МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

Кваліфікаційна робота магістра

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПІДТРИМКИ ВОЛОНТЕРСЬКОЇ ДІЯЛЬНОСТІ

Здобувач освіти гр.ІН.м-13 Максим КОЛОМОЙЦЕВ

Науковий керівник Ігор ШЕЛЕХОВ
вчене звання, наук. ступінь

Завідувач кафедри Ігор ШЕЛЕХОВ
вчене звання, наук. ступінь

Суми-2022

Сумський державний університет
(назва вузу)

Факультет *ЕлІТ* Кафедра *Комп'ютерних наук*
Спеціальність *«122 - Комп'ютерні науки»*

Затверджую:
зав.кафедрою _____
_____
"_____" _____ 20___р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра студентові

Коломойцев Максим Сергійович
(прізвище, ім'я, по батькові)

1 Тема проекту_____

затверджена наказом по університету від «___»_____2022 р. №_____

2 Термін здачі студентом закінченого проекту _____

3 Вхідні дані до проекту _____
_____
_____
_____

4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) _____
_____
_____
_____
_____
_____

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
_____
_____
_____
_____
_____
_____
_____
_____

6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Аналіз предметної області	Шелехов І.В.		
Постановка задачі та вибір методів розробки	Шелехов І.В.		
Практична реалізація додатку	Шелехов І.В.		

Дата видачі завдання _____.

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів випускної проекту	Термін виконання етапів проекту	Примітка
1	Дослідження предметної області	01.09.22-05.09.22	
2	Порівняння з конкурентами	05.09.22-15.09.22	
3	Встановлення мети та задачі	15.09.22-25.09.22	
4	Створення дизайну	25.09.22-05.10.22	
5	Проектування бази даних	05.10.22-10.10.22	
6	Проектування каркасу	10.10.22-25.10.22	
7	Створення функціональних блоків	25.10.22-09.11.22	
8	Тестування	09.11.22-16.11.22	
9	Заповнення контентом	16.11.22-26.11.22	
10	Завантаження інформаційної технології на сервер	26.11.22-31.11.22	

Студент – дипломник _____
(підпис)

Керівник проєкту _____
(підпис)

РЕФЕРАТ

Записка: 97 стр., 85 рис., 1 додаток, 23 джерела.

Об'єкт дослідження: структура подачі інформації користувачу, яку подає інформаційна технологія.

Мета: створити інформаційну технологію по волонтерській допомозі для спрощення та зручності у використанні наданої інформації.

Предмет дослідження: об'єднання та уніфікування взаємодії між волонтером та користувачем.

Гіпотеза: використання створеної інформаційної технології та надання контролю збору коштів волонтером.

Практичне значення: практичне значення у цьому проєкті є створення інформаційної технології по волонтерській допомозі, яке буде задовольняти всі поставлені потреби.

При виконанні даної технології повинно бути вирішено такі як:

- Розповсюдження інформації, щодо відкритих фінансових зборів;
- Відкритість інформації, де і як можна придбати військове спорядження;
- Реалізація можливості подання запиту волонтеру на придбання військового спорядження;
- Полегшення пошуку та комунікації із волонтерами.

Результати: результатом проведеної роботи є створена інформаційна технологія по волонтерській допомозі.

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ, ВОЛОНТЕРСТВО, ПРОДАЖ, ПЛАТІЖНА СИСТЕМА, ВОЛОНТЕР, КОРИСТУВАЧ, ТОВАР, ЗБІР КОШТІВ, ОЦІНКА, КОМЕНТАР, ПОЖЕРТВУВАННЯ, ПЕРЕГЛЯД ТОВАРІВ, КОНТРОЛЬ ЗБОРУ КОШТІВ, ТЕХНОЛОГІЯ.

ЗМІСТ

ВСТУП

На сьогоднішній час люди постійно стикаються з інформацією про фінансові збори постраждалим від війни чи збори на військові потреби. Все це стало частиною нашого життя. Звісно, зіткнутися із проблемою в пошуку додаткового спорядження може кожен. В Україні досить мало ресурсів, що допомагають у пошуку та підбору речей у цьому напрямку.

Волонтерська діяльність стала невіддільною частиною життя кожного та важко зараз знайти людину, яка так чи інакше не приймала в цьому участі. Така популярність цієї галузі породила пласт населення, який має на меті лише наживатися на людях з добрими намірами. Фільтрувати подібні випадки важко, адже волонтери в більшості своєму роздрібнені та не мають централізованої системи для організації зборів та публікації результатів з наявністю коментарів.

Роздрібненість волонтерської спільноти – не єдина проблема даної галузі. Деякі види спорядження або навіть техніки розкидані по різних куточках інтернету, що збільшує труднощі при пошуках конкретного товару. Спеціальна інформаційна технологія могла б вмістити в собі не лише волонтерське об’єднання небайдужих, а й потенціал інтернет-магазину з відкритим набором товарів, де в режимі онлайн можна слідкувати за зборами на той чи інший товар в якого потребують не лише в зонах проведення військових дій.

Для того, щоб віднайти спеціальну військову техніку та придбати її, на перевірних сайтах, потрібно прикласти багато зусиль. У багатьох випадках потрібно знаходити волонтерів, які або допоможуть у придбанні речей, або підкажуть ресурси для пошуку. Додатково наявні труднощі з доставлянням речей від виробника.

Для полегшення роботи у цьому напрямку та спрощенню співпраці, було прийнято рішення по розробці інформаційній технології по волонтерській допомозі.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень і публікацій

Всесвітнє дослідження волонтерської діяльності, виявило, що люди, які відчували найвищий рівень щастя, були учасниками різних волонтерських активностей. Для порівняння, відомо, що інша форма альтруїзму, філантропія у формі грошових пожертв (одним з яких є волонтерство), має подібний ефект. Інше дослідження показало, що допомога іншим пов'язана з покращенням психічного здоров'я, а не з вигодами від отримання допомоги [1]. Студенти, які займалися волонтерством 1–9 годин на тиждень, мали менше шансів бути депресивними, ніж ті, хто не займався волонтерством. Серед людей старше 65 років волонтерство знижує ризик депресії.

Волонтерство – це коли хтось жертвує своїм вільним часом заради іншого. Допомога близьким друзям і родичам не є добровільною. Але робити щось, що допоможе довкіллю та іншим людям – є. Волонтерство може бути як офіційним [2], організованим організацією, так і неформальним. Але що точно їх об'єднує – це завжди вільний вибір тих, хто витрачає свій час.

Кожен має право займатися волонтерством. Волонтери можуть бути будь-якого віку, походження, займати різні галузі та мати різний рівень достатку. Вони можуть навчатися, працювати або на пенсії. Це можуть бути працівники компанії, які добровільно віддають себе у свій час. Вони можуть бути медичними чи юридичними фахівцями, які надають свій час безкоштовно.

Волонтери не являють собою найманих працівників і на них не поширюється трудове законодавство. Важливо розрізняти оплачуваний персонал і волонтерів.

Щоб виділити ці два поняття потрібно встановити різницю:

- Волонтерство – це особистий вибір людини;
- Роль волонтера – це не те саме, що роль працівника
- Волонтери не можуть бути заміною оплачуваного персоналу.

Революція Гідності в Києві та інших великих містах з листопада 2013 року по лютий 2014 року мала тривалі наслідки для українського громадянського суспільства. Волонтерські організації народжуються для задоволення потреб протестувальників. На Майдані Незалежності програмісти, медики, юристи, роздавачі їжі тощо створили власні організації та разом виконували свої обов'язки.

Пізніше багато волонтерських сил відіграли ключову роль у створенні Червоного хреста для підтримки регулярних сил на сході України зокрема лікарі-волонтери, які стали основою та натхненням для пізніших волонтерських груп [3]. Волонтерські організації співпрацюють з паралельними організаціями, коли українська влада не може допомогти своїм громадянам.

Волонтерська організація Save Ukrainian Lives забезпечує українських бійців на передовій розчинами крові та іншими медичними засобами. Організація співпрацює з MedAutoMaidan, який раніше надавав медичну допомогу протестувальникам проти влади Януковича.

Українська волонтерська організація «Return Alive» забезпечує українських бійців військовими речами, в тому числі необхідними медикаментами, які не змогла забезпечити центральна влада. Обладнання, яке організація надає солдатам у зонах бойових дій, включає окуляри нічного бачення та інфрачервоні окуляри. Return Alive також надає ліки, щоб запобігти сильній кровотечі.

З початком повномасштабного вторгнення, волонтерством почала займатися велика частина населення [4]. Друзі об'єднувалися в команди, команди в громади, громади в організації, які можливо не мають навіть назви, але мають ціль.

Зі збільшенням волонтерської аудиторії непомітно з'явилася потреба в об'єднанні волонтерів в межах однієї великої інформаційної технології.

Системи управління процесами та інформаційний технологічний процес не стоять на місці. Кожного дня з'являються технології, що полегшують процес роботи шляхом автоматизації звичних процесів та оптимізації часу на їх використання.

Інформаційні технології з часом розвивалися. Вони стали способом для спільноти розробників стандартними інструментами, які покращують взаємодію з користувачем і, перш за все, безпеку. Наприклад, HTML5 і CSS3 зробили процес

створення інформаційних технологій простішим, зручнішим і на сам перед зручнішим для користувачів. Сертифікати SSL роблять інформаційну технологію і його програми більш безпечними для всіх. Сьогодні ці стандарти все частіше використовуються у створенні все новіших інформаційних технологій.

За останнє десятиліття стандарти безпеки покращилися й продовжують вдосконалюватися. Однією з причин є збільшення кількості кібератак. Оскільки можливості інформаційних технологій продовжують розширюватися, схоже, що зловмисники, а також компанії та широка громадськість бачать у ньому можливість.

Сьогодні безпека є однією з головних проблем розробників. Крім SSL, в колі інформаційних технологій і в мобільних додатках реалізована двофакторна аутентифікація, яка ще поки набирає обертів.

Значна частина змін у галузі створення інформаційних технологій пов’язана з поширенням мобільних пристроїв [5]. Це означає, що більшість людей переглядають інформаційних технологій на екранах розміром із долоню.

Десять років тому більшість інформаційних технологій складалися з багатьох сторінок, з’єднаних в ієрархічну структуру, але сьогодні підхід змінюється. Це все тому, що інформаційні технології повільно завантажуються на мобільних пристроях. Перегляд на мобільному телефоні може викликати розчарування. Якщо користувачеві доводиться чекати, поки сторінка завантажиться, йому потрібно чекати лише один раз, а не кожного разу, коли він натискає елемент.

Підхід Single Page Application (SPA) сьогодні найбільш популярний серед розробників. Тому, що він зручний для мобільних пристроїв і дозволяє моделювати продуктивність мобільних інформаційних технологій, які покращують взаємодію з користувачем. SPA чудово підходять для адаптивної інформаційної технології. Сторінки завантажуються лише один раз, що забезпечує кращий досвід користувача (UX) і швидшу реакцію на натискання по посиланню.

Для полегшення роботи у цьому напрямку та спрощенню співпраці, було прийнято рішення розробити інформаційну технологію по волонтерській допомозі.

1.2 Аналіз програмних продуктів

Більшість людей знає, як виникають нові проєкти в передбачуваному світі: збирається команда, аналізується ринок, створюється прогноз й пишеться бізнес-план. Потім збираються ресурси, і план приводиться в дію.

Аналіз проєкту має вирішальне значення для компаній і керівників проєктів, щоб зробити свої проекти більш успішними та стійкими. Хоча очевидно, що проблеми та виклики постануть на вашому шляху, ви можете тримати ситуацію під контролем за допомогою правильного інструменту та підходу.

Для аналізу програмних продуктів конкурентів інформаційної технології по волонтерській допомозі було обрано:

– «Spivdiia.org.ua» [6];
– «Uahelpers.com» [7];
– «Volonter.org» [8];

Розберемо кожен із представлених варіантів продуктів конкурентів інформаційної технології.

1.2.1 Аналіз «Spivdiia.org.ua»

Перш за все було проведено аналіз програмного продукту «Spivdiia.org.ua». Він являє собою веб-додаток, що дозволяє подати запит на отримання допомоги, чи запит на надання допомоги.

Дана інформація представлена на головній сторінці веб-додатку (рис.1.1).

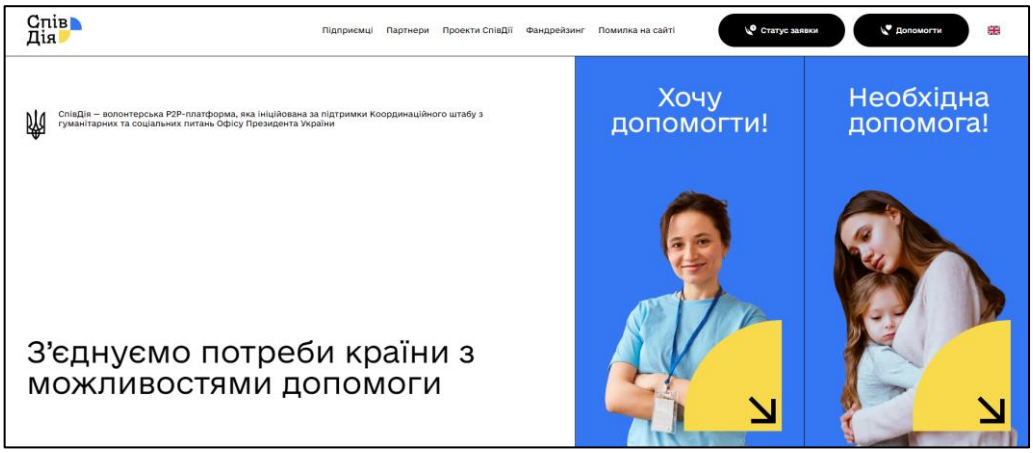


Рисунок 1.1 – «Spivdiia.org.ua» головна сторінка

Наступною є блок з інформацією про проєкти «Spivdiia.org.ua» та як вони взаємодіють між собою. До тем проєктів відноситься: хаб, діти, шелтер, психологічна підтримка та інше (рис.1.2). Відкритість інформації – значна перевага будь-якої інформаційної системи.

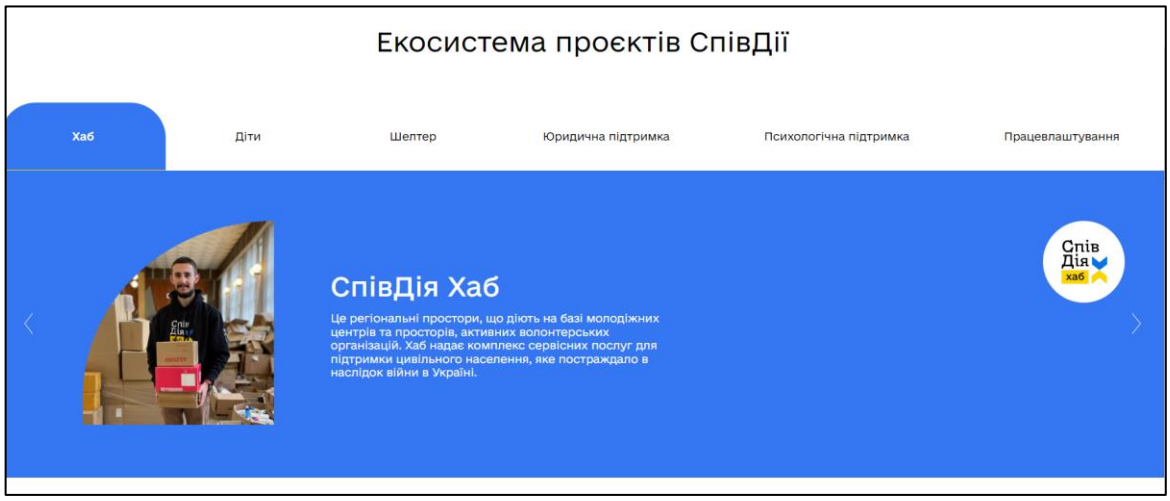


Рисунок 1.2 – «Spivdiia.org.ua» екосистеми проєктів

На сторінках веб-додатку можна знайти функціонал для подання заявки на допомогу іншим. Якщо людина може придбати певні речі, засоби гігієни, їжа чи інше, то можна залишити заявку (рис.1.3).

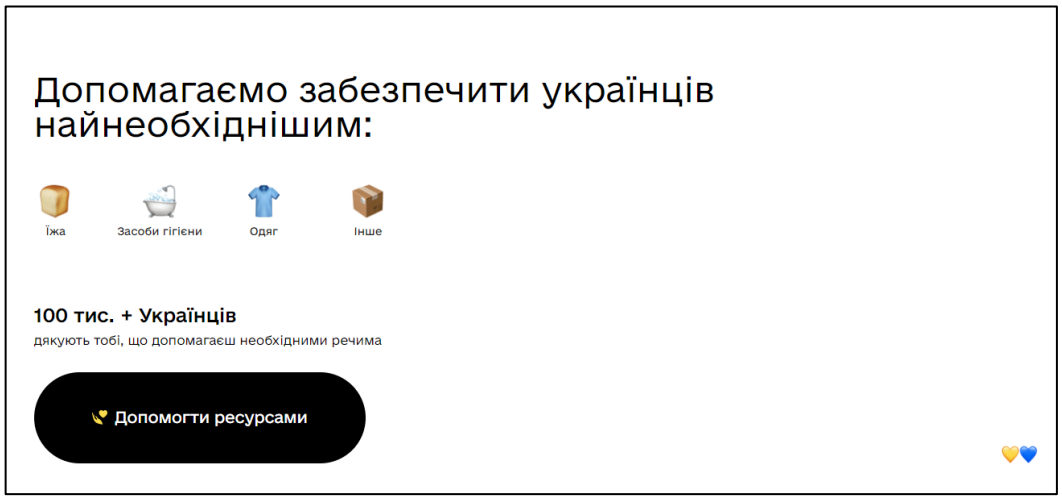


Рисунок 1.3 – «Spivdiia.org.ua» функціонал для допомоги іншим

Приклад такої заявки представлено на рисунку 1.4, користувач веб-додатку повинен заповнити всю необхідну інформацію, починаю з ПІП а закінчуючи переліком речей для допомоги.

На мою думку, заявка виконана добре, але не зрозумілий момент із часом обробкою запита від користувача.

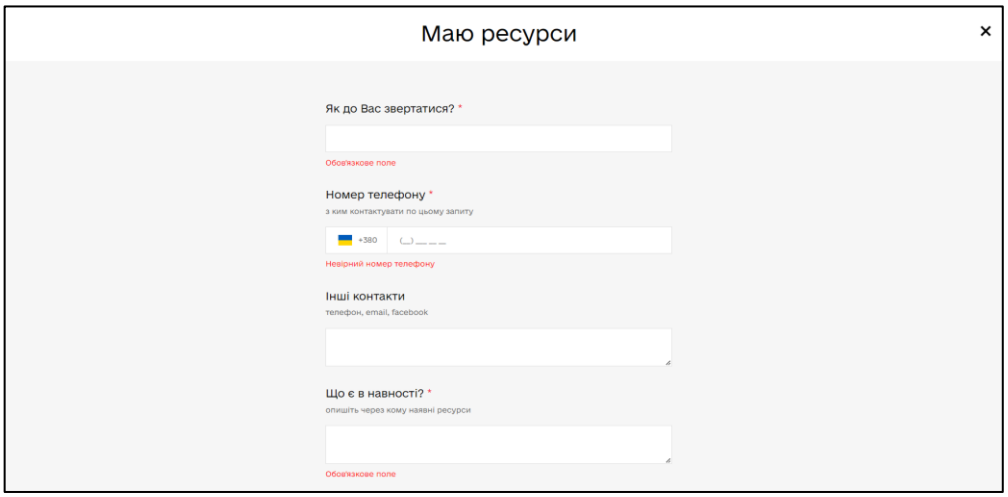


Рисунок 1.4 – «Spivdiia.org.ua» приклад заяви

Крім цього, на сторінках веб-додатку присутній функціонал для збору коштів. Якщо людина може допомогти фінансово, замість допомоги речами, то можна зробити внесок (рис.1.5).

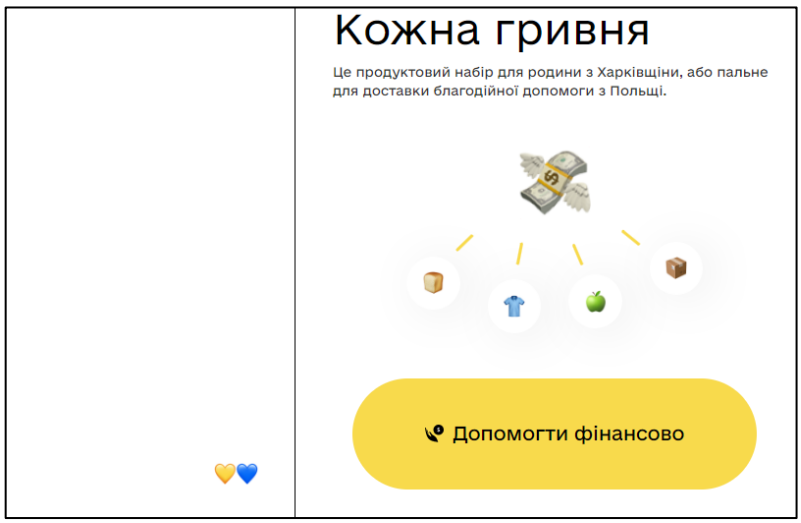


Рисунок 1.5 – «Spivdiia.org.ua» функціонал для фінансової допомоги

Користувачеві, кім власної суми, на вибір дається можливість варіанти суми донату (рис.1.6). До переваг можна віднести те, що від кожною сумою вказується чому вона дорівнює. Як, наприклад, 7000 гривень дорівнює 20 продуктовим наборам на 20 людей.

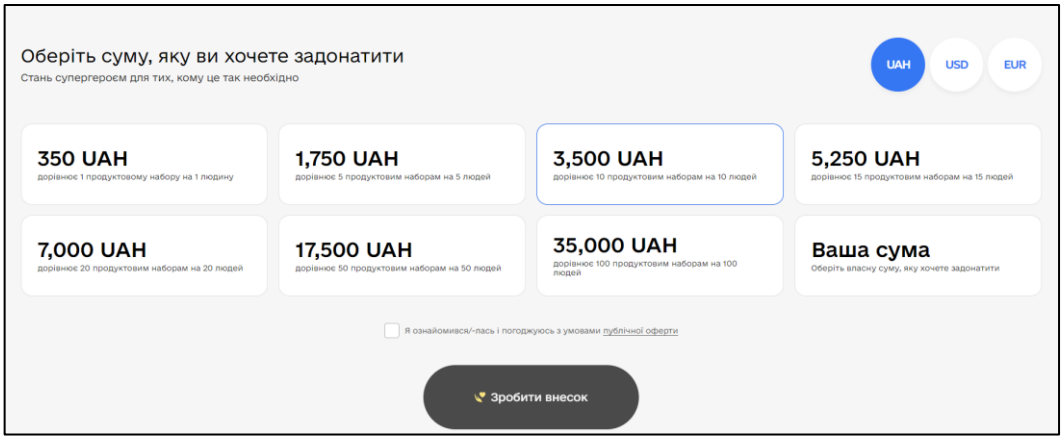


Рисунок 1.6 – «Spivdiia.org.ua» функціонал для фінансової допомоги

Отже, було проведено аналіз «Spivdiia.org.ua», що є гарним приладом якісної реалізації веб-додатку. До переваг можна віднести зручність використання та якісний дизайн.

1.2.2 Аналіз «Uahelpers.com»

Наступним прикладом та аналогом буде веб-додаток «Uahelpers.com». На головній сторінці представлена інформація що до волонтерської тематики (рис.1.7).

