

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій

«До захисту допущено»

В.о. завідувача кафедри

_____ Світлана ВАЩЕНКО

_____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня бакалавр

зі спеціальності 122 «Комп'ютерні науки»,
освітньо-професійної програми «Інформаційні технології проектування»
на тему: «Web-додаток підтримки волонтерської діяльності»

Здобувача (ки) групи ІТ-92-1/2 Сидоренко Валерії Олександрівни
(шифр групи) (прізвище, ім'я, по батькові)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

(підпис)

Валерія СИДОРЕНКО
(Ім'я та ПРІЗВИЩЕ здобувача)

Керівник

к. т. н., доц. Вікторія АНТИПЕНКО
(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ)

(підпис)

Суми – 2023

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

В. о. зав. кафедри ІТ

_____ Світлана ВАЩЕНКО
«__» _____ 2023 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Сидоренко Валерії Олександрівні

1. Тема роботи Web-додаток підтримки волонтерської діяльності

керівник роботи Антипенко Вікторія Петрівна, к.т.н., доцент,

затверджені наказом по університету від «29» травня 2023 р. №0588-VI

2. Строк подання студентом роботи «7» червня 2023 р.

3. Вхідні дані до роботи технічне завдання на розробку web-додатку підтримки волонтерської діяльності

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) аналіз предметної області, проектування web-додатку, розробка web-додатку

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) актуальність, постановка задачі, аналіз продуктів-аналогів, порівняння продуктів-аналогів, функціональні вимоги, функціональне моделювання з точки зору користувача, моделювання діаграми послідовності, моделювання використання web-додатку, засоби реалізації, реалізація, архітектура web-додатку, фізична модель даних, демонстрація роботи web-додатку, апробація, висновки

6. Консультанти розділів роботи:

| Розділ | Консультант | Підпис, дата | |
|--------|-------------|----------------|------------------|
| | | Завдання видав | Завдання прийняв |
| | | | |

7. Дата видачі завдання «8» лютого 2023 р**КАЛЕНДАРНИЙ ПЛАН**

| № п/п | Назва етапів кваліфікаційної роботи | Строк виконання етапів роботи | Примітка |
|-------|---------------------------------------|-------------------------------|----------|
| 1 | Оформлення планування робіт | 06.03.2023- 10.03.2023 | |
| 2 | Оформлення технічного завдання | 13.03.2023- 31.03.2023 | |
| 3 | Проведення аналізу предметної області | 03.04.2023- 14.04.2023 | |
| 4 | Проведення проектування web-додатку | 17.04.2023- 12.05.2023 | |
| 5 | Розробка та тестування web-додатку | 13.05.2023- 05.06.2022 | |
| 6 | Завантаження web-додатку на хостинг | 07.05.2023 | |
| 7 | Оформлення пояснювальної записки | 06.03.2023- 07.06.2023 | |

Студент

(підпис)

Валерія СИДОРЕНКО

Керівник роботи

(підпис)

к.т.н., доц. Вікторія
АНТИПЕНКО

РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Web-додаток підтримки волонтерської діяльності».

Пояснювальна записка складається зі вступу, трьох розділів, висновків, списку використаних джерел із 20 найменувань та чотирьох додатків. Загальний обсяг пояснювальної записки складає 74 сторінки, у тому числі 34 сторінок основного тексту, 3 сторінки списку використаних джерел, 32 сторінки додатків.

Кваліфікаційну роботу бакалавра присвячено розробці web-додатку підтримки волонтерської діяльності.

У першому розділі наведено огляд останніх публікації за волонтерською тематикою. Проаналізовано певні продукти-аналоги розроблюваного web-додатку, визначено їх переваги та недоліки. Також визначено мету та задачі проекту та визначаються засоби його реалізації.

У другій частині виконано структурно-функціональне моделювання, визначено варіанти використання web-додатку та спроектовано базу даних. У результаті були змодельовані такі діаграми: контекстна діаграма IDEF0 та її декомпозиція, діаграма послідовності, діаграма варіантів використання та логічна модель бази даних.

У третьому розділі описано архітектуру даного програмного продукту та процес його розробки. Продемонстровано використання web-додатку підтримки волонтерської діяльності. Коротко наведено результати тестування.

Ключові слова: WEB-ДОДАТОК, БАЗА ДАНИХ, ВОЛОНТЕРСТВО, LARAVEL, ОГолошення, ДОПОМОГА, ГРОШОВИЙ ЗБІР, РОЗРОБКА.

ЗМІСТ

| | |
|--|----|
| ВСТУП..... | 6 |
| 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ..... | 8 |
| 1.1 Огляд останніх досліджень і публікацій..... | 8 |
| 1.2 Аналіз програмних продуктів-аналогів | 9 |
| 1.3 Постановка задачі..... | 16 |
| 2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ | 18 |
| 2.1 Структурно-функціональне моделювання | 18 |
| 2.2 Проектування інформаційної системи | 21 |
| 2.3 Проектування моделі бази даних | 22 |
| 3 РОЗРОБКА WEB-ДОДАТКУ ПІДТРИМКИ ВОЛОНТЕРСЬКОЇ ДІЯЛЬНОСТІ | 24 |
| 3.1 Архітектура web-додатку | 24 |
| 3.2 Програмна реалізація | 25 |
| 3.3 Використання web-додатку | 29 |
| ВИСНОВКИ..... | 39 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 40 |
| ДОДАТОК А..... | 43 |
| ДОДАТОК Б..... | 53 |
| ДОДАТОК В | 65 |
| ДОДАТОК Г..... | 66 |

ВСТУП

Некомерційний web-додаток – це тип онлайн ресурсу, який не створений із метою отримання прибутку. Їх можуть розроблювати з різних причин, включаючи особисте самовираження, активізм, освіту, хобі тощо.

Однією з головних переваг створення некомерційного web-додатку є те, що він дозволяє окремим особам або організаціям ділитися інформацією та ідеями, не турбуючись про отримання прибутку. А сьогодні це особливо важливо для багатьох. Наприклад, для окремих осіб або організацій, які просувають справу чи повідомлення, у які вони вірять, оскільки можуть робити це без тиску з метою монетизації свого контенту.

Незалежно від методу створення некомерційного web-додатку, слід враховувати декілька ключових факторів. По-перше, важливо вибрати доменне ім'я. Воно повинне відповідати вмісту web-додатку та легко запам'ятовуватися. По-друге, дизайн і макет розробки мають бути зручними для користувачів і візуально привабливими. Нарешті, важливо регулярно оновлювати web-додаток свіжим вмістом, щоб підтримувати зацікавленість відвідувачів і викликати в них бажання повернутися.

Загалом, створення некомерційних web-додатків може бути корисним досвідом, який дозволяє окремим особам і організаціям ділитися власними точками зору, прагненнями та переживаннями з більшою аудиторією. Вони пропонують платформу для обміну інформацією та ідеями з іншими як для особистого самовираження, так і для активізму, освіченості тощо.

Особливу увагу в наш час варто приділити web-додаткам, які б допомагали людям, які постраждали під час війни. Річ не тільки про грошову допомогу, а й звичайне вирішення побутових потреб. Допомога армії України та громадянам країни

наразі є нагальним питанням, що і спонукає братися за розробки таких некомерційних web-додатків, які будуть поширювати волонтерство серед мас.

Тому метою даного проекту є створення web-додатку для підтримки волонтерської діяльності. Його використання забезпечить її належну організацію шляхом автоматизації процесу взаємодії волонтерів та людей, яким необхідна допомога.

Для досягнення поставленої мети треба виконати такі задачі:

- визначити актуальність дослідження та цільову аудиторію роботи;
- провести аналіз існуючих відповідних тематиці web-додатків-аналогів і виділити їх переваги та недоліки;
- обрати методи та технології для розробки web-додатку підтримки волонтерської діяльності;
- спроектувати модель та структуру даного web-додатку;
- створити прототип web-додатку підтримки волонтерської діяльності;
- розробити функціонал представленого web-додатку;
- виконати тестування web-додатку підтримки волонтерської діяльності.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень і публікацій

Із початком широкомасштабної війни в Україні волонтерство вийшло на новий рівень. Навіть стало буденною справою. Допомога армії, громадянам, які постраждали від бойових дій, наближає нашу перемогу та доводить, що українці – незламні.

У джерелі [1] приділено увагу волонтерській діяльності під час війни. Наголошено на тому, що сьогодні такі добровольці – потужний локомотив громадянського суспільства, який допомагає наближати перемогу України. Зазначається, що кожен охочий може долучитися та стати корисним, бо велика справа складається з маленьких вчинків.

Джерело [2] також зазначає, що згідно з рейтингом «World Giving Index» (WGI), опублікованим у 2022 році, Україна входить до десятки країн, які розвивають благодійність та має один із найвищих темпів зростання WGI – всесвітнього індексу благодійності за останні роки. Але також приділяється увага саме питанню організації волонтерської діяльності. Зазначається, що в перші місяці війни ця благодійна діяльність характеризувалася хаосом, який потребував систематизації.

Отже, базуючись на всьому вищезазначеному, можна зробити висновок, що розробка web-додатку підтримку діяльності волонтерської діяльності є дійсно актуальною. Сьогодні забезпечення потреб населення України є досить важливим завданням у такий важкий час. Застосування даного web-додатку дозволить збільшити видимість волонтерської організації та її роботи. Це дозволить їй охопити ширшу аудиторію та потенційно залучити більше підтримки. Демонструючи місію, цінності та досягнення організації, використання такого web-додатку може сприяти виклику ентузіазму та зацікавленості у її діяльності та надихнути інших взяти у цьому участь, а, отже, допомогти більшій кількості людей.

1.2 Аналіз програмних продуктів-аналогів

Перед розробкою прототипу вищевказаного web-додатку необхідно провести пошук та аналіз web-додатків-аналогів волонтерських організацій.

Основними критеріями аналізу є такі:

- оформлення (дизайн web-додатку, підбір кольорів, фонових зображень, читабельність тексту);
- дотримання стилю;
- зручність навігації;
- якість і доцільність ілюстрованого матеріалу.

Перший аналог – web-додаток волонтерської допомоги «Гуманітарна та медична допомога Україні» [3]. Цей онлайн ресурс для волонтерів представляє web-додаток з можливостями робити донати, переглядати звітність даної організації, контактну інформацію, ознайомлюватися з новинами тощо. Також відвідувач може дізнатися інформацію як надіслати допомогу з іншої країни. Дизайн даного онлайн ресурсу доцільно пропрацьовано (рис. 1.1). Було використано чотири різні шрифти. На web-додатку для фону використовується жовтий, синій, білий та чорний кольори. Гама тексту оформлена чорним, білим і червоним забарвленням. Кольори досить гармонічно взаємодіють між собою. Перевагами даного онлайн ресурсу є те, що він повністю відповідає тематиці, оформлення є актуальним для нашого часу, ілюстрований матеріал підібрано професійно, стиль витримано згідно до теми. Серед його недоліків варто зазначити те, що навігація web-додатку не продумана, а також стрічка новин досить виділяється зі стилю основної частини, що, у свою чергу, є незручним для сприйняття користувачу (рис. 1.2).



Рисунок 1.1 – Головна сторінка web-додатку helpukraine.center

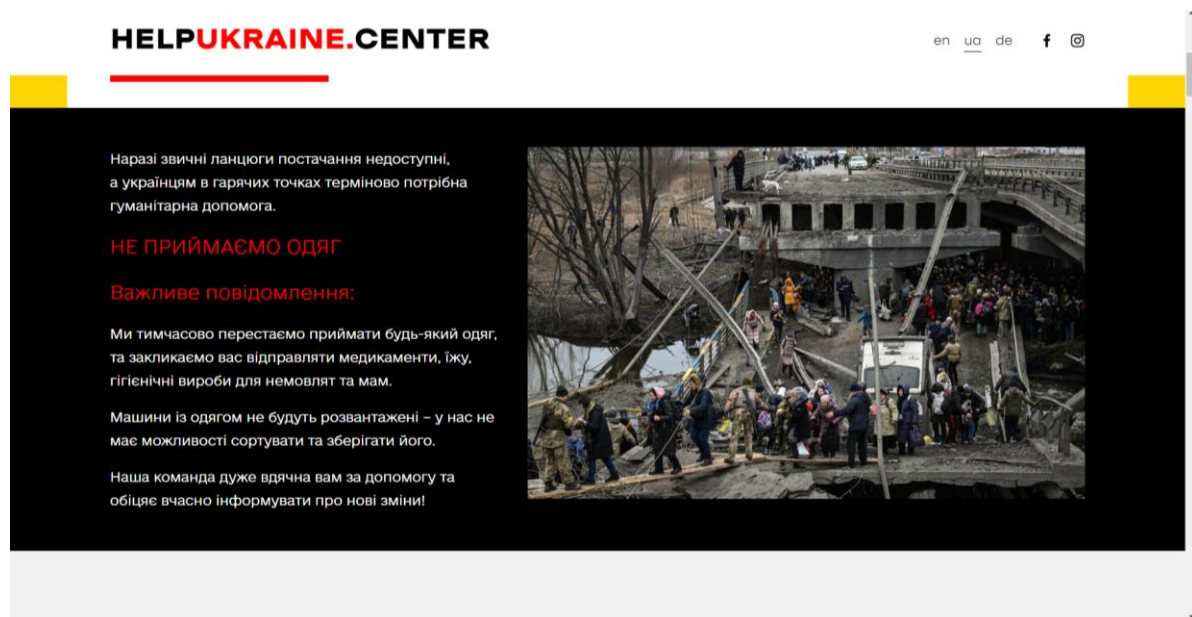


Рисунок 1.2 – Сторінка новин web-додатку helpukraine.center

Другий аналог – web-додаток організації «Повернись живим» [4]. Даний онлайн ресурс представляю собою фонд допомоги військовим. Його головна сторінка представлена на рисунку 1.3. Тут є можливість ознайомитися з новинами з фронту, переглянути звітність фонду, дізнатися докладніше про нього та в яких напрямках вони працюють. Є можливість перерахувати кошти на армію. При створені web-

додатку було використано п'ять різних, але гармонічних між собою поєднаних, шрифтів. Кольори підбрано згідно тематики military. Перевагами даного онлайн ресурсу є те, що його структура та стиль оформлення відповідає тематиці, вдало підбрано шрифти, навігація дає можливість знайти те, що сховано (рис. 1.4) тощо. Недостатня якість ілюстрованого матеріалу (в деяких новинах) є головним недоліком представленого web-додатку.

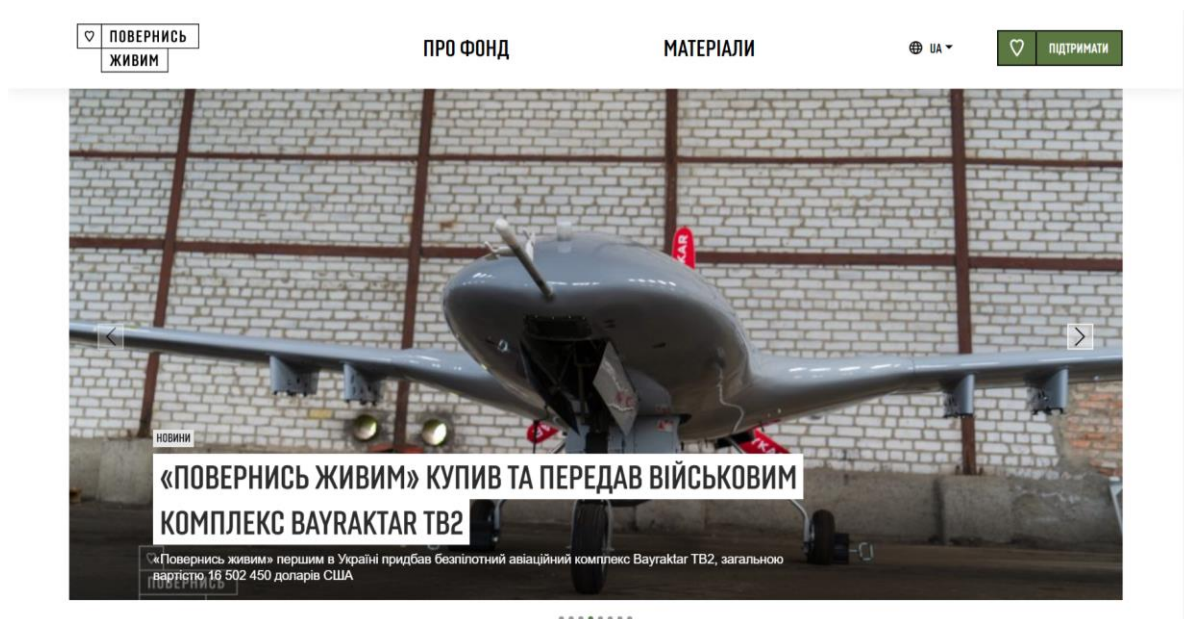


Рисунок 1.3 – Головна сторінка web-додатку savelife.in.ua

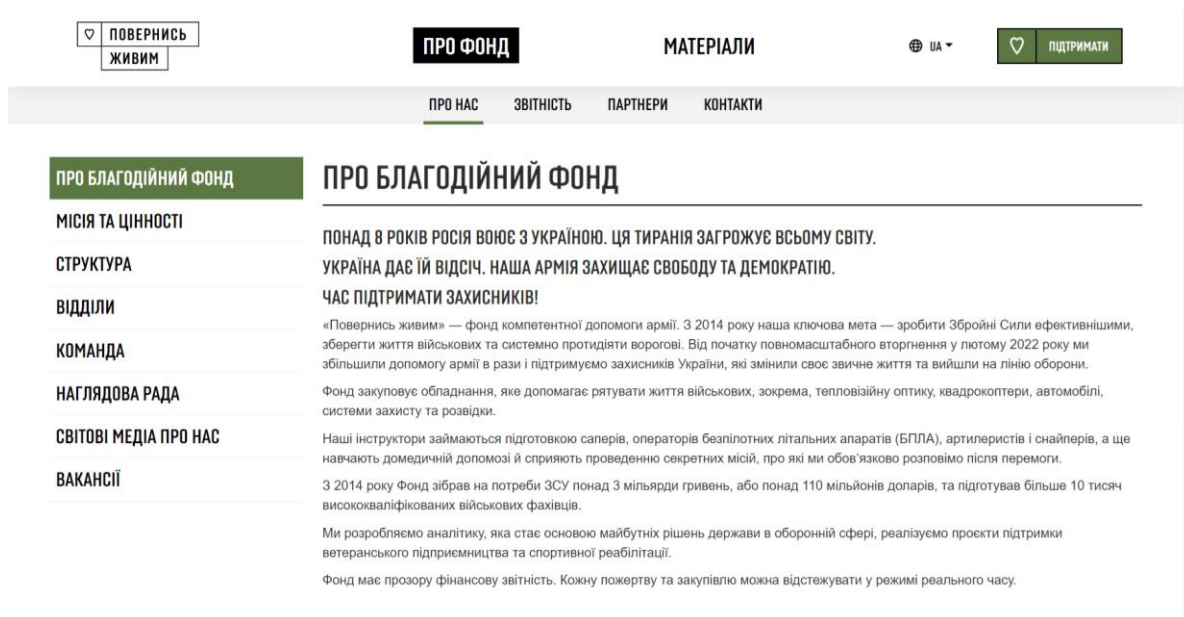


Рисунок 1.4 – Інформаційна сторінка web-додатку savelife.in.ua

Третій аналог – web-додаток благодійної організації, яка допомагає армії країни «ARMY SOS» [5]. Це невелика інформаційна платформа з можливістю зробити внесок для закупівлі новітнього обладнання та спорядження. Його головну сторінку зображено на рисунку 1.5. Також можна продивитися звіти фонду, хто працює в цій команді та як можна зв'язатися з даною організацією. Оформлення сторінок є простим, але в той же час інформативним. Вдало підібрана кольорова палітра. На головній сторінці міститься тільки основна інформація (рис. 1.5), додаткову можна знайти у меню зверху, використовуючи меню (рис. 1.5). Тобто немає нагромадження непотрібними даними відразу, що є досить зручним для користувачів. Перевагами даного web-додатку є його інформативність і кольорова гамма (рис. 1.6). Його головний недолік – це навігація, оскільки зверху не вистачає пунктів, так як у футері цього онлайн ресурсу можна знайти повну інформацію щодо переміщення по ньому.

