : кваліфікаційна робота бакалавра / І. В. Гейко ; наук. кер. В. В. Булатецький ; Волинський національний університет імені Лесі Українки. Луцьк, 2025. 58 с.
31. React: A JavaScript library for building user interfaces. Official Documentation. URL: <https://react.dev> (дата звернення: 02.06.2026).
32. Banks A., Porcello E. Learning React: Modern Patterns for Developing React Apps. 2nd ed. Sebastopol : O'Reilly Media, 2020. 310 p.
33. Node.js Tools and Documentation. Node.js Foundation. URL: <https://nodejs.org> (дата звернення: 02.06.2026).
34. Бабенко О. В. Метод багатокритеріального аналізу для вибору SQL або NoSQL баз даних залежно від умов проєкту : кваліфікаційна робота бакалавра / О. В. Бабенко ; наук. керівник Д. І. Загородня ; Західноукраїнський національний університет. Тернопіль : ЗУНУ, 2024. 70 с. URL: <https://dspace.wunu.edu.ua/items/7abb6aae-b7b8-4ca1-8115-3a9dd747d300> (дата звернення: 02.06.2026).
35. Prisma ORM: Next-generation Node.js and TypeScript ORM. Official Reference. URL: <https://www.prisma.io> (дата звернення: 02.06.2026).
36. Fielding R. T. Architectural Styles and the Design of Network-based Software Architectures: Doctoral dissertation. Irvine : University of California, 2000. 162 p.

37. Fielding R., Nottingham M., Reschke J. HTTP Semantics. RFC 9110. Internet Engineering Task Force (IETF), 2022. URL: <https://datatracker.ietf.org/doc/html/rfc9110> (дата звернення: 02.06.2026).
38. Застосування системного підходу при виконанні ІТ-проектів / О. А. Журан, Є. Ю. Беркунський, О. В. Гайдаєнко, Г. С. Морозова, А. Ю. Павленко. Електротехнічні та комп'ютерні системи. 2024. № 40 (116). DOI: <https://doi.org/10.15276/eltecs.40.116.2024.4> (дата звернення: 02.06.2026).
39. Wiegers K., Beatty J. Software Requirements. 3rd ed. Redmond : Microsoft Press, 2013. 637 p.
40. Miles R., Hamilton K. Learning UML 2.0. Sebastopol : O'Reilly Media, 2006. 269 p.
41. Bass L., Clements P., Kazman R. Software Architecture in Practice. 4th ed. Boston : Addison-Wesley, 2021. 464 p. URL: <https://www.oreilly.com/library/view/software-architecture-in/9780136885979/> (дата звернення: 02.06.2026).
42. Петрик М. Р., Мудрик Ф. Я. Архітектура програмного забезпечення (на базі використання CASE-засобів IBM Rational) : навчальний посібник. Тернопіль : ТНТУ імені Івана Пулюя, 2017. 104 с.
43. Fowler M. Patterns of Enterprise Application Architecture. Boston: Addison-Wesley, 2003. 533 p.
44. Date C. J. An Introduction to Database Systems. 8th ed. Boston : Pearson, 2004. 1024 p.
45. Основи баз даних : навчальний посібник / С. Л. Рзаєва, І. В. Машкіна, П. М. Складанний та ін. Київ : КУБГ, 2025. 248 с. URL: <https://elibrary.kubg.edu.ua/id/eprint/54741> (дата звернення: 02.06.2026).
46. Garcia-Molina H., Ullman J. D., Widom J. Database Systems: The Complete Book. 2nd ed. Upper Saddle River : Pearson, 2012. 1216 p.
47. Зінов'єва О. Г. Використання CASE-засобів для проектування інформаційних систем. Українські студії в європейському контексті. 2023.

- № 7. С. 220–227. URL: <https://dspace.elgort.kharkov.ua> (дата звернення: 02.06.2026).
48. Bierman G., Abadi M., Torgersen M. Understanding TypeScript. *European Conference on Object-Oriented Programming*. Springer, Cham, 2014. P. 257–281.
49. Chacon S., Straub B. Pro Git. 2nd ed. Berkeley : Apress, 2014. 456 p. URL: <https://git-scm.com/book/en/v2> (дата звернення: 02.06.2026).
50. OWASP Top 10 Reference. Open Web Application Security Project. URL: <https://owasp.org/www-project-top-ten/> (дата звернення: 02.06.2026).
51. Martin R. C. Clean Architecture: A Craftsman's Guide to Software Structure and Design. Boston : Prentice Hall, 2018. 404 p.
52. Jones M., Bradley J., Sakimura N. JSON Web Token (JWT). RFC 7519. Internet Engineering Task Force (IETF), 2015. URL: <https://datatracker.ietf.org/doc/html/rfc7519> (дата звернення: 02.06.2026).
53. Myers G. J., Sandler C., Badgett T. The Art of Software Testing. 3rd ed. Hoboken : John Wiley & Sons, 2012. 256 p.
54. Авраменко А. С., Авраменко В. С., Косенюк Г. В. Тестування програмного забезпечення : навчальний посібник. Черкаси : ЧНУ імені Богдана Хмельницького, 2017. 284 с. URL: <https://eprints.cdu.edu.ua/1482/1/testyvan.pdf> (дата звернення: 02.06.2026).
55. Євдокімов О. І. Рекомендаційна система персоналізації користувацького досвіду : кваліфікаційна робота на здобуття освітнього ступеня «магістр» : спец. 121 «Інженерія програмного забезпечення». Миколаїв : ЧНУ ім. Петра Могили, 2024. 95 с. URL: <https://krs.chmnu.edu.ua/jspui/handle/123456789/3758> (дата звернення: 02.06.2026).