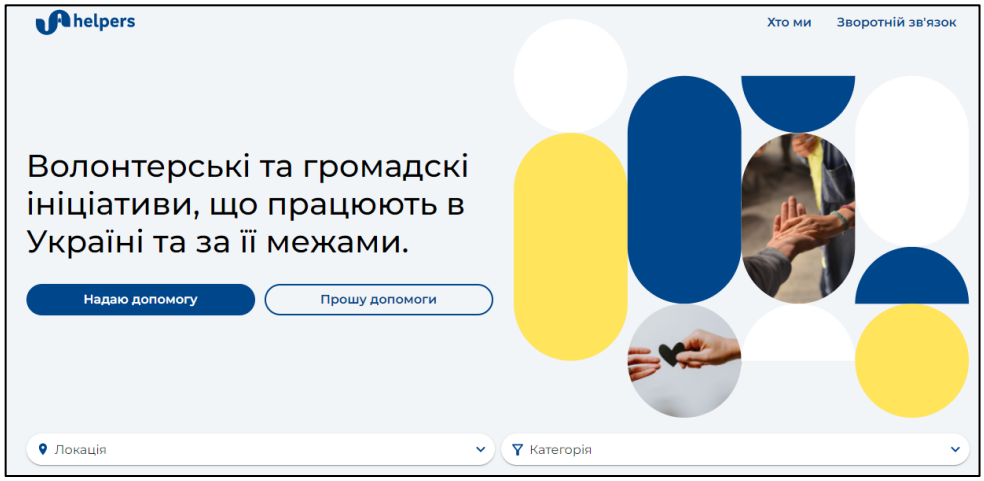


Рисунок 1.7 – «Uahelpers.com» головна сторінка

Наступним блоком на головній сторінці «Uahelpers.com» є інформаційний блок і створеними та доданими пропозиціями та проханнями (рис.1.8). Кожна із «плиток» надає коротку інформацію, контакти, локацію та тип пропозиції чи прохання.

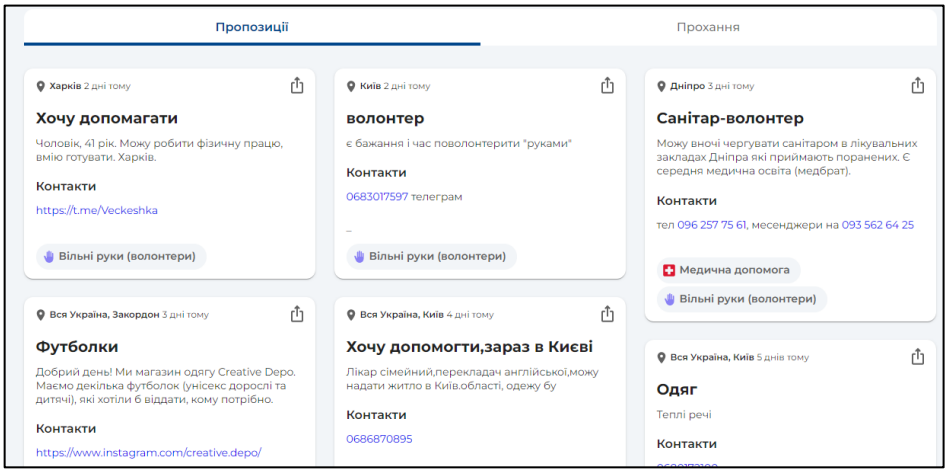


Рисунок 1.8 – «Uahelpers.com» блок із пропозиціями та проханнями

Наступний блок складає з себе інформацію про самий веб-додаток та розробників. Із цього блоку можна дізнатися основну інформацію що до призначення та напрямок використання (рис.1.9).

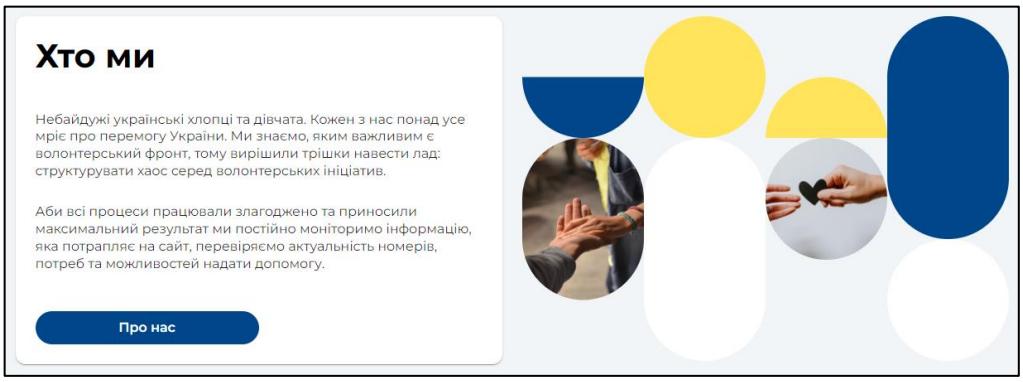


Рисунок 1.9 – «Uahelpers.com» інформаційний блок

Натискаючи на кнопку «Надаю допомоги» чи «Потребую допомогу» користувач перенаправляється на сторінку для заповнення відповідної заявки. Приклад заявки представлений на рисунку 1.10.

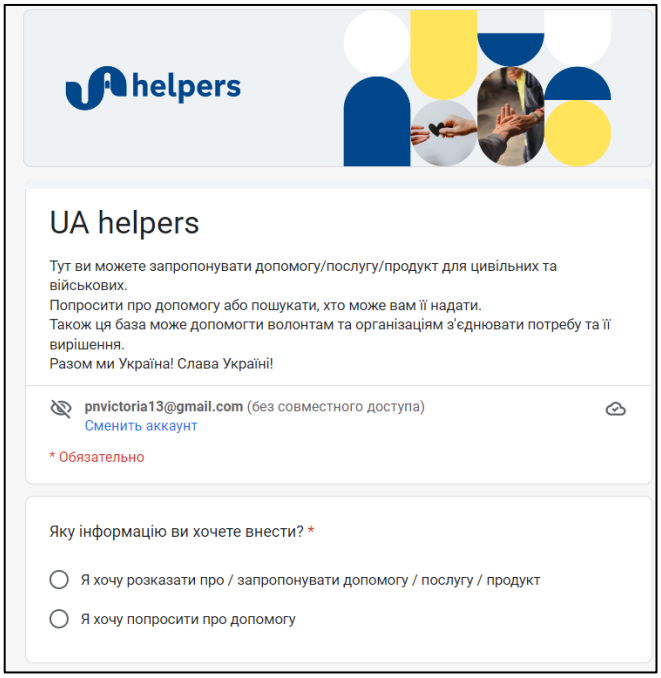


Рисунок 1.10– «Uahelpers.com» форма для залишення заявки на допомогу чи послугу

До переваг додатку можна віднести й функціонал, що дозволяє користувачеві залишити заявку у разі виникнення помилки із роботи веб-додатку. Для цього потрібно перейти за відповідним посиланням (рис.1.11).

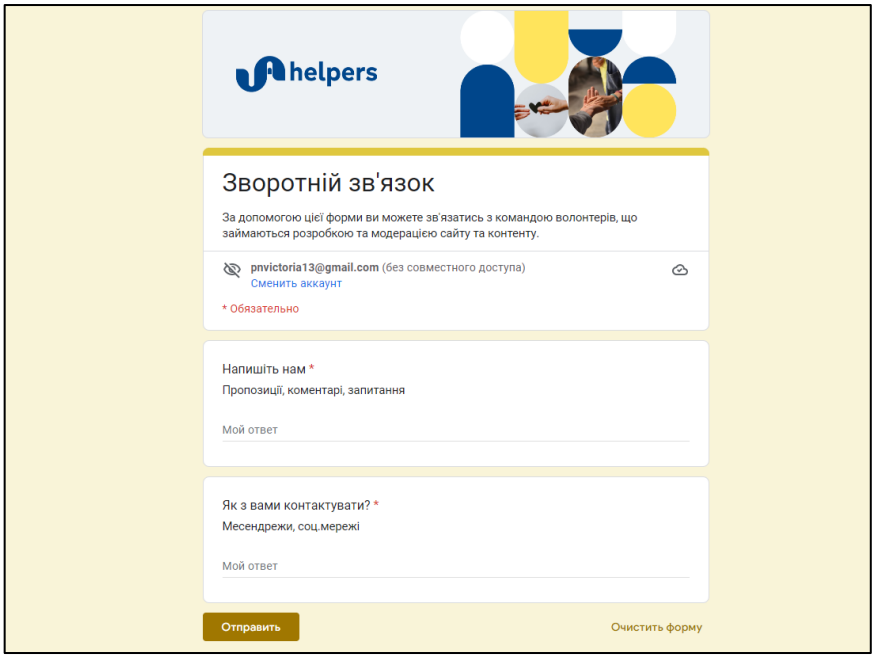


Рисунок 1.11– «Uahelpers.com» форма для заявки про помилку

Отже, проаналізувавши веб-додаток «Uahelpers.com» можна зробити зробити висновок, що даний додаток має як свої переваги так й недоліки. До недоліків можна віднести малу кількість інформації та формат додатку. Адже веб-додаток «Uahelpers.com» виконаний як односторінковий додаток.

До недоліків можна також віднести використання не вбудованого заповнення заявки, а використання сторонніх сервісів. Використання вбудованих функцій значно краще та якісніше у використанні.

1.2.3 Аналіз «Volonter.org»

Останнім аналогом інформаційної технології кваліфікаційної роботи буде веб-додаток «Volonter.org»(рис.1.12). Даний веб-додаток використовується для волонтерської допомоги за різними напрямками. Приклади категорій представлено на рисунку 1.13

На головній сторінці знаходиться основна інформація про веб-додаток та роботу, з якою він пов’язаний.

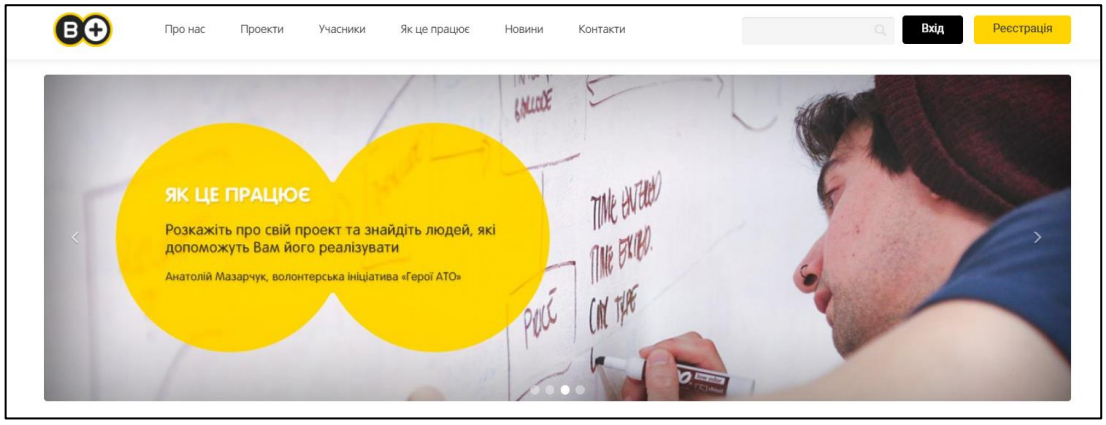


Рисунок 1.12– «Volonter.org» головна сторінка

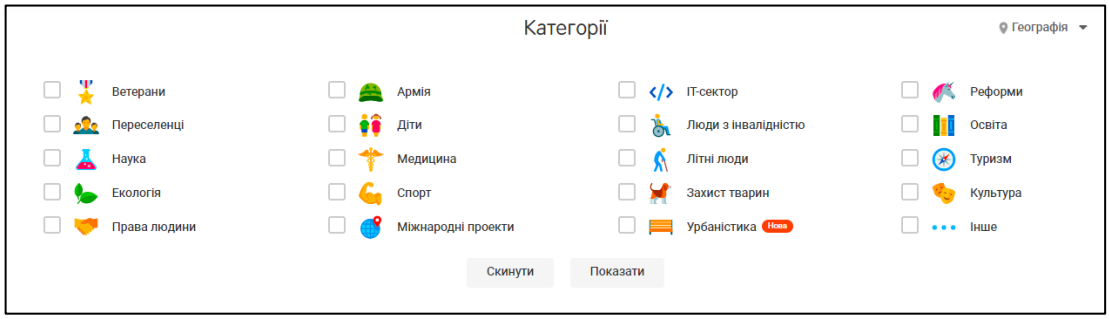


Рисунок 1.13– «Volonter.org» категорії

У веб-додатку користувач може переглянути проекти команди, що саме було виконано по волонтерській діяльності. Вони складають з себе заголовок, посилання на користувача, що над цим працює та коротка інформація (рис.1.14).

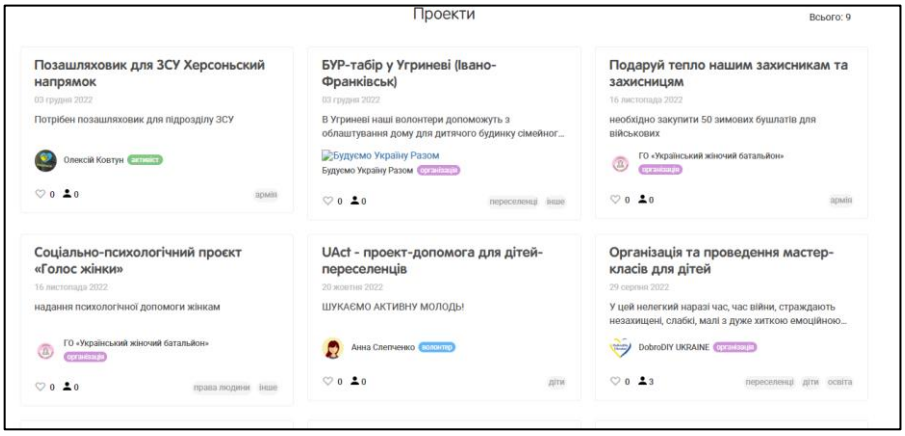


Рисунок 1.14– «Volonter.org» проекти веб-додатку

Крім того, на даному веб-додатку можна подивитися додані актуальні новини. Приклад новини веб-додатку представлено на рисунку 1.15.

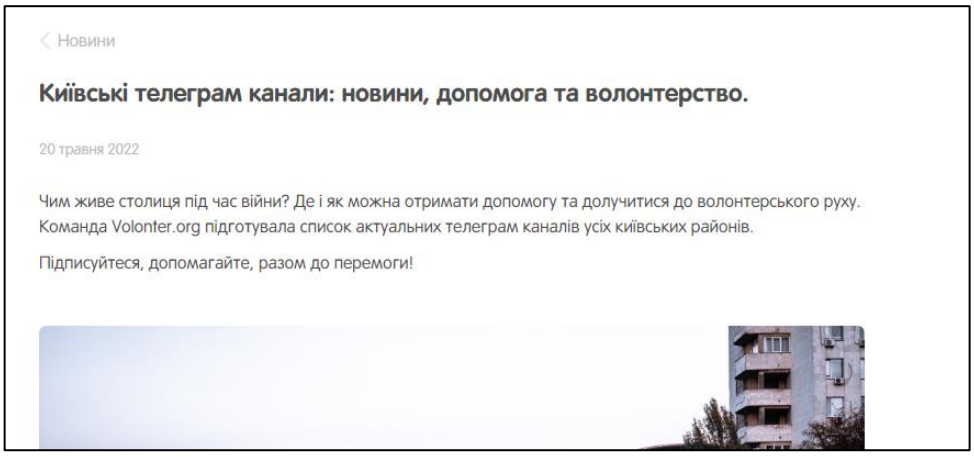


Рисунок 1.15– «Volonter.org» форма

Отже, проаналізувавши «Volonter.org», можна зробити висновок, що головною перевагою даного веб-додатку з поміж інших є можливість реєструватися та подавати заявки вже в системі.

1.3. Мета та задачі дослідження

Головною метою розробки інформаційної технології по волонтерській допомозі є спрощення співпраці та зручності у використанні наданої інформації. При виконанні даної технології повинно бути вирішено такі як:

– Розповсюдження інформації, що до відкритих фінансових зборів;

– Відкритість інформації, де й як можна придбати військове спорядження;

– Реалізація можливості подання запиту волонтеру на придбання військового спорядження;

– Полегшення пошуку та комунікації із волонтерами.

До цільової аудиторії можна віднести:

– волонтери;

– люди, що потребують допомоги в покупці спорядження чи іншого;

– люди, що зацікавленні в представленій інформації.

Розглянемо також основні функції до інформаційної технології:

1. Реєстрація як звичайний користувач або волонтер;
2. Редагування особистої інформації;
3. Додавання оголошень, що може допомогти придбати волонтер;
4. Фільтрація на сторінці з списом волонтерів;
5. Фільтрація за оголошеннями, що можуть допомогти придбати волонтери;
6. Волонтер може змінювати статус сторінки, чи може він допомогти в придбанні, чи ні;
7. Волонтер може додавати оголошення, для збору коштів;
8. Поскаржитися на користувача;
9. Подати запит на придбання речей, якщо волонтер у відповідному статусі;
10. Можливість формування автоматичної відповіді та розсилок на електронну пошту через адміністративну панель;

2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

2.1. Розробка візуальної частини інформаційної системи

Vue.js — це революційний фреймворк, який розробники програмного забезпечення впроваджують для створення користувальницьких інтерфейсів різної складності. Ця технологія допомагає розробляти ресурсні програми з багатим інтерфейсом користувача, оскільки її основна бібліотека зосереджена на рівні перегляду. Продовжуйте читати, щоб дізнатися більше про його переваги та те, як він може змінити ваш бізнес.

Таблиця 2.1 – Переваги використання Vue.js

№	Назва	Детальна інформація
1	Створення прототипів для спеціалістів початкового рівня	Для дизайнерів з невеликим досвідом створити прототип за допомогою Vue не складе великих труднощів [9]. Створення компонентів за допомогою шаблонів JS, HTML, CSS і HTML дозволяє користувачам легко розпочинати свої проекти. Крім того, за допомогою інтерфейсу користувача Vue інженер-програміст може налаштувати та організувати програму так, щоб швидко скористатися перевагами її компонентів.
2	Інтеграція програми	Фреймворк Vue є чудовим варіантом, коли вам потрібно інтегрувати поточні MPA та SPA, відтворені сервером.
3	Анімація та взаємодія	Анімація допомагає вам легко привернути увагу клієнта, а також утримати його. Тому це одна з важливих частин програм, які використовуються сьогодні. Фреймворк Vue.js пропонує API із плавним переходом між переглядом і регульованою архітектурою [18].

№	Назва	Детальна інформація
4	Легкість навчання	Одним із найважливіших факторів, які допомагають людям вирішити, чи вивчати певну технологію, є її крива навчання. І у випадку з фреймворком Vue.js саме це робить його таким популярним. Щоб вивчити Vuejs, вам не потрібно володіти такими інструментами, як TypeScript, JSX або будь-якою іншою бібліотекою, на відміну від інших інтерфейсних технологій.
5	Легкочитаний код	Частини майбутньої веб-сторінки або програми, створені у рамках Vue.js, називаються компонентами. Вони позначають захоплені елементи інтерфейсів і можуть бути закодовані в JS, CSS і HTML.
6	Найкраще підходить для проведення модульного тестування	Модульні тести використовуються для перевірки незалежності роботи невеликих частин програми. Цей компонентний підхід допомагає оптимізувати роботу програми.
7	Широкий вибір інструментів	Фреймворк Vue.js має безліч інструментів, які покращують його функціонування. Vue CLI пропонує безліч інноваційних характеристик. Розробники програмного забезпечення можуть розширити його та поділитися з спільнотою. Таким чином, проекти, які впроваджують CLI, можуть залишатися оновленими протягом тривалого часу.
8	Гнучкість і інтегрованість	Індикатором інтенсивної еволюції будь-якого програмного засобу є його інтегрованість із найпопулярнішими та найпоширенішими програмами. Фреймворк Vuejs виявився добре інтегрованим.

№	Назва	Детальна інформація
		Порівняно з монолітними фреймворками конкурентів, Vue.js надзвичайно адаптивний.
9	Прогресивність	Якщо вам потрібно підвищити продуктивність одного зі своїх проектів, гарною ідеєю буде переробити його у рамках Vue.js [10,13]. Завдяки своїм прогресивним функціям Vue.js може постійно потрапляти в базу даних коду без необхідності переписувати весь елемент.
10	Комплексна документація	Ретельна документація та навчальні матеріали дуже допоможуть у процесі розробки. Ці вказівки показують можливості, які пропонує фреймворк, включно з усіма останніми вдосконаленнями та оновленнями, які були представлені у Vue.js.

Vue.js — чудовий вибір, коли вам потрібен гнучкий масштабований інструмент для вашого проекту JS. Завдяки своїй структурі фреймворк простий у вивченні навіть початківцям і дозволяє легко відстежувати помилки. В основному він популярний для створення SPA та веб-інтерфейсів, але також може використовуватися для створення мобільних і веб-додатків. Якщо вам потрібна порада щодо розробки Vue.js, зверніться до нашої команди [10-12, 17].

Якщо у вас є існуюча програма, і ви хочете додати до неї інтерактивності, Vue може допомогти в цьому. Оскільки він заснований на JavaScript, його можна легко інтегрувати в будь-який проект за допомогою JS. Крім того, він сумісний із багатьма серверними технологіями та фреймворками, такими як Laravel, Express, Rails і Django. Також додайте сюди легкість навчання, щоб охопити аспект командного навчання.

2.2. Розробка функціональної частини інформаційної системи

PHP — це широко використовувана мова сценаріїв загального призначення з відкритим вихідним кодом, яка особливо підходить для розробки інформаційної технології та може бути вбудована в HTML.

Найкраще у використанні PHP те, що він дуже простий для початківців, але має багато додаткових функцій для професійних програмістів [13-15]. Сервер можна легко налаштувати для обслуговування всіх файлів HTML за допомогою PHP.

PHP має безліч переваг, які потрібно розглянути ближче. Ось деякі із них:

– Кросплатформенність. PHP не залежить від платформи. Він працює на всіх платформах, включаючи Mac, Windows і Linux, тому не потрібно використовувати певну операційну систему;

– Відкритий код. PHP є відкритим кодом. Оригінальний код доступний кожному, хто хоче його використовувати в розробці. Це одна з причин, чому один із їхніх фреймворків, Laravel, настільки популярний;

– Легкий у вивченні. Для новачків не складно вивчити PHP, якщо у є знання програмування і бажання трохи навчитися;

– PHP синхронізується з усіма базами даних. PHP може підключатися до будь-якої бази даних, пов’язаної чи не пов’язаної з головною системою. Таким чином, наявна можливість швидко підключитися до MySQL, Postgress, MongoDB та інших [16];

– Спільнота підтримки. PHP має дуже опорову спільноту розробників. Офіційна документація містить інструкції щодо використання цієї функції, що полегшує розв’язання проблем.

Laravel надає багатий набір функцій, який включає основні функції фреймворків PHP. Laravel має достатній набір функцій для полегшення та прискорення розробки інформаційних систем та технологій.

Laravel пропонує такі переваги:

– Завдяки фреймворку Laravel покращується масштабування створеної інформаційної технології;

– Laravel повторно використовує компоненти з інших фреймворків під час розробки інформаційних технологій [19], заощаджуючи багато часу під час розробки інформаційної технології;

– Laravel містить структуру імен та інтерфейси, які допомагають організувати ресурси та керувати ними.

Отже, для створення функціональної частини було обрано Laravel за ряд переваг. Інформаційні технології створені за допомогою Laravel добре масштабуються та мають добре структуровану кодову базу. Модульна система Laravel та надійне керування зв’язками дозволяють розробникам легко орієнтуватися в структурі та швидко розширювати функціональність своїх програмних продуктів. Laravel містить інструменти, які допомагають швидше створювати інформаційні технології та економити час розробки. Після створення інформаційної технології, кодова база буде добре структурована та її буде легко підтримувати.

3 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ

3.1 Структура інформаційної технології

3.1.1. Загальна інформація про структуру інформаційної технології для не зареєстрованих користувачів

Загальна структура інформаційної технології по волонтерській допомозі являє собою набір функціонала, що буде представлений сторінками:

– Головна: сторінка на якій буде представлена загальна інформація, статистика, топ товарів в каталозі, топ волонтерів по проведеній роботі та кнопки швидкого переходу.

– Волонтери: сторінка на якій представлена коротка інформація про кожного зареєстрованого в системі волонтера. При натисканні на блок із обраним волонтером, відбувається перехід на сторінку із повною інформацією. Доступний функціонал для фільтрації за такими параметрами як ФІО, рейтинг, категорії, статус волонтера.