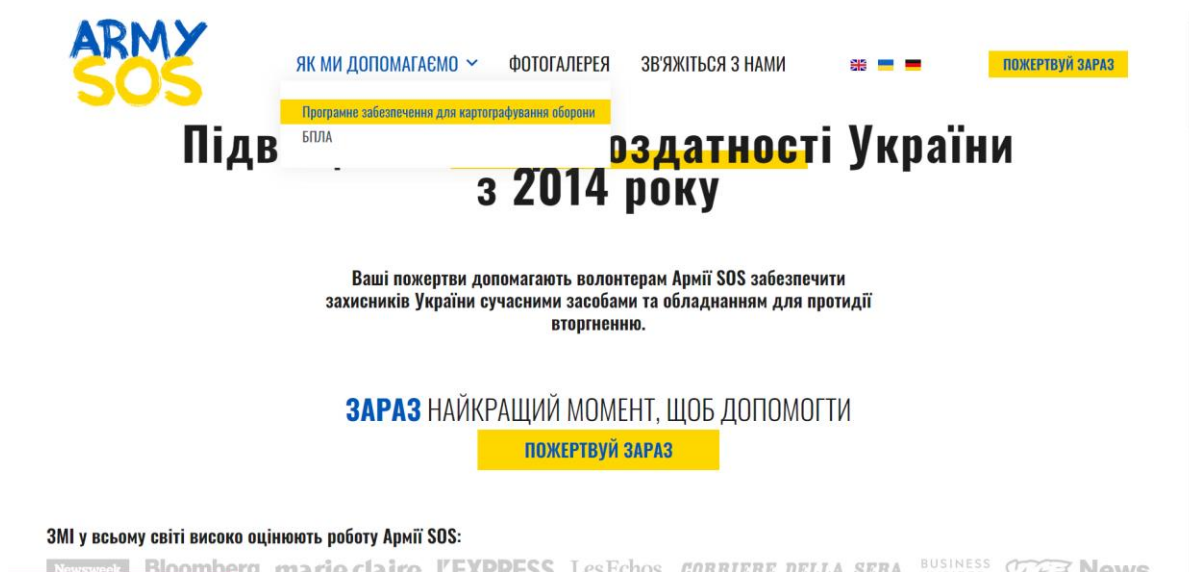


Рисунок 1.5 – Головна сторінка та меню web-додатку armysos.com.ua

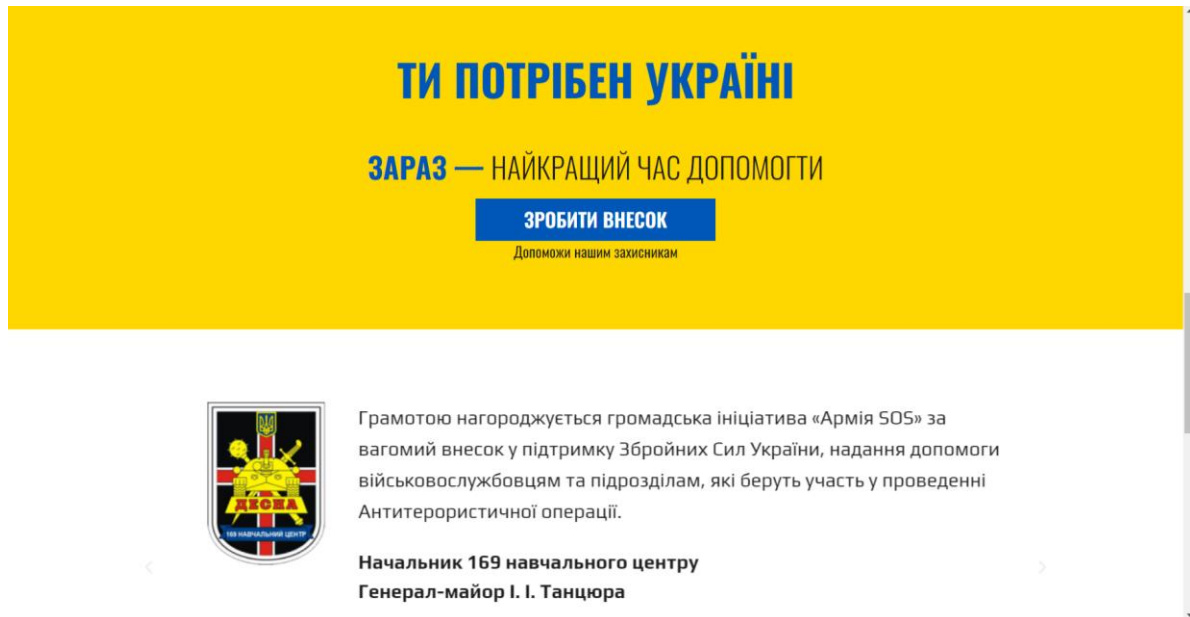


Рисунок 1.6 – Інформаційна сторінка web-додатку armysos.com.ua

Четвертий аналог – web-додаток громадської організації «Мрія вільних людей» [6]. Це онлайн ресурс одеської волонтерської організації (рис. 1.7). Він несе в більшості своїй інформаційний характер та соціальну направленість. Має відеоматеріали. Є короткий опис чим займається організація, яку допомогу надає населенню та військовим, що необхідно на даний момент та на що збирається матеріальна допомога (рис. 1.8). На його сторінках також можна знайти цікаву інформацію від самого волонтерського центру. Даний web-додаток спроектовано професійно. Використано три різні шрифти. Кольорова палітра підібрана гармонійно. Його головні переваги – це оформлення та читабельність тексту. Головним недоліком web-додатку є досить багато незадіяного простору. Також кольорове оформлення новин не відповідає основній палітрі онлайн ресурсу (рис. 1.9).

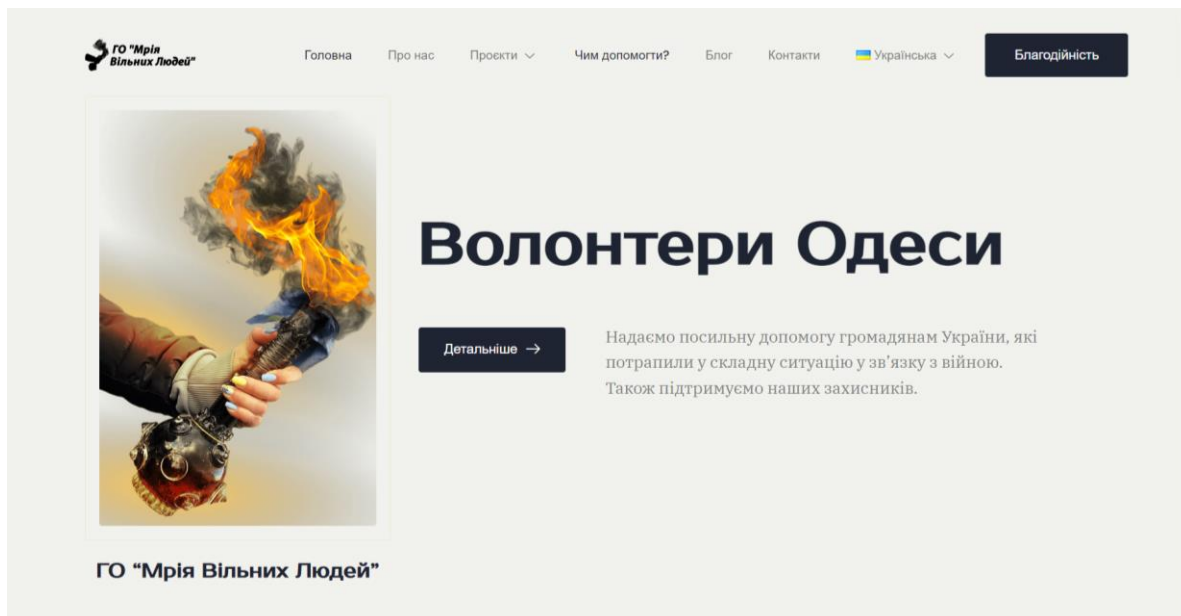


Рисунок 1.7 – Головна сторінка web-додатку mriya.od.ua

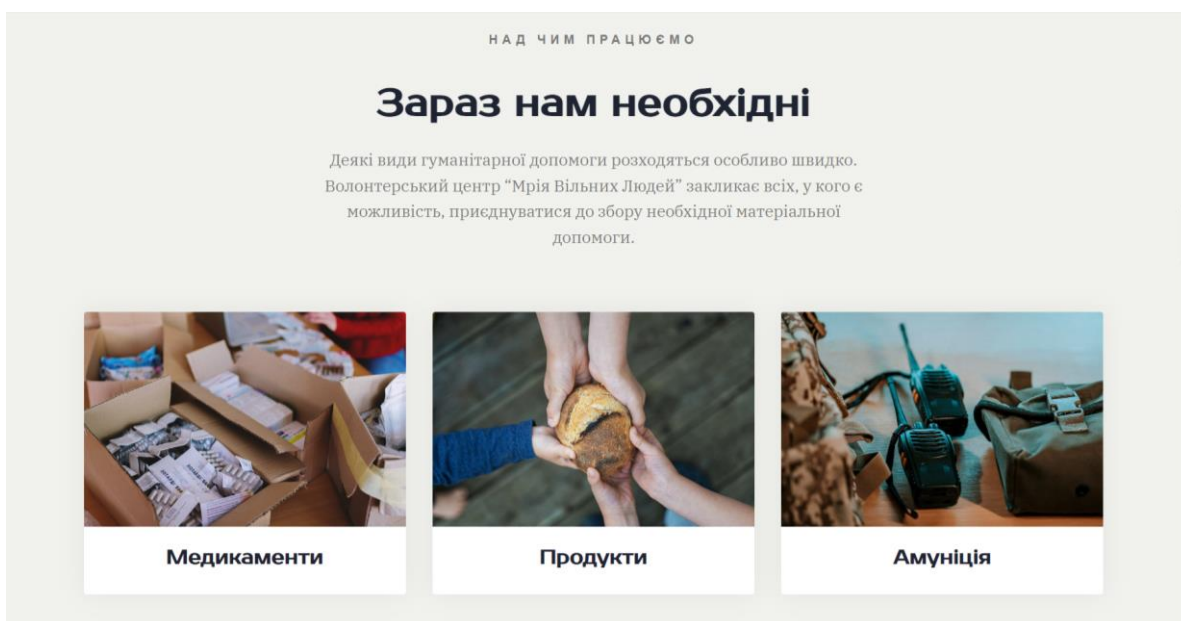


Рисунок 1.8 – Інформаційна сторінка web-додатку mriya.od.ua

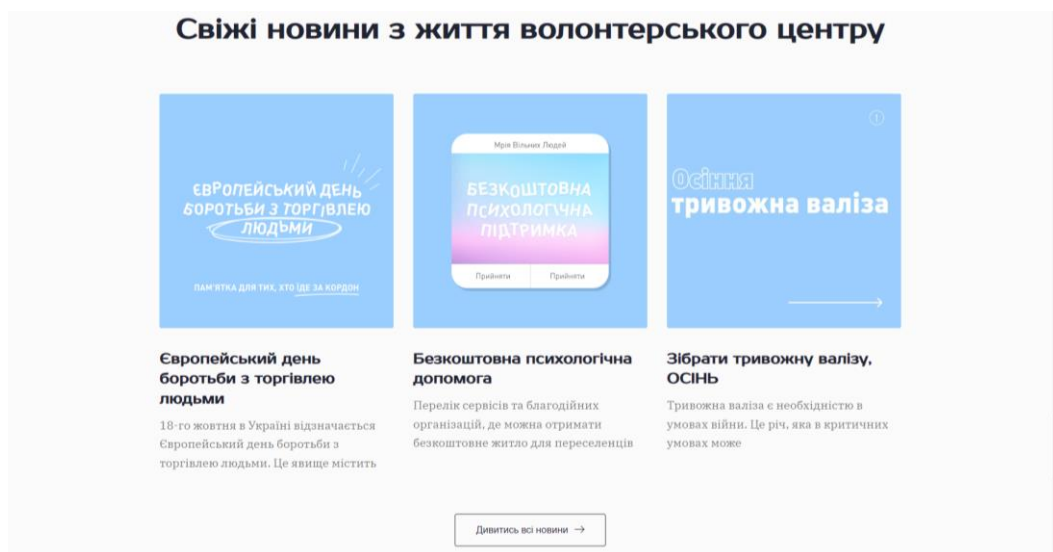


Рисунок 1.9 – Сторінка новин web-додатку mriya.od.ua

Після детального аналізу web-додатків-аналогів, було визначено їх основні переваги та недоліки. У таблиці 1.1 представлені його результати.

Таблиця 1.1 – Порівняльна таблиця характеристик web-додатків-аналогів

| Характеристика/ Онлайн ресурс | helpukrai ne.center | savelife.in .ua | armyos.com .ua | mriya.od.ua |
|----------------------------------|------------------------|--------------------|-------------------|-------------|
| Зручний інтерфейс | + | + | + | + |
| Сучасний дизайн | - | + | + | - |
| Функціональність | + | - | + | + |
| Інтерактивність | - | - | + | + |
| Навігація | + | + | + | + |
| Реєстрація користувачів | - | - | - | - |
| Інформаційна сторінка | + | + | + | + |
| Сторінка новин | + | + | + | + |
| Наявність відгуків | + | - | - | - |

Після аналізу аналогів, можна підвести підсумок. Більша кількість вищезазначених онлайн ресурсів мають вдалу та повну інформативну складову. Але для отримання програмного продукту, який буде подобатися користувачам, потрібно промалювати дизайн продумано та якісно. Із аналогів можна взяти інформаційну складову, структуру, кольорову гамму та ілюстративний матеріал, доповнивши власний web-додаток новим функціоналом, що зробить розроблюваний продукт унікальним та дозволить подолати вищезазначені недоліки.

1.3 Постановка задачі

Метою даного дослідження є створення web-додатку для підтримки волонтерської діяльності. Його використання забезпечить її належну організацію шляхом автоматизації процесу взаємодії волонтерів та людей, яким необхідна допомога. Це залучить велику кількість небайдужих для підтримки постраждалих під час війни та в цілому країні.

Основні вимоги до створюваного web-додатку є наступними:

- організувати зручну навігацію web-додатку для швидкої орієнтації по його сторінках;
- створити зручну форму для реєстрації користувачів;
- організувати блок звітності для підвищення довіри до даного онлайн ресурсу;
- створити інтуїтивно зрозумілу форми подання заявок на допомогу;
- розробити чат для онлайн комунікації волонтерів та людей, які потребують допомоги;
- забезпечити застосування пошуку заявок за назвою та фільтрацію за регіоном і категорією;

- створити зручну адміністративну панель для швидкого підтвердження/відхилення заявок та створення грошових зборів;
- розробити симулятор для проведення грошових зборів;
- розробити сучасний дизайн для привернення уваги відвідувачів web-додатку.

Для досягнення мети проекту потрібно виконати наступні задачі:

- визначити актуальність дослідження та цільову аудиторію роботи;
- провести аналіз існуючих відповідних тематиці web-додатків-аналогів і виділити їх переваги та недоліки;
- обрати методи та технології для розробки web-додатку підтримки волонтерської діяльності;
- спроектувати модель та структуру даного web-додатку;
- створити прототип web-додатку підтримки волонтерської діяльності;
- розробити функціонал представленого web-додатку;
- виконати тестування web-додатку підтримки волонтерської діяльності.

Вимоги до всього проекту, структура програмного продукту, що розробляється, тип програмного забезпечення та функціональні можливості системи описані в технічних завданнях на розробку проекту (додаток А).

Для реалізації web-додатку було обрано такі технології, як HTML [7] – мову розмітки гіпертексту, для створення структури web-сторінок, каскадні таблиці стилів CSS [8] використовувалися для оформлення, надання візуальних ефектів, а також адаптованості сторінок, скриптову мову програмування JavaScript [9] для забезпечення динамічності та асинхронної взаємодії з серверною частиною, Laravel [10] – фреймворк для PHP, який полегшить розробку, і Bootstrap [11] – фреймворк CSS для респонсивного дизайну. Для реалізації бази даних було обрано систему управління базами даних (СУБД) MySQL [12].

2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ

2.1 Структурно-функціональне моделювання

Для моделювання функціональності системи та організації бізнес-процесів використовуються об'єктно-орієнтовані та структуровані методи. І IDEF0 [13] є одним із найпопулярніших інструментів для цього. Це метод графічного опису систем і процесів організаційної діяльності як набору взаємозалежних функцій. Він дозволяє вивчати останні без прив'язки їх до об'єктів, які забезпечують їх виконання. У свою чергу, це є досить зручно.

У стандарті IDEF0 об'єкти, інформаційні та матеріальні потоки, трансформовані в бізнес-процеси, відображаються за допомогою управління.

Для графа контексту визначено такі дані:

- вхідні дані: запит на публікацію оголошення про допомогу;
- управління: правила реєстрації та авторизації користувачів, інструкція публікації оголошень, положення про доброчесність;
- механізми: web-додаток, апаратне забезпечення, база даних;
- вихідні дані: розміщене оголошення про допомогу.

На рисунку 2.1 представлено функціональну діаграму IDEF0 web-додатку підтримки волонтерської допомоги.

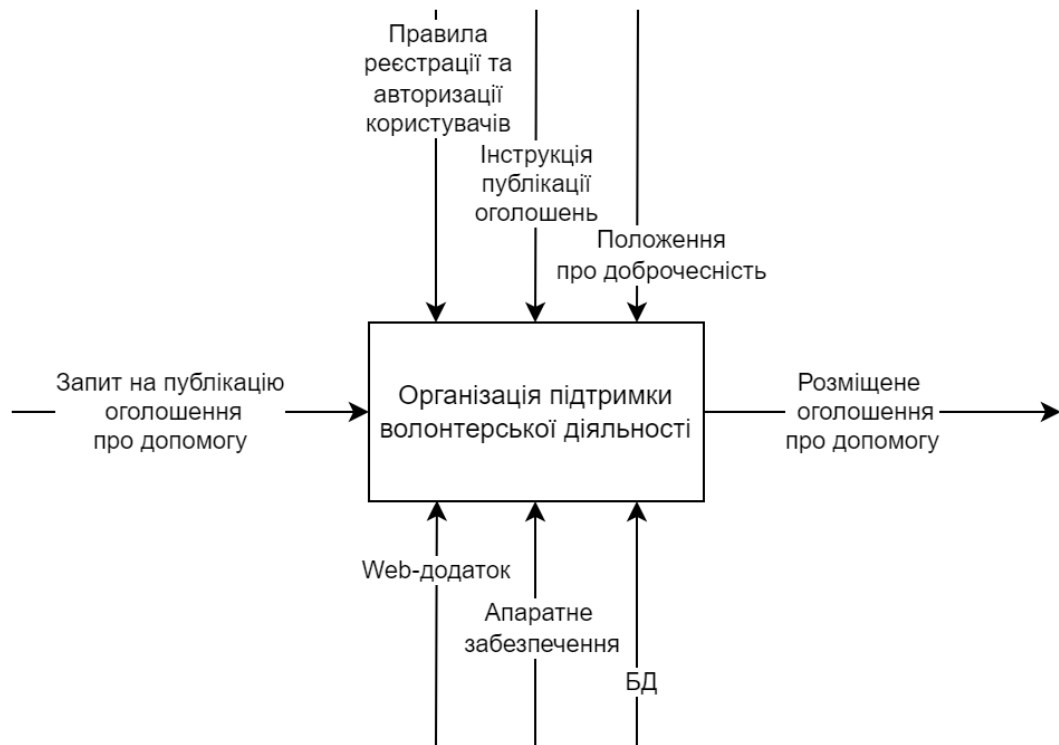


Рисунок 2.1 – Функціональна діаграма

Для уточнення внутрішнього потоку діаграми IDEF0, було виконано її декомпозицію (рис. 2.2). Таблиця введення та виведення даних (табл. 2.1) для підпроцесів публікації оголошень була створена для полегшення представлення інформації.