– Оголошення: сторінка із оголошеннями зареєстрованих користувачів. На даній сторінці користувача матимуть змогу виконувати пошук та відповідати на них.

– Каталог: сторінка із представленими товарами, що можуть придбати волонтери. Також до кожного із товарів додана інформація із посиланнями, де саме можна їх придбати. На цій сторінці також представлений функціонал для фільтрації товарів за назвою, країною виробника, категорією, статусом волонтера.

– Збір коштів: сторінка із представленими відкритими зборами волонтерів. При натисканні на обраний блок, то відбувається перехід на сторінку із детальною інформацією про збір: назва приладу чи спорядження, опис, посилання на прилад.

– Про нас: загальна коротка інформація про розробників інформаційної технології по волонтерській допомозі та інформація про призначенні розробленої технології.

– Реєстрація/Авторизація: сторінка для реєстрації нових користувачів та авторизації вже зареєстрованих раніше користувачів.

3.1.2. Загальна інформація про структуру інформаційної технології для зареєстрованих користувачів волонтерів

Зареєстровані користувачі інформаційної технології по волонтерській допомозі матимуть додатковий функціонал в залежності від типу користувача. Для зареєстрованих користувачів волонтерів буде доданий такий набір функціонала, що буде представлений сторінками:

– Додати оголошення: на даній сторінці користувач типу «волонтер» має можливість додати оголошення, що до речей які він може придбати. Також це належати до інформації яку він може просто розповсюдити для інших волонтерів.

– Додати підтримку: на даній сторінці користувач типу «волонтер» має можливість додати оголошення про збір коштів. Користувач може додати помітку, що це терміновий збір.

– Кабінет: сторінка із загальною інформацією про користувача системи.

– Чат: сторінка для комунікації із користувачами системи.

– Редагувати: сторінка для редагування особистої інформації. Також користувач має можливість змінити пароль на цій сторінці.

– Вихід: вихід з авторизованої сторінки.

3.1.3. Загальна інформація про структуру інформаційної технології для звичайних зареєстрованих користувачів

Звичайні зареєстровані користувачі на відміну він не зареєстрованих користувачів мають додаткову функцію:

– Додати пошук: сторінка на якій користувачі мають можливість створити оголошення для пошуку описаного спорядження.

3.2 Розподіл функцій за групами користувачів інформаційної технології

Адміністратори:

– Можливість формування автоматичної відповіді на електрону пошту через адмін-панель;

– Додавання нових адміністраторів;

– Редагування даних;

– Розгляд запитів від зареєстрованих користувачів;

– Написання користувачеві через адміністративну панель;

– Блокування користувачів.

Волонтер:

– Реєстрація та авторизація як волонтер;

– Редагування особистої інформації;

– Додавання оголошення, що може допомогти придбати волонтер;

– Додавання оголошення, для збору коштів;

– Фільтрація на сторінці за списом волонтерів;

– Фільтрація за оголошеннями, що можуть допомогти придбати волонтери;

– Волонтер може змінювати статус сторінки, чи може він допомогти в придбанні, чи ні;

– Поскаржитися на користувача;

– Подати запит на придбання речей, якщо волонтер у відповідному статусі.

Звичайний користувач інформаційної системи:

– Реєстрація та авторизація як звичайний користувач;

– Редагування особистої інформації;

– Додавання оголошення, для пошуку спорядження;

– Фільтрація на сторінці із списком волонтерів;

– Фільтрація за оголошеннями, що можуть допомогти придбати волонтери;

– Поскаржитися на користувача;

– Подати запит на придбання речей, якщо волонтер у відповідному статусі.

3.3. Проектування бази даних

Для якісної та повноцінної роботи інформаційної технології було розроблено ER-діаграма інформаційної технології по волонтерській допомозі. Дана структура

допоможе розробити систему взаємодію всіх моделей, що використовуватимуться в функціоналі [20-23].

Детальна інформація за кожною із таблицею представлено в таблиці 3.1.

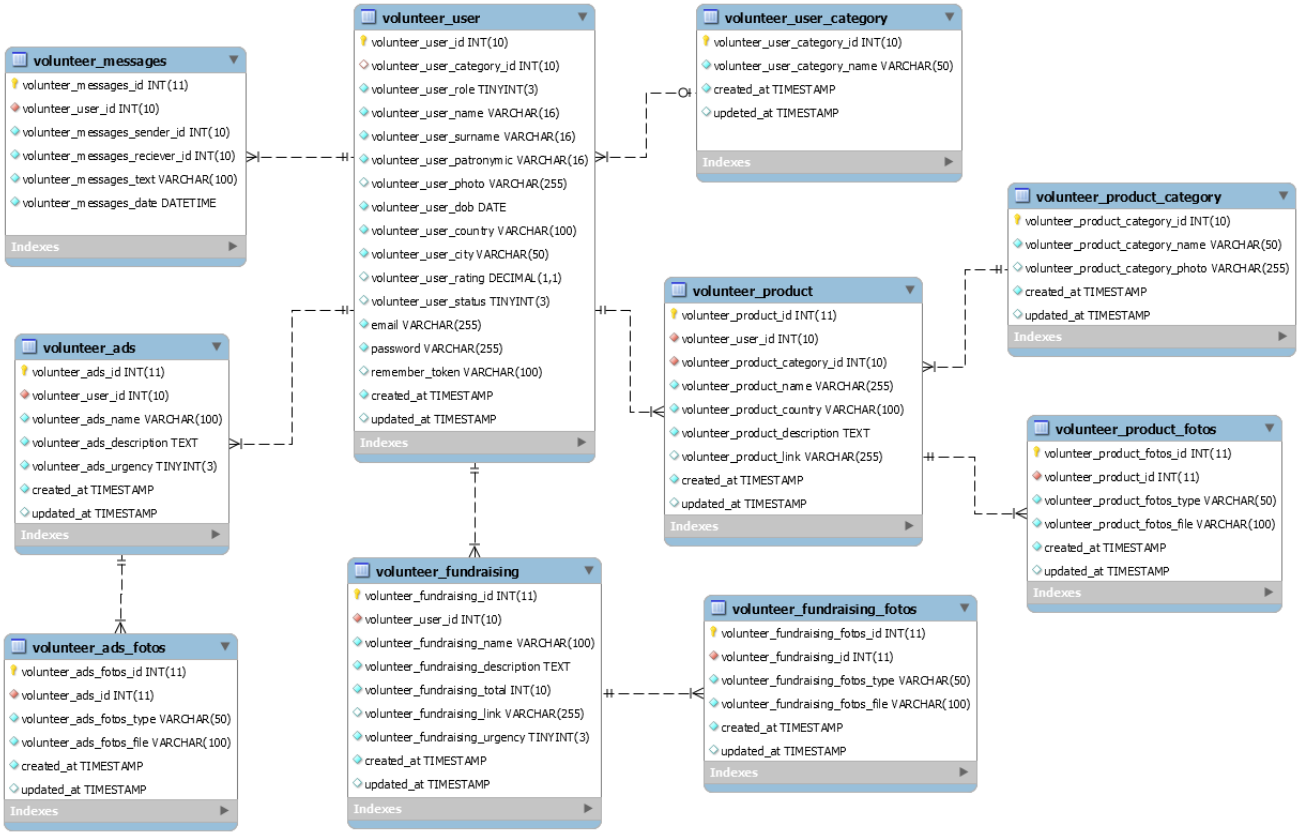


Рисунок 3.1 – ER-діаграма інформаційної технології по волонтерській допомозі

Таблиця 3.1 – Інформація за моделями ER-діаграми інформаційної технології по волонтерській допомозі

№	Таблиця	Поле	Зміст	Тип	Ключі	Умови
1	volunteer_user	volunteer_user_id	Ідентифікатор аккаунта користувача технології	INTEGER	PK	Не пустий
		volunteer_user_category_id	Ідентифікатор категорії аккаунта користувача технології	INTEGER	FK	Не пустий
		volunteer_user_role	Роль користувача технології	INTEGER	FK	Не пустий
		volunteer_user_name	Ім’я користувача технології	VARCHAR(16)		Не пустий
		volunteer_user_surname	Прізвище користувача технології	VARCHAR(16)		Не пустий
		volunteer_user_patronym		VARCHAR(16)		Не пустий
		volunteer_user_photo	Фото користувача технології	VARCHAR(225)		Не пустий
		volunteer_user_dob	Дата народження користувача технології	DATE		Не пустий
		volunteer_user_country	Назва країни	VARCHAR(100)		Не пустий
		volunteer_user_city	Назва міста	VARCHAR(50)		Не пустий
		volunteer_user_rating	Рейтинг в системі користувача технології	DECIMAL(1,1)		Не пустий
		volunteer_user_status	Статус користувача	INTEGER		Не пустий
		volunteer_user_email	Електронна пошта користувача технології	VARCHAR(50)		Не пустий
		volunteer_user_password	Пароль користувача технології	DECIMAL(1,1)		Не пустий

№	Таблиця	Поле	Зміст	Тип	Ключі	Умови
		volunteer_user_token	користувача технології	VARCHAR(100)		Не пустий
		volunteer_user_created_at	Дата реєтрції користувача технології	TIMESTAMP		Не пустий
		volunteer_user_updated_at	Дата оновлення інформації користувача технології	TIMESTAMP		Не пустий
2	volunteer_messages	volunteer_messages_id	Ідентифікатор повідомлення користувача технології	INTEGER	PK	Не пустий
		volunteer_user_id	Ідентифікатор аккаунта користувача технології	INTEGER	FK	Не пустий
		volunteer_messages_sender_id	Ідентифікатор аккаунта відправника повідомлення	INTEGER		Не пустий
		volunteer_messages_recieve_id	Ідентифікатор аккаунта отримувача повідомлення	INTEGER		Не пустий
		volunteer_messages_text	Текст повідомлення	VARCHAR(100)		Не пустий
		volunteer_messages_date	Дата відправлення повідомлення	DATETIME		Не пустий
3	volunteer_ads	volunteer_ads_id	Ідентифікатор оголошення користувача технології	INTEGER	PK	Не пустий
		volunteer_user_id	Ідентифікатор аккаунта користувача технології	INTEGER	FK	Не пустий
		volunteer_ads_name	Назва оголошення користувача	VARCHAR(100)		Не пустий

№	Таблиця	Поле	Зміст	Тип	Ключі	Умови
		volunteer_ads_description	Опис оголошення користувача	TEXT		Не пустий
		volunteer_ads_urgency	Статус терміновості оголошення	TINYINT		Не пустий
		volunteer_ads_created_at	Дата реєтрції оголошення користувача технології	TIMESTAMP		Не пустий
		volunteer_ads_updated_at	Дата оновлення інформації оголошення користувача технології	TIMESTAMP		Не пустий
3	volunteer_ads_fotos	volunteer_ads_fotos_id	Ідентифікатор фото оголошення користувача	INTEGER	PK	Не пустий
		volunteer_ads_id	Ідентифікатор оголошення користувача	INTEGER	FK	Не пустий
		volunteer_ads_fotos_type	Тип фото оголошення користувача	VARCHAR(100)		Не пустий
		volunteer_ads_fotos_file	Файл оголошення користувача	VARCHAR(100)		Не пустий
		volunteer_ads_created_at	Дата реєтрації фото оголошення користувача	TIMESTAMP		Не пустий
		volunteer_ads_updated_at	Дата оновлення фото оголошення користувача технології	TIMESTAMP		Не пустий

№	Таблиця	Поле	Зміст	Тип	Ключі	Умови
4	volunteer_user_category	volunteer_user_category_id	Ідентифікатор категорії аккаунта користувача технології	INTEGER	PK	Не пустий
		volunteer_user_category_name	Назва категорії аккаунта користувача технології	VARCHAR(100)		Не пустий
		volunteer_ads_created_at	Дата реєтрції назви категорії	TIMESTAMP		Не пустий
		volunteer_ads_updated_at	Дата оновлення назви категорії	TIMESTAMP		Не пустий
5	volunteer_fundrasing	volunteer_fundrasing_id	Ідентифікатор збору коштів	INTEGER	PK	Не пустий
		volunteer_user _id	Ідентифікатор аккаунта користувача технології	INTEGER	FK	Не пустий
		volunteer_fundrasing_name	Назва збору коштів	VARCHAR(100)		Не пустий
		volunteer_fundrasing_description	Опис збору коштів	TEXT		Не пустий
		volunteer_fundrasing_total	Загальний збір коштів	INTEGER		Не пустий
		volunteer_fundrasing_link	Посилання на збір коштів	VARCHAR(100)		Не пустий
		volunteer_fundrasing_urgency	Терміновість збору коштів	TINYINT		Не пустий
		volunteer_fundrasing _created_at	Дата реєтрції сбору коштів	TIMESTAMP		Не пустий
		volunteer_fundrasing _updated_at	Дата оновлення збору коштів	TIMESTAMP		Не пустий
6	volunteer_fundrasing_fotos	volunteer_fundrasing_fotos_id	Ідентифікатор фото збору коштів	INTEGER	PK	Не пустий

№	Таблиця	Поле	Зміст	Тип	Ключі	Умови
		volunteer_fundrasing_id	Ідентифікатор збору коштів	INTEGER	FK	Не пустий
		volunteer_fundrasing_fotos_type	Тип фото збору коштів	VARCHAR(100)		Не пустий
		volunteer_fundrasing_fotos_file	Файл збору коштів	VARCHAR(100)		Не пустий
		volunteer_fundrasing_fotos_created_at	Дата реєтрції фото збору коштів	TIMESTAMP		Не пустий
		volunteer_fundrasing_fotos_updated_at	Дата оновлення фото збору коштів	TIMESTAMP		Не пустий
7	volunteer_product	volunteer_product_id	Ідентифікатор продукту користувача	INTEGER	PK	Не пустий
		volunteer_user_id	Ідентифікатор аккаунта користувача технології	INTEGER	FK	Не пустий
		volunteer_product_category_id	Ідентифікатор категорії продукту	INTEGER		Не пустий
		volunteer_product_name	Назва продукту	VARCHAR(100)		Не пустий
		volunteer_product_country	Країна походження продукту	VARCHAR(100)		Не пустий
		volunteer_product_description	Опис продукту	TEXT		Не пустий
		volunteer_product_link	Посилання на продукт	VARCHAR(100)		Не пустий
		volunteer_product_created_at	Дата реєтрції продукту	TIMESTAMP		Не пустий
		volunteer_product_updated_at	Дата оновлення продукту	TIMESTAMP		Не пустий
8	volunteer_product_category	volunteer_product_category_id	Ідентифікатор категорії продукту	INTEGER	PK	Не пустий

№	Таблиця	Поле	Зміст	Тип	Ключі	Умови
		volunteer_product_category_name	Назва категорії продукту	VARCHAR(100)		Не пустий
		volunteer_product_category_photo	Фото категорії продукту	VARCHAR(100)		Не пустий
		volunteer_product_category_created_at	Дата реєтрції назви категорії продукту	TIMESTAMP		Не пустий
		volunteer_product_category_updated_at	Дата оновлення категорії продукту	TIMESTAMP		Не пустий
9	volunteer_product_fotos	volunteer_product_fotos_id	Ідентифікатор фото продукту	INTEGER	PK	Не пустий
		volunteer_product_id	Ідентифікатор продукту	INTEGER	FK	Не пустий
		volunteer_product_fotos_type	Тип фото продукту	VARCHAR(100)		Не пустий
		volunteer_product_fotos_file	Файл фото продукту	VARCHAR(100)		Не пустий
		volunteer_product_fotos_created_at	Дата реєтрції фото продукту	TIMESTAMP		Не пустий
		volunteer_product_fotos_updated_at	Дата оновлення фото продукту	TIMESTAMP		Не пустий

4 ПРАКТИЧНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ

4.1 Програмна реалізація

Програмна реалізація інформаційної технології по волонтерській допомозі представлена на рисунку 4.1. На даному рисунку зображено проект, що складаться із багатьох модулів. Всі ці модулі та підключені бібліотеки взаємодіють за технологією MVC.

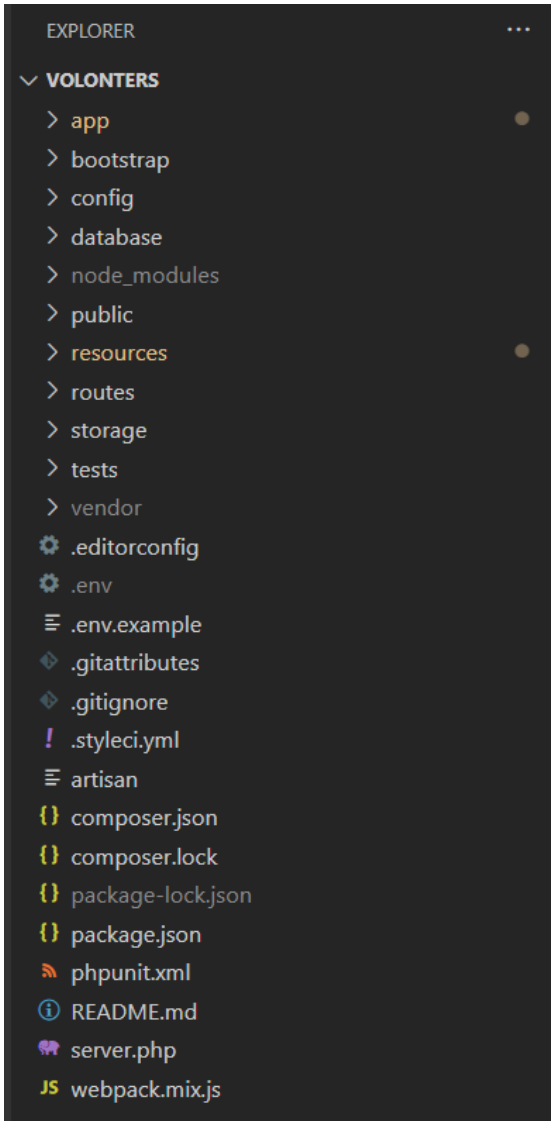


Рисунок 4.1 – «volonters» програмна реалізація

Також для роботи із даними було виконано фізичну реалізацію бази даних(рис.4.2-4.16).

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id	bigint(20)		UNSIGNED	Нет	*Нет*		AUTO_INCREMENT
2	volunteer_title	varchar(255)	utf8mb4_unicode_ci		Нет	*Нет*		
3	volunteer_text	text	utf8mb4_unicode_ci		Нет	*Нет*		

Рисунок 4.2 – «volunteer_about» таблиця

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	volunteer_ads_id	int(10)		UNSIGNED	Нет	*Нет*		AUTO_INCREMENT
2	volunteer_ads_role_id	int(10)		UNSIGNED	Да	*NULL*		
3	volunteer_user_id	int(10)		UNSIGNED	Нет	*Нет*		
4	volunteer_country_id	int(10)		UNSIGNED	Да	*NULL*		
5	volunteer_ads_name	varchar(255)	utf8mb4_unicode_ci		Нет	*Нет*		
6	volunteer_ads_description	text	utf8mb4_unicode_ci		Нет	*Нет*		
7	volunteer_ads_urgency	tinyint(4)			Да	*NULL*		
8	volunteer_ads_link	varchar(255)	utf8mb4_unicode_ci		Да	*NULL*		
9	volunteer_cost	bigint(20)			Нет	0		
10	volunteer_type	varchar(255)	utf8mb4_unicode_ci		Нет	*Нет*		
11	volunteer_freeze	tinyint(1)			Нет	0		

Рисунок 4.3 – «volunteer_ads» таблиця

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	volunteer_ads_role_id	int(10)		UNSIGNED	Нет	*Нет*		AUTO_INCREMENT
2	volunteer_ads_role_name	varchar(255)	utf8mb4_unicode_ci		Нет	*Нет*		

Рисунок 4.4 – «volunteer_ads_role» таблиця

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id	bigint(20)		UNSIGNED	Нет	*Нет*		AUTO_INCREMENT
2	volunteer_user_id	int(10)		UNSIGNED	Нет	*Нет*		
3	volunteer_ads_id	int(10)		UNSIGNED	Нет	*Нет*		
4	volunteer_text	text	utf8mb4_unicode_ci		Нет	*Нет*		

Рисунок 4.5 – «volunteer_complaints_ads» таблиця

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id	bigint(20)		UNSIGNED	Нет	*Нет*		AUTO_INCREMENT
2	volunteer_user_id	int(10)		UNSIGNED	Нет	*Нет*		
3	volunteer_to_user_id	int(10)		UNSIGNED	Нет	*Нет*		
4	volunteer_text	text	utf8mb4_unicode_ci		Нет	*Нет*		

Рисунок 4.6 – «volunteer_complaints_user» таблиця

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	volunteer_country_id	int(10)		UNSIGNED	Нет	*Нет*		AUTO_INCREMENT
2	volunteer_country_title	varchar(255)	utf8mb4_unicode_ci		Нет	*Нет*		

Рисунок 4.7 – «volunteer_country» таблиця

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id	bigint(20)		UNSIGNED	Нет	*Нет*		AUTO_INCREMENT
2	volunteer_to_users_id	bigint(20)		UNSIGNED	Нет	*Нет*		
3	volunteer_from_users_id	bigint(20)		UNSIGNED	Нет	*Нет*		
4	created_at	timestamp			Да	*NULL*		

Рисунок 4.8 – «volunteer_dialogs» таблиця

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id	bigint(20)		UNSIGNED	Нет	*Нет*		AUTO_INCREMENT
2	volunteer_users_id	bigint(20)		UNSIGNED	Нет	*Нет*		
3	volunteer_dialogs_id	bigint(20)		UNSIGNED	Нет	*Нет*		
4	volunteer_text	text	utf8mb4_unicode_ci		Нет	*Нет*		
5	created_at	timestamp			Да	*NULL*		

Рисунок 4.9 – «volunteer_dialogs_messages» таблиця

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id	bigint(20)		UNSIGNED	Нет	*Нет*		AUTO_INCREMENT
2	volunteer_user_id	int(10)		UNSIGNED	Нет	*Нет*		
3	volunteer_ads_id	int(10)		UNSIGNED	Нет	*Нет*		
4	volunteer_sum	int(11)			Нет	*Нет*		
5	created_at	timestamp			Да	*NULL*		

Рисунок 4.10 – «volunteer_donate» таблиця

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	volunteer_user_evaluations_id	int(10)		UNSIGNED	Нет	*Нет*		AUTO_INCREMENT
2	volunteer_user_id	int(10)		UNSIGNED	Нет	*Нет*		
3	volunteer_user_evaluations_reciever_id	int(10)		UNSIGNED	Нет	*Нет*		
4	volunteer_user_evaluations_value	int(11)			Нет	*Нет*		

Рисунок 4.11 – «volunteer_evalutations» таблиця

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	volunteer_fotos_id	int(10)		UNSIGNED	Нет	*Нет*		AUTO_INCREMENT
2	volunteer_ads_id	int(10)		UNSIGNED	Нет	*Нет*		
3	volunteer_fotos_file	varchar(255)	utf8mb4_unicode_ci		Нет	*Нет*		

Рисунок 4.12 – «volunteer_photos» таблиця

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id	bigint(20)		UNSIGNED	Нет	*Нет*		AUTO_INCREMENT
2	volunteer_title	varchar(255)	utf8mb4_unicode_ci		Нет	*Нет*		
3	volunteer_url	varchar(255)	utf8mb4_unicode_ci		Нет	*Нет*		
4	volunteer_file	varchar(255)	utf8mb4_unicode_ci		Нет	*Нет*		

Рисунок 4.13 – «volunteer_socials» таблиця

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	volunteer_user_role_id	int(10)		UNSIGNED	Нет	*Нет*		AUTO_INCREMENT
2	volunteer_user_role_name	varchar(50)	utf8mb4_unicode_ci		Нет	*Нет*		

Рисунок 4.14 – «volunteer_user_role» таблиця

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	volunteer_user_status_id	int(10)		UNSIGNED	Нет	*Нет*		AUTO_INCREMENT
2	volunteer_user_status_title	varchar(50)	utf8mb4_unicode_ci		Нет	*Нет*		

Рисунок 4.15 – «volunteer_user_status» таблиця

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	volunteer_user_id	int(10)		UNSIGNED	Нет	*Нет*		AUTO_INCREMENT
2	volunteer_user_name	varchar(255)	utf8mb4_unicode_ci		Нет	*Нет*		
3	volunteer_user_surname	varchar(255)	utf8mb4_unicode_ci		Нет	*Нет*		
4	volunteer_user_patronymic	varchar(255)	utf8mb4_unicode_ci		Нет	*Нет*		
5	volunteer_user_role_id	int(10)		UNSIGNED	Нет	*Нет*		
6	volunteer_email	varchar(255)	utf8mb4_unicode_ci		Нет	*Нет*		
7	volunteer_password	varchar(255)	utf8mb4_unicode_ci		Нет	*Нет*		
8	volunteer_user_photo	varchar(255)	utf8mb4_unicode_ci		Нет	/images/no-photo.png		
9	volunteer_user_status_id	int(10)		UNSIGNED	Нет	1		
10	volunteer_user_text	text	utf8mb4_unicode_ci		Да	*NULL*		
11	volunteer_country_id	int(10)		UNSIGNED	Да	*NULL*		
12	volunteer_is_ban	tinyint(1)			Нет	0		

Рисунок 4.16 – «volunteer_user» таблиця

Отже, результатом програмної реалізації інформаційної технології по волонтерській допомозі є структурно-вірно розроблена система взаємодії усіх функцій та модулів між собою.