Таблиця 2.1 – Вхідні та вихідні дані декомпозиції процесу публікації оголошень

| Підпроцес | Вхідні дані | Управління | Механізми | Вихідні дані |
|----------------------------|---|--|--|-----------------------------------|
| Реєстрація/ авторизація | Запит на публікацію оголошення про допомогу | Правила реєстрації та авторизації користувачів | Web-додаток, апаратне забезпечення, БД | Ідентифікація користувача |
| Визначення типу допомоги | | Інструкція публікації оголошень, положення про доброчесність | | Вибір категорії |
| Введення місця знаходження | | | | Зазначення місцезнаходження |
| Написання опису потреби | | | | Розміщене оголошення про допомогу |

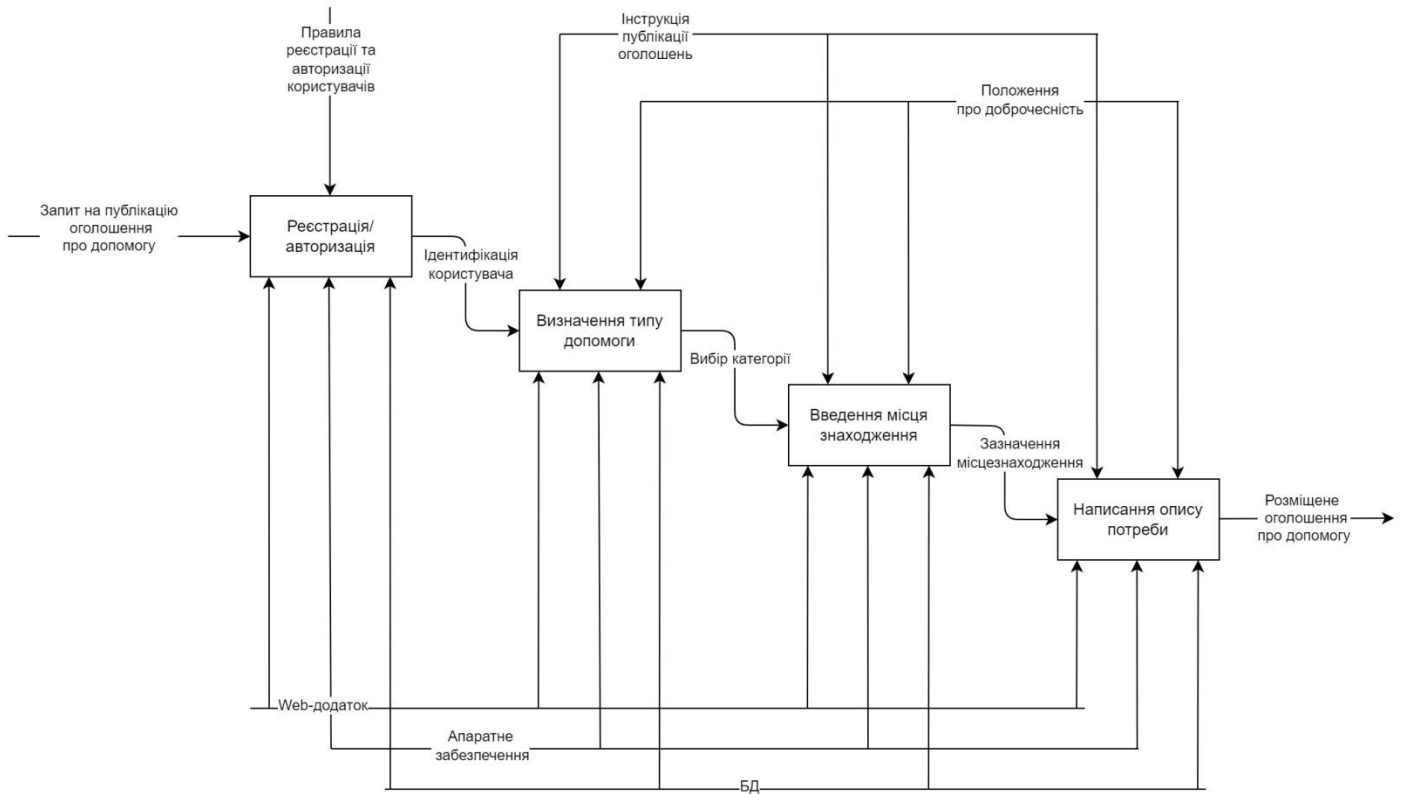


Рисунок 2.2 – Діаграма декомпозиції 1-го рівня

Для демонстрації процесу створення грошових зборів розроблена діаграма послідовності (рис. 2.3). На ній присутні два актори (Адміністратор та Представник волонтерської організації). Вони взаємодіють особисто за межами представленого web-додатку. А результатом їх комунікації є створений грошовий збір. Саме він розміщується у web-додатку волонтерської допомоги.

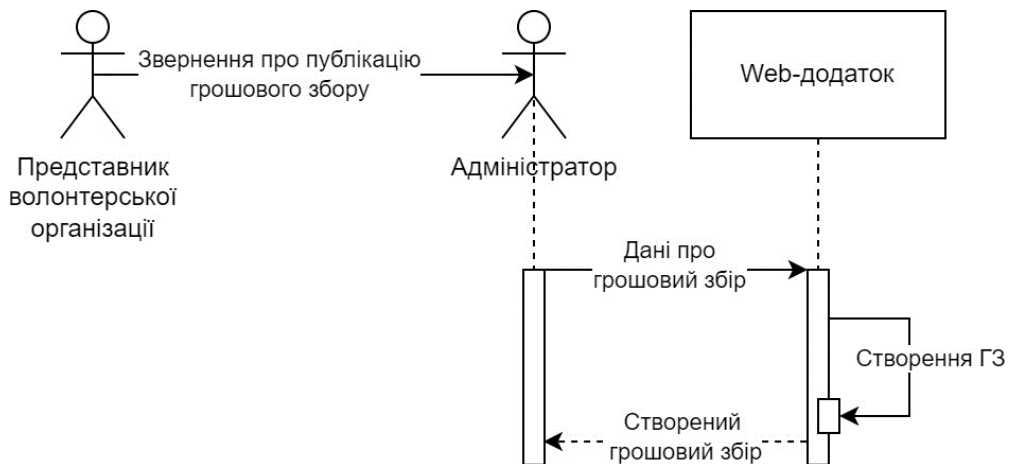


Рисунок 2.3 – Діаграма послідовності процесу створення грошових зборів

2.2 Проектування інформаційної системи

Виконавши моделювання процесів та проаналізувавши структуру майбутнього web-додатку, необхідно розробити діаграму варіантів використання (Use Case diagram) [14]. Це найбільш загальне уявлення функціонального призначення розроблюваного програмного продукту.

На даній діаграмі застосовуються два типи основних сутностей: варіанти використання та дійові особи, між якими встановлюються відносини.

Акторами розроблюваного web-додатку виступають користувач, гість, адміністратор та база даних (БД).

Можливі дії є наступними:

- розміщення оголошення (створення оголошення, редагування оголошення, видалення оголошення);
- авторизація;
- реєстрація;
- перегляд контактної інформації автора оголошення;
- написання публічних коментарів до оголошень;
- переказ грошей на благодійність;
- перегляд оголошень;
- перегляд звітності;
- розміщення грошових зборів (створення грошових зборів, редагування грошових зборів, видалення грошових зборів);
- підтвердження/відхилення заявок на публікацію.

Use Case діаграма web-додатку підтримки волонтерської діяльності представлена на рисунку 2.4.

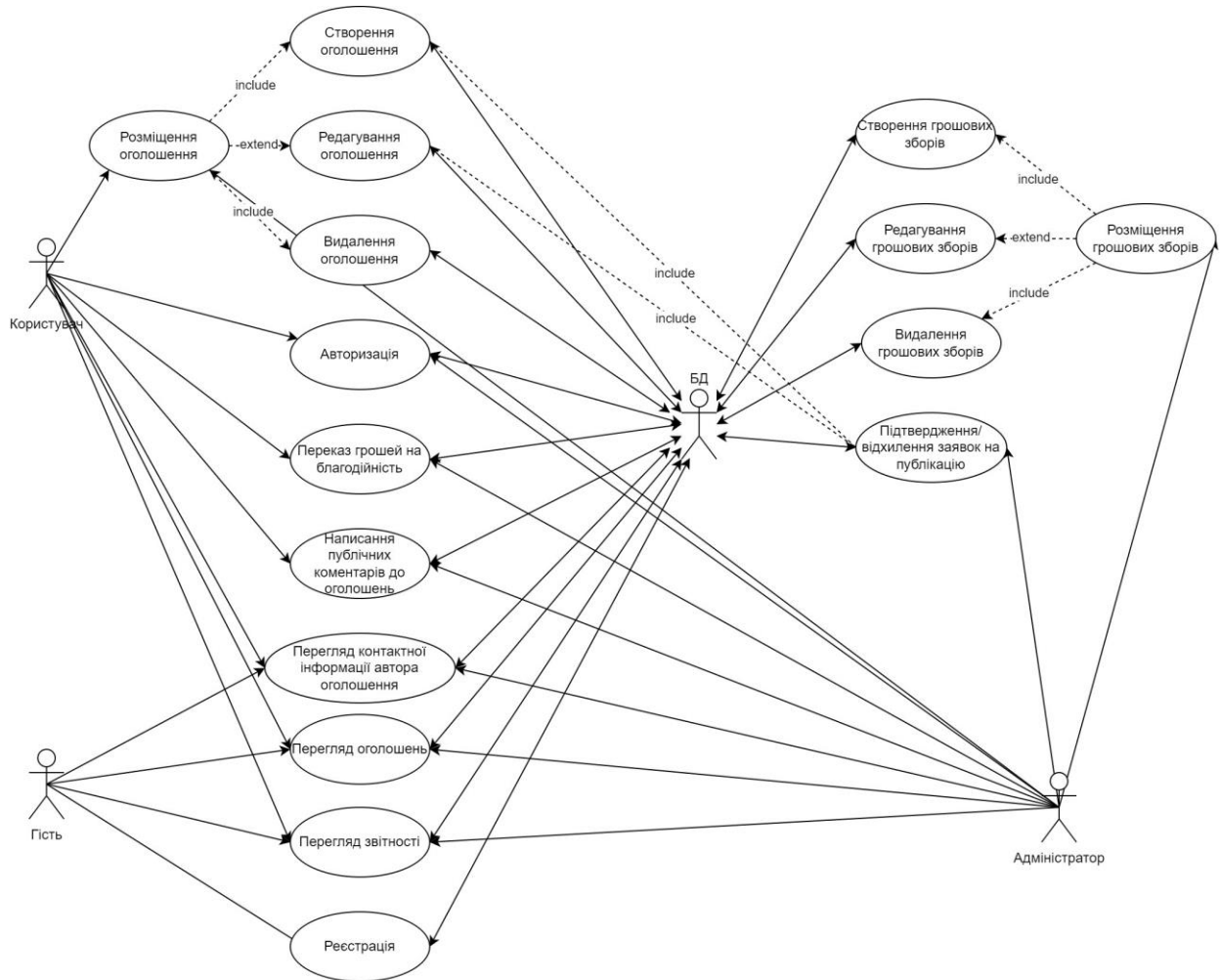


Рисунок 2.4 – Діаграма варіантів використання

2.3 Проектування моделі бази даних

Для роботи з інформацією даного web-додатку було вирішено підключити базу даних, у якій будуть зберігатися дані зареєстрованих користувачів, їх оголошення, грошові збори та інше.

У БД створено 11 таблиць, кожна з них відповідає за окремий набір структурованих даних. А саме:

- users – користувачі;
- reset_codes – коди відновлення пароля;

- confirmation_codes – коди підтвердження пошти;
- applications – заявки на допомогу;
- application_images – зображення для заявок у категорію «Надаю допомогу»;
- reject_messages – коментарі при відхиленні заявок;
- comments – коментарі до заявки;
- reports – звіти до заявки;
- categories – категорії заявок;
- fundraisers – збори коштів;
- fundraiser_users – учасники грошових зборів.

Логічна модель бази представлена на рисунку 2.5.

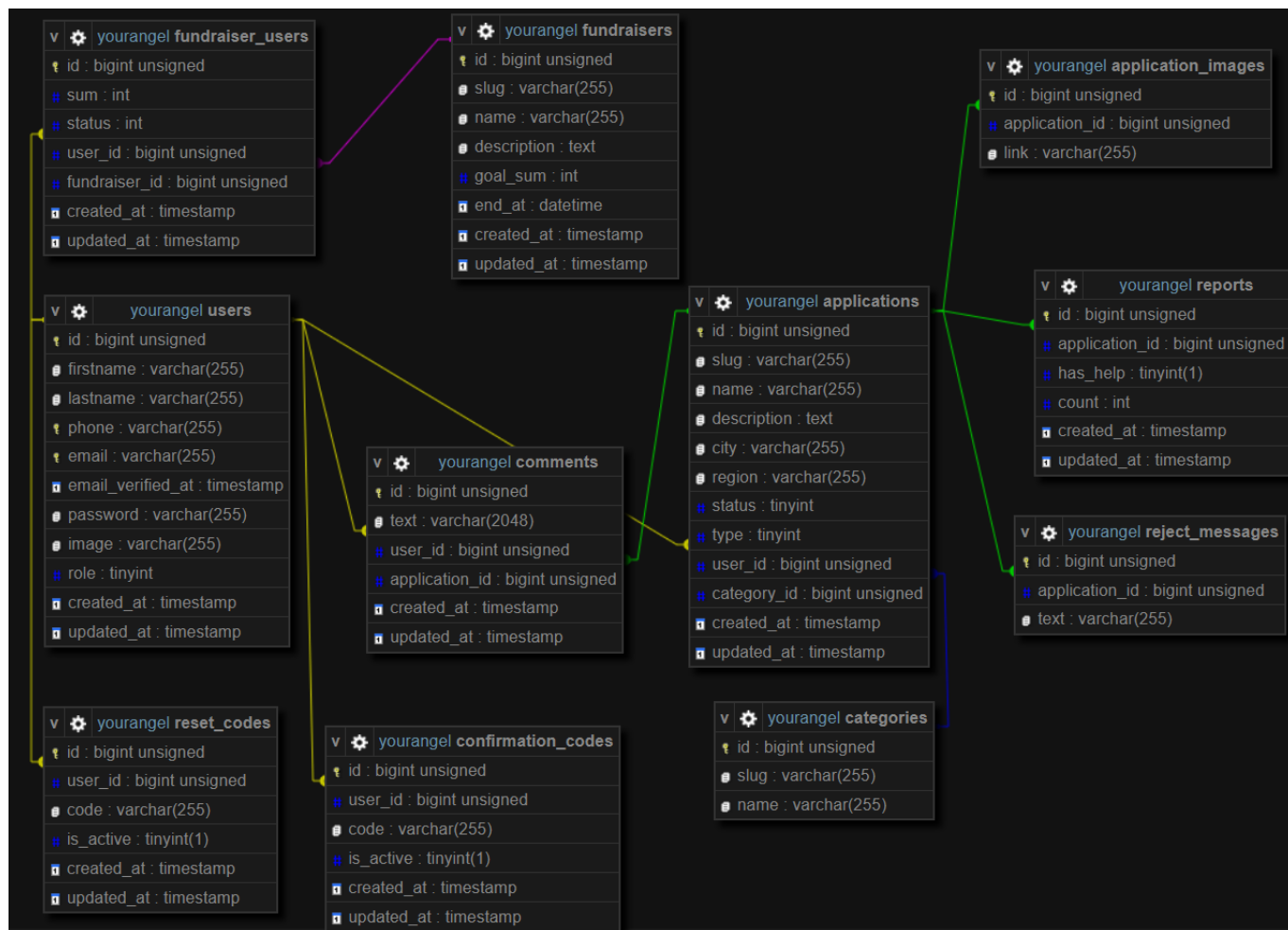


Рисунок 2.5 – Логічна модель бази даних

3 РОЗРОБКА WEB-ДОДАТКУ ПІДТРИМКИ ВОЛОНТЕРСЬКОЇ ДІЯЛЬНОСТІ

3.1 Архітектура web-додатку

Web-додатки, написані на Laravel, мають архітектуру Model-View-Controller (MVC) [15]. Це дозволяє відокремити їх логіку від інтерфейсу користувача. Використовуючи цю архітектуру, web-додатки легше оновлювати, оскільки кожна сторінка має власний файл View, кожна таблиця в базі даних має Model, а кожна функція, яка належить до спільного «об'єкта», має власний Controller. Це також значно спрощує кодування, оскільки структура містить тисячі різних інструментів.

Процес роботи web-додатку можна розділити на кілька етапів при відкритті сторінки:

1. Користувач відкриває сторінку за посиланням.
2. Маршрутизатор запускається, визначаючи URI та викликаючи метод із контролера, указанного для запиту.
3. Контролер створює об'єкт певної моделі та виконує інші функції.
4. Модель заповнюється даними з бази даних, або ж навпаки, дані з моделі додаються в базу.
5. Контролер рендерить представлення, тобто шаблон сторінки, і передає модель.
6. Користувач бачить сторінку з дизайном та інформацією з бази даних.

Архітектура даного web-додатку представлена на рисунку 3.1.

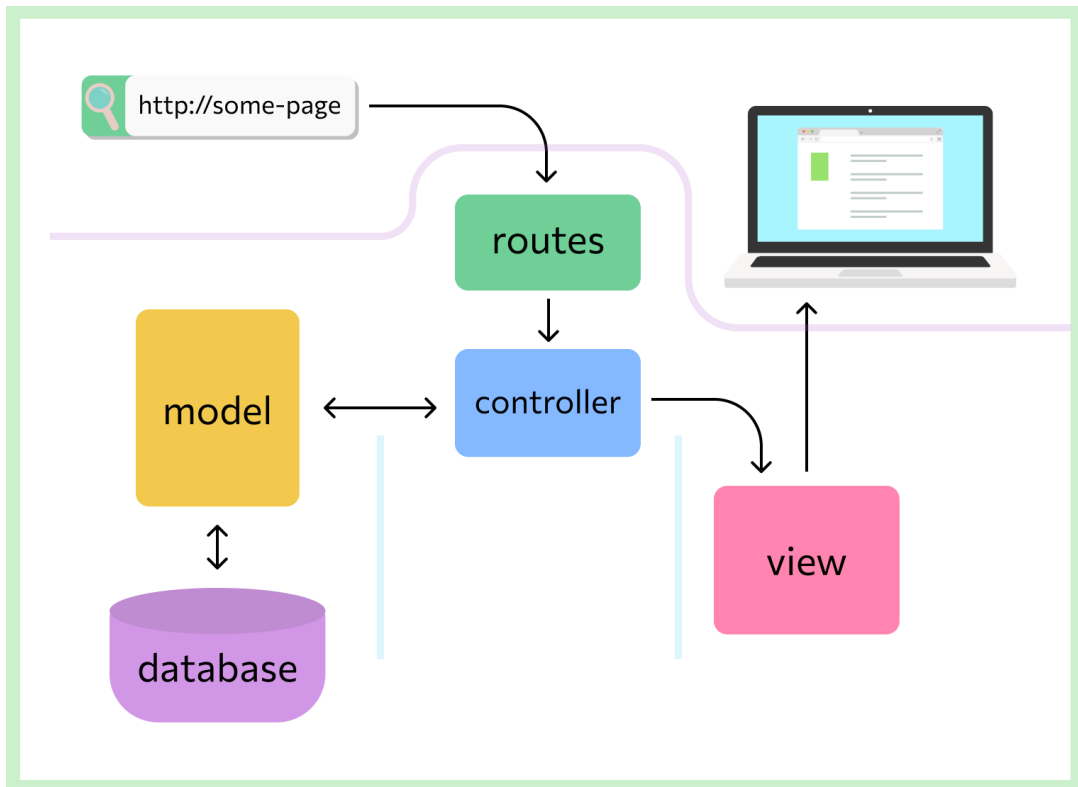


Рисунок 3.1 – Архітектура web-додатку

3.2 Програмна реалізація

Перед початком створення web-додатку необхідно обрати середовище та технології розробки. Для реалізації web-додатку було обрано такі технології, як HTML та CSS для створення структури та зовнішнього вигляду, PHP для обробки логіки на сервері, SQL – для взаємодії з базою даних, JavaScript – для створення інтерактивних елементів, Laravel – фреймворк для PHP, який полегшує розробку, і Bootstrap – фреймворк CSS для респонсивного дизайну. Для написання коду було обрано Visual Studio Code [16].

Першим етапом розробки є створення бази даних MySQL у СУБД PhpMyAdmin. За логічною моделлю БД розроблено відповідні таблиці (рис. 3.2). Службова таблиця migrations була створена автоматично для роботи з Laravel.

| Таблиця ▲ | Дія | Рядки ? | Тип | Зіставлення | Розмір | Фрагментовані |
|----------------------|--------------|---------|--------|--------------------|---------|---------------|
| ■ applications | ★ 🔍 ☰ 🔍 📄 🗑️ | 4 | InnoDB | utf8mb4_unicode_ci | 48.0 КБ | - |
| ■ application_images | ★ 📄 ☰ 🔍 📄 🗑️ | 0 | InnoDB | utf8mb4_unicode_ci | 32.0 КБ | - |
| ■ categories | ★ 🔍 ☰ 🔍 📄 🗑️ | 6 | InnoDB | utf8mb4_unicode_ci | 16.0 КБ | - |
| ■ comments | ★ 📄 ☰ 🔍 📄 🗑️ | 0 | InnoDB | utf8mb4_unicode_ci | 48.0 КБ | - |
| ■ confirmation_codes | ★ 🔍 ☰ 🔍 📄 🗑️ | 1 | InnoDB | utf8mb4_unicode_ci | 32.0 КБ | - |
| ■ fundraisers | ★ 🔍 ☰ 🔍 📄 🗑️ | 1 | InnoDB | utf8mb4_unicode_ci | 16.0 КБ | - |
| ■ fundraiser_users | ★ 🔍 ☰ 🔍 📄 🗑️ | 4 | InnoDB | utf8mb4_unicode_ci | 48.0 КБ | - |
| ■ migrations | ★ 🔍 ☰ 🔍 📄 🗑️ | 11 | InnoDB | utf8mb4_unicode_ci | 16.0 КБ | - |
| ■ reject_messages | ★ 🔍 ☰ 🔍 📄 🗑️ | 1 | InnoDB | utf8mb4_unicode_ci | 32.0 КБ | - |
| ■ reports | ★ 🔍 ☰ 🔍 📄 🗑️ | 3 | InnoDB | utf8mb4_unicode_ci | 32.0 КБ | - |
| ■ reset_codes | ★ 📄 ☰ 🔍 📄 🗑️ | 0 | InnoDB | utf8mb4_unicode_ci | 32.0 КБ | - |
| ■ users | ★ 🔍 ☰ 🔍 📄 🗑️ | 2 | InnoDB | utf8mb4_unicode_ci | 48.0 КБ | - |

Рисунок 3.2 – Список таблиць бази даних

Усі сторінки мають однакову шапку та підвал, для цього було розроблено два відповідних шаблони, які потім підключено до кожної сторінки (рис. 3.3-3.4).

```

resources > views > templates > footer.blade.php
1 <div class="container">
2   <footer class="py-3 my-4">
3     <ul class="nav justify-content-center border-bottom pb-3 mb-3">
4       <li class="nav-item"><a href="{{route('index')}}" class="nav-link px-2 text-body-secondary">Головна</a></li>
5       <li class="nav-item"><a href="{{route('applications.index')}}" class="nav-link px-2 text-body-secondary">Заявки</a></li>
6       <li class="nav-item"><a href="{{route('reports.index')}}" class="nav-link px-2 text-body-secondary">Звіти</a></li>
7       <li class="nav-item"><a href="{{route('fundraisers.index')}}" class="nav-link px-2 text-body-secondary">Грошові збори</a></li>
8       <li class="nav-item"><a href="{{route('applications.create')}}" class="nav-link px-2 text-body-secondary">Нова заявка</a></li>
9       <li class="nav-item"><a href="{{route('applications.my')}}" class="nav-link px-2 text-body-secondary">Мої заявки</a></li>
10    </ul>
11    <p class="text-center text-body-secondary">© 2023 YourAngel</p>
12  </footer>
13 </div>
14

```

Рисунок 3.3 – Шаблон footer.blade.php

```

header.blade.php
resources > views > templates > header.blade.php
1 <header class="p-3 mb-3 border-bottom">
2   <div class="container">
3     <div class="d-flex flex-wrap align-items-center justify-content-center justify-content-lg-start">
4       <a href="{{route('index')}}" style="font-weight: 800;font-size: 14pt;margin-right: 20px;"
5         class="d-flex align-items-center mb-2 mb-lg-0 text-dark text-decoration-none">
6         
7         YourAngel
8       </a>
9
10      <ul class="nav col-12 col-lg-auto me-lg-auto mb-2 justify-content-center mb-md-0">
11        <li><a href="{{route('index')}}" class="nav-link px-2 link-dark">Головна</a></li>
12        <li><a href="{{route('applications.index')}}" class="nav-link px-2 link-dark">Заявки</a></li>
13        <li><a href="{{route('reports.index')}}" class="nav-link px-2 link-dark">Звіти</a></li>
14        <li><a href="{{route('fundraisers.index')}}" class="nav-link px-2 link-dark">Грошові збори</a></li>
15      </ul>
16
17      <form class="col-12 col-lg-auto mb-3 mb-lg-0 me-lg-3" action="{{route('applications.index')}}">
18        <input type="hidden" name="type" value="{{$_GET['type'] ?? 'need'}}">
19        <input type="search" name="search" class="form-control" placeholder="Пошук..." aria-label="Search">
20      </form>
21
22      <div class="dropdown text-end">
23        @auth
24          <a href="#" class="d-block link-dark text-decoration-none dropdown-toggle" id="dropdownUser1"
25            data-bs-toggle="dropdown" aria-expanded="false">
26            
28          </a>
29          <ul class="dropdown-menu text-small" aria-labelledby="dropdownUser1" style="">
30            @if(auth()->user()->isAdmin())
31              <li><a class="dropdown-item bg-body-secondary"
32                href="{{route('admin.index')}}">Адмін-панель</a></li>
33            @endif
34            <li><a class="dropdown-item" href="{{route('applications.create')}}">Створити заявку</a></li>
35            <li><a class="dropdown-item" href="{{route('applications.мy')}}">Мої заявки</a></li>
36            <li>
37              <hr class="dropdown-divider">
38            </li>
39            <li><a class="dropdown-item" href="{{route('auth.logout')}}">Вихід</a></li>
40          </ul>
41        @else
42          <a href="{{route('auth.login')}}" class="btn btn-outline-secondary">Вхід</a>
43          <a href="{{route('auth.register')}}" class="btn btn-primary">Реєстрація</a>
44        @endauth
45      </div>
46    </div>
47  </div>
48 </header>

```

Рисунок 3.4 – Шаблон header.blade.php

Далі створюємо структуру сторінки за допомогою HTML. Приклад структури головної сторінки зображено на рисунку 3.5.

```

1 @extends('templates.layout')
2
3 @section('title','Головна | YourAngel')
4
5 @section('content')
6     <div class="px-4 pt-5 my-5 text-center border-bottom">
7         <h1 class="display-4 fw-bold text-body-emphasis">ЛАСКАВО ПРОСИМО</h1>
8         <div class="col-lg-6 mx-auto">
9             <p class="lead mb-4">
10                 Ми тут, щоб надати підтримку та допомогу людям, які постраждали внаслідок війни. Україна зазнала великих випробуван
11             </p>
12             <div class="d-grid gap-2 d-sm-flex justify-content-sm-center mb-5">
13                 <a href="{{route('applications.index')}}" class="btn btn-primary btn-lg px-4 me-sm-3">Усі заявки</a>
14                 <a href="{{route('applications.create')}}" type="button" class="btn btn-outline-secondary btn-lg px-4">Створити нов
15             </div>
16         </div>
17         <div class="overflow-hidden" style="max-height: 30vh;">
18             <div class="container px-5">
19                 
21         </div>
22     </div>
23
24     <div class="container col-xxl-8 px-4 py-5">
25         <div class="row flex-lg-row-reverse align-items-center g-5 py-5">
26             <div class="col-10 col-sm-8 col-lg-6">
27                 
29             <div class="col-lg-6">
30                 <h1 class="display-5 fw-bold text-body-emphasis lh-1 mb-3">Опинився в скрутній ситуації?</h1>
31                 <p class="lead">Тоді тобі до нас! Небайдужі люди з радістю допоможуть вирішити будь-яку проблему.
32                 Розміщуй оголошення про необхідну допомогу та отримуй те, чого критично не вистачає.</p>
33             </div>
34         </div>
35     </div>
36
37     <div class="container col-xl-10 col-xxl-8 px-4 py-5">

```

Рисунок 3.5 – Створення головної сторінки

Для перегляду, створення, редагування та видалення заявок створюємо контролер `ApplicationController` (рис. 3.6). `ApplicationController` має 8 основних методів: `index`, `my`, `create`, `show`, `edit`, `update`, `destroy`. `Index` призначений для відображення заявок на сторінці «Заявки». Враховує фільтри та пошук. `My` відображає заявки в особистому кабінеті. Це є досить зручним для користувачів. `Create` показує форму створення заявки. `Store` призначений для збереження останньої в базі. `Edit` відображає форму редагування заявки. `Update` оновлює останню в базі. `Destroy` деактивує заявку та зберігає результат опитування.

Рисунок 3.6 – Файл ApplicationController.php

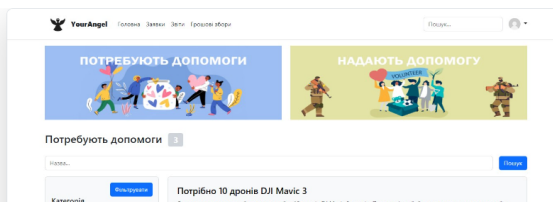
У додатку Г надано програмний код для основних модулів web-додатку підтримки волонтерської діяльності.

3.3 Використання web-додатку

При завантаженні web-додатку волонтерської допомоги «YourAngel» за посиланням: <https://yourangelua.000webhostapp.com/>, користувач потрапляє на головну сторінку (рис. 3.7). Там представлені інформаційні блоки та зображення, які описують діяльність даного ресурсу. Із головної сторінки користувач має можливість перейти до наступних розділів за допомогою меню (рис. 3.8).

ЛАСКАВО ПРОСИМО

Ми тут, щоб надати підтримку та допомогу людям, які постраждали внаслідок війни. Україна зазнала великих випробувань, і наша мета - зібрати разом руки та серця, щоб допомогти відновити життя тих, хто потребує допомоги найбільше.

[Усі заявки](#)
[Створити нову](#)


Опинився в скрутній ситуації?

Тоді тобі до нас! Небайдужі люди з радістю допоможуть вирішити будь-яку проблему. Розміщуй оголошення про необхідну допомогу та отримуй те, чого критично не вистачає.



Приєднуйся до нас

Не втрачайте часу - приєднуйтесь до нашої волонтерської спільноти вже сьогодні! Реєструйся, щоб допомагати та отримувати допомогу від інших вже зараз!

[Зареєструватись](#)

Ви приймаєте умови користування сайтом

Рисунок 3.8 – Меню web-додатку

Для отримання доступу до всього наявного функціоналу користувач має зареєструватися. Або здійснити вхід у вже існуючий акаунт. Форми реєстрації та авторизації представлені на рисунках 3.9-3.10.

Реєстрація

Ім'я

Тарас

Прізвище

Шевченко

Email

example@mail.com

Номер телефону

+380998887766

Пароль

.....

Зареєструватись

Вже зареєстровані? [Увійти](#)

Рисунок 3.9 – Форма реєстрації

Вхід

Email

example@mail.com

Пароль

[Забули пароль?](#)

.....

Увійти

Немає акаунту? [Зареєструватися](#)

Рисунок 3.10 – Форма авторизації

Наступним пунктом меню є «Заявки» (рис. 3.11). Це сторінка з оголошеннями від людей, які потребують допомоги (категорія «Потребують допомоги») та користувачів, які мають можливість допомагати (категорія «Надають допомогу»). Для полегшення пошуку необхідної заявки у web-додатку присутня фільтрація. Наприклад, за категоріями допомоги та місцем розташування автора оголошення, та форма для пошуку за назвою.

The screenshot displays the 'YourAngel' website interface. At the top, there is a navigation bar with the logo and menu items: 'Головна', 'Заявки', 'Звіти', and 'Грошові збори'. A search bar is located on the right side of the header.

Below the header, there are two main banners: 'ПОТРЕБУЮТЬ ДОПОМОГИ' (Need help) and 'НАДАЮТЬ ДОПОМОГУ' (Offer help). The 'ПОТРЕБУЮТЬ ДОПОМОГИ' banner is highlighted with a blue background and a '3' badge, indicating three active requests.

The main content area is divided into a filter sidebar on the left and a list of requests on the right. The filter sidebar includes sections for 'Фільтрувати' (Filter), 'Категорія' (Category), and 'Область' (Region), each with a list of checkboxes and counts.

The list of requests includes:


- Для прихистку внутрішньопереселеній осіб потребуються продовольчі товари** (Need food for internally displaced persons). Location: Ходорів, Львівська область. Date: 05 червня, 2023.
- Необхідне авто** (Need a car). Location: Бахмут, Донецька область. Date: 05 червня, 2023.
- Потребую одяг на хлопчика 3-4роки** (Need clothes for a 3-4 year old boy). Location: Чернігів, Чернігівська область. Date: 05 червня, 2023.

At the bottom of the page, there is a footer with the text '© 2023 YourAngel' and a navigation bar with links: 'Головна', 'Заявки', 'Звіти', 'Грошові збори', 'Нова заявка', and 'Мої заявки'.


Рисунок 3.11 – Сторінка «Заявки» категорія «Потребують допомоги»

Клікнувши на оголошення відкривається його окрема сторінка (рис. 3.12). Окрім опису потреби можна ознайомитися з контактними даними автора та залишити коментар. Функція публічного листування під оголошеннями доступна тільки для зареєстрованих користувачів. Волонтер може не тільки написати коментар, а й за потреби видалити його.

YourAngel Головна Заявки Звіти Грошові збори



Користувач
Віктор Кобилинський
+380507878272
 На сайті з 23 травня 2023 року

Місцезнаходження
📍 **Хмельницький**
 Хмельницька область 

📅 23 травня, 2023 Автівки


Безкоштовно віддам автомобіль на потреби ЗСУ


Маю в наявності Honda CR-V 2001 року. Машина на ходу та не потребує ремонту. Передам автомобіль особисто військовим. Прошу людей, які хочуть машину для себе, не турбувати. Телефонуйте або пишіть в телеграм.

Коментарі


Напишіть ваш коментар

Додати




Валерія Сидоренко
 1 секунда тому Ви 

маю таку саму машину, крута)



Віктор Кобилинський
 1 хвилина тому Власник

Дякую, очікую на дзвінок



Your Angel
 2 хвилини тому Адмін

Ми можемо скontaktувати з військовими, які потребують автівку

Головна Заявки Звіти Грошові збори [Нова заявка](#) [Мої заявки](#)

© 2023 YourAngel

Рисунок 3.12 – Сторінка оголошення

Розділ оголошень наповнюється користувачами web-додатку з їх особистого кабінету (рис. 3.13). Останній дозволяє створити заявку(рис. 3.14), відредагувати її та видалити. Для безпеки інших користувачів усі оголошення після створення та внесення змін проходять перевірку адміністратором web-додатку.

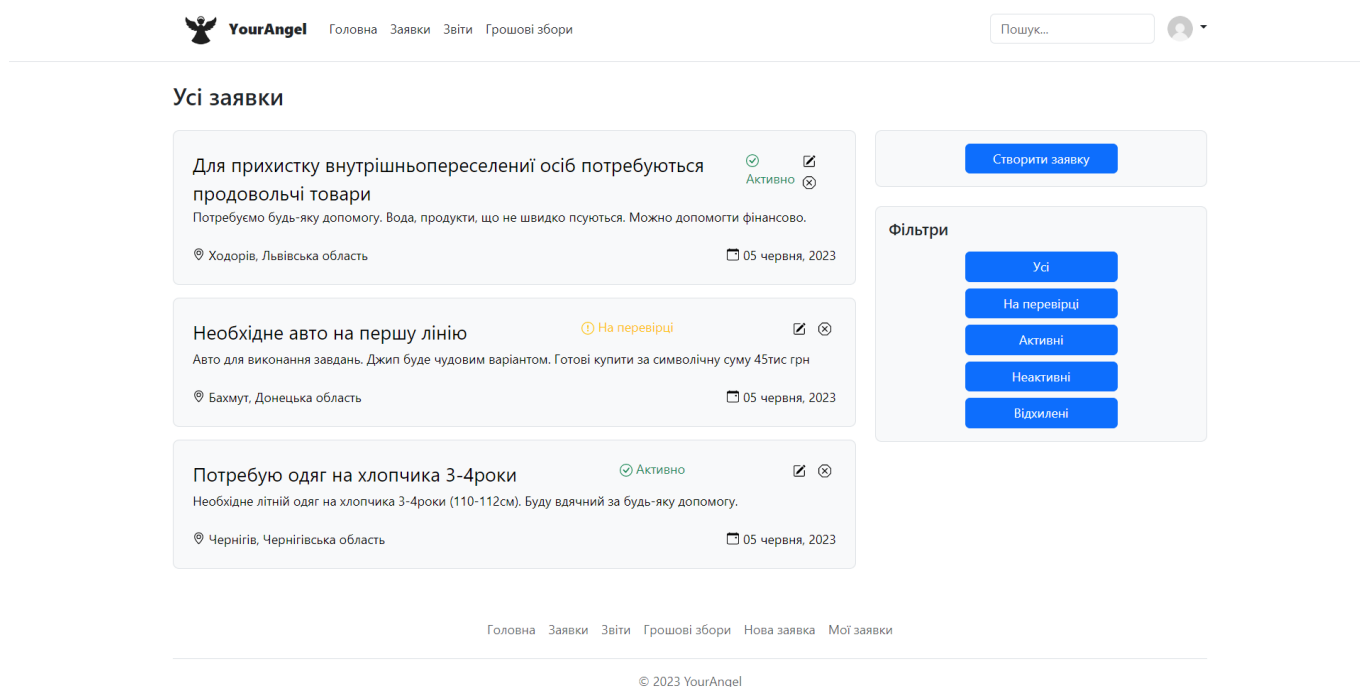


Рисунок 3.13 – Особистий кабінет користувача

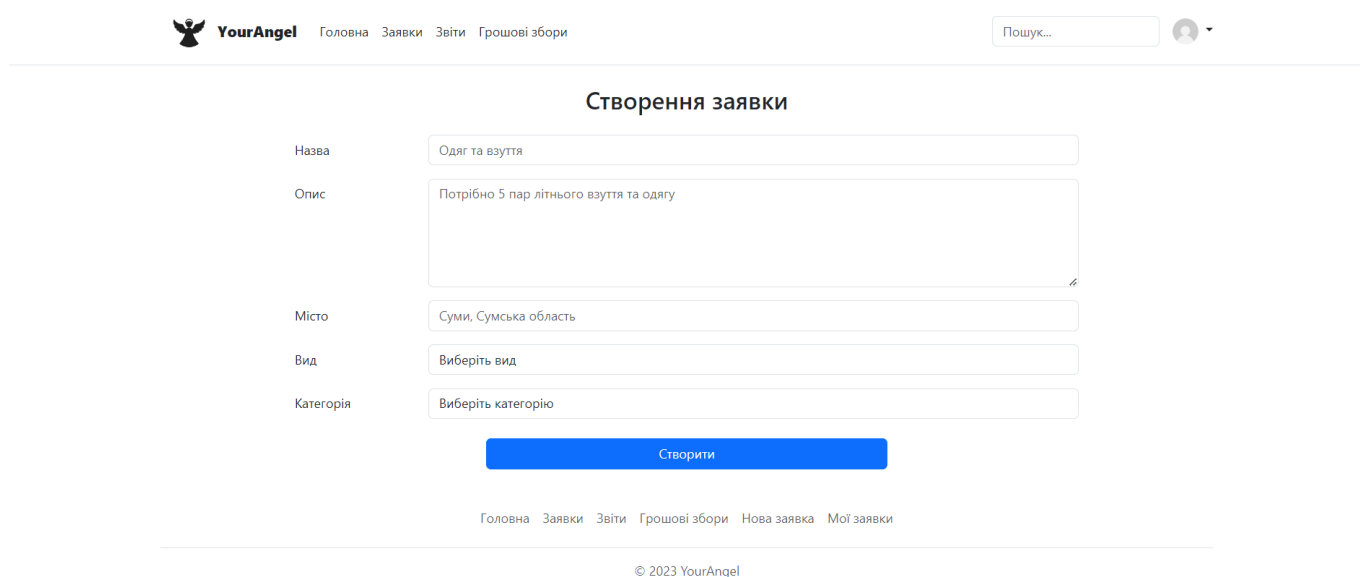


Рисунок 3.14 – Форма створення оголошення

Для створення звітності (рис. 3.15), яка допомогла б показати успішність даного проекту, розроблено форми з запитаннями (рис. 3.16-3.17). для збору інформації. Якщо у користувача виникає потреба видалити оголошення, то він має відповісти на декілька питань, які потім будуть відображені у розділі «Звіти». Для розвитку довіри до даного ресурсу у аналітиці web-додатку відображається графік кількості створених заявок за останній тиждень (рис. 3.15.).