4.2 Використання інформаційної технології незареєстрованими користувачами

Переходячи за посиланням на інформаційну технологію по волонтерській допомозі користувач опиняється на головній сторінці. Перший блок головної сторінки – коротка інформація та посилання на сторінки реєстрації (рис.4.17).

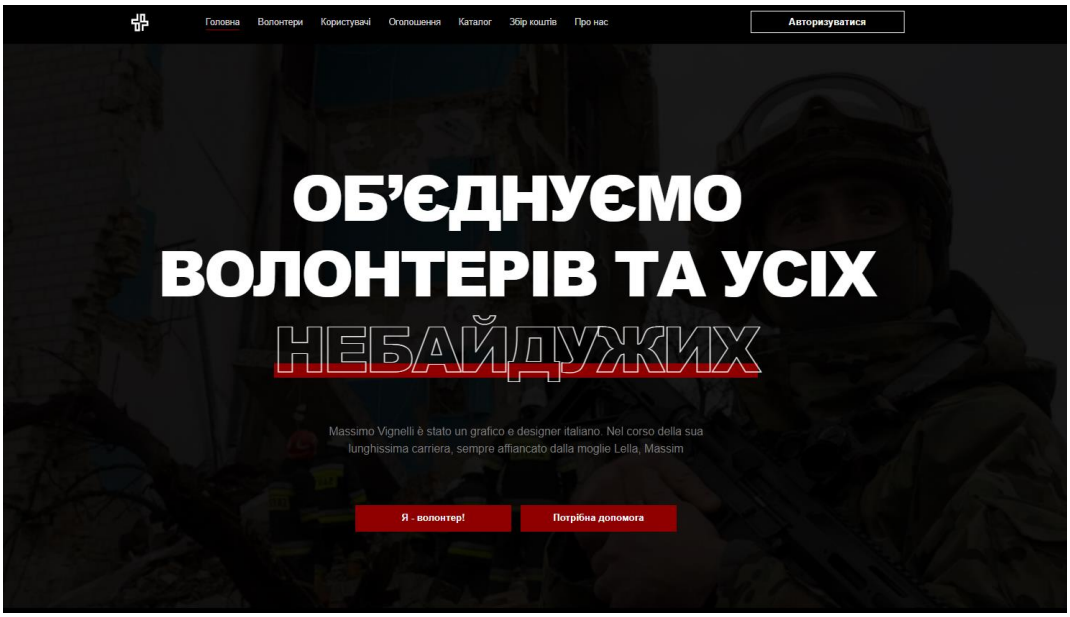


Рисунок 4.17 – Головна сторінка інформаційної технології по волонтерській допомозі (перший блок)

Одразу на головній сторінці трішки нижче знаходяться найактивніші волонтери, на сторінки яких можна перейти для детального перегляду (рис.4.18).

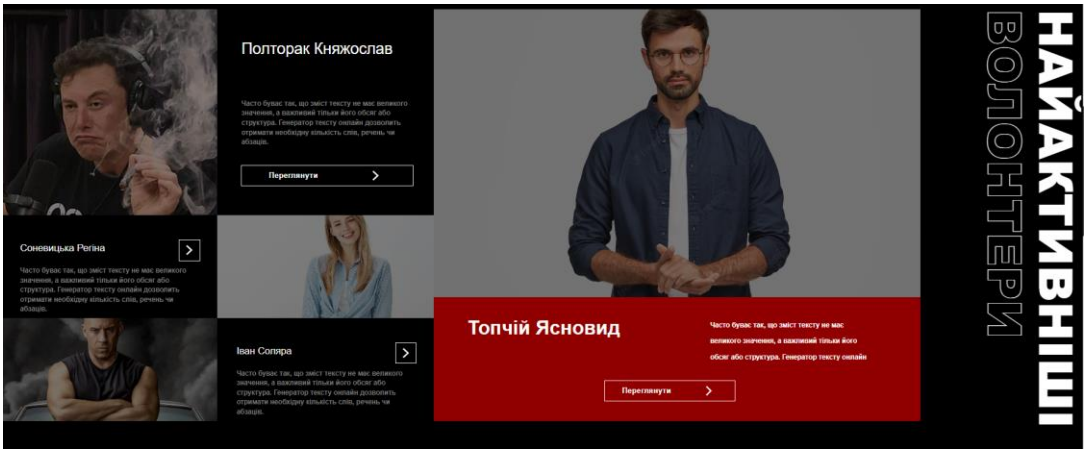


Рисунок 4.18 – Головна сторінка інформаційної технології по волонтерській допомозі (другий блок)

Для кращої контрастності на загальній картині сайту та щоб підкреслити виділений елемент сайту, додано анімацію виділення яка реагує на наведення курсору миші користувачем (рис.4.19-4.20).

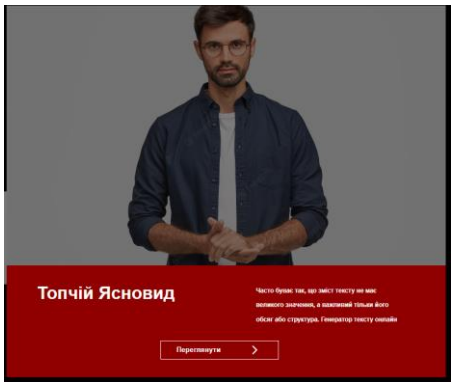


Рисунок 4.19 – Елемент сайту, на який користувач не навів курсор миші

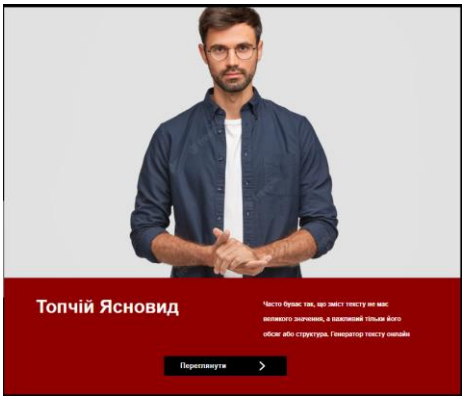


Рисунок 4.20 – Виділений елемент сайту

Нижче на цій сторінці розміщена панель термінових зборів для швидкого переходу по потрібний збір. Елементи цієї сторінки таклж мають спеціальне виділення, яке реагує на наведення курсора миші користувачем (рис.4.21-4.22).

Рисунок 4.21 – Частина головної сторінки «Термінові збори» (третій блок)

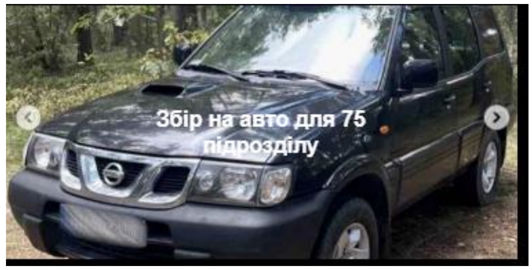


Рисунок 4.22 – Виділений елемент «Термінові збори»

Наступним блоком на головній сторінці є «Інформація про нас», користувач має змогу дізнатися деталі та додаткову інформацію що до інформаційної технології (рис.4.23).

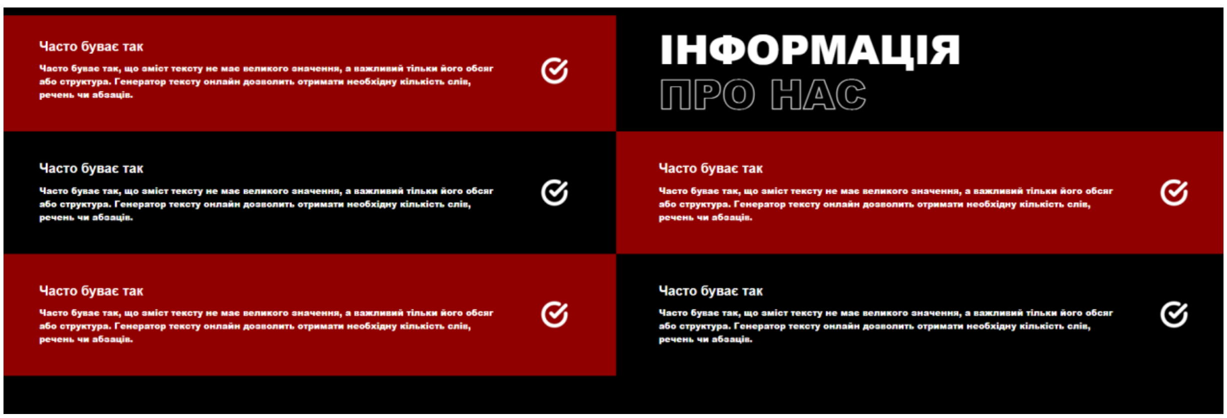


Рисунок 4.23 – Блок сайту «Інформація про нас»

Останнім блоком головної сторінки є «Статистика системи», в якій розміщена статистика про кількість оголошень, зібраних коштів, кількість зареєстрованих волонтерів та кількість лотів в каталозі (рис.4.24).

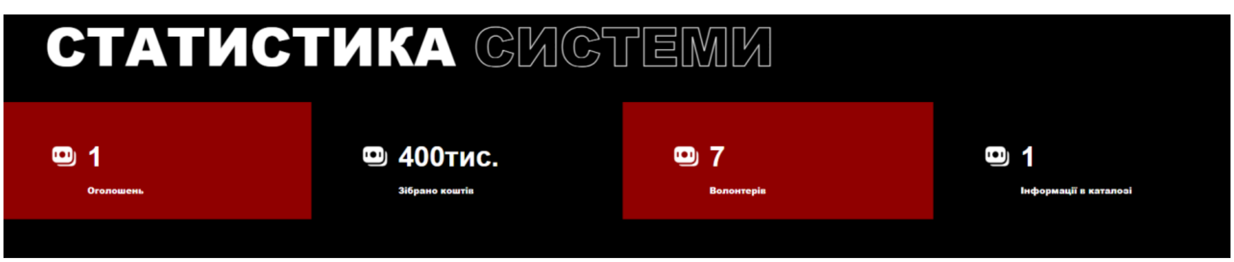


Рисунок 4.24 – Виділений елемент сайту

На сторінці «Волонтери» розміщена інформація про волонтерів з можливістю фільтрувати порядок відображення волонтерів за різними категоріями (рис.4.25).

Що до фільтрації, це відповідає тому, яку категорію продуктів може привезти волонтер, про яку вже має перевірену інформацію. Також можна виконувати фільтрацію за країною та статусом користувача-волонтера.

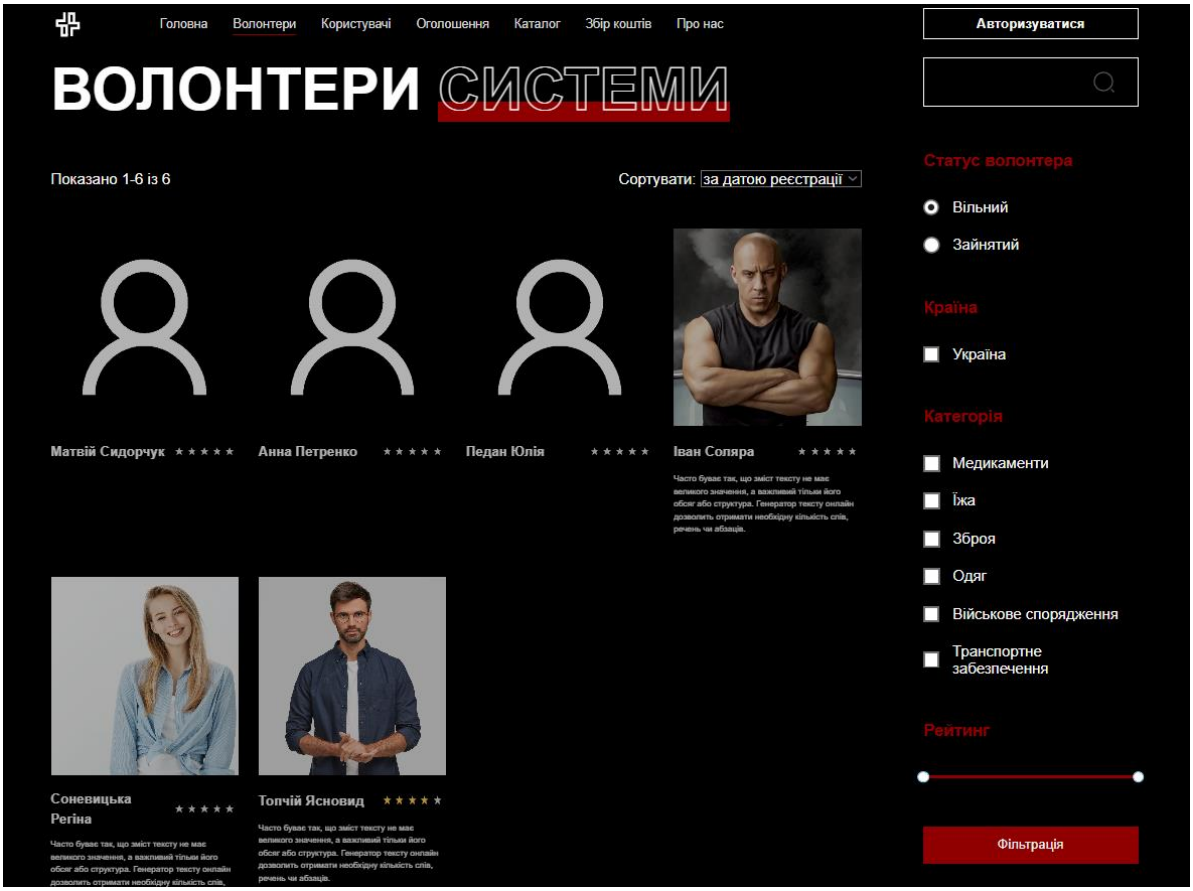
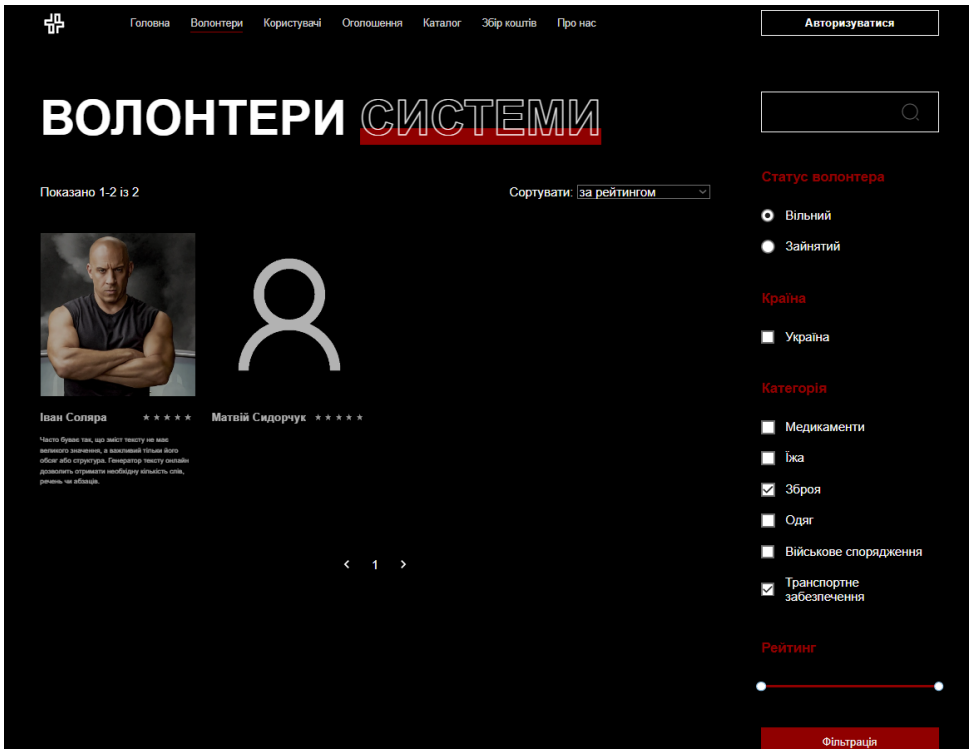


Рисунок 4.25 – Сторінка «Волонтери системи»

На рисунку 4.26 представлений набір відфільтрованих користувачів-волонтерів за категорями, а саме «зброя». Окрім цього, був обраний статус «вільний», що також надає додаткову інформацію, чи зможемо чи співпрацювати з ним.



Таблиц 4.26 – Фільтрація набору волонтерів за категоріями

Крім головної фільтрації, на сторінці присутня додаткове сортування за рейтингом чи реєстраційною датою. Як ми бачимо на рисунку 4.27 користувачі змінили своє положення.

Рисунок 4.27 – Сортування набору волонтерів

Натиснувши на бажаного волонтера, користувач переходить на його особисту сторінку. Перейшовши на обрану сторінку, можна переглянути детальну інформацію про волонтера, його рейтинг, його додані раніше збори та каталог (рис.4.28).

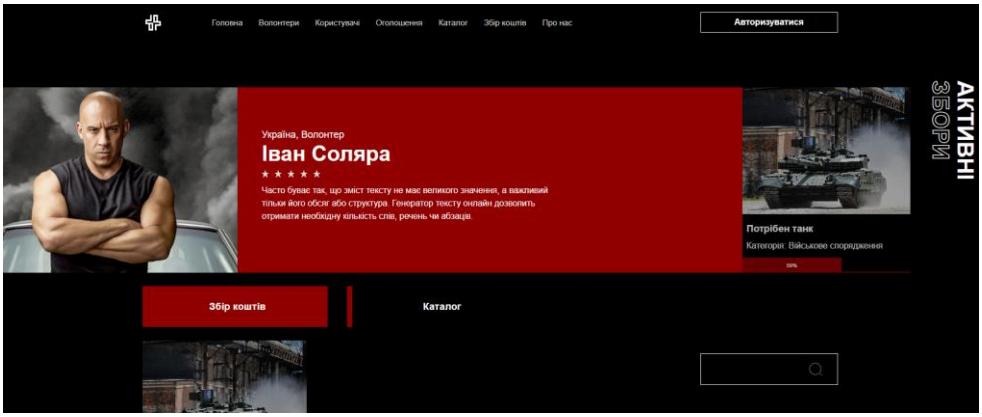


Рисунок 4.28 – Особиста сторінка волонтера

Перейшовши на вкладку «каталог» волонтера, користувач може ознайомитися із тим, що може надати волонтер (рис.4.29). Крім цього, на даній сторінці також присутня фільтрація.

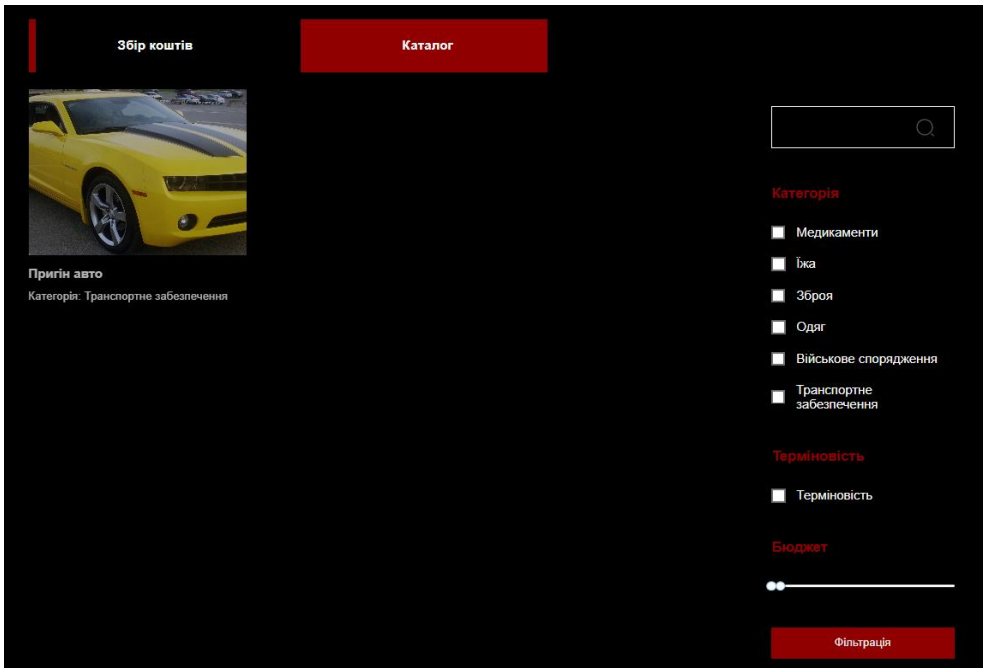


Рисунок 4.29 – Каталог волонтера

На сторінці «Користувачі системи» розміщені користувачі інформаційної технології (рис.4.30). Даний тип користувача відрізняється від волонтера тим, що вони можуть додавати лише оголошення по пошуку речей. На даній сторінці присутня функція фільтрації аналогічно до функції фільтрації на сторінці волонтерів.

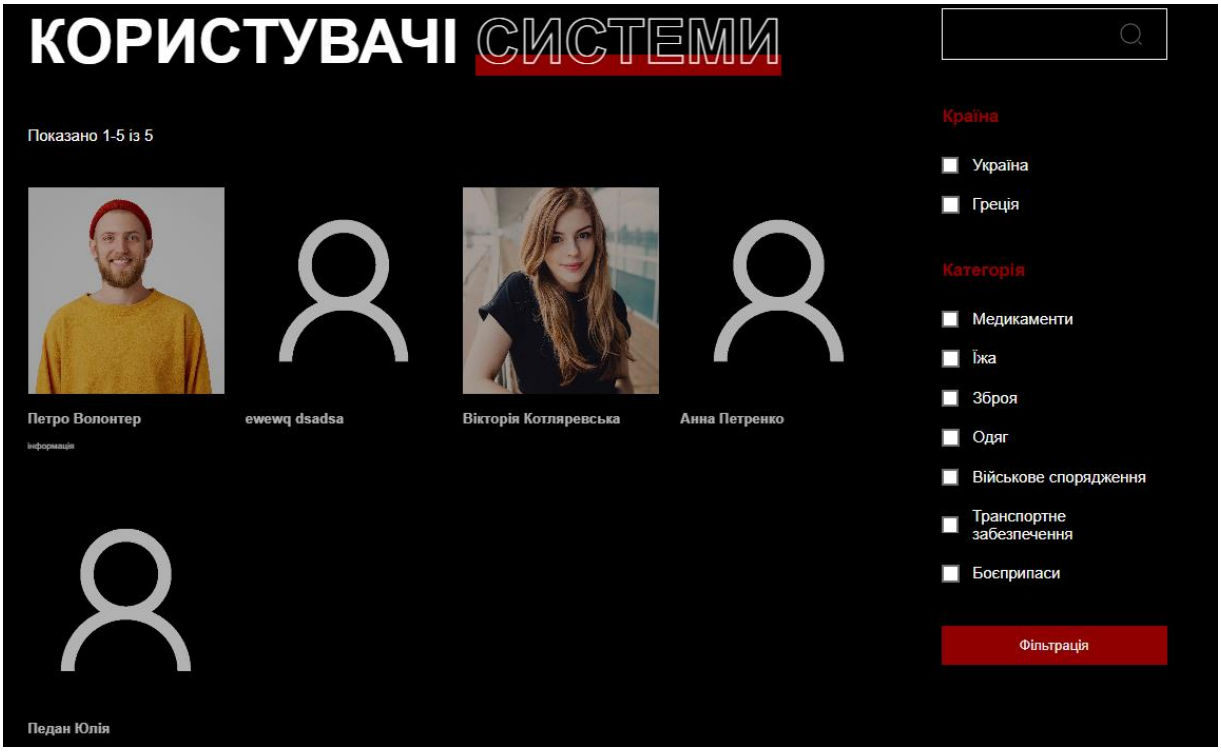


Рисунок 4.30 – Сторінка користувачів інформаційної технології

Перейдемо на сторінку «Оголошення». Саме тут, на рисунку 4.42, відображаються додані речі, які шукають звичайні зареєстровані користувачі інформаційної технології. На даній сторінці також присутні функція фільтрації. Від інших вона відрізняється новою категорією, а саме терміновість.