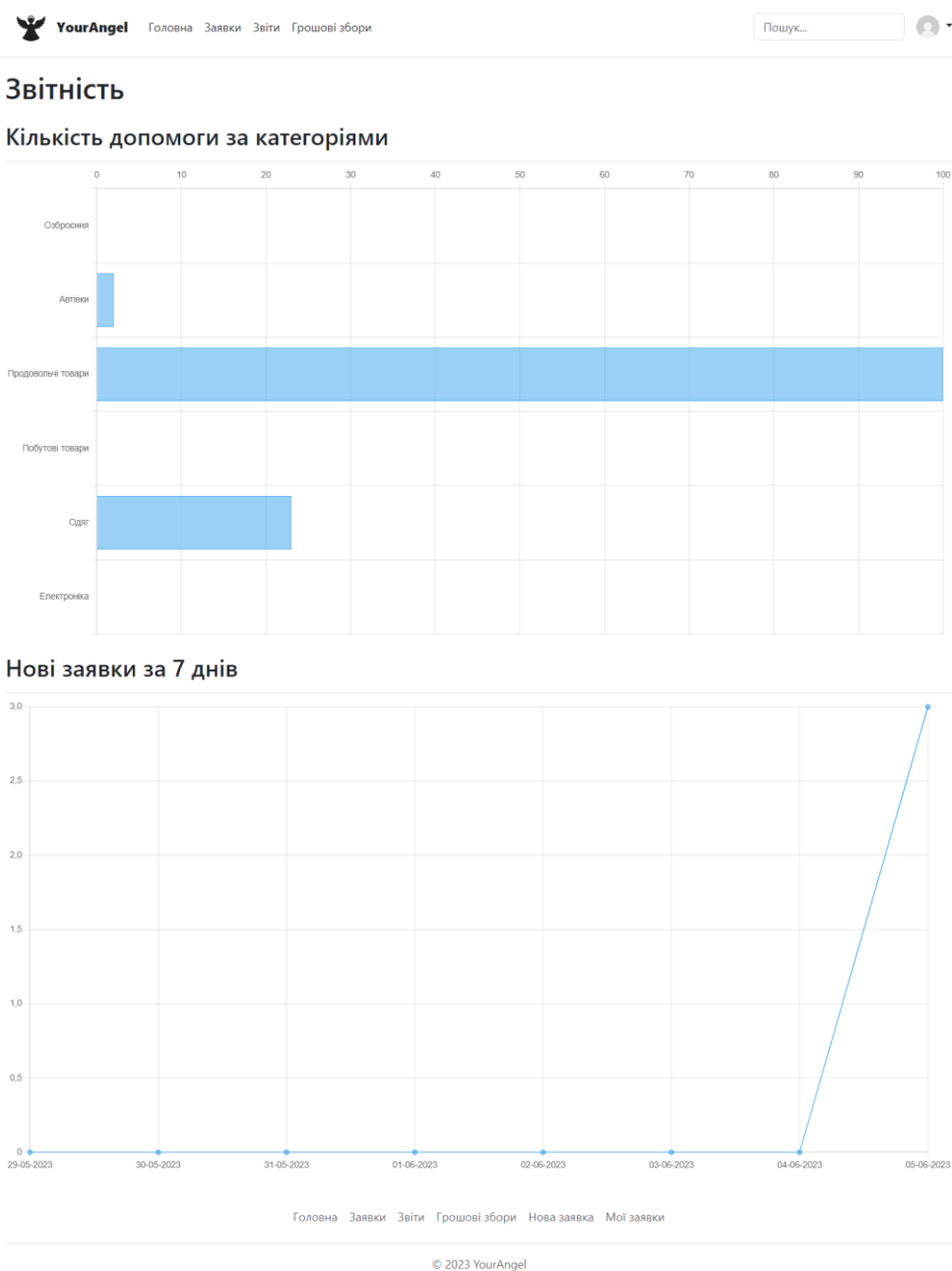


Рисунок 3.15 – Сторінка звітності

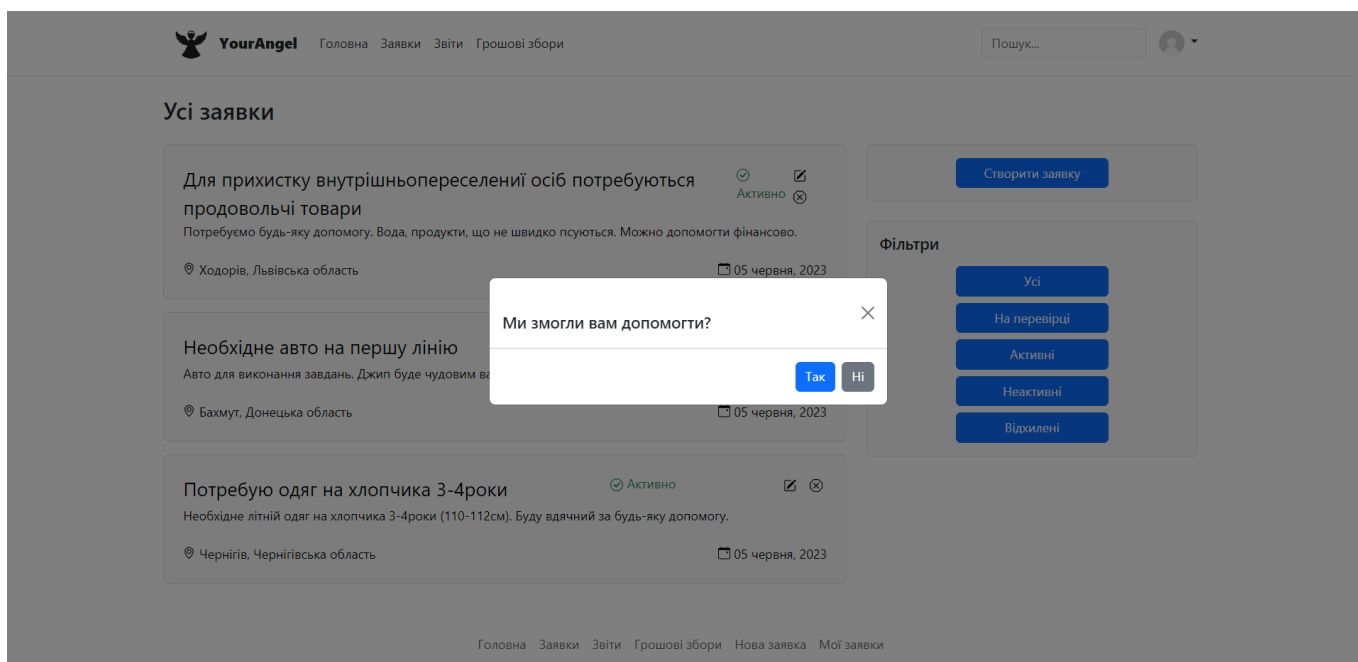


Рисунок 3.16 – Діалогове вікно опитування при видаленні оголошення. Частина 1

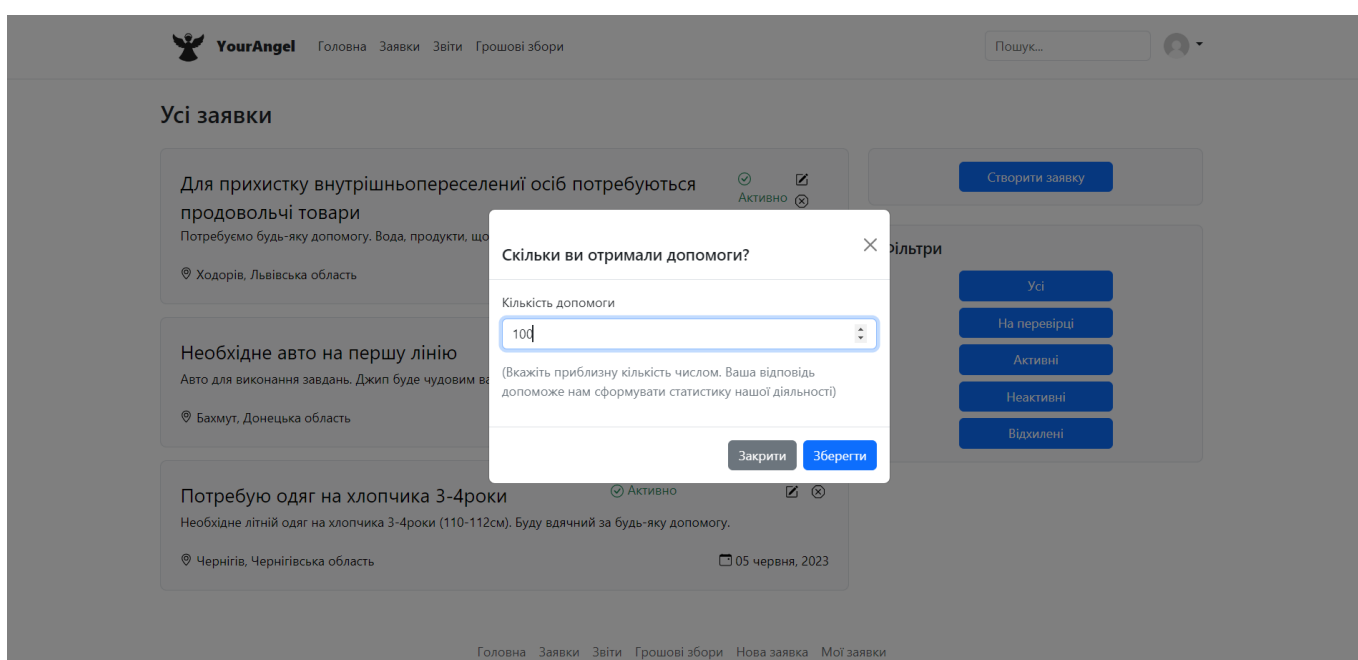


Рисунок 3.17 – Діалогове вікно опитування при видаленні оголошення. Частина 2

У web-додатку передбачена адміністративна панель, за допомогою якої можна створювати грошові збори (рис. 3.18) та підтверджувати/відхиляти

заявки (рис. 3.19). Оскільки платформа «YourAngel» розроблена для взаємодії людей і не позиціонує себе, як повноцінна волонтерська організація, то такий функціонал як грошові збори призначений, щоб допомагати іншим волонтерським організаціям.

The screenshot shows the 'YourAngel' website interface for creating a monetary collection. The header includes the logo and navigation links: Головна, Заявки, Звіти, Грошові збори. A search bar and user profile icon are on the right. The main heading is 'Створення грошового збору'. The form consists of four input fields: 'Назва' (Name) with the value 'Збір на 10 дронів', 'Опис' (Description) with the value 'Опис збору', 'Сума' (Amount) with the value '10000', and 'Дата закінчення' (End Date) with the value 'дд.мм.рррр'. A blue 'Створити' button is positioned below the form. At the bottom, there are navigation links: Головна, Заявки, Звіти, Грошові збори, Нова заявка, Мої заявки, and a copyright notice '© 2023 YourAngel'.

Рисунок 3.18 – Форма створення грошових зборів

The screenshot shows the 'YourAngel' website interface with a modal dialog box titled 'Відхилення заявки' (Request Rejection). The dialog box has a close button (X) in the top right corner. It contains a text input field for the reason of rejection, with the text 'Причина відмови' (Reason for refusal) above it and 'Ми не зможемо опублікувати Ваше оголошення. Воно не ві...' (We cannot publish your advertisement. It is not vi...) inside the field. Below the input field are two buttons: 'Закрити' (Close) and 'Відхилити' (Reject). The background shows the 'Заявки на перевірку' (Requests for review) section with a listing for 'Хочу нове авто' (I want a new car) and a 'Фільтри' (Filters) section with buttons for 'На перевірку' (For review), 'Активні' (Active), 'Неактивні' (Inactive), and 'Відхилені' (Rejected). The footer includes navigation links: Головна, Заявки, Звіти, Грошові збори, Нова заявка, Мої заявки, and a copyright notice '© 2023 YourAngel'.

Рисунок 3.19 – Процес відхилення заявок

Оскільки адміністративна панель передбачає створення грошових зборів, то через головне меню можна перейти до сторінки «Грошові збори» (рис. 3.20). Для переказу коштів розроблений симулятор. Зареєстрований користувач нажимає кнопку «Поповнити» у діалоговому вікні (рис. 3.21), вводить суму для поповнення та натискає «Оплатити». Для підключення повноцінних платіжних систем необхідно відкривати спеціальні рахунки, що на даному етапі не є доцільним.

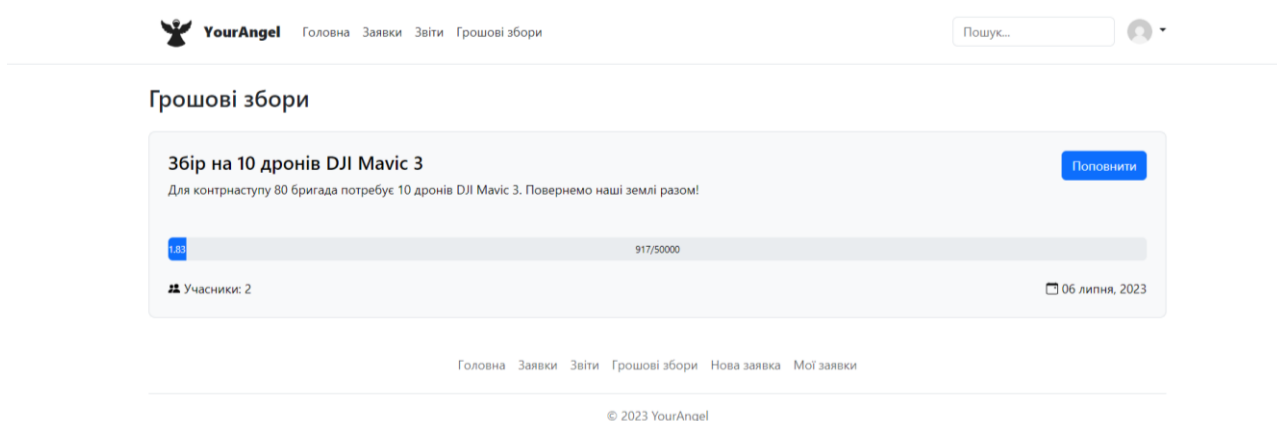


Рисунок 3.20 – Сторінка «Грошові збори»

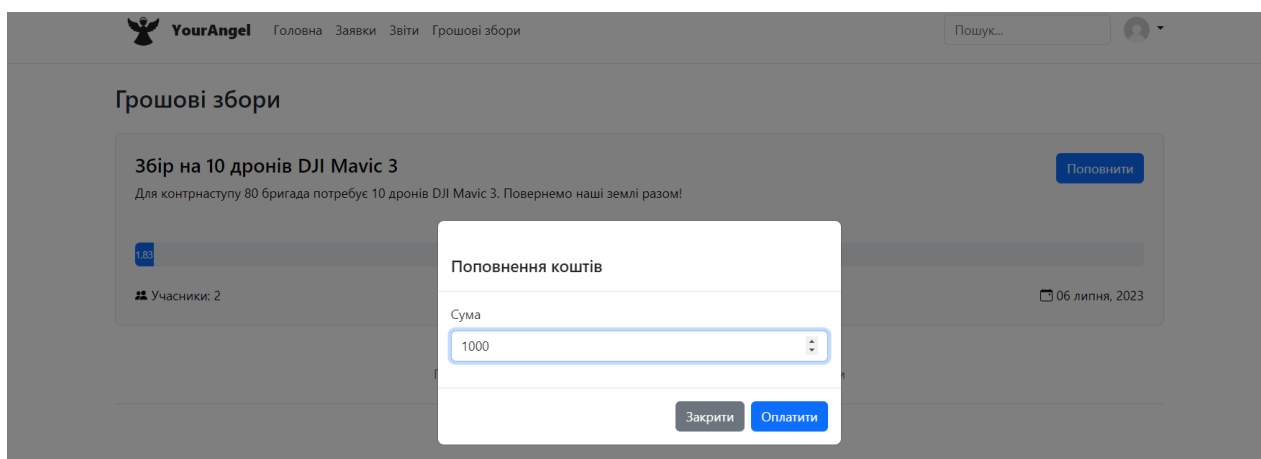


Рисунок 3.21 – Симулятор донатів

Для впевненості у працездатності web-додатку було проведено його тестування. У процесі перевірялась коректність відображення контенту, функціональність форм та кнопок, взаємодія з БД та інші аспекти роботи даного програмного продукту. У результаті перевірки помилок роботи web-додатку не виявлено.

ВИСНОВКИ

У результаті виконання дипломної роботи було розроблено web-додаток підтримки волонтерської діяльності.

Під час виконання даної роботи було проведено аналіз предметної області, досліджено актуальність розробки, проаналізовано продукти-аналоги, сформовано мету та задачі проєкту. Розроблено технічне завдання, де описані функціональні та технічні вимоги до створення web-додатку підтримки волонтерської діяльності (додаток А). Також проведено планування робіт щодо реалізації цього проєкту (додаток Б).

У другій частині виконано структурно-функціональне моделювання, визначено варіанти використання web-додатку та спроектовано базу даних. У результаті були змодельовані такі діаграми: контекстна діаграма IDEF0 та її декомпозиція, діаграма послідовності, діаграма варіантів використання та логічна модель бази даних.

У третьому розділі описано архітектуру web-додатку та процес розробки. Продемонстровано використання web-додатку підтримки волонтерської діяльності. Коротко наведено результати тестування.

Web-додаток було розміщено на хостингу та відкрито для публічного доступу за посиланням: <https://yourangelua.000webhostapp.com/>.

Використання розробленого програмного продукту дозволить охопити ширшу аудиторію та потенційно залучити більше підтримки.

Результати роботи були представлені на конференції «ІМА-2023» (додаток В).

У додатку Г надано лістинг для основних модулів web-додатку підтримки волонтерської діяльності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Здоров'я і суспільство в умовах війни // Збірник наукових статей / Центральнуукраїнський інститут розвитку людини Університету «Україна». Кропивницький, 2022. С. 149-151. [Електронний ресурс] – Режим доступу до ресурсу: https://cnej.gov.md/sites/default/files/zbirnik_statey-2022._1.pdf#page=150 (дата звернення: 30.04.2023).
2. Формування соціальної згуртованості територіальної громади шляхом розвитку волонтерської діяльності: Львів, 2023. [Електронний ресурс] – Режим доступу до ресурсу: https://er.ucu.edu.ua/bitstream/handle/1/3714/Tsebenko_mag.pdf?sequence=1 (дата звернення: 30.04.2023).
3. Humanitarian and medical aid to ukraine [Електронний ресурс] – Режим доступу до ресурсу: <https://helpukraine.center/> (дата звернення: 02.05.2023).
4. Повернись живим [Електронний ресурс] – Режим доступу до ресурсу: <https://savelife.in.ua/> (дата звернення: 02.05.2023).
5. Армія SOS [Електронний ресурс] – Режим доступу до ресурсу: <https://armysos.com.ua/uk/> (дата звернення: 03.05.2023).
6. Мрія вільних людей [Електронний ресурс] – Режим доступу до ресурсу: <https://mriya.od.ua/> (дата звернення: 03.05.2023).
7. HTML Довідник тегів // W3schoolsUA. Українською [Електронний ресурс] – Режим доступу до ресурсу: <https://w3schoolsua.github.io/tags/index.html#gsc.tab=0> (дата звернення: 03.05.2023).
8. CSS Підручник // W3schoolsUA. Українською [Електронний ресурс] – Режим доступу до ресурсу: <https://w3schoolsua.github.io/css/index.html#gsc.tab=0> (дата звернення: 03.05.2023).
9. JavaScript Tutorial // W3schools [Електронний ресурс] – Режим доступу до ресурсу: <https://www.w3schools.com/js/> (дата звернення: 03.05.2023).

10. The PHP Framework for Web Artisans [Електронний ресурс] – Режим доступу до ресурсу: <https://laravel.com/docs/10.x> (дата звернення: 03.05.2023).
11. Get started with Bootstrap // Bootstrap [Електронний ресурс] – Режим доступу до ресурсу: <https://getbootstrap.com/docs/5.3/getting-started/introduction/> (дата звернення: 03.05.2023).
12. SQL Tutorial // W3schools [Електронний ресурс] – Режим доступу до ресурсу: <https://www.w3schools.com/sql/default.asp> (дата звернення: 03.05.2023).
13. МЕТОДОЛОГІЯ IDEF0 // Stud.com.ua [Електронний ресурс] – Режим доступу до ресурсу: https://stud.com.ua/87184/ekonomika/metodologiya_idef0 (дата звернення: 15.03.2023).
14. Мова UML. Діаграма використання [Електронний ресурс] – Режим доступу до ресурсу: <http://p4ilka.blogspot.com/2018/12/uml.html> (дата звернення: 16.05.2023).
15. MVC Framework - Introduction // Tutorialspoint [Електронний ресурс] – Режим доступу до ресурсу: https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm (дата звернення: 30.05.2023).
16. Visual Studio Code [Електронний ресурс] – Режим доступу до ресурсу: <https://code.visualstudio.com/docs> (дата звернення: 01.05.2023).
17. Постановка цілей по SMART – приклади, критерії [Електронний ресурс] – Режим доступу до ресурсу: <https://pdatu.edu.ua/images/vihovna-robota/psiholog/ps10.pdf> (дата звернення: 10.03.2023).
18. What is a Work Breakdown Structure // WorkBreakdownStructure [Електронний ресурс] – Режим доступу до ресурсу: <https://www.workbreakdownstructure.com/> (дата звернення: 14.03.2023).
19. Структуризація проекту [Електронний ресурс] – Режим доступу до ресурсу: https://www.oa.edu.ua/download/Lektsija_5.PDF (дата звернення: 14.03.2023).

20. Що таке діаграма Ганта та як нею правильно користуватися [Електронний ресурс] – Режим доступу до ресурсу: <https://nachasi.com/creative/2020/09/03/gantt-chart/> (дата звернення: 14.03.2023).

ДОДАТОК А

ТЕХНІЧНЕ ЗАВДАННЯ

на розробку

«Web-додаток підтримки волонтерської діяльності»

1. Призначення й мета web-додатку підтримки волонтерської діяльності

1.1 Призначення web-додатку

Web-додаток призначений для підтримки волонтерської діяльності.

1.2 Мета створення web-додатку

Головна мета проекту – це створення web-додатку для підтримки волонтерської діяльності, використання якого забезпечить її належну організацію шляхом автоматизації процесу взаємодії волонтерів та людей, яким необхідна допомога.

1.3 Цільова аудиторія

Цільовою аудиторією даного проекту є свідомі громадяни, які мають можливість і бажання добровільно допомагати, та люди або організації, які потребують цієї помічі.

2 Вимоги до проекту

2.1 Вимоги до проекту в цілому

2.1.1 Вимоги до структури й функціонування

Web-додаток для підтримки волонтерської діяльності повинен бути реалізований за допомогою web-інструментів та забезпечувати визначений набір функціональних можливостей.

Кінцевий продукт даного проекту має бути представлений web-додатком, який має інтуїтивно-зрозумілий інтерфейс та увесь необхідний функціонал для забезпечення організації добровільної волонтерської допомоги.

2.1.2 Вимоги до персоналу

Адміністратори web-додатку не повинні мати особливих технічних навичок як для роботи з ним, так і його підтримки. Єдиною вимогою є наявність базового вміння користування персональним комп'ютером та web-браузером.

2.1.3 Вимоги до збереження інформації

Уся інформація надана у web-додатку повинна зберігатися у базі даних реалізованій засобами системи управління базами даних MySQL.

2.1.4 Вимоги до розмежування доступу

Розроблюваний web-додаток має бути загальнодоступним у мережі Інтернет. Права доступу до інформації розмежовані за групами користувачів: адміністратор, відвідувач та зареєстрований користувач. Перший має необмежений доступ до даних із правами підтвердження та відхилення заявок на допомогу, та можливістю створювати грошові збори. Доступ до адміністративної панелі надається за спеціальним логіном та паролем.

Будь-який відвідувач web-додатку може переглянути усі оголошення та звітність. Для цього він може скористатися пошуковою системою та фільтрами. Зареєстрованому користувачу доступні всі можливості відвідувача, а також дозвіл на розміщення власних заявок на допомогу, можливість залишити коментар до оголошення та участь у грошових зборах.

2.2 Структура web-додатку

2.2.1 Загальна інформація про структуру web-додатку

До структури web-додатку входять усі його web-сторінки, які є загальнодоступними, та адміністративна панель для адміністраторів web-додатку.

Перелік сторінок web-додатку наступний:

- «Головна» сторінка містить навігаційне меню, яке закріплене на кожній сторінці web-додатку, інформаційні блоки про діяльність даного онлайн сервісу, контактні дані для звернення з питань роботи web-додатку, кнопки переходу до особистого кабінету або реєстрації;
- на сторінці «Заявки» розміщені оголошення, які пройшли модерацію та чекають на підтримку небайдужих. Для зручності пошуку необхідної заявки, на сторінці передбачений пошук за назвою, фільтрація по областях та категоріях;
- сторінка «Звітність» містить графічні представлення успішної діяльності користувачів web-додатку. Конкретні цифри гуманітарної, грошової допомоги, яка була зібрана через даний онлайн сервіс;
- «Грошові збори» сторінка для здійснення фінансової допомоги;
- для переходу в особистий кабінет користувач має пройти форму входу або реєстрації. Після успішної авторизації він має можливість взаємодіяти зі своїми вже створеними заявками та розміщувати нові оголошення;
- адмін-панель web-додатку містить сторінки управління оголошеннями (підтвердження/відхилення) та користувачами, із можливістю заблокувати акаунт ненадійного учасника спільноти.

2.2.2 Навігаційне меню

Для зручної навігації повинно бути створене меню, яке забезпечить швидке переміщення користувача по всім доступним сторінкам web-додатку. Воно має бути закріплене й розташовуватися зверху (у шапці) на кожній сторінці.

2.2.3 Управління контентом

Управління оголошеннями web-додатку має здійснюватися за допомогою адміністративної панелі. Усе інформаційне наповнення онлайн сервісу має міститися у базі даних.

2.2.4 Дизайн web-додатку

Дизайн web-додатку має бути виконаний у мінімалістичному та сучасному стилі. Основними кольорами є білий та чорний з патріотичними деталями жовто-блакитного кольору.

Види та розміри шрифтів мають бути комфортними для перегляду. Інформаційні блоки, графічні матеріали та інші елементи web-сторінок повинні мати зручне та логічне розташування. Шаблон майбутнього програмного продукту зображено на рисунку А.1.

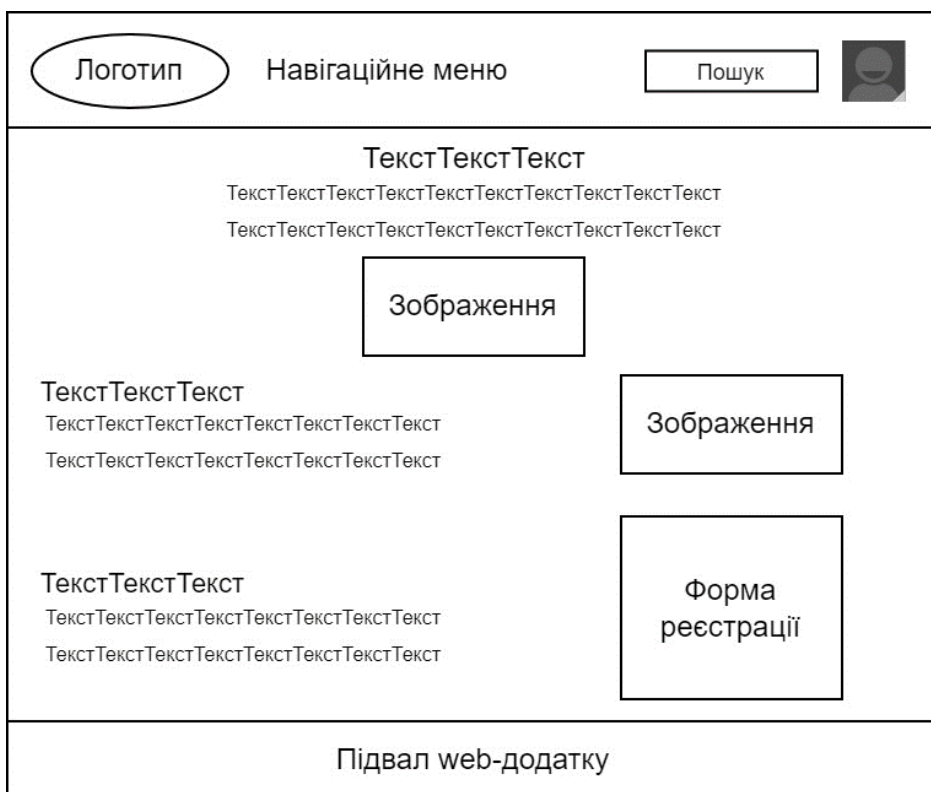


Рисунок А.1 – Схема головної сторінки

2.2.4 Система навігації (карта web-додатку)

Карта web-додатку зображена на рисунку А.2.

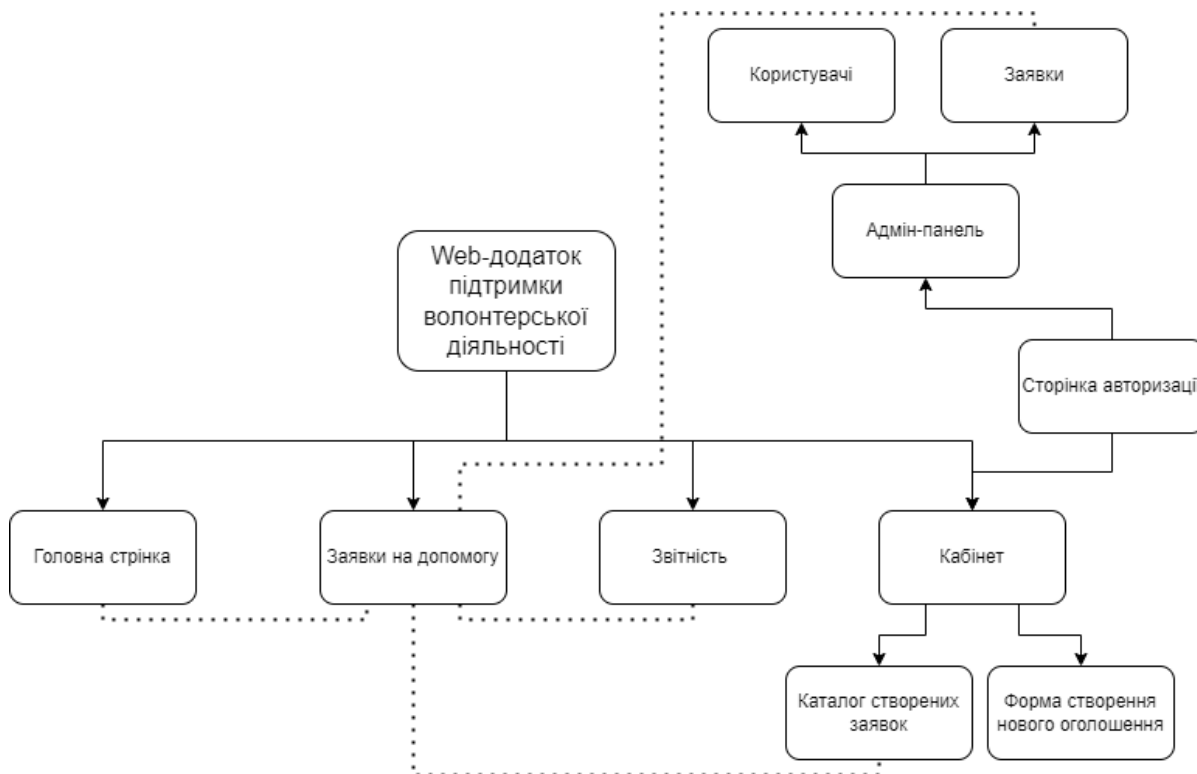


Рисунок А.2 – Система навігації

2.3 Вимоги до видів забезпечення

2.3.1 Вимоги до лінгвістичного забезпечення

Увесь текст у web-додатку має бути виконаний українською мовою.

2.3.2 Вимоги до програмного забезпечення

Програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам:

- web-браузер: Internet Explorer 7.0 і вище, або Firefox 3.5 і вище, або Opera 9.5 і вище, або Chrome 2 і вище, Microsoft Edge 38 і вище.
- включена підтримка JavaScript, Flash і cookies.

2.3.3 Вимоги до апаратного забезпечення

Для користування web-додатком можна використовувати комп'ютер (ПК) або смартфон. Апаратне забезпечення серверної частини ПК повинне задовольняти наступним вимогам:

- процесор, базова частота якого є не нижче 2,4 ГГц та частота пам'яті від 2400 МГц;
- оперативна пам'ять не менше Гб;
- жорсткий диск ємністю не менше 120 Гб;
- ОС: Microsoft Windows XP та вище, або Linux (Peppermint OS, elementary OS, Pop OS, Linux Lite, Endless OS, Ubuntu LTS Release с GNOME, Deepin, Kubuntu LTS Release, Linux Mint Cinnamon Edition, Zorin OS), або MacOS 10.7 та вище;
- web-сервер Apache з модулем MOD_REWRITE;
- не менш 500 МБ вільного місця на диску.

Мобільні пристрої повинні мати наступне:

- Android версія 5 та вище, від 2 Гб оперативної пам'яті, мінімальний чотирьохядерний з частотою 1,5 ГГц (32- або 64-розрядний);
- iOS версія 10 та вище, від 2 Гб оперативної пам'яті, мінімальний двоядерний 64-розрядний процесор.

2.4 Вимоги до функціонування системи

2.4.1 Потреби користувача

Потреби користувача, визначені на основі рішення замовника, представлені у таблиці А.1.

Таблиця А.1 – Потреби користувача

| ID | Потреби користувача | Джерело |
|-----------|--|-----------------------|
| UN-01 | Перегляд розміщених оголошень | Клієнт, Відвідувач |
| UN-02 | Пошук, фільтрація оголошень | Клієнт, Відвідувач |
| UN-03 | Перегляд контактів авторів оголошень | Клієнт, Відвідувач |
| UN-04 | Перегляд звітності роботи web-додатку | Клієнт, Відвідувач |
| UN-05 | Створення, редагування та видалення заявок на допомогу | Клієнт |
| UN-06 | Перегляд інформації про web-додаток та його діяльність | Клієнт, Відвідувач |
| UN-07 | Написання коментарів | Клієнт |
| UN-08 | Участь у грошових зборах | Клієнт |
| UN-09 | Створення грошових зборів | Адміністратор |
| UN-10 | Підтвердження та відхилення заявок, що надходять | Адміністратор |

2.4.2 Функціональні вимоги

Проаналізувавши потреби користувачів web-додатку було визначено наступні вимоги:

- наявність реєстрації та авторизації користувачів;
- можливість в зручному та зрозумілому вигляді ознайомитися зі звітністю роботи web-додатку;
- пошук та фільтрація за назвою, типом допомоги, геолокацією;
- можливість онлайн зв'язатися з автором оголошення, щоб запропонувати допомогу;

- наявність панелі адміністратора для підтвердження або відхилення нових оголошень про допомогу;
- наявність грошових зборів з можливістю задонатити;
- можливість автоматизованого формування графіків для звітування про роботу web-додатку.

3 Склад і зміст робіт зі створення web-додатку підтримки волонтерської діяльності

Детальний опис етапів створення web-додатку наведено в таблиці А.2.

Таблиця А.2 – Етапи створення web-додатку

| № | Склад і зміст робіт | Строк розробки |
|----------|---|-----------------------|
| 1 | Розробка шаблону web-додатку | 5 днів |
| 2 | Верстка сторінок web-додатку | 9 днів |
| 3 | Розробка модулю створення оголошень | 6 днів |
| 4 | Розробка модулю зв'язку між користувачами | 6 днів |
| 5 | Розробка модулю формування звітності | 6 днів |
| 6 | Розробка бази даних | 7 днів |
| 7 | Розробка адміністративної панелі | 6 днів |
| 8 | Текстове наповнення даними | 4 дні |
| 9 | Графічне наповнення даними | 3 дні |
| 10 | Alpha-тестування | 5 днів |
| 11 | Beta-тестування | 4 дні |
| 12 | Розміщення на хостингу | 1 день |
| 13 | Перевірка працездатності | 3 дні |
| 14 | Написання супровідної документації | 9 днів |
| 15 | Реліз web-додатку | 2 дні |
| | Загальна тривалість робіт | 76 днів |

4 Вимоги до складу й змісту робіт із введення web-додатку в експлуатацію

Для створення умов функціонування, при яких гарантується відповідність створюваного web-додатку вимогам технічного завдання (ТЗ) та можливість його ефективної роботи, в організації замовника повинен бути проведений певний комплекс заходів.

Для переносу web-додатку на хостинг необхідно, щоб параметри хостинга відповідали вимогам, зазначеним у ТЗ.

ДОДАТОК Б

Планування робіт

Благодійність вже давно стала звичайною справою для певних жителів України. Допомога хворим дітям, підтримка безпритульних тварин, донати на армію з 2014 року тощо. Цим займались та займаються певні організації та фонди. Але з початком повномасштабної війни в Україні благодійність вийшла на новій рівень. Дехто не може уявити свій день без доната на Збройні сили України або волонтерської діяльності на користь нашої Батьківщини. Тому зараз особливо важливо вдосконалювати процеси волонтерської діяльності, щоб охочих допомагати ставало більше. І це все звичайно наблизить нашу перемогу.

Деталізація мети проекту методом SMART. Для успішності та конкурентоспроможності даного проекту треба на концептуальному етапі правильно визначити його мету за допомогою SMART-методу [17]. Формат постановки SMART-мети для даного проекту наступний:

«Розробити web-додаток підтримки волонтерської діяльності до 05.06.2023 для залучення більшої кількості людей до благодійності та швидкого закриття потреб громадян країни через створення оголошень (заявок на допомогу) та відкритого чату користувачів».

Результати цього представлено в таблиці Б.1.

Таблиця Б.1 – Деталізація мети проекту методом SMART

| | |
|-------------------------|--|
| Specific (конкретна) | Створити web-додаток підтримки волонтерської діяльності, який містить такі функції як створення оголошень, відкритий чат для контактування користувачів, автоматизоване створення звітності. |
|-------------------------|--|

Продовження табл. Б.1.

| | |
|---------------------------------------|---|
| Measurable (вимірювана) | Розробити web-додаток, який збільшить кількість волонтерів серед людей віком від 18 до 65 років на 25%, успішне закриття 200 оголошень на місяць. |
| Achievable (досяжна, узгоджена) | Для виконання проекту у розробника наявні необхідні знання HTML, CSS, мови програмування JavaScript, PHP, баз даних MySQL та навичок написання документації. Ураховуючи доступні ресурсні можливості та обмеження мета є такою, яку можливо досягти. Є затверджене технічне завдання. |
| Relevant (реалістична) | Створений web-додаток автоматизує процес взаємодії волонтерів та людей, яким необхідна допомога, що збільшить кількість охочих допомагати та зменшить терміни закриття оголошень. |
| Time-framed (обмежена в часі) | Ціль має часове обмеження. Термін досягнення мети проекту визначено за регламентом навчального закладу і має бути виконаним до проведення захисту кваліфікаційної роботи, а саме до 5го червня 2023 року. |

Планування змісту робіт. WBS [18] (Work Breakdown Structure – ієрархічна структура робіт) – це графічний вигляд елементів проекту, які згруповані ієрархією у єдине ціле з продуктом проекту. Структура декомпозиції робіт орієнтована на досконале їх виконання по частинам. WBS і сама є ключовою складовою проекту. Вона спрямована на організацію командної роботи. Елементами декомпозиції можуть бути продукти, дані та послуги. Більше того, WBS забезпечує необхідний каркас для подальшої ретельної оцінки термінів, контролю та графіків роботи при створенні календарного плану.

На найвищому (першому) рівні розміщений продукт проекту. Основні дії та заходи, які забезпечують досягнення мети проекту, зафіксовані на другому рівні

декомпозиції. Остання виконується до тих пір, поки роботи не стануть елементарними задачами. Їх також називають простими.