Перейдемо на обране оголошення інформаційної технології (рис.4.33). На даній сторінці відображається детальна інформація та посилання на приклад, що саме шукає користувач.

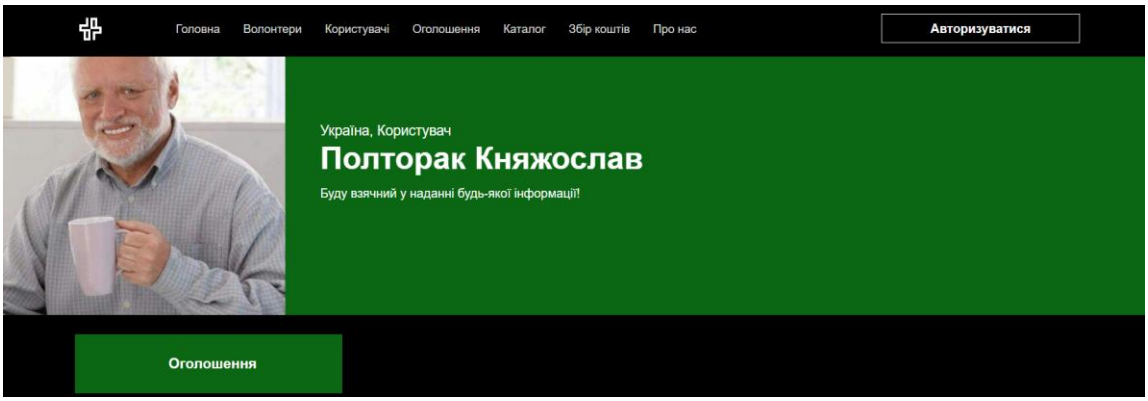


Рисунок 4.31 – Особиста сторінка звичайного користувача

Перейдемо на сторінку «Оголошення». Саме тут, на рисунку 4.42, відображаються додані речі, які шукають звичайні зареєстровані користувачі інформаційної технології. На даній сторінці також присутні функція фільтрації. Від інших вона відразняється новою категорією, а саме терміновість.

Перейдимо на обране оголошення інформаційної технології (рис.4.33). На даній сторінці відображається детальна інформація та посилання на приклад, що саме шукає користувач.

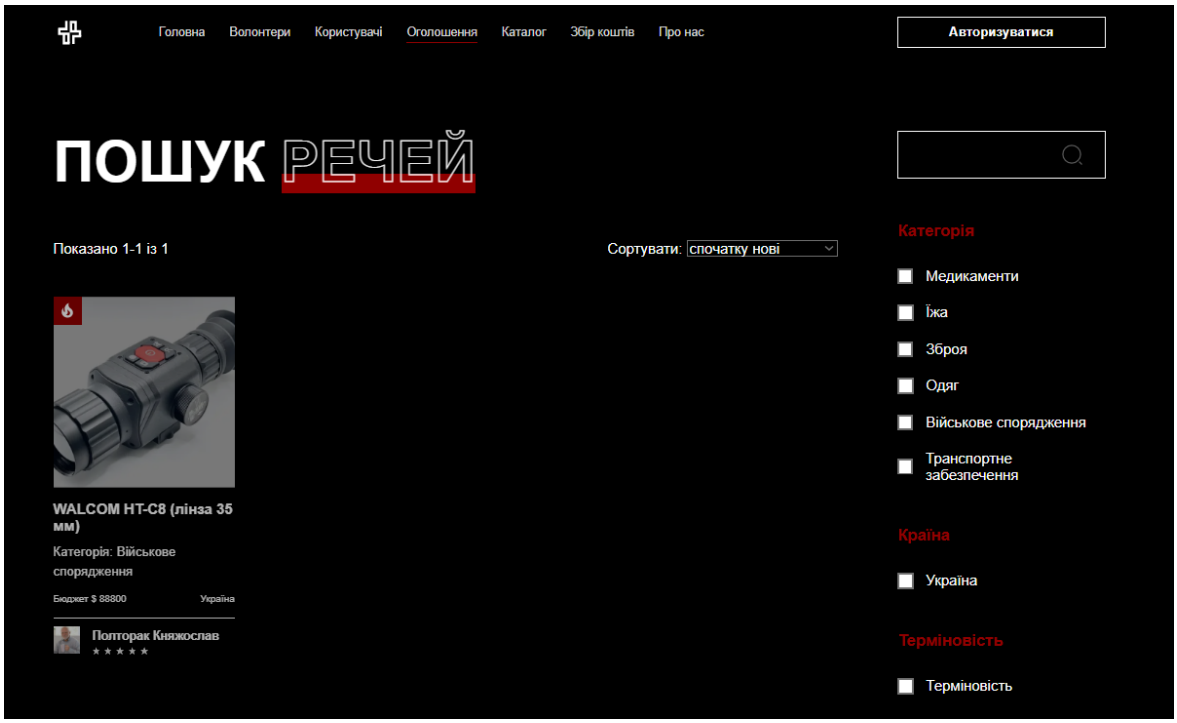


Рисунок 4.32 – Сторінка з оголошеннями

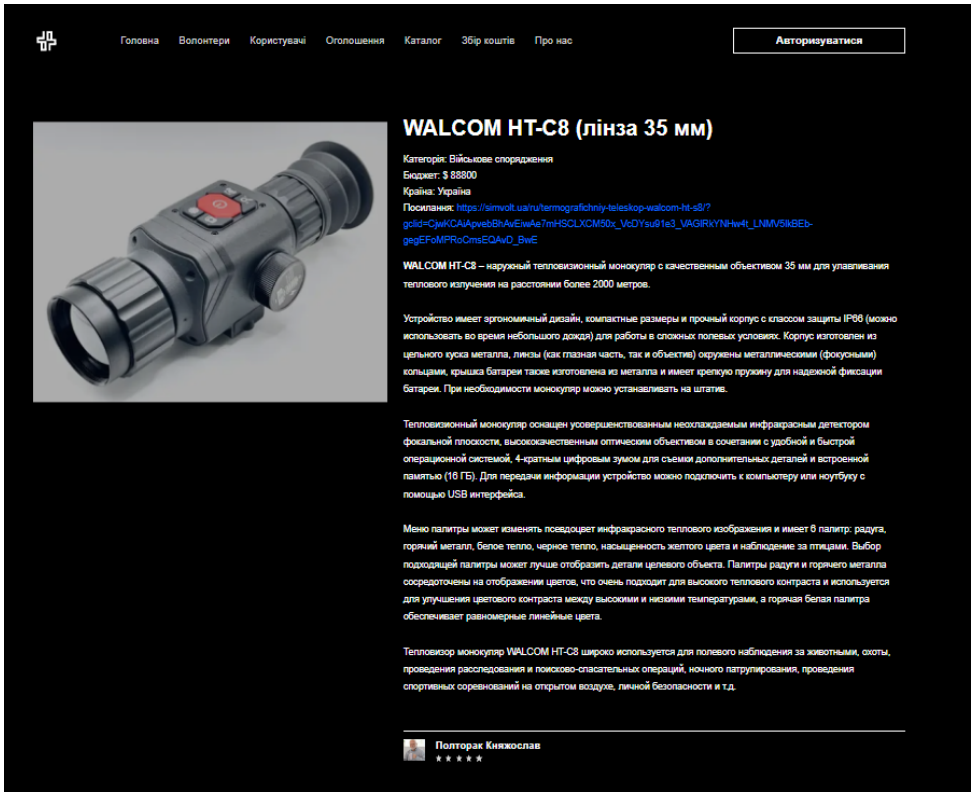


Рисунок 4.33 – Сторінка обраного оголошення

Наступною є сторінка «Каталог». На цій сторінкці будь-які користувачі інформаційної технології можуть переглянути інформацію. Речі із каталогу додані волонтерами, що саме вони можуть привезти та купити (рис.4.34).

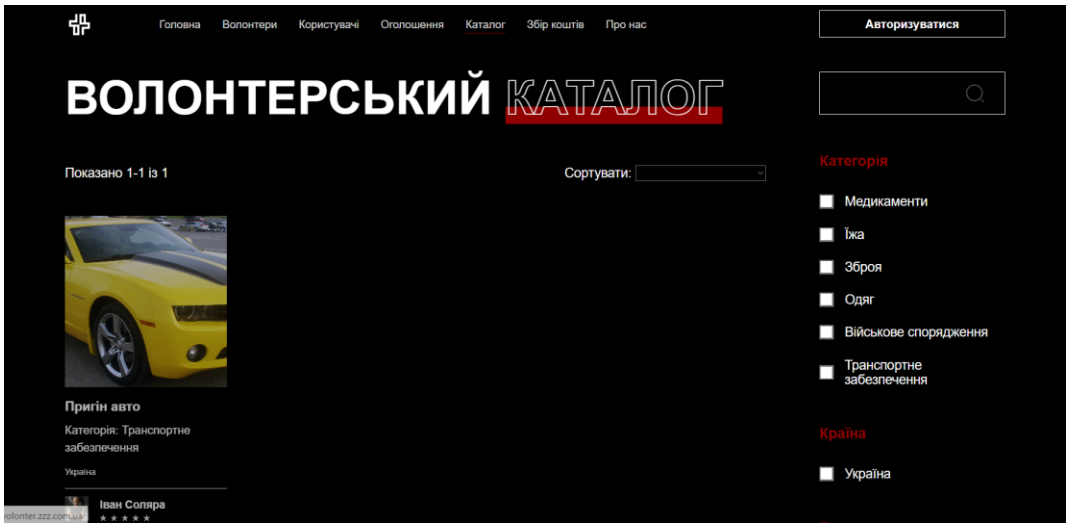


Рисунок 4.34 – Сторінка із речами, що можуть привезти волонтери

Перейшовши на сторінку обраного продукту із категорії, можна переглянути детальну інформацію (рис.4.35).

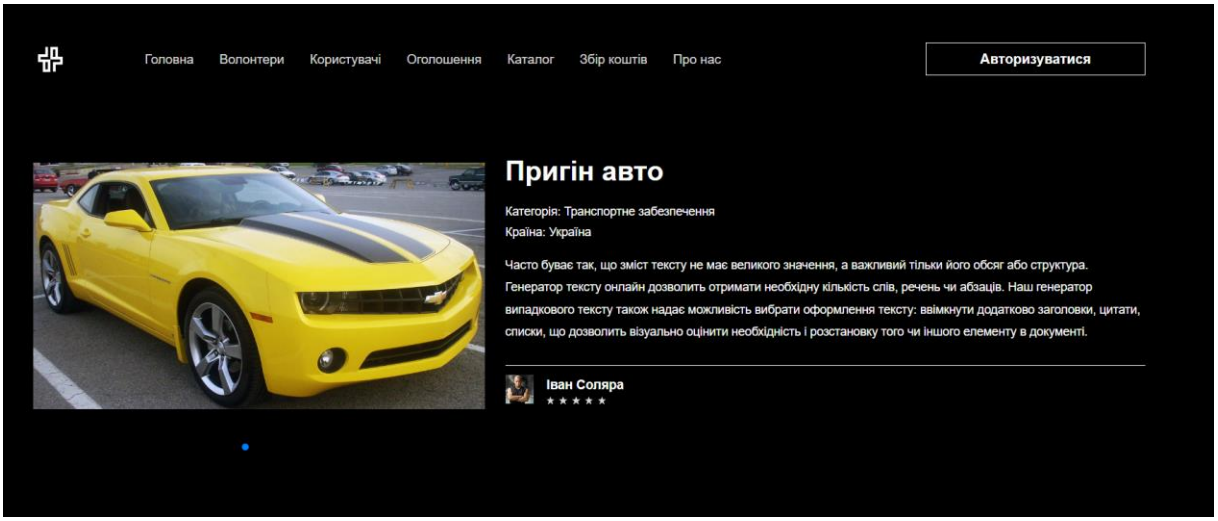


Рисунок 4.35 – Детальна інформація про продукт

Наступною сторінкою є «Збір коштів» (рис.4.36). Тут знаходиться інформація що до активних зборів створеними волонтерами інформаційної технології. Детальна інформація про обраний збір представлено на рисунку 4.37.

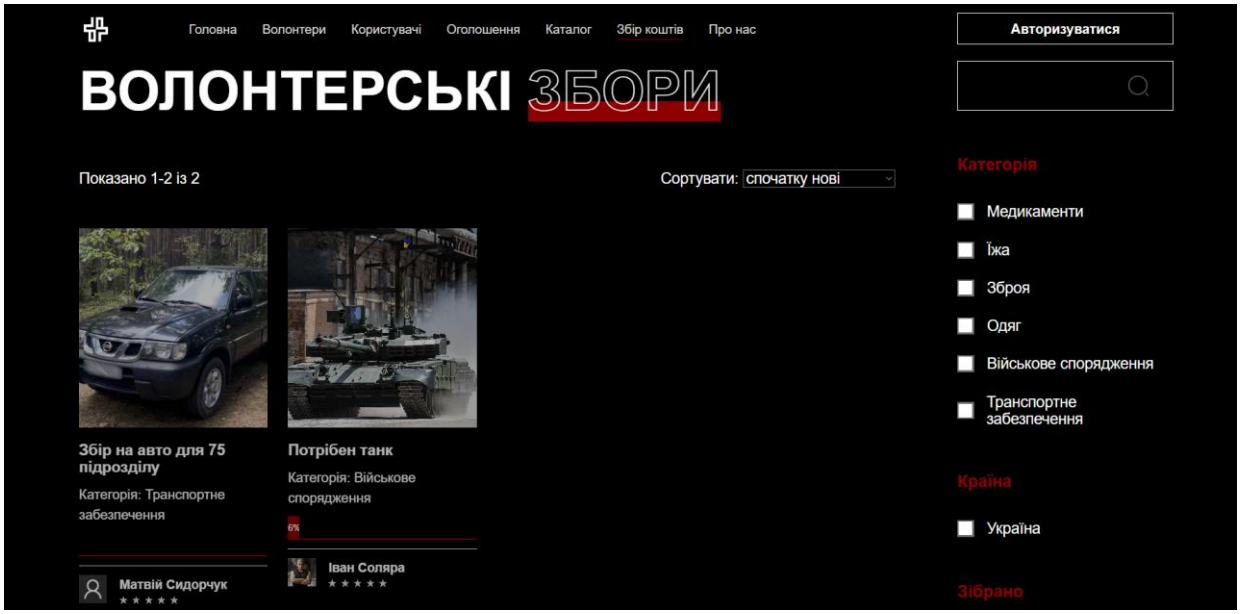


Рисунок 4.36 – Сторінка із зборами коштів

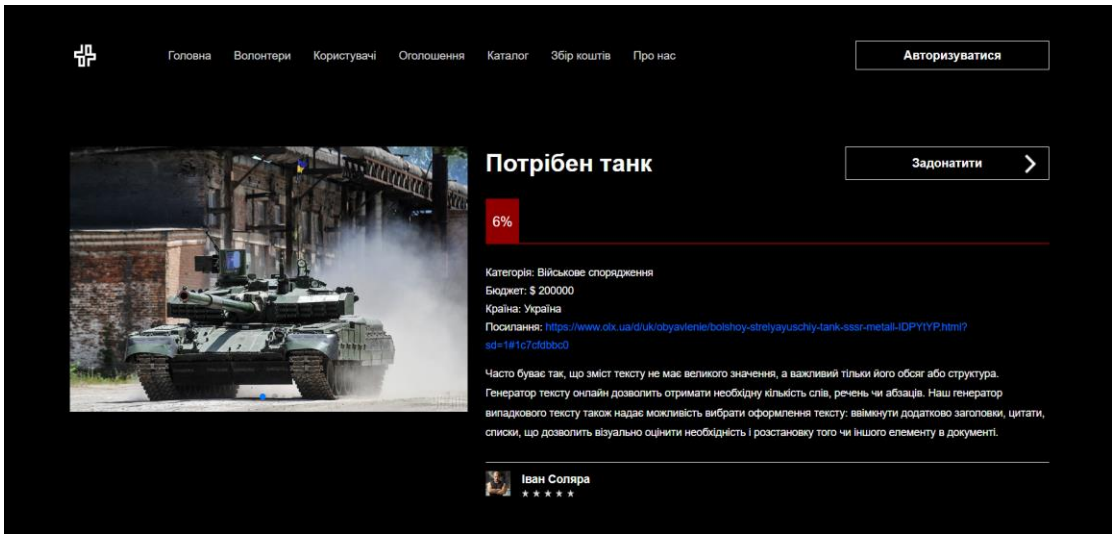


Рисунок 4.37 – Детальна інформація про збір коштів

На даній сторінці можна також внести оплату. Саме це представлено на рисунку 4.38-39. Задонатити може будь-який користувач, навіть не зареєстрований.

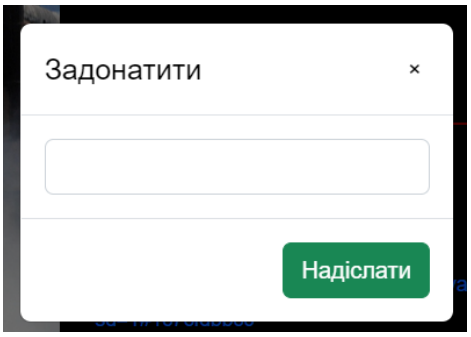


Рисунок 4.38 – Шаблон для донату

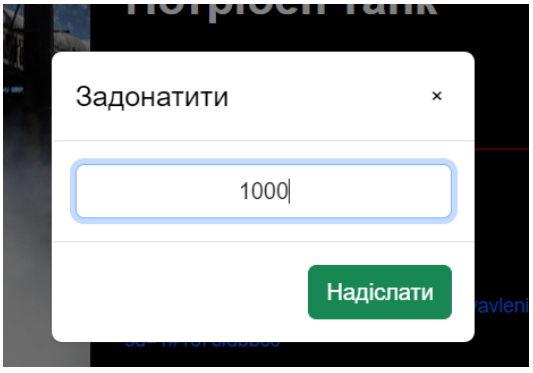


Рисунок 4.39 – Прилад виконання донату

Для реєстрації та авторизації користувача потрібно перейти на відповідні сторінки. Користувач повинен заповнити всю необхідну особисту інформацію (рис.4.40-42).

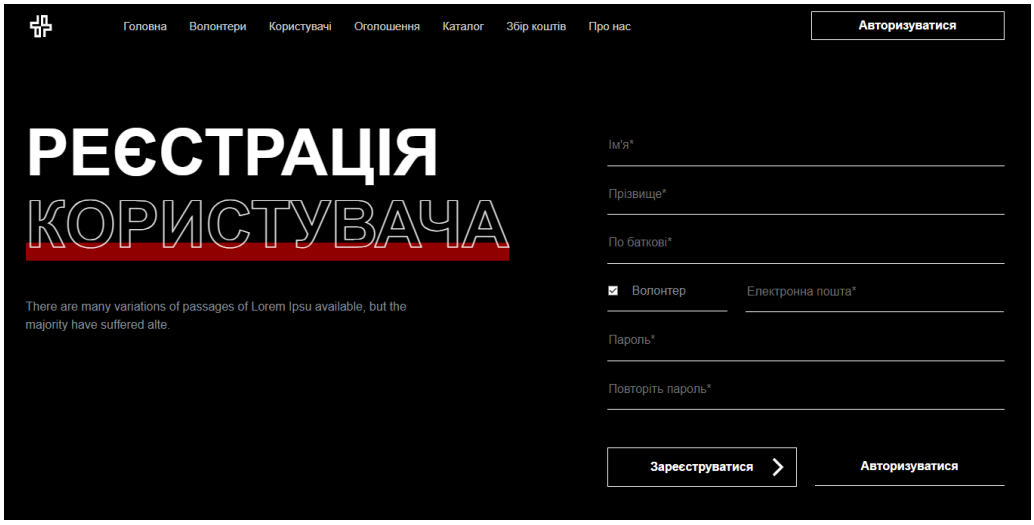


Рисунок 4.40 – Сторінка реєстрації

При введені невірної інформації чи не введені інформації в обов’язкові поля, користувач отримує повідомлення про помилку (рис.4.41).

Рисунок 4.41 – Приклад помилки

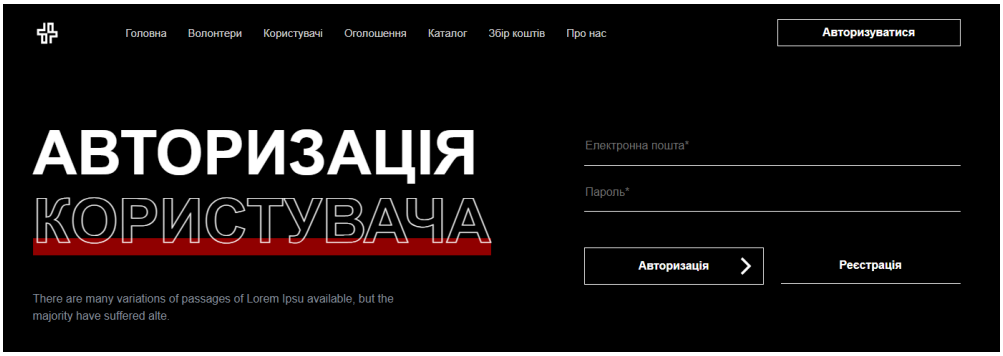


Рисунок 4.42 – Сторінка авторизації

Отже, після проведеної роботи можна зробити висновок, що інформаційна технологія задовольняє всі потреби та виконує всі потрібні функції з боку незареєстрованого користувача.

4.3 Використання інформаційної технології користувачами-волонтерами

Після авторизації, користувач-волонтер автоматично переходить на особисту сторінку та має можливість одразу розпочати роботу з акаунтом (рис.4.43).

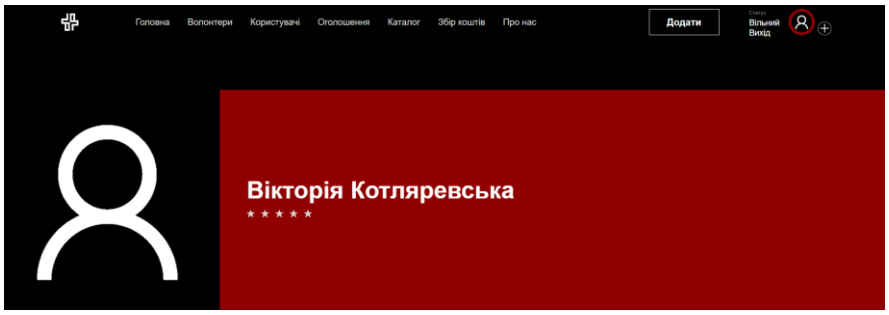


Рисунок 4.43 – Особиста сторінка користувача-волонтера

Користувач-волонтер може редагувати інформацію, яка представлена на особистій сторінці. При завантаженні фото-аватару акаунта, він автоматично центрується та обрізається за налаштуваннями при розробці (рис.4.44).

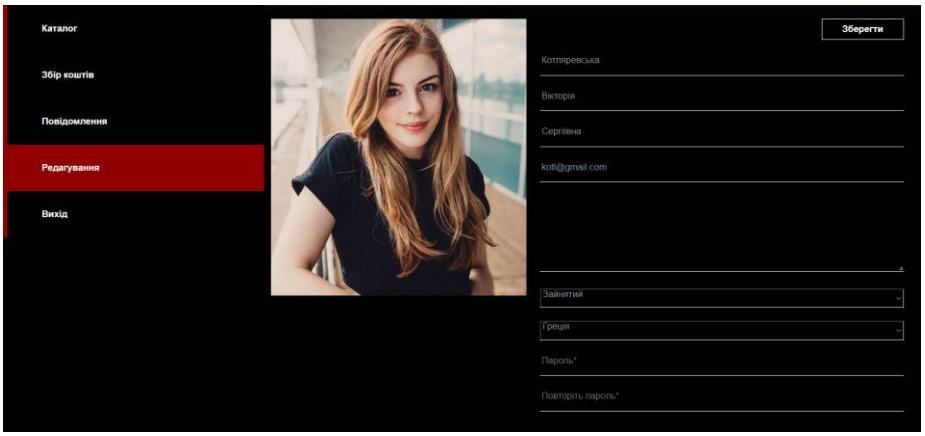


Рисунок 4.44 – Редагування особистої сторінки

На рисунку 4.45 показано зміни в відображенні фото після редагування особистої сторінки. На рисунку відображено фото, яке редагувалося при завантаженні, враховуючи, налаштування за замовчуванням.

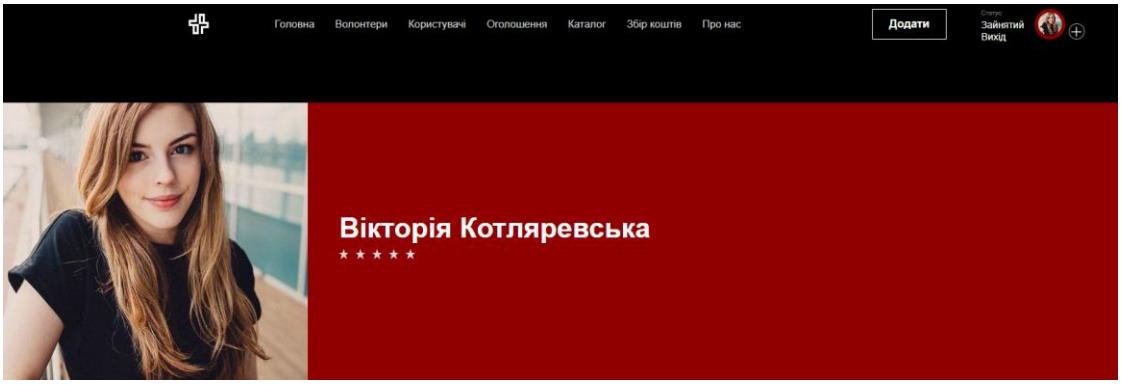


Рисунок 4.45 – Завантажене фото на особисту сторінку

Користувач-волонтер може додавати послуги або розпочинати збір коштів. Для цього потрібно натиснути «Додати» та вибрати бажану категорію для створення, як це показано на рисунку 4.46.

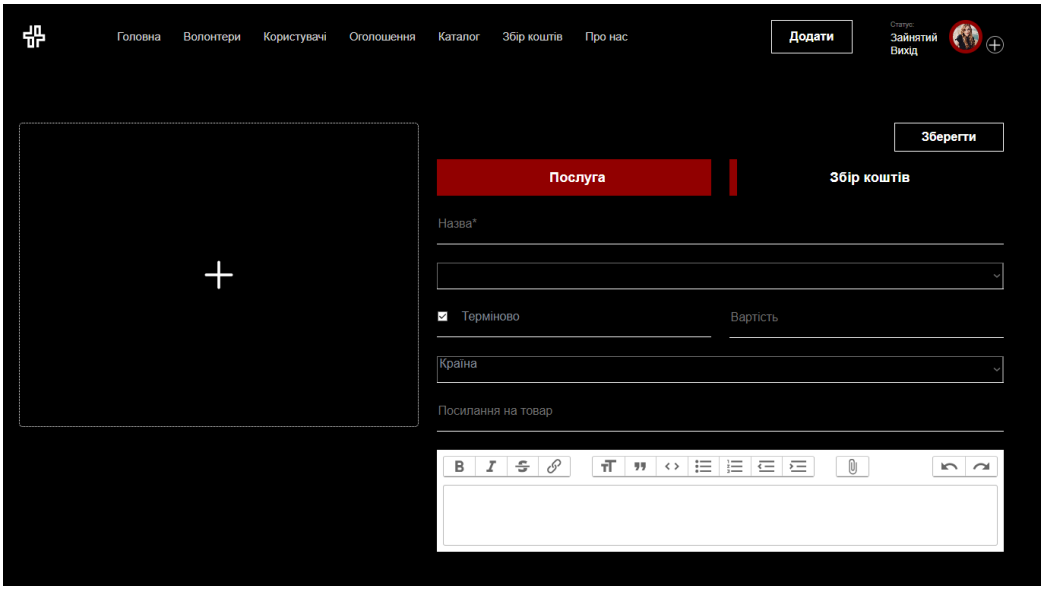


Рисунок 4.46 – Додавання послуги або збору коштів

Для успішного створення послуги потрібно заповнити всі текстові поля, вибрати категорію та країну. Обов’язковим для заповнення також є завантаження фото, для цього відведене спеціальне місце зліва. Терміновість послуги обирається також при створенні послуги (рис.4.47).

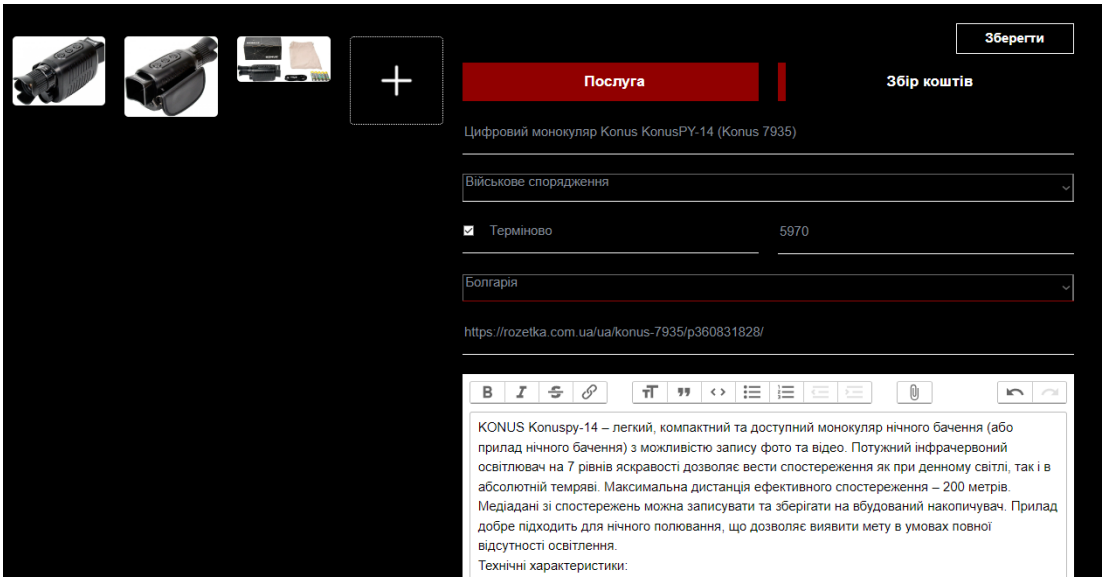


Рисунок 4.47 – Заповнений шаблон створення послуги

На рисунку 4.48 зображена послуга, яка одразу відображається в каталозі товарів особистої сторінки. У каталозі товарів особистої сторінки можна фільтрувати товари за категоріями, або вести пошук через текстове поле.

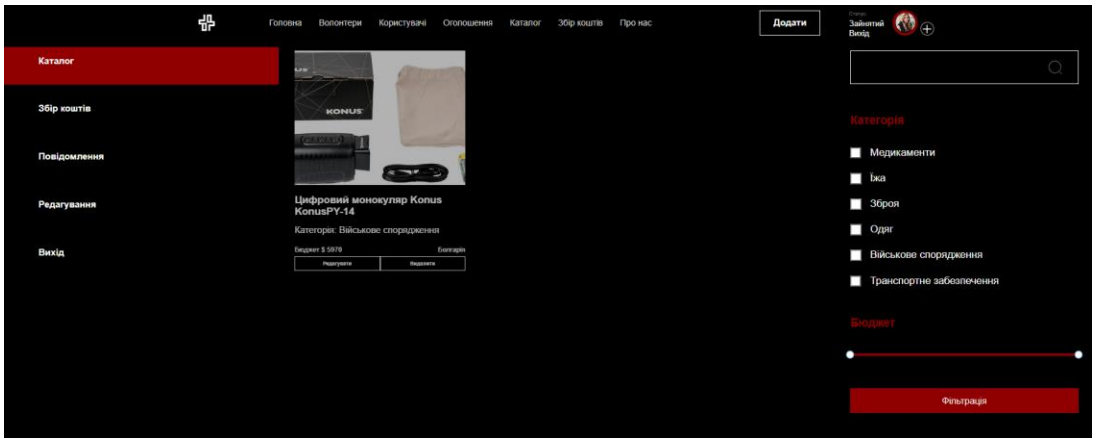


Рисунок 4.48 – Каталог товарів особистої сторінки

За бажанням можна оновити інформацію чи актуалізувати її. Для оновлення інформації доступні всі поля, що і при створенні послуги. Після внесення змін в опис послуги, система оповіщення проінформує користувача про збереження нової інформації, як це показано на рисунку 4.49.

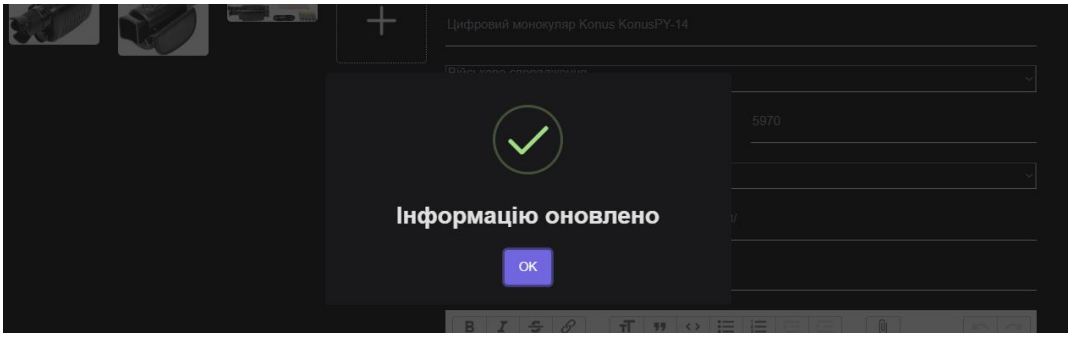


Рисунок 4.49 – Система сповіщення користувача про оновлення інформації

На сторінці «Оголошення» розміщений функціонал пошуку речей звичайними користувачами. Пошук речей показано у вигляді каталогу з вичерпною кількістю інформації. Для спрощення пошуку конкретних речей в оголошеннях присютнє сортування течей (рис.4.50).

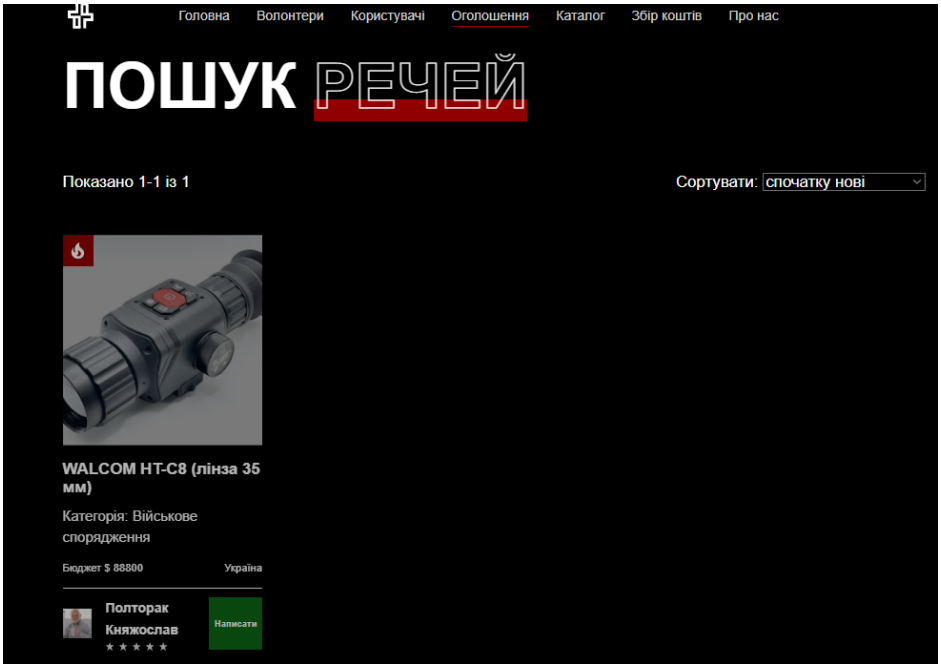


Рисунок 4.50 – Сторінка «Оголошення»

Користувач-волонтер через особисту сторінку має можливість вести листування з іншими користувачами. Система листування реалізована у вигляді діалогу з широким набором функцій редагування тексту (рис.4.51).

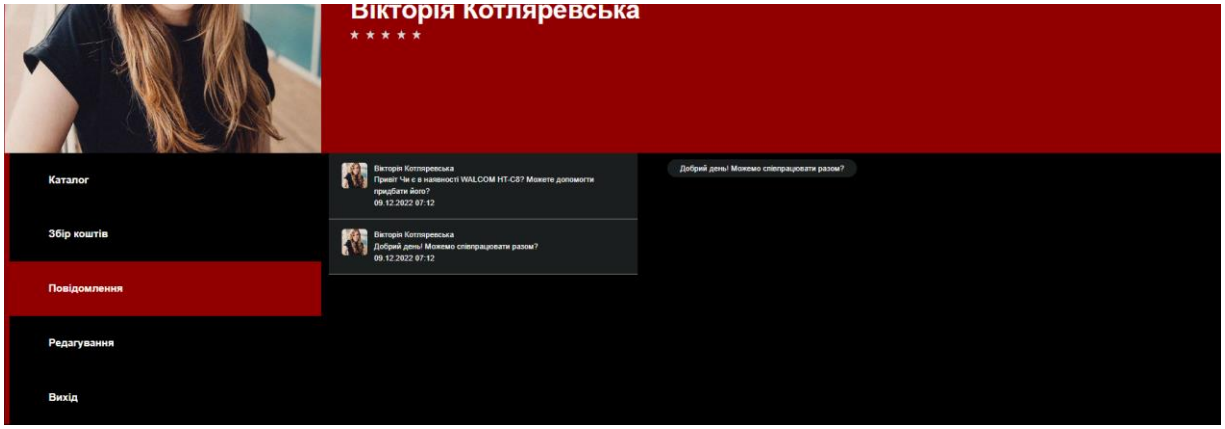


Рисунок 4.51 – Листування через особисту сторінку

Користувач може почати листування з іншим користувачем, перейшовши на особисту сторінку іншого користувача та натиснувши на «Написати». Також, перейшовши на сторінку іншого користувача, є можливість поскаржитися на нього, натиснувши «Поскаржитися», як це показано на рисунку 4.52).

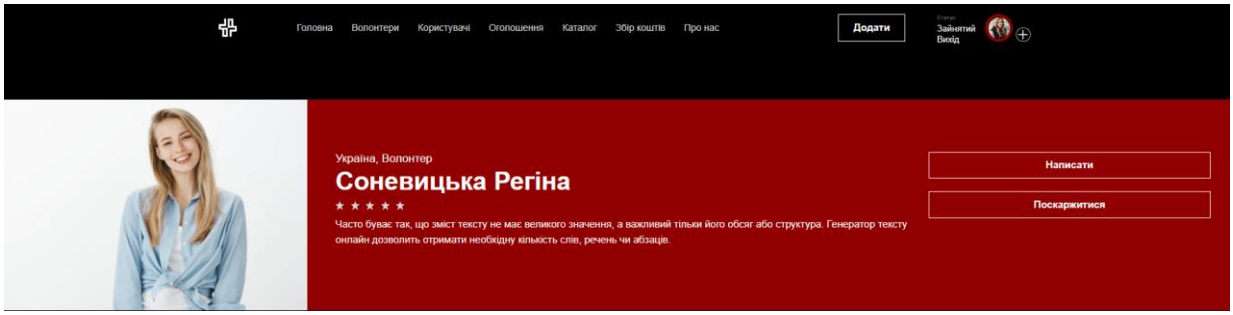


Рисунок 4.52 – Особиста сторінка іншого користувача

Скарга на іншого користувача створюється в текстовому виді довільної форми та надсилається до адміністраторів інформаційної технології. Приклад подання скарги на іншого користувача показано на рисунку 4.53.

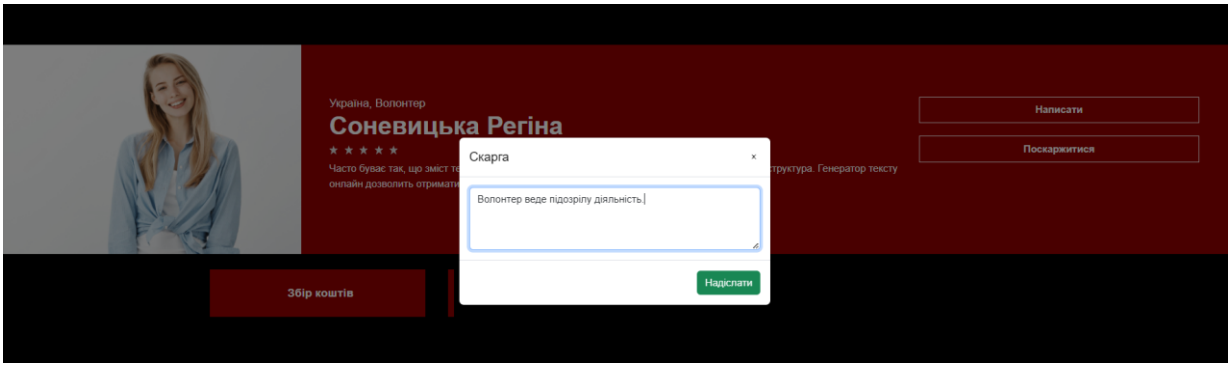


Рисунок 4.53 – Подання скарги на іншого користувача

На рисунку 4.54 показано збір коштів на товар із категорії «Транспортне забезпечення». Користувач волонтер може оглянути інформацію про збір коштів та задонатити або поскаржитися на цей збір. Скаргу можна подати окремо як на збір коштів, так і на користувача який створив цей збір.

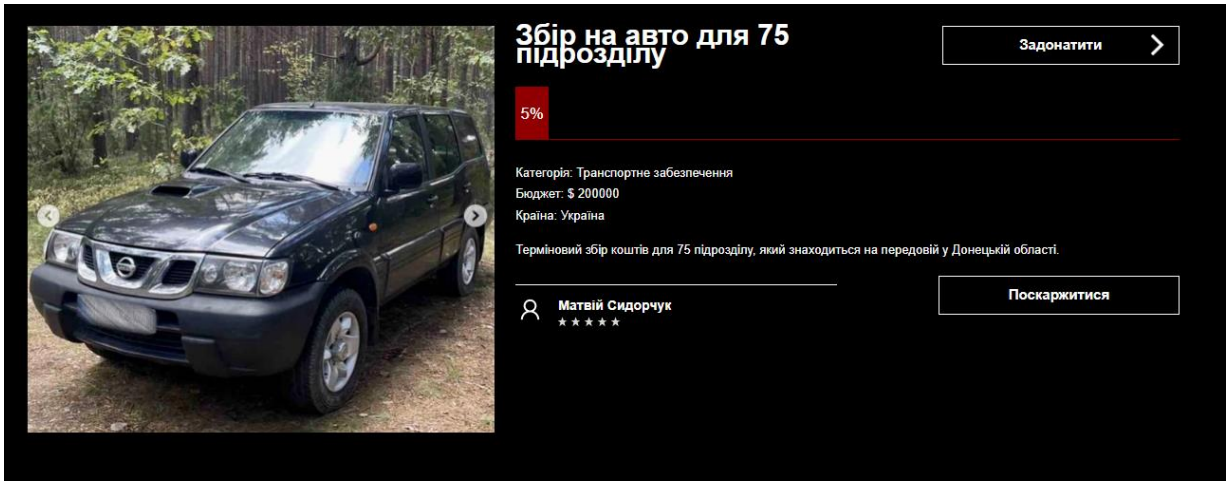


Рисунок 4.54 – Сторінка збору коштів на авто

Скарга подається текстом та одразу надсилається адміністраторам інформаційної технології на обробку, як це показано на рисунку 4.55.

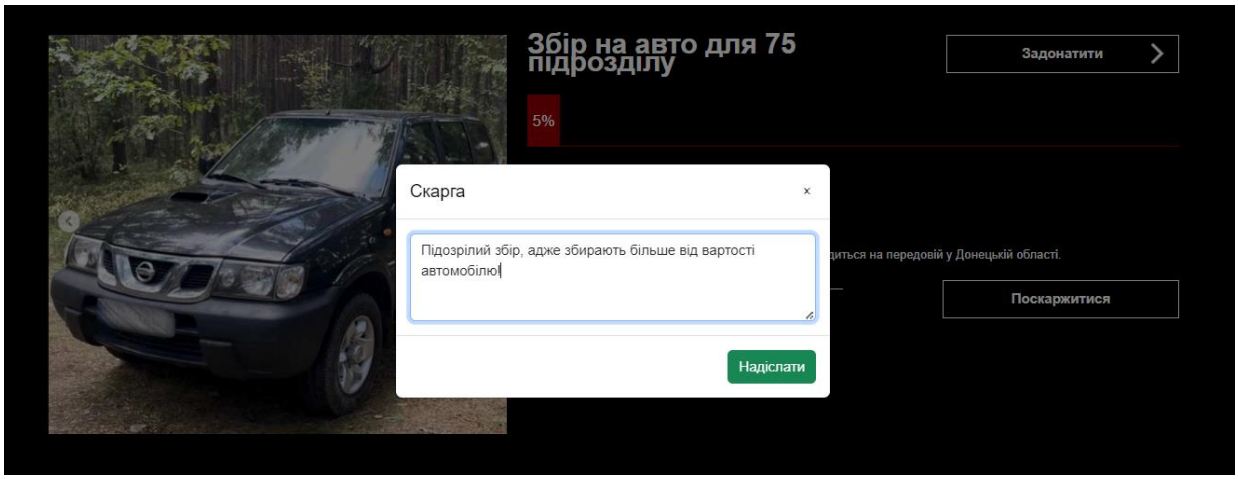


Рисунок 4.55 – Подання скарги на збір коштів

Отже, після проведеної роботи можна зробити висновок, що інформаційна технологія задовольняє всі потреби та виконує всі потрібні функції з боку користувача-волонтера.

4.4 Використання інформаційної технології звичайними користувачами

Окрім створення сторінки із звичайного користувача із сторінки реєстрації присутня можливість створити новий акаунт на основі вже створеного. Наприклад, волонтер може створити додаткову сторінку звичайного користувача. Для цього потрібно натиснути на плюс біля аватару користувача (рис.4.56).

Даний функціонал необхідний для вірного розділення функціоналу між користувачами.

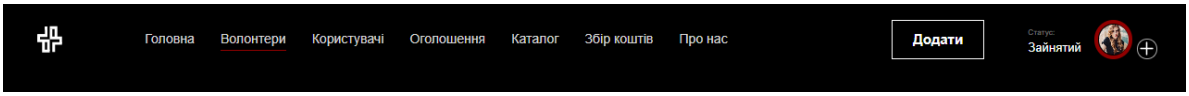


Рисунок 4.56 – Хедер авторизованого користувача-волонтера

Після натискання користувач перенаправляється на сторінку для створення нового акаунту. Частина інформація із створеного вже акаунту переноситься до кожного із полів відповідно (рис.4.57).