Елементарні роботи – це дії, які мають однозначний чіткий результат. На кожну з них призначена одна відповідальна конкретна особа, для якої можна обчислити витрати праці й тривалість виконання. На рисунку Б.1 представлено WBS проекту розробки web-додатку підтримки волонтерської діяльності.

Планування структури виконавців. Наступним етапом після декомпозиції процесів є розробка організаційної структури виконавців або OBS [19]. Остання є аббревіатурою від Organizational Breakdown Structure. Вона визначається як графічна структура відображення відповідальних осіб. Вони всі є учасниками, які беруть участь у реалізації проекту.

У ролі відповідальних осіб виступають співробітники, які відповідають за організацію та виконання однієї елементарної роботи, яка зазначена у WBS. Кожну з них можна розглядати як окремий проект.

На рисунку Б.2 представлено організаційну структуру планування даного проекту. Список виконавців, які функціонують в проекті описано в таблиці Б.2.

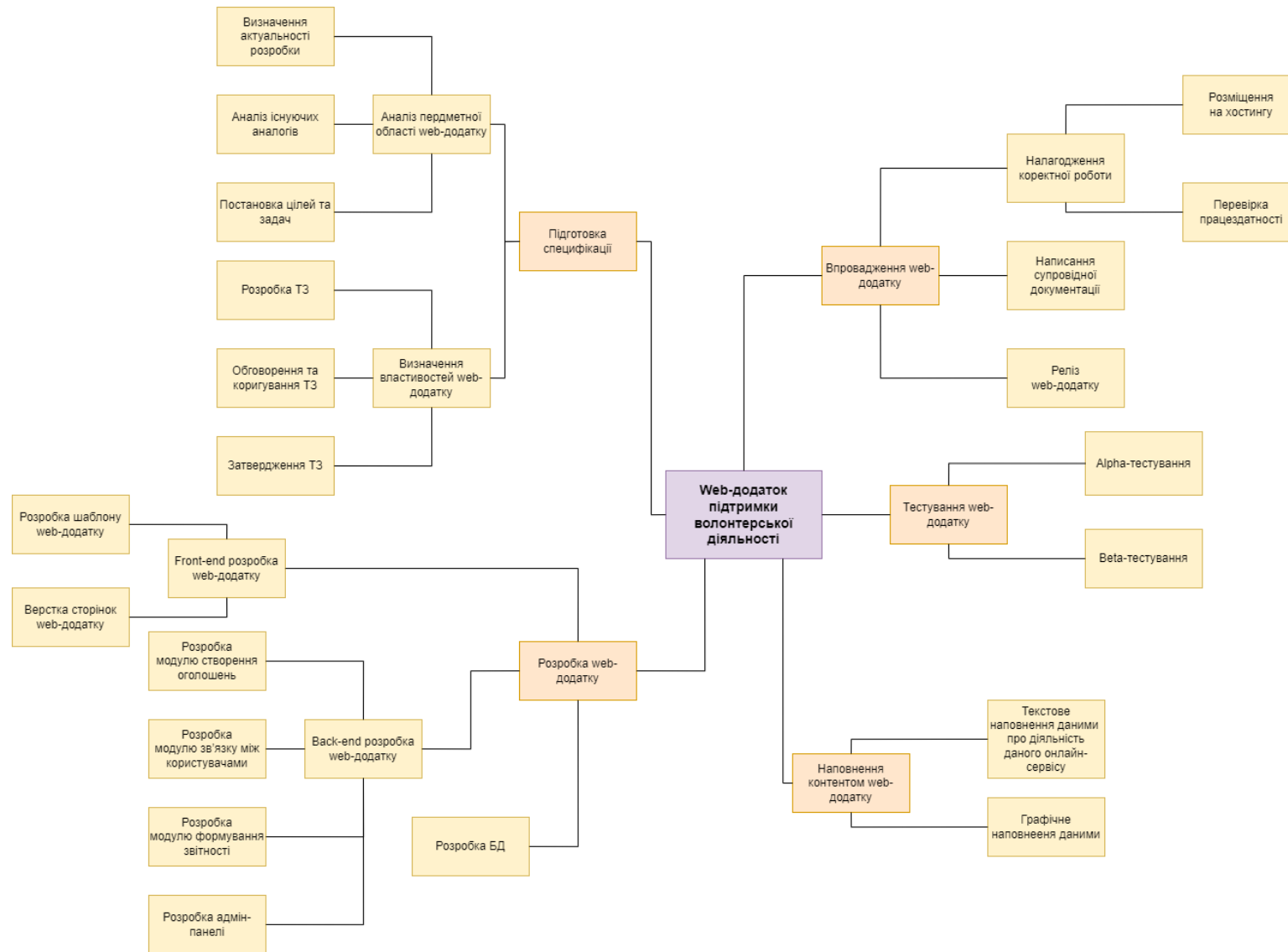


Рисунок Б.1 – WBS-структура робіт проекту

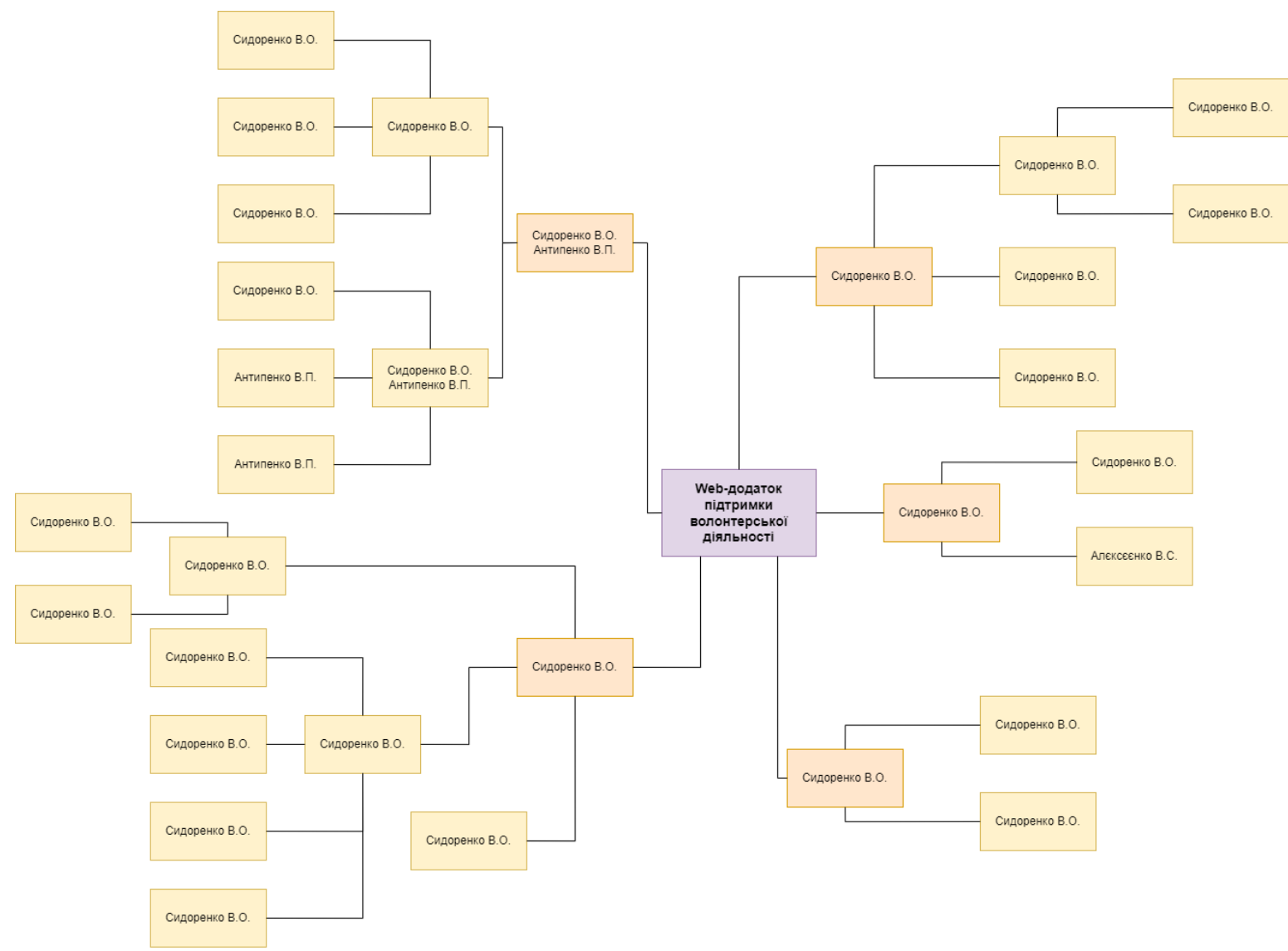


Рисунок Б.2 – OBS-структура робіт проект

Таблиця Б.2 – Виконавці проекту

| Роль | Ім'я | Проектна роль |
|-------------------------------|-----------------|--|
| Розробник | Сидоренко В.О. | Виконує front-end та back-end розробку |
| Проектувальник | Сидоренко В.О. | Виконує проектування бази даних та розробляє структуру web-додатку. |
| Тестувальник в рамках проекту | Сидоренко В.О. | Відповідає за Alpha-тестування функціоналу та дизайну web-додатку. |
| Тестувальник поза проекту | Алексєєнко В.С. | Відповідає за Beta-тестування функціоналу та дизайну web-додатку. |
| Керівник проекту | Антипенко В.П. | Формує завдання на розробку проекту та затверджує документацію. |
| Менеджер проекту | Сидоренко В.О. | Відповідає за виконання термінів, розподіл ресурсів та завдань між учасниками. Виконує збір та аналіз даних. |

Діаграма Ганта. Побудова календарного графіку (діаграми Ганта) [20] є одним з важливих етапів планування проекту, що виглядає як розклад виконання робіт з реальним розподілом дат. Завдяки йому можна отримати достовірне уявлення про тривалість процесів з обмеженнями у ресурсах, урахуванням вихідних днів та свят.

Календарний графік даного проекту представлено на рисунку Б.3.

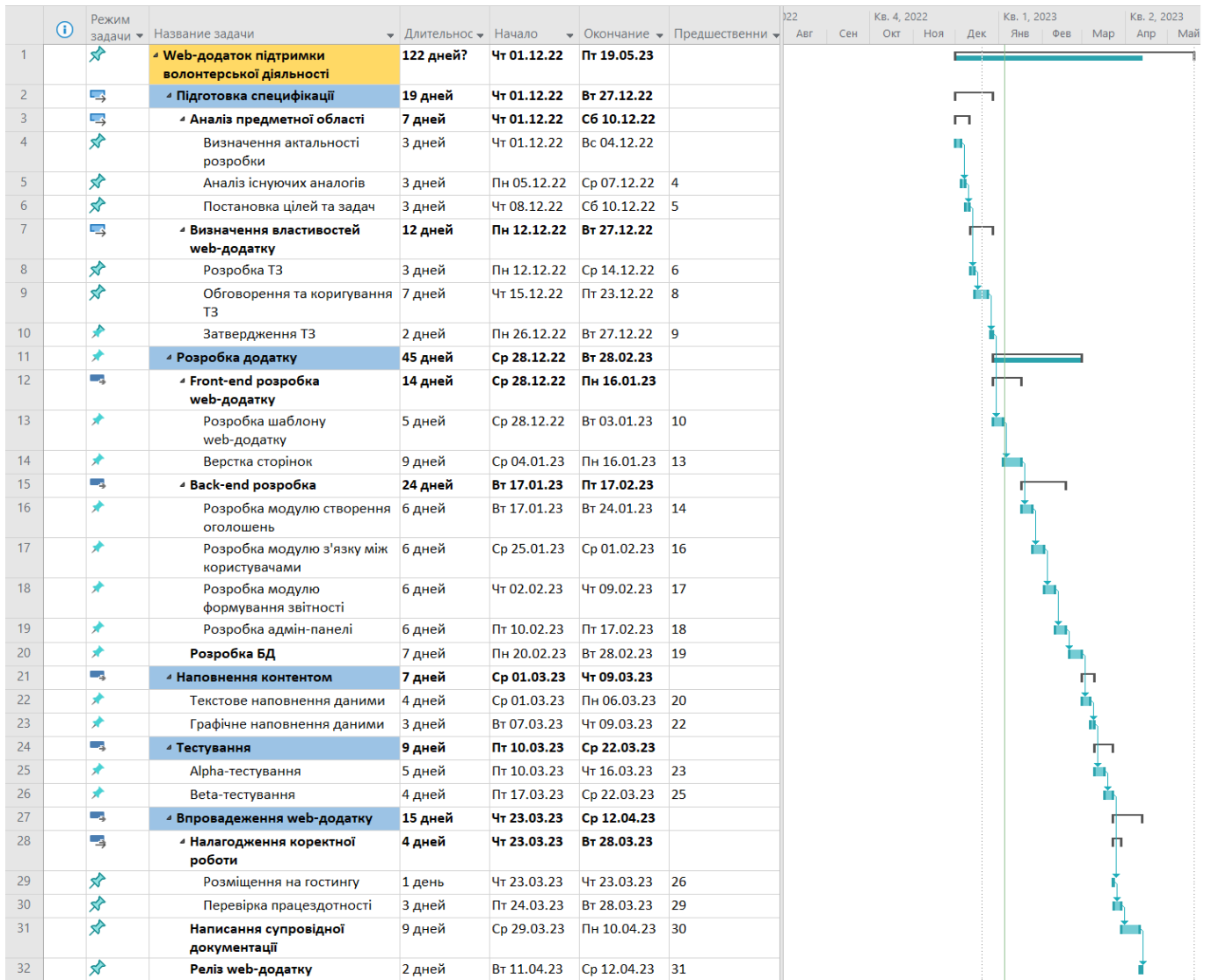


Рисунок Б.3 – Календарний графік проект

Управління ризиками проекту. Під час виконання якісної оцінки ризиків треба визначити ризики, які мають бути усунені якнайшвидше. У залежності від ступеня важливості ризику – реагування буде відповідне. Наступним етапом є виконання кількісного оцінювання ризиків. Кількісне та якісне оцінювання можуть виконувати одночасно або окремо, що залежить від ступеня забезпечення проекту. Таблиця Б.3 включає в себе потенційні ризики майбутнього продукту. У таблиці Б.4 представлено шкалу для класифікації ризиків за величиною впливу на проект та ймовірністю виникнення.

Таблиця Б.3 – Ідентифікація ризиків проекту

| № | Назва ризику |
|----|--|
| 1 | Непорозуміння між розробником та замовником |
| 2 | Поява альтернативного продукту |
| 3 | Нечітке завдання на розробку |
| 4 | Низька кваліфікація розробників |
| 5 | Неоптимальний розподіл часу |
| 6 | Вибір не ефективної технології розробки |
| 7 | Помилки проектування |
| 8 | Збої в роботі програмного забезпечення |
| 9 | Відсутність резервних копій даних |
| 10 | Реалізація непотрібного функціоналу |
| 11 | Аварійні відключення світла |
| 12 | Загроза ракетних та артилерійських обстрілів |
| 13 | Невиконання моніторингу проекту |

Таблиця Б.4 – Шкала оцінювання ризиків за ймовірністю виникнення та величиною впливу.

| Оцінка | Ймовірність виникнення | Вплив ризику | Тип ризику |
|--------|------------------------|--------------|-------------|
| 1 | Низька | Низький | Прийнятні |
| 2 | Середня | Середній | Виправдані |
| 3 | Висока | Високий | Недопустимі |

У результаті планування реагування було отримано матрицю ймовірності виникнення ризиків та впливу ризику, що зображена на рисунку Б.4.

Зеленим кольором на матриці позначають прийнятні ризики, жовтим – виправдані, а червоним – недопустимі.

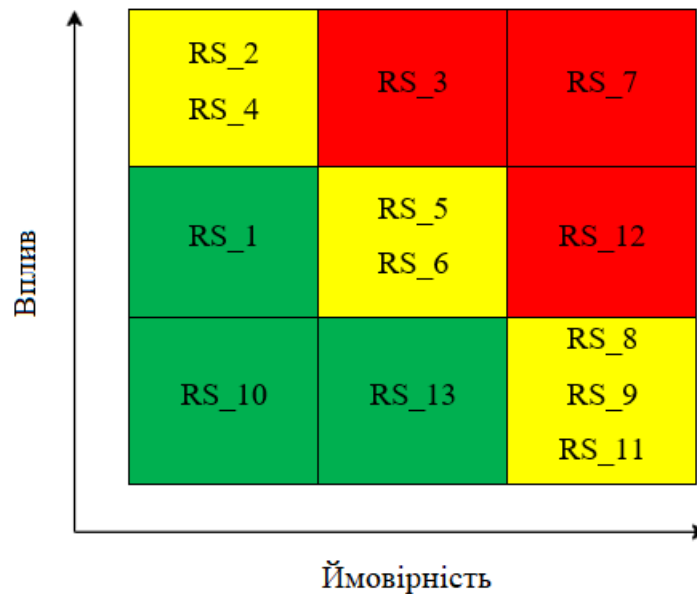


Рисунок Б.4. – Матриця ймовірності

Класифікація ризиків за рівнем, відповідно до отриманого значення індексу, представлена у таблиці Б.5. У таблиці Б.6 описано ризики та стратегії реагування на кожен з них.

Таблиця Б.5 – Шкала оцінювання за рівнем ризику.

| № | Назва | Межі | Ризики, які входять(назва) |
|---|-------------|-------------------|----------------------------|
| 1 | Прийнятні | $1 \leq R \leq 2$ | 1, 10, 13 |
| 2 | Виправдані | $3 \leq R \leq 4$ | 2, 4, 5, 6, 8, 9, 11 |
| 3 | Недопустимі | $6 \leq R \leq 9$ | 3, 7, 12 |

Таблиця Б.6 – Ризики та стратегії реагування

| ID | Статус ризику | Опис ризику | Ймовірність виникнення | Вплив ризику | Ранг ризику | План А | Тип стратегії реагування | План Б |
|------|---------------|---|------------------------|--------------|-------------|---|--------------------------|---|
| RS_1 | Відкритий | Непорозуміння між розробником та замовником | Низька | Середній | 2 | Налагодження гарних ділових стосунків між розробником та замовником. Створення комфортних умов для праці. | Попередження | Оговорення причини непорозуміння та спокійне їх вирішення. |
| RS_2 | Відкритий | Поява альтернативного продукту | Низька | Високий | 3 | Приділити достатню увагу дослідженню аналогів та реалізація унікальних рішень у предметній області. | Прийняття | Розробка додаткового унікального функціоналу. |
| RS_3 | Відкритий | Нечітке завдання на розробку | Середня | Високий | 6 | Ясно і однозначно обговорити із замовником усі види вимог. Періодичний контроль замовником етапів роботи. | Попередження | Внесення правок у розробку. |
| RS_4 | Відкритий | Низька кваліфікація розробників | Середня | Середній | 4 | Переглянути онлайн-ресурси для підвищення рівня знань. | Пом'якшення | Врахувати час на підготовку працівників. Видати літературу, переглянути онлайн-уроки. |

Продовження табл. Б.6

| ID | Статус ризику | Опис ризику | Ймовірність виникнення | Вплив ризику | Ранг ризику | План А | Тип стратегії реагування | План Б |
|------|---------------|---|------------------------|--------------|-------------|---|--------------------------|---|
| RS_5 | Відкритий | Неоптимальний розподіл часу | Середня | Середній | 4 | Звернути особливу увагу на правильність розподілу часу. Правильно визначити пріоритети виконання робіт. Чітко дотримуватися календарного плану. | Пом'якшення | Змінити порядок пріоритетів робіт. Знайти способи оптимізації роботи з вже існуючою розстановкою. Обговорити варіанти внесення поправок до термінів реалізації із замовником. |
| RS_6 | Відкритий | Вибір не ефективної технології розробки | Середня | Середній | 4 | Проаналізувати методи та засоби, для виконання проекту. Обрати зрозумілу та легку в використанні технологію розробки. | Пом'якшення | Виділити час та ресурси на пошуки покращення обраної технології. Застосувати допоміжні ресурси. |
| RS_7 | Відкритий | Помилки проектування | Висока | Високий | 9 | На етапі проектування тісно співпрацювати із замовником та на певних етапах демонструвати поточні результати. | Пом'якшення | Здійснювати проміжний контроль результатів в ході виконання проекту. |
| RS_8 | Відкритий | Збої в роботі програмного забезпечення | Низька | Високий | 3 | Підготувати резерв програмних засобів. Залучити спеціаліста для усунення збоїв. | Попередження | Замінити програмне забезпечення. |