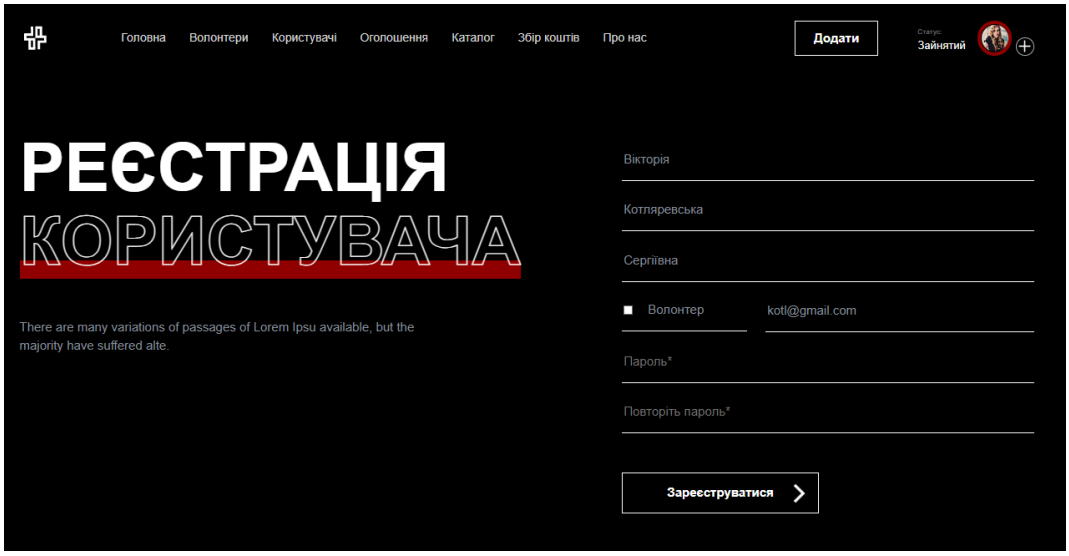


Рисунок 4.57 – Сторінка для створення нового акаунту

Також після створення нового акаунту змінюється хедер, але користувач у будь-який час може переключитися на іншу сторінку (рис.4.58).

Після створення нового акаунту, у представленому випадку сторінка звичайного користувача, змінюється відображення всієї інформаційної технології (рис.4.59). Основний колір, червоний, змінюється на зелений. Саме цей колір відповідає сторінці звичайного користувача інформаційної технології.

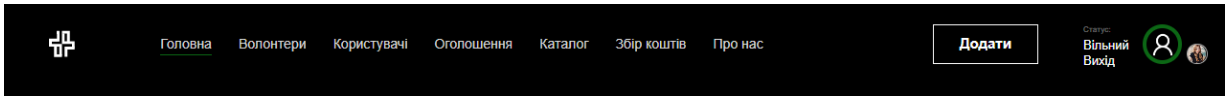


Рисунок 4.58 – Хедер з нового створеного акаунту

Рисунок 4.59 – Приклад зміни кольору

Перейдемо на сторінку редагування основної інформації про користувача (рис.4.60). Як ми бачимо, більшість інформацію збігається із сторінкою про волонтерів.

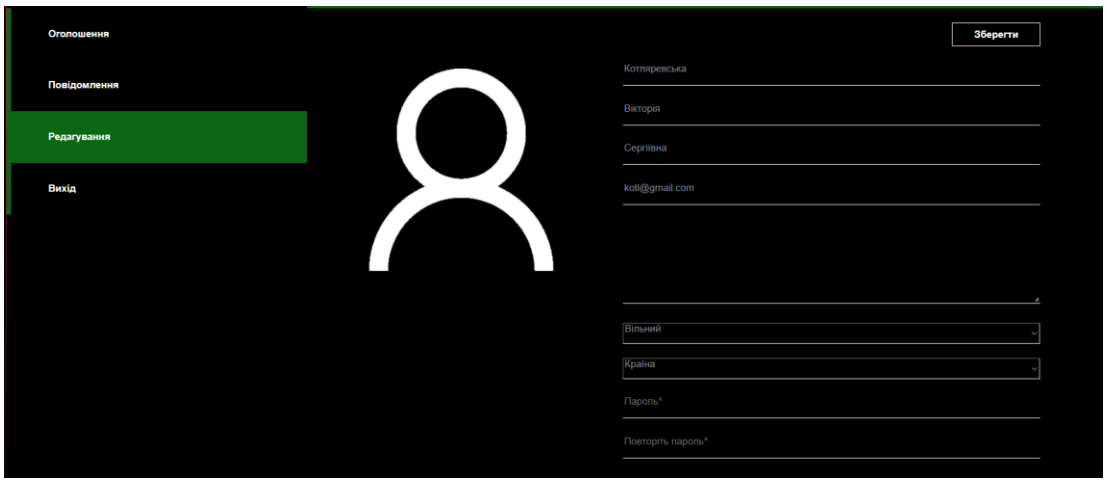


Рисунок 4.60 – сторінка редагування основної інформації

Головна відмінність звичайного акаунту – можливість створювати оголошення. Приклад заповнення шаблону для створення оголошення представлено на рисунку 4.61.

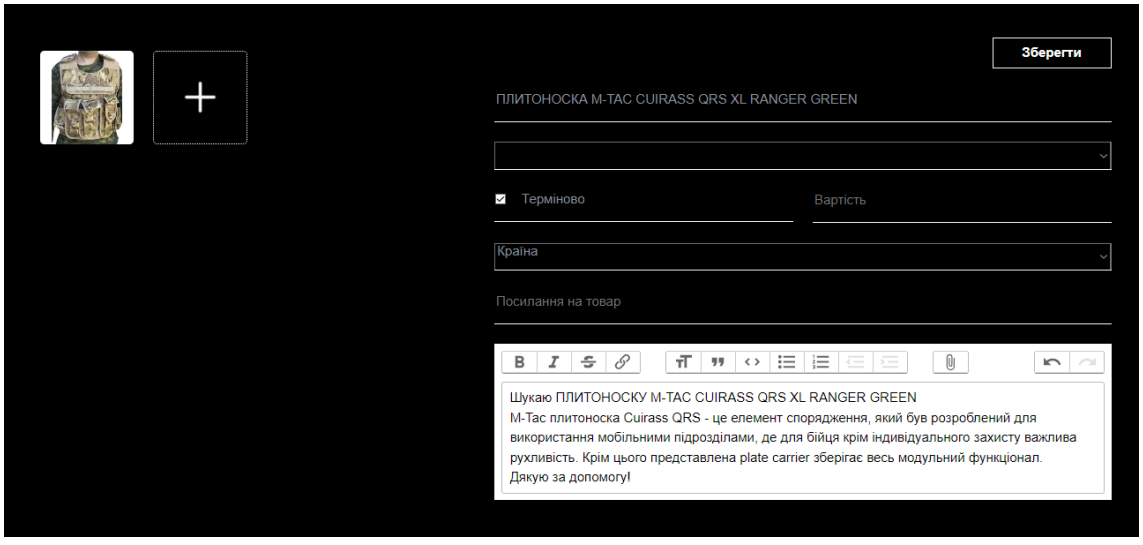


Рисунок 4.61 – Приклад створення оголошення

При введені некоректної інформації користувач отримує повідомлення, що до помилки при створенні (рис.4.62).

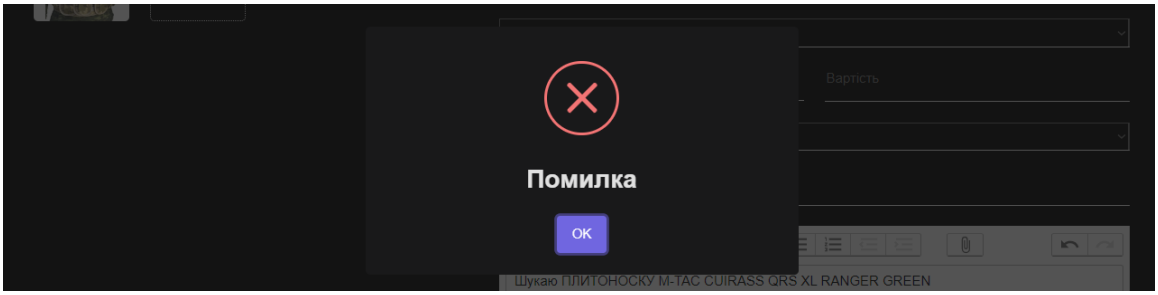


Рисунок 4.62 – Помилка при створенні

У результаті було створеного нове оголошення, що представлено на рисунку 4.63. Також переглянути створене оголошення можна на особистій сторінці користувача інформаційної технології (рис.4.63).

Рисунок 4.63 – Створене оголошення

Рисунок 4.64 – Відображення оголошення на сторінці користувача

Отже, після проведеної роботи можна зробити висновок, що інформаційна технологія задовольняє всі потреби та виконує всі потрібні функції з боку звичайного користувача.

4.5 Адміністрування інформаційної системи

Адміністрування інформаційної технології по волонтерській допомозі проходить на спеціально відведеній сторінці. На головній сторінці відображено статистичні данні інформаційної технології, модуль створення редагування та видалення посилань на соціальні мережі та категорій волонтерських напрямків (рис. 4.65).

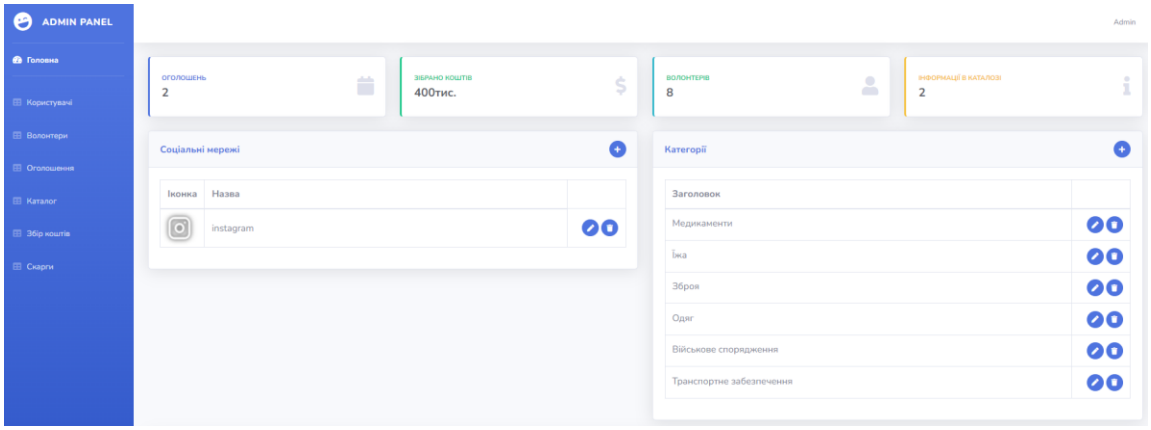


Рисунок 4.65 – Головна сторінка панелі адміністратора

На сторінці з волонтерами адміністратор може переглядати інформацію про волонтерів (аватар, ПІП та Email) та блокувати або розблоковувати їх (рис. 4.66).

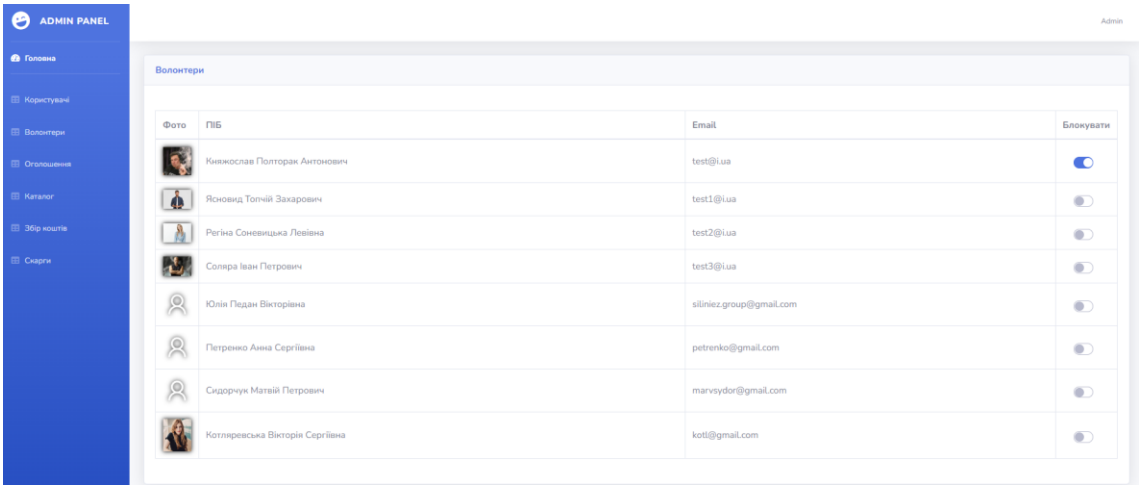


Рисунок 4.66 – Панель волонтерів

За допомогою панелі оголошень, адміністратор має можливість переглядати інформацію про набори та видаляти ті, які належать заблокованим авторам. Особливість в тому, якщо блокується користувач з різних причинам, то і блокуються всі збори коштів від цього користувача, а його оголошення не відображаються (рис. 4.67).

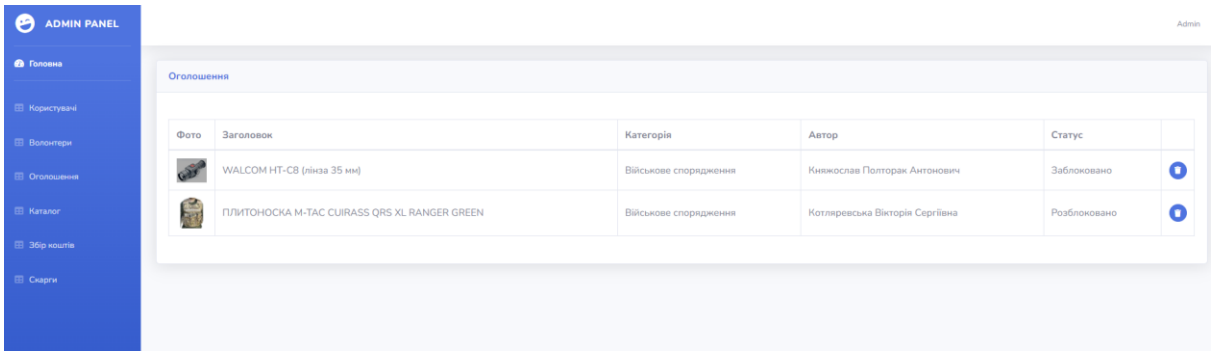


Рисунок 4.67 – Панель оголошень

На сторінці «Каталог» адміністратору надається інформація про активні лоти продажу, статус користувача, що створив цей збір, та можливість видаляти лоти з каталогу (рис. 4.68).

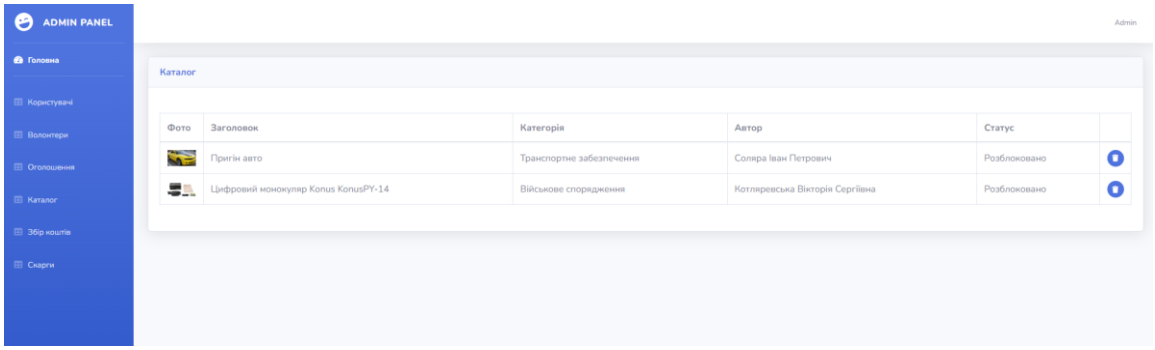


Рисунок 4.68 – Сторінка «Каталог»

«Збір коштів» надає адміністратору інформацію про активні збори коштів, статус користувача, який створив цей збір, та можливість видаляти збори коштів з каталогу (рис. 4.69).

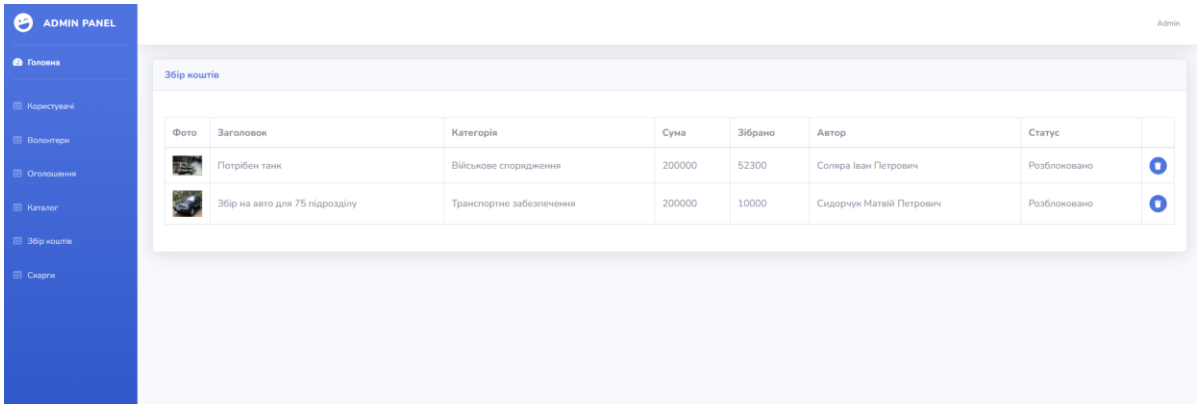


Рисунок 4.69 – Сторінка «Збір коштів»

На сторінці «Скарги» є можливість переглядати текст скарги, на яку послугу чи збір коштів створенно скаргу та дату створення скарги, як це показано на рисунку 4.70.

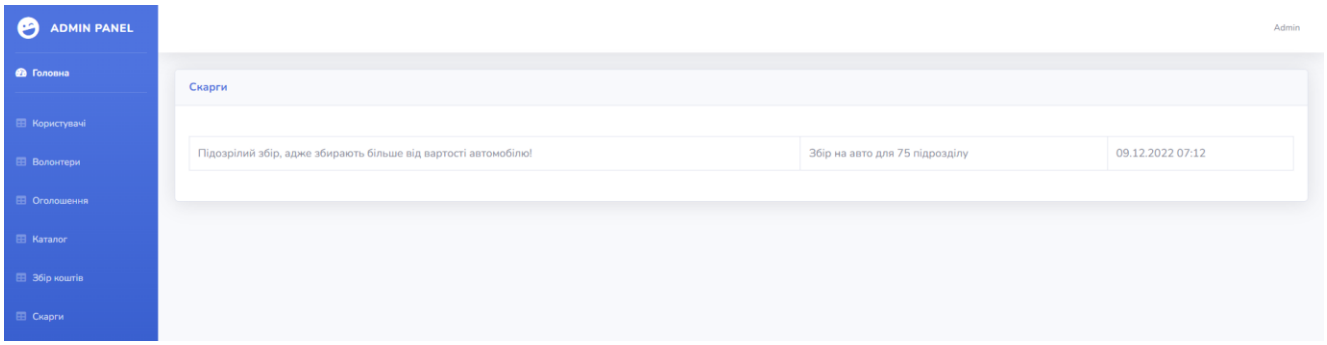


Рисунок 4.70 – Сторінка «Скарги»

Отже, адміністративна панель виконує поставлені задачі з адмініструванням інформаційної технології та дає адміністратору всі потрібні можливості для виконання поставленої мети.

ВИСНОВКИ

При виконанні кваліфікаційної роботи магістра, було створено інформаційну технологію по волонтерській допомозі.

Робота була розподілена на етапи зі своїми задачами та підзадачами. Детально проаналізувавши предметну область, було оглянуто останні дослідження та проведено порівняння з схожими інформаційними технологіями.

Після постановки задачі та визначення мети інформаційної технології, вирішено використовувати Vue.js для розробки візуальної частини інформаційної технології та Laravel для розробки функціональної частини.

Проведена робота над створенням супутньої документації зі створенням відповідних схем та таблиць. Розроблене проектування бази даних та представлено її у вигляді ER-діаграми та створено описання для неї у вигляді таблиці. При створенні супутньої документації, враховувалися функціональні можливості інформаційної технології та потреби які інформаційна технологія повинна задовольняти.

Після розробки інформаційної технології, система була наповнена контентом та проведено демонстрацію роботи інформаційної технології.

У результаті отримано інформаційну технологію по волонтерській допомозі, які задовольняла потреби користувачів та виконувала всі поставлені задачі.

СПИСОК ЛІТЕРАТУРИ

1. What is Project Analysis and Why it is Important? [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: https://www.smarttask.io/blog/project-analysis(дата звернення: 10.09.2022);

2. K. Dempsey-Brench, A. Shantz, Skills-based volunteering: A systematic literature review of the intersection of skills and employee volunteering. Human Resource Management Review. 32 (2022), doi:10.1016/j.hrmr.2021.100874.Тут буде інфа про проекти (статі для актуальності)

3. 1. M. Baillie Smith, B. Fadel, A. O'Loghlen, S. Hazeldine, Volunteering Hierarchies in the Global South: Remuneration and Livelihoods. Voluntas. 33, 93–106 (2022).Тут буде інфа про волонтерві (статі для актуальності)

4. 1. M. Meyer, P. Rameder, Who Is in Charge? Social Inequality in Different Fields of Volunteering. Voluntas. 33, 18–32 (2022).

5. 1. T. Walker et al., Determinants of Volunteering Within a Social Housing Community. Voluntas. 33, 188–200 (2022).Spivdiia [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: https://spivdiia.org.ua (дата звернення: 12.09.2022);

6. Spivdiia [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: https://spivdiia.org.ua (дата звернення: 11.09.2022);

7. Uahelpers [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: https://uahelpers.com (дата звернення: 12.09.2022);

8. Volonter.org [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: https://volonter.org (дата звернення: 15.09.2022);

9. 10 Reasons Why Vue.js [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: https://www.sam-solutions.com/blog/why-vue-js/ (дата звернення: 18.09.2022);

10. The Good and the Bad of Vue.js Framework Programming [Електронний ресурс]. – 2022. – Режим доступу до ресурсу:

https://www.altexsoft.com/blog/th/engineering/pros-and-cons-of-vue-js/ (дата звернення: 10.12.2022);

11. Comparison with Other Frameworks [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: https://v2.vuejs.org/v2/guide/the/comparison.html (дата звернення: 12.12.2022);

12. Vue.js for Web Designers [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: https://www.linkedin.com/learning/vue-js -web-designers (дата звернення: 12.09.2022);

13. Is Vue a Good Fit for My Project? - 7 Questions to Ask Before Choosing Vue [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: https://www.netguru.com/blog/th/is-vue-good-fit-for-my-project (дата звернення: 12.11.2022);

14. Jon Duckett: PHP & MySQL: The Server-side Web Development / Wiley, 2022. – 672 с.

15. Peter MacIntyre, Sávio Resende: Development in Depth/php architect, 2020. – 228 с.

16. Doug Bierer: PHP 7 the Programming Cookbook / Packt Publishing, 2028. – 610 с.

17. Nico Anastasio: The PHP the Good Practices handbook / Nico Anastasio, 2022. – 73 с.

18. Joel Murach, Ray Harris: Murach's PHP / Mike Murach, 2029. – 866 с.

19. Doug Bierer, Cal Evans: PHP 8 Programming Tips, Tricks and Best Practices: A practical guide to PHP 8 features, usage changes, and advanced programming techniques / Packt Publishing, 2021. – 528 с.

20. Jeffrey E. F. Friedl: Mastering The Regular Expressions / O'Reilly Media, 2019. – 544 с.

21. Walter Shields: SQL QuickStart Guide: The Simplified Beginner's Guide to Managing, Analyzing, and Manipulating Data With SQL Paperback / ClydeBank Media LLC, 2019. – 249 с.

22. Brad Williams, Justin Tadlock, John James Jacoby: Professional WordPress Plugin Development / Wrox, 2020. – 480 c.

23. Guy Harrison, Steven Feuerstein: MySQL Stored Procedure Programming: Building High-Performance Web Applications in MySQL / O'Reilly Media, 2020. – 640 c.

ДОДАТОК А. КОД РЕАЛІЗАЗІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

UserController.php

Контролер користувачів.

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\DB;

use App\Models\User;
use App\Models\UserStatus;
use App\Models\Country;

class UserController extends AuthController
{
    protected $publicStorage = "/uploads/ads-files/";
    protected $userStorage = "/uploads/user-files/";

    public function getCurrentUser($id = null)
    {
        $user = User::find($id);
        if ($id == null) {
            $user = auth()->user();
        }
        if ($user) {
            return response()->json([
                'user' => $this->authUser($user),
                'token' => $user->createToken("API TOKEN")->plainTextToken
            ]);
        } else {
            return null;
        }
    }

    public function get(Request $request)
    {
        $model = User::with('role', 'ads')->where('volunteer_user_role_id', '!=', 3);
        if ($request->roles) {
            $model = $model->whereIn('volunteer_user_role_id', $request->roles);
        }

        if ($request->name) {
            $model = $model->where(function ($query) use ($request) {
                $query->where('volunteer_user_name', 'LIKE', '%' . $request->name . '%')
                    ->orWhere('volunteer_user_surname', 'LIKE', '%' . $request->name . '%')
                    ->orWhere('volunteer_user_patronymic', 'LIKE', '%' . $request->name . '%');
            });
        }

        if ($request->status_id) {
            $model = $model->where('volunteer_user_status_id', $request->status_id);
        }
```

```
    if (isset($request->countries) && count($request->countries) > 0) {
        $model = $model->whereIn('volunteer_country_id', $request->countries);
    }

    if (isset($request->categories) && count($request->categories) > 0) {
        $model = $model->whereHas('ads', function ($query) use ($request) {
            $query->whereIn('volunteer_ads_role_id', $request->categories);
        });
    }

    if ($request->is_ban == 'false') {
        $model = $model->where('is_ban', 0);
    }

    if (isset($request->sort) && $request->sort != 'rate') {
        $model = $model->orderBy($request->sort, 'desc');
    }

    $data = $model->get();

    if ($request->rate) {
        $data = $data->filter(function ($model) use ($request) {
            return $model->rate >= $request->rate[0] && $model->rate <= $request->rate[1];
        });
    }

    $data = $data->toArray();

    if (isset($request->sort) && $request->sort == 'rate') {
        usort($data, function($a, $b) {return $a['rate'] < $b['rate'];});
    }

    return response()->json(collect($data));
}

public function getUserStatus()
{
    $model = UserStatus::get();
    return response()->json($model);
}

public function getId(Request $request, $id)
{
    $data = User::with(
        'ads.preview',
        'ads.category',
        'role',
        'country',
        'activeFundraising.category',
        'activeFundraising.preview'
    )
        ->with('ads', function ($query) {
            $query->withCount(['donate' => function ($query) {
                $query->select(DB::raw('SUM(sum)'));
            }]);
        })->find($id);
    return response()->json($data);
}

public function profile()
```

```
    {
        $data = User::with(
            'role',
            'country',
            'activeFundraising.category',
            'activeFundraising.preview',
            'ads'
        )->find(Auth::user()->volunteer_user_id);
        return response()->json($data);
    }

    function profileUpdate(Request $request)
    {
        $model = User::find(Auth::id());
        $data = $request->all();
        if (isset($request['new_photo']) && $request['new_photo'] != "null" && $request['new_photo'] != "undefined") {
            $name = $this->userStorage . uniqid() . '.' . $request['new_photo']->getClientOriginalExtension();
            $request['new_photo']->move(public_path() . $this->userStorage, $name);
            $data['volunteer_user_photo'] = $name;
        }
        $model->update($data);

        return response('ok', 200);
    }

    function banUser(Request $request, $id)
    {
        $model = User::find($id);
        $model->update([
            'is_ban' => $request->is_ban == true ? 1 : 0
        ]);
    }

    function getCountry(Request $request)
    {
        $country = User::select('volunteer_country_id')
            ->where('volunteer_user_role_id', $request->volunteer_user_role_id)
            ->where('is_ban', 0)
            ->distinct('volunteer_country_id')
            ->pluck('volunteer_country_id');
        $data = Country::whereIn('volunteer_country_id', $country)->get();
        return response()->json($data);
    }
}
```

AdsController.php

Контролер оголошень.

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\DB;

use App\Models\Ads;
use App\Models\Fotos;
use App\Models\AdsRole;
use App\Models\Country;
```

```
class AdsController extends Controller
{
    protected $publicStorage = "/uploads/ads-files/";

    public function post(Request $request)
    {
        $ads = new Ads();
        $data = $request->all();
        $data['volunteer_user_id'] = Auth::id();
        $data['volunteer_ads_urgency'] = $data['volunteer_ads_urgency'] ? 1 : 0;
        if($data['freeze']) {
            $data['freeze'] = $data['freeze'] == 'true' ? 1 : 0;
        }
        $response = $ads->create($data);
        $uploadedFiles = $request->adsPhotos;
        foreach ($uploadedFiles as $file) {
            $foto = new Fotos();
            $image = substr($file, strpos($file, ',') + 1);
            $imageName = $this->publicStorage . uniqid() . '.' . explode('/', mime_content_type($file))[1];
            \File::put(public_path() . $imageName, base64_decode($image));
            $foto->create([
                "volunteer_ads_id" => $response->volunteer_ads_id,
                "volunteer_fotos_file" => $imageName
            ]);
        }
        return response()->json([
            'volunteer_ads_id' => $response->volunteer_ads_id
        ]);
    }

    public function update(Request $request, $id)
    {
        $ads = Ads::find($id);
        $data = $request->all();
        if(isset($data['volunteer_ads_urgency'])) {
            $data['volunteer_ads_urgency'] = $data['volunteer_ads_urgency'] ? 1 : 0;
        }
        if($data['freeze']) {
            $data['freeze'] = $data['freeze'] == 'true' ? 1 : 0;
        }

        $ads->update($data);
        if ($request->adsPhotos) {
            Fotos::where('volunteer_ads_id', $id)->delete();
            foreach ($request->adsPhotos as $file) {
                $foto = new Fotos();
                if (strripos($file, 'base64')) {
                    $image = substr($file, strpos($file, ',') + 1);
                    $imageName = $this->publicStorage . uniqid() . '.' . explode('/', mime_content_type($file))[1];
                    \File::put(public_path() . $imageName, base64_decode($image));
                } else {
                    $imageName = $file;
                }
                $foto->create([
                    "volunteer_ads_id" => $id,
                    "volunteer_fotos_file" => $imageName
                ]);
            }
        }
```

```
    return response('ok', 200);
}

public function get(Request $request)
{
    $model = Ads::with(
        'photos',
        'preview',
        'category',
        'country',
        'user'
    )->withCount([
        'donate' => function ($query) {
            $query->select(DB::raw('SUM(sum)'));
        }
    ])->where('type', $request->type);


    if ($request->name) {
        $model = $model->where('volunteer_ads_name', 'LIKE', '%' . $request->name . '%');
    }

    if (isset($request->categories) && count($request->categories) > 0) {
        $model = $model->whereIn('volunteer_ads_role_id', $request->categories);
    }

    if ($request->user_id) {
        $model = $model->where('volunteer_user_id', $request->user_id);
    }

    if ($request->term) {
        $model = $model->where('volunteer_ads_urgency', $request->term);
    }

    if($request->rate) {
        $model = $model->where(function($query) use ($request) {
            $query->where('volunteer_cost', '>=', intval($request->rate[0]))
            ->where('volunteer_cost', '<=', intval($request->rate[1]))
            ->orWhereNull('volunteer_cost');
        });
    }

    if(isset($request->sort)) {
        $sort = explode(':', $request->sort);
        $model = $model->orderBy($sort[0], $sort[1]);
    }

    $data = $model->get();

    if ($request->ready) {
        $data = $data->filter(function($model) {
            return $model->donate_count >= $model->volunteer_cost;
        });
    }

    return response()->json($data);
}

public function getId($id)
{
```

```
        $data = Ads::with('user', 'photos', 'category', 'country', 'user')->withCount([
            'donate' => function ($query) {
                $query->select(DB::raw('SUM(sum)'));
            }
        ])->find($id);
        return response()->json($data);
    }

    function delete($id)
    {
        Ads::find($id)->delete();
        return response('ok', 200);
    }

    function getCategory()
    {
        $data = AdsRole::get();
        return response()->json($data);
    }

    function getCountry(Request $request)
    {
        $country = Ads::select('volunteer_country_id')
            ->where('type', $request->type)
            ->distinct('volunteer_country_id')
            ->pluck('volunteer_country_id');
        $data = Country::whereIn('volunteer_country_id', $country)->get();
        return response()->json($data);
    }
}
```

AuthController.php

Контролер авторизації. В даному контроллері відбувається реєстрація та авторизація користувачів, а також перемикання між двома аккаунтами одного користувача.

```
<?php
namespace App\Http\Controllers;

use App\Models\User;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Hash;

use Illuminate\Support\Facades\Validator;

class AuthController extends Controller
{
    public function authUser($user)
    {
        if ($user->volunteer_user_role_id == 1) {
            $anotherRole = User::select('volunteer_user_id', 'volunteer_user_photo')->where([
                ['volunteer_user_role_id', '2'],
                ['email', $user->email]
            ])->first();
        } else if ($user->volunteer_user_role_id == 2) {
            $anotherRole = User::select('volunteer_user_id', 'volunteer_user_photo')->where([
```

```
            ['volunteer_user_role_id', '1'],
            ['email', $user->email]
        ])->first();
    } else {
        $anotherRole = null;
    }
    return [
        'authUser' => $user,
        'anotherRole' => $anotherRole
    ];
}

public function login(Request $request)
{
    try {
        $validateUser = Validator::make($request->all(),
        [
            'email' => 'required|email',
            'password' => 'required'
        ]);

        if($validateUser->fails()) {
            return response()->json([
                'status' => false,
                'message' => 'validation error',
                'errors' => $validateUser->errors()
            ], 401);
        }

        if(!Auth::attempt(['email' => $request->email, 'password' => $request->password])) {
            return response()->json([
                'status' => false,
                'message' => 'Email & Password does not match with our record.',
            ], 401);
        }

        $user = User::where('email', $request->email)->first();

        return response()->json([
            'status' => true,
            'message' => 'User Logged In Successfully',
            'user' => $this->authUser($user),
            'token' => $user->createToken("API TOKEN")->plainTextToken
        ], 200);

    } catch (\Throwable $th) {
        return response()->json([
            'status' => false,
            'message' => $th->getMessage()
        ], 500);
    }
}

public function logout()
{
    Auth::logout();
}

public function register(Request $request)
{
```

```
        $this->validate($request, [
            'volunteer_user_name' => 'required',
            'volunteer_user_surname' => 'required',
            'volunteer_user_patronymic' => 'required',
            'email' => 'required',
            'password' => 'required'
        ]);
        $user = new User;

        $data = $request->all();
        $data['volunteer_user_role_id'] = $data['volunteer_user_role_id'] ? 2 : 1;
        $data['password'] = Hash::make($data['password']);

        if(!User::where('email', $data['email'])->where('volunteer_user_role_id', $data['volunteer_user_role_id'])->exists()) {
            $user->create($data);
            return response('ok', 200);
        } else {
            return response()->json([
                'message' => 'Користувач вже зареєстрований'
            ], 500);
        }
    }
```

EvaluationController.php

Контролер оцінки користувачів.

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

use App\Models\Evaluations;

class EvaluationController extends Controller
{
    public function post(Request $request, $id)
    {
        if(Evaluations::where('volunteer_user_id', Auth::id())->where('volunteer_user_evaluations_reciever_id', $id)-
>exists()) {
            $evaluation = Evaluations::where('volunteer_user_id', Auth::id())->where('volunteer_user_evaluations_reciever_id',
$id)->first();
            $evaluation->volunteer_user_evaluations_value = $request->evaluation;
            $evaluation->save();
        } else {
            $evaluation = new Evaluations();
            $evaluation->volunteer_user_id = Auth::id();
            $evaluation->volunteer_user_evaluations_reciever_id = $id;
            $evaluation->volunteer_user_evaluations_value = $request->evaluation;
            $evaluation->save();
        }
    }
}
```

User.php

Модель користувачів.

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Sanctum\HasApiTokens;

class User extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable;

    protected $table = 'volunteer_user';
    protected $primaryKey = 'volunteer_user_id';

    protected $fillable = [
        'volunteer_user_role_id',
        'volunteer_user_status_id',
        'volunteer_country_id',
        'volunteer_user_name',
        'volunteer_user_surname',
        'volunteer_user_patronymic',
        'volunteer_user_photo',
        'volunteer_user_text',
        'email',
        'password',
        'is_ban'
    ];

    protected $appends = [
        'rate'
    ];

    function getRateAttribute() {
        return $this->evaluations->sum('volunteer_user_evaluations_value') > 0 ? round($this->evaluations-
>sum('volunteer_user_evaluations_value') / count($this->evaluations)) : $this->evaluations-
>sum('volunteer_user_evaluations_value');
    }

    public function role()
    {
        return $this->belongsTo('App\Models\UserRole', 'volunteer_user_role_id');
    }

    public function status()
    {
        return $this->belongsTo('App\Models\UserStatus', 'volunteer_user_status_id');
    }

    public function country()
    {
        return $this->belongsTo('App\Models\Country', 'volunteer_country_id');
    }

    public function vMessages()
    {
        return $this->hasMany('App\Models\Messages', 'volunteer_user_id');
    }
```

```
    public function evaluations()
    {
        return $this->hasMany('App\Models\Evaluations', 'volunteer_user_evaluations_reciever_id');
    }

    public function ads()
    {
        return $this->hasMany('App\Models\Ads', 'volunteer_user_id');
    }

    public function activeFundraising()
    {
        return $this->hasOne('App\Models\Ads', 'volunteer_user_id')->where('type', 'fundraising');
    }

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array<int, string>
     */
    protected $hidden = [
        'password',
        'remember_token',
    ];

    /**
     * The attributes that should be cast.
     *
     * @var array<string, string>
     */
    protected $casts = [
        'email_verified_at' => 'datetime',
        'evaluations_count' => 'integer'
    ];

    protected $guarded = ['volunteer_user_id'];
}
```

Ads.php

Модель оголошень.

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Ads extends Model
{
    protected $table = 'volunteer_ads';
    protected $primaryKey = 'volunteer_ads_id';

    protected $fillable = [
        'volunteer_ads_role_id',
        'volunteer_user_id',
        'volunteer_country_id',
        'volunteer_ads_name',
        'volunteer_ads_description',
        'volunteer_ads_urgency',
```

```
        'volunteer_ads_link',
        'volunteer_cost',
        'type',
        'freeze'
    ];

    protected $casts = [
        'volunteer_cost' => 'integer',
        'donate_count' => 'integer'
    ];

    public function user()
    {
        return $this->belongsTo('App\Models\User', 'volunteer_user_id');
    }
    public function country()
    {
        return $this->belongsTo('App\Models\Country', 'volunteer_country_id');
    }
    public function category()
    {
        return $this->belongsTo('App\Models\AdsRole', 'volunteer_ads_role_id');
    }
    public function photos()
    {
        return $this->hasMany('App\Models\Fotos', 'volunteer_ads_id');
    }
    public function preview()
    {
        return $this->hasOne('App\Models\Fotos', 'volunteer_ads_id');
    }
    public function donate()
    {
        return $this->hasMany('App\Models\Donate', 'ads_id');
    }
}
```

app.js

Вступний файл js, його функція - представляти глобальні бібліотеки, загальні стилі і методи, і навіть ставити маршрути.

```
require('./bootstrap');

import Vue from 'vue'
import { BootstrapVue, IconsPlugin } from 'bootstrap-vue'

import VueSweetalert2 from 'vue-sweetalert2';

import router from "./routes/app"
import store from "./store.js";

import 'bootstrap/dist/css/bootstrap.css'
import 'bootstrap-vue/dist/bootstrap-vue.css'

import 'vue-slider-component/theme/default.css'
import 'vue-slider-component/theme/material.css'
import 'vue-slider-component/theme/antd.css'
```

```
import AppComponent from './views/Main'

Vue.use(BootstrapVue)
Vue.use(IconsPlugin)
Vue.use(VueSweetalert2);

Vue.prototype.$http = axios

const app = new Vue({
   el: '#app',
   components: {
     AppComponent
   },
   router,
   store
});
```

routes/app.js

Файл маршрутизації сайту.

```
import Vue from "vue";
import Router from "vue-router";
import store from '../store';

Vue.use(Router);

function lazyLoad(view) {
   return () => import(`../views/${view}.vue`);
}

let router = new Router({
   mode: "history",
   base: process.env.BASE_URL,
   routes: [
      {
         path: "/",
         name: "home",
         component: lazyLoad("Home"),
      },
      {
         path: "/login",
         name: "login",
         component: lazyLoad("Login"),
      },
      {
         path: "/register",
         name: "register",
         component: lazyLoad("Register"),
      },
      {
         path: "/accounts/:id",
         name: "accounts",
         component: lazyLoad("Volunteers"),
      },
      {
         path: "/users/:id",
         name: "users-item",
         component: lazyLoad("VolunteersItem"),
      },
```

```
{
   path: "/ads",
   name: "ads",
   component: lazyLoad("Ads"),
},
{
   path: "/ads/:id",
   name: "ads-item",
   component: lazyLoad("AdsItem"),
},
{
   path: "/ads/:id/edit",
   name: "edit-ads",
   component: lazyLoad("CreateAds"),
},
{
   path: "/catalog",
   name: "catalog",
   component: lazyLoad("Catalog"),
},
{
   path: "/fundraising",
   name: "fundraising",
   component: lazyLoad("Fundraising"),
},
{
   path: "/about",
   name: "about",
   component: lazyLoad("About"),
},
{
   path: "/profile",
   name: "profile",
   component: lazyLoad("profile/Index"),
   meta: {
      requiresAuth: true
   },
   children: [
      {
         path: "/profile",
         name: "profile-edit",
         component: lazyLoad("profile/Edit"),
      },
      {
         path: "/profile/ads",
         name: "profile-ads",
         component: lazyLoad("profile/Ads"),
      },
      {
         path: "/profile/ads",
         name: "profile-ads",
         component: lazyLoad("profile/Ads"),
      },
      {
         path: "/profile/catalog",
         name: "profile-catalog",
         component: lazyLoad("profile/Catalog"),
      },
      {
         path: "/profile/fundraising",
```

```
                name: "profile-fundraising",
                component: lazyLoad("profile/Fundraising"),
            },
            {
                path: "/profile/chat",
                name: "profile-chats",
                component: lazyLoad("profile/Chats"),
                children: [
                    {
                        path: "/profile/chat/:id",
                        name: "chat-item",
                        component: lazyLoad("profile/ChatItem"),
                    },
                ]
            }
        ]
    },
    {
        path: "/create-ads",
        name: "create-ads",
        component: lazyLoad("CreateAds"),
        meta: {
            requiresAuth: true
        },
    }
  ],
});


router.beforeEach((to, from, next) => {
  if(to.matched.some(record => record.meta.requiresAuth)) {
    if (store.getters.isLoggedIn) {
      store.dispatch('update');
      next();
    } else {
      next('/login');
    }
  } else {
    next();
  }
})

export default router;
```

VolunteersItem.vue

Сторінка користувача.

```
<template>
  <div>
    <b-modal v-model="complaintModal" centered size="xs" title="Скарга">
      <b-form-textarea rows="3" style="height: 100px;" v-model="complaint"></b-form-textarea>
      <template #modal-footer="{ ok }">
        <b-button size="xs" variant="success" @click="sendComplaint()">
          Надіслати
        </b-button>
      </template>
    </b-modal>

    <b-modal v-model="messageModal" centered size="xs" title="Повідомлення">
```

```
        <b-form-textarea rows="3" style="height: 100px;" v-model="message"></b-form-textarea>
        <template #modal-footer="{ ok }">
          <b-button size="xs" variant="success" @click="sendMessage()">
            Надіслати
          </b-button>
        </template>
      </b-modal>

      <b-row class="header">
        <b-col md="3" lg="3" xl="3" xs="12" class="p-0">
          <img class="avatar" v-if="data.volunteer_user_photo" :src="data.volunteer_user_photo" alt="">
        </b-col>
        <b-col :class="['general', data.volunteer_user_role_id == 1 ? 'bg-green' : 'bg-red']">
          <b-row class="w-100">
            <b-col md="8" lg="8" xl="8" xs="12">
              <div class="content">
                <div class="country">
                  <span v-if="data.country">{{data.country.volunteer_country_title}},
</span>{{data.role.volunteer_user_role_name}}
                </div>
                <div class="name">
                  {{data.volunteer_user_name}} {{data.volunteer_user_surname}}
                </div>
                <star-rating
                  v-if="data.volunteer_user_role_id == 2"
                  v-model="data.rate"
                  :rating="data.rate"
                  class="rate"
                  :star-size="15"
                  :show-rating="false"
                  :padding="9"
                  :active-on-click="true"
                  :read-only="authUser() && authUser().authUser.volunteer_user_id != data.volunteer_user_id ? false :
true"
                  :round-start-rating="true"
                  @rating-selected="saveRate()"
                ></star-rating>
                <p class="mt-2">{{data.volunteer_user_text}}</p>
              </div>
            </b-col>
            <b-col cols="4">
              <button v-if="authUser() && authUser().authUser.volunteer_user_id != data.volunteer_user_id"
@click="messageModal = !messageModal">Написати</button>
              <button v-if="authUser() && authUser().authUser.volunteer_user_id != data.volunteer_user_id"
@click="complaintModal = !complaintModal">Поскаржитися</button>
            </b-col>
          </b-row>
        </b-col>
        <b-col cols="2" class="p-0 hide-block" v-if="data.active_fundraising">
          <div class="adss-item">
            <a :href="`/ads/${data.active_fundraising.volunteer_ads_id}`">
              <img :src="data.active_fundraising.preview.volunteer_fotos_file" alt="">
            </a>
            <div class="name px-2">
              <span>{{data.active_fundraising.volunteer_ads_name}}</span>
            </div>
            <div class="description px-2" v-if="data.active_fundraising.category">
              Категорія: {{data.active_fundraising.category.volunteer_ads_role_name}}
            </div>
            <div class="progress">
```

```
            <div class="progress_line" style="width: 59%">59%</div>
          </div>
        </div>
      </b-col>
      <b-col cols="1" class="p-0 hide-block" v-if="data.active_fundraising">
        <div class="head-title">
          Активні <br><span>збори</span>
        </div>
      </b-col>
    </b-row>
    <b-container fluid="xl">
      <div :class="['navigation', data.volunteer_user_role_id == 1 ? 'user' : '']">
        <button
          v-if="data.volunteer_user_role_id == 2"
          :class="filter.type == 'fundraising' ? 'active' : ''"
          @click="filter.type = 'fundraising'"
        >Збір коштів</button>
        <button
          v-if="data.volunteer_user_role_id == 2"
          :class="filter.type == 'catalog' ? 'active' : ''"
          @click="filter.type = 'catalog'"
        >Каталог</button>
        <button
          v-if="data.volunteer_user_role_id == 1"
          :class="filter.type == 'ads' ? 'active' : ''"
          @click="filter.type = 'ads'"
        >Оголошення</button>
      </div>
      <b-row>
        <b-col md="9" lg="9" xl="9" xs="12" order="2" order-md="1">
          <b-row v-if="filter.type == 'fundraising'">
            <b-col md="4" lg="4" xl="4" xs="12" v-for="(item, i) in filterFundraising" :key="i">
              <FundraisingItem :item="item" />
            </b-col>
          </b-row>
          <b-row v-if="filter.type == 'catalog' || filter