Продовження табл. Б.6

| ID | Статус ризику | Опис ризику | Ймовірність виникнення | Вплив ризику | Ранг ризику | План А | Тип стратегії реагування | План Б |
|-------|---------------|--|------------------------|--------------|-------------|---|--------------------------|--|
| RS_9 | Відкритий | Відсутність резервних копій даних | Низька | Високий | 3 | Налаштувати автоматичне збереження даних. Зберігати дані на різних носіях інформації. | Попередження | Робити копію даних після кожного виконаного етапу. |
| RS_10 | Відкритий | Реалізація непотрібного функціоналу | Низька | Низький | 1 | Попередити замовника про можливість додаткового функціоналу. | Використання | Обговорити вигоди і збитки від можливих змін проекту. |
| RS_11 | Відкритий | Аварійні відключення світла | Висока | Низький | 3 | Передбачити у календарному плані неможливість роботи через відключення світла. | Попередження | Оснащення розробників додатковими джерелами живлення. |
| RS_12 | Відкритий | Загроза ракетних та артилерійських обстрілів | Середня | Високий | 6 | Оснастити безпечне місце роботи. | Прийняття | Організувати дистанційний спосіб роботи. |
| RS_13 | Відкритий | Невиконання моніторингу проекту | Середня | Низький | 2 | Здійснювати проміжний контроль результатів в ході виконання проекту. | Перенесення | Надання проміжних результатів виконання проекту після кожного етапу. |

ДОДАТОК В

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

Web application to support volunteer activities

V. Sydorenko, *student of IT-92-1/2*; V. Antypenko, *Associate Professor*
Sumy State University, Sumy, Ukraine

A non-commercial web application is a type of online resource that is not created for profit. They can be developed for various reasons, including personal expression, activism, education, hobbies etc. One of the main advantages of creating a non-commercial web application is that it allows individuals or organizations to share information and ideas without worrying about making a profit. And today it is especially important for many ones. Nowadays, more attention should be paid to web applications that would help Ukrainian people who suffered during the russian invasion. It is not only about monetary assistance, but also the usual solutions to household needs. Helping the army of Ukraine and the citizens of our country is currently an urgent issue, which prompts us to develop such non-commercial web applications that will spread volunteerism among the masses. Therefore, the aim of this project is creating a web application to support volunteer activities. The proper organization of last one will be ensured by use of this online recourse due to automating the process of interaction between volunteers and people who need help. The developed web application would provide information about upcoming events, training opportunities, other important updates etc. Using such tool will help volunteers to stay up-to-date as well as allow them to participate in the activities of any organization. In general, this web application can serve as a platform for communication between volunteers. For instance, to coordinate their actions and stay connected even when they are not physically together. Such web application can also be a useful tool for attracting new volunteers. For example, through the possibility of obtaining relevant information about the organization and its mission. Having an easy way for people to register to do this form of charity online is also an advantage of using his online resource. All these can help expand the pool of available volunteers. And that means bringing new energy and ideas to concrete organization. Thus, the development of a web application to support volunteer activities is really relevant. Today, meeting the needs of the Ukrainian population is a very important task within such difficult time. Using this online service will increase the visibility of the volunteer organization and its work. This will allow it to reach a wider audience and potentially attract more attention, so get more support. By demonstrating the organization's mission, values and achievements, using such web app will helps to generate enthusiasm and interest in its activities and inspire others to get involved, thereby helping more people.

ДОДАТОК Г

ЛІСТИНГ ПРОГРАМНОГО КОДУ ОСНОВНИХ МОДУЛІВ WEB-ДОДАТКУ

index.blade.php

```

<?php

include_once ("head.php");
include_once ("header.php");

?>
@extends('templates.layout')

@section('title','Головна | YourAngel')

@section('content')
    <div class="px-4 pt-5 my-5 text-center border-bottom">
        <h1 class="display-4 fw-bold text-body-emphasis">ЛАСКАВО ПРОСИМО</h1>
        <div class="col-lg-6 mx-auto">
            <p class="lead mb-4">
                Ми тут, щоб надати підтримку та допомогу людям, які постраждали внаслідок
                війни. Україна зазнала великих випробувань, і наша мета - зібрати разом руки та серця, щоб
                допомогти відновити життя тих, хто потребує допомоги найбільше.
            </p>
            <div class="d-grid gap-2 d-sm-flex justify-content-sm-center mb-5">
                <a href="{{route('applications.index')}}" class="btn btn-primary btn-lg px-4
                me-sm-3">Усі заявки</a>
                <a href="{{route('applications.create')}}" type="button" class="btn btn-
                outline-secondary btn-lg px-4">Створити нову</a>
            </div>
        </div>
        <div class="overflow-hidden" style="max-height: 30vh;">
            <div class="container px-5">
                
            </div>
        </div>
    </div>

    <div class="container col-xxl-8 px-4 py-5">
        <div class="row flex-lg-row-reverse align-items-center g-5 py-5">
            <div class="col-10 col-sm-8 col-lg-6">
                
            </div>
            <div class="col-lg-6">
                <h1 class="display-5 fw-bold text-body-emphasis lh-1 mb-3">Опинився в скрутній
                ситуації?</h1>
                <p class="lead">Тоді тобі до нас! Небайдужі люди з радістю допоможуть вирішити
                будь-яку проблему.
                Розміщуй оголошення про необхідну допомогу та отримуй те, чого критично не
                вистачає.</p>
            </div>
        </div>
    </div>

    <div class="container col-xl-10 col-xxl-8 px-4 py-5">
        <div class="row align-items-center g-lg-5 py-5">
            <div class="col-lg-7 text-center text-lg-start">

```

```

        <h1 class="display-4 fw-bold lh-1 text-body-emphasis mb-3">Приєднуйся до
нас</h1>
        <p class="col-lg-10 fs-4">Не втрачайте часу - приєднуйтеся до нашої
волонтерської спільноти вже сьогодні!
        Реєструйся, щоб допомагати та отримувати допомогу від інших вже зараз!</p>
    </div>
    <div class="col-md-10 mx-auto col-lg-5">
        <form class="p-4 p-md-5 border rounded-3 bg-body-tertiary"
action="{{route('auth.register')}}" method="POST">
            @csrf
            <div class="form-floating mb-2">
                <input type="text" name="firstname" class="form-control"
id="floatingFirstname" placeholder="Ім'я">
                <label for="floatingFirstname">Ім'я</label>
                @if ($errors->has('firstname'))
                    <span class="text-danger text-left">{{ $errors->first('firstname')
}}</span>
                @endif
            </div>
            <div class="form-floating mb-2">
                <input type="text" name="lastname" class="form-control"
id="floatingLastname" placeholder="Шевченко">
                <label for="floatingLastname">Прізвище</label>
                @if ($errors->has('lastname'))
                    <span class="text-danger text-left">{{ $errors->first('lastname')
}}</span>
                @endif
            </div>
            <div class="form-floating mb-2">
                <input type="email" name="email" class="form-control"
id="floatingEmail" placeholder="name@example.com">
                <label for="floatingEmail">Email</label>
                @if ($errors->has('email'))
                    <span class="text-danger text-left">{{ $errors->first('email')
}}</span>
                @endif
            </div>
            <div class="form-floating mb-2">
                <input type="text" name="phone" class="form-control"
id="floatingPhone" placeholder="name@example.com">
                <label for="floatingPhone">Номер телефону</label>
                @if ($errors->has('phone'))
                    <span class="text-danger text-left">{{ $errors->first('phone')
}}</span>
                @endif
            </div>
            <div class="form-floating mb-2">
                <input type="password" name="password" class="form-control"
id="floatingPassword" placeholder=".....">
                <label for="floatingPassword">Пароль</label>
                @if ($errors->has('password'))
                    <span class="text-danger text-left">{{ $errors->first('password')
}}</span>
                @endif
            </div>
            <button class="w-100 btn btn-lg btn-primary"
type="submit">Зареєструватись</button>
            <hr class="my-4">
            <small class="text-body-secondary">Ви приймаєте умови користування
сайтом</small>
        </form>
    </div>
</div>
</div>
@endsection

```

header.blade.php

```

<header class="p-3 mb-3 border-bottom">
  <div class="container">
    <div class="d-flex flex-wrap align-items-center justify-content-center justify-
content-lg-start">
      <a href="{{route('index')}}" style="font-weight: 800;font-size: 14pt;margin-right:
20px;"
        class="d-flex align-items-center mb-2 mb-lg-0 text-dark text-decoration-none">
        
        YourAngel
      </a>

      <ul class="nav col-12 col-lg-auto me-lg-auto mb-2 justify-content-center mb-md-0">
        <li><a href="{{route('index')}}" class="nav-link px-2 link-
dark">Головна</a></li>
        <li><a href="{{route('applications.index')}}" class="nav-link px-2 link-
dark">Заявки</a></li>
        <li><a href="{{route('reports.index')}}" class="nav-link px-2 link-
dark">Звіти</a></li>
        <li><a href="{{route('fundraisers.index')}}" class="nav-link px-2 link-
dark">Грошові збори</a></li>
      </ul>

      <form class="col-12 col-lg-auto mb-3 mb-lg-0 me-lg-3"
action="{{route('applications.index')}}">
        <input type="hidden" name="type" value="{{$_GET['type'] ?? 'need'}}">
        <input type="search" name="search" class="form-control" placeholder="Пошук..."
aria-label="Search">
      </form>

      <div class="dropdown text-end">
        @auth
          <a href="#" class="d-block link-dark text-decoration-none dropdown-toggle"
id="dropdownUser1"
          data-bs-toggle="dropdown" aria-expanded="false">
            
          </a>
          <ul class="dropdown-menu text-small" aria-labelledby="dropdownUser1"
style="">
            @if(auth()->user()->isAdmin())
              <li><a class="dropdown-item bg-body-secondary"
href="{{route('admin.index')}}">Адмін-панель</a></li>
            @endif
            <li><a class="dropdown-item"
href="{{route('applications.create')}}">Створити заявку</a></li>
            <li><a class="dropdown-item" href="{{route('applications.my')}}">Мої
заявки</a></li>
            <li>
              <hr class="dropdown-divider">
            </li>
            <li><a class="dropdown-item"
href="{{route('auth.logout')}}">Вихід</a></li>
          </ul>
        @else
          <a href="{{route('auth.login')}}" class="btn btn-outline-
secondary">Вхід</a>
          <a href="{{route('auth.register')}}" class="btn btn-
primary">Реєстрація</a>
        @endauth
      </div>
    </div>
  </div>
</header>

```

footer.blade.php

```
<div class="container">
  <footer class="py-3 my-4">
    <ul class="nav justify-content-center border-bottom pb-3 mb-3">
      <li class="nav-item"><a href="{{route('index')}}" class="nav-link px-2 text-body-
secondary">Головна</a></li>
      <li class="nav-item"><a href="{{route('applications.index')}}" class="nav-link px-
2 text-body-secondary">Заявки</a></li>
      <li class="nav-item"><a href="{{route('reports.index')}}" class="nav-link px-2
text-body-secondary">Звіти</a></li>
      <li class="nav-item"><a href="{{route('fundraisers.index')}}" class="nav-link px-2
text-body-secondary">Грошові збори</a></li>
      <li class="nav-item"><a href="{{route('applications.create')}}" class="nav-link
px-2 text-body-secondary">Нова заявка</a></li>
      <li class="nav-item"><a href="{{route('applications.my')}}" class="nav-link px-2
text-body-secondary">Мої заявки</a></li>
    </ul>
    <p class="text-center text-body-secondary">© 2023 YourAngel</p>
  </footer>
</div>
```

ApplicationController.php

```
<?php
```

```
namespace App\Http\Controllers;

use App\Http\Requests\DeleteApplicationRequest;
use App\Http\Requests\StoreApplicationRequest;
use App\Http\Requests\UpdateApplicationRequest;
use App\Mail\VerificationMail;
use App\Models\Application;
use App\Models\ApplicationImage;
use App\Models\Category;
use App\Models\ConfirmationCode;
use App\Models\Report;
use Exception;
use Illuminate\Http\RedirectResponse;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\App;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Mail;
use Illuminate\Support\Str;
use Illuminate\View\View;

class ApplicationController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index(Request $request): View
    {
        $applications = Application::query()->where('status', Application::ACTIVE);

        if ($request->has('search')) {
            $search = strtolower($request->search);
            $applications->where('name', 'LIKE', "%$search%");
        } else {
            // Фільтр за категорією
            if ($request->has('category')) {
                $applications->whereHas('category', function ($query) use ($request) {
                    $query->whereIn('slug', $request->category);
                });
            }

            // Фільтр за регіоном
            if ($request->has('region')) {
                $applications->whereIn('region', $request->region);
            }
        }
    }
}
```

```

    }
}

// Фільтр за типом
$type = Application::TYPE_NEED;
if ($request->has('type')) {
    if ($request->type == 'provide') {
        $type = Application::TYPE_PROVIDE;
    }
}
$applications->where('type', $type);

$applications = $applications->latest()->paginate(15);
$categories = Category::query()->withCount(['applications' => function ($query) use
($type) {
    $query->where('type', $type);
}])->orderBy('name', 'DESC')->get();

$title = ($type === Application::TYPE_NEED) ? 'Потребують допомоги' : 'Надають
допомогу';

$regionsCount = Application::where('type', $type)->where('status',
Application::ACTIVE)->groupBy('region')
->pluck(DB::raw('count(*)'), 'region')
->toArray();

return view('applications.index')->with([
    'title' => $title,
    'applications' => $applications,
    'categories' => $categories,
    'regionsCount' => $regionsCount
]);
}

public function my(Request $request): View
{
    $applications = auth()->user()->applications();
    if ($request->has('status')) {
        $status = $this->getStatus($request->status);
        $applications->where('status', $status);
    }
    $applications = $applications->latest()->paginate(15);

    return view('applications.my')->with([
        'title' => $this->getTitle($status ?? 100),
        'applications' => $applications
    ]);
}

/**
 * Show the form for creating a new resource.
 */
public function create(): View
{
    $categories = Category::orderBy('name', 'DESC')->get();

    return view('applications.create')->with([
        'categories' => $categories
    ]);
}

/**
 * Store a newly created resource in storage.
 */
public function store(StoreApplicationRequest $request): RedirectResponse
{
    $application = new Application($request->validated());
    $images = [];

```

```

if ($request->hasFile('images')) {
    foreach ($request->file('images') as $image) {
        $link = $image->storePublicly('images', 'public');
        $images[] = new ApplicationImage(['link' => $link]);
    }
}
DB::beginTransaction();
try {
    auth()->user()->applications()->save($application);
    $application->updateSlug();
    $application->save();
    if (!empty($images)) {
        $application->images()->saveMany($images);
    }
    DB::commit();
} catch (Exception $e) {
    DB::rollBack();
    return redirect()->back()->with([
        'error' => 'Виникла помилка при створенні заявки, спробуйте пізніше'
    ]);
}

return redirect()->route('applications.my')->with([
    'success' => 'Заявку успішно створено'
]);
}

/**
 * Display the specified resource.
 */
public function show(string $slug): View
{
    $application = Application::with(['user', 'category', 'images', 'comments'])
        ->where('slug', $slug)
        ->firstOrFail();

    $rejected = $application->status == Application::REJECTED;
    $moderation = $application->status == Application::MODERATION;
    $noAccess = !auth()->check() || !$application->isOwner() && !auth()->user()-
>isAdmin();

    if ($rejected || ($moderation && $noAccess)) {
        abort(404);
    }

    return view('applications.show')->with([
        'application' => $application
    ]);
}

/**
 * Show the form for editing the specified resource.
 */
public function edit(Application $application): View
{
    if ($application->status == Application::INACTIVE) {
        abort(403);
    }
    $categories = Category::orderBy('name', 'DESC')->get();

    return view('applications.edit')->with([
        'application' => $application,
        'categories' => $categories
    ]);
}

/**
 * Update the specified resource in storage.

```

```

*/
public function update(UpdateApplicationRequest $request, Application $application):
RedirectResponse
{
    if ($application->status == Application::INACTIVE) {
        abort(403);
    }

    $application->fill($request->validated());
    $originalImages = $application->images()->get();

    $images = [];
    if ($request->hasFile('images')) {
        foreach ($request->file('images') as $image) {
            $link = $image->storePublicly('images', 'public');
            $images[] = new ApplicationImage(['link' => $link]);
        }
    }
    DB::beginTransaction();
    try {
        if (!empty($images)) {
            $application->images()->delete();
            $application->images()->saveMany($images);
        }

        if (empty($images) && !$application->isDirty()) {
            return redirect()->back()->with([
                'error' => 'Немає жодних змін у заявці'
            ]);
        }

        if ($application->isDirty('name')) {
            $application->updateSlug();
        }
        $application->status = Application::MODERATION;
        $application->save();

        DB::commit();
    } catch (Exception $e) {
        DB::rollBack();
        return redirect()->back()->with([
            'error' => 'Виникла помилка при редагуванні заявки, спробуйте пізніше'
        ]);
    }

    return redirect()->route('applications.my')->with([
        'success' => 'Заявку успішно відредаговано і подано на розгляд'
    ]);
}

/**
 * Remove the specified resource from storage.
 */
public function destroy(Request $request, Application $application): RedirectResponse
{
    if ($application->status == Application::INACTIVE) {
        abort(403);
    }

    DB::beginTransaction();
    try {
        if ($application->status == Application::ACTIVE) {
            $validated = $request->validate([
                'has_help' => 'required|in:0,1',
                'count' => 'required_if:help,1|integer|max:100000'
            ]);
            if ($request->help == '0') {
                $validated['count'] = 0;
            }
        }
    }
}

```



```

    }
    $report = new Report($validated);
    $application->report()->save($report);
  }
  $application->status = Application::INACTIVE;
  $application->save();
  DB::commit();
} catch (\Throwable $e) {
  DB::rollBack();
  return redirect()->back()->with([
    'error' => 'Виникла помилка при деактивації заявки, спробуйте пізніше'
  ]);
}

return redirect()->route('applications.my')->with([
  'success' => 'Заявку успішно деактивовано'
]);
}

protected function getStatus(string $status): string
{
  return match ($status) {
    'active' => Application::ACTIVE,
    'inactive' => Application::INACTIVE,
    'rejected' => Application::REJECTED,
    'moderation' => Application::MODERATION,
    default => 100,
  };
}
}

```

LoginController.php

```
<?php
```

```

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use App\Http\Requests\LoginRequest;
use App\Http\Requests\RegisterRequest;
use App\Models\User;
use Illuminate\Http\RedirectResponse;
use Illuminate\Support\Facades\Auth;
use Illuminate\View\View;

class LoginController extends Controller
{
  public function showLoginPage(): View
  {
    return view('auth.login');
  }

  public function login(LoginRequest $request): RedirectResponse
  {
    if (!Auth::attempt($request->validated())) {
      return redirect()->route('auth.login')->withErrors([
        'password' => 'Неправильний пароль',
      ]);
    }
    if (!Auth::user()->hasVerifiedEmail()) {
      $user = Auth::user();
      Auth::logout();
      $this->sendVerification($user);
      return redirect()->route('auth.login')->with([
        'error' => 'Вам надіслано лист підтвердження'
      ]);
    }

    return redirect()->route('applications.my');
  }
}

```

```

    }

    public function logout(): RedirectResponse
    {
        Auth::logout();
        return redirect()->route('auth.login');
    }
}

```

RegisterController.php

```

<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use App\Http\Requests\RegisterRequest;
use App\Mail\VerificationMail;
use App\Models\ConfirmationCode;
use App\Models\User;
use Illuminate\Auth\Notifications\VerifyEmail;
use Illuminate\Http\RedirectResponse;
use Illuminate\Support\Carbon;
use Illuminate\Support\Facades\Mail;
use Illuminate\Support\Facades\URL;
use Illuminate\Support\Str;
use Illuminate\View\View;

class RegisterController extends Controller
{
    /**
     * Відображення сторінки реєстрації
     */
    public function showRegistrationPage(): View
    {
        return view('auth.register');
    }

    /**
     * Реєстрація користувача
     */
    public function register(RegisterRequest $request): RedirectResponse
    {
        $user = User::query()->create($request->validated());
        $this->sendVerification($user);
        //$user->markEmailAsVerified();

        return redirect()->route('auth.login')
            ->with('success', 'На вашу пошту надіслано лист підтвердження');
    }
}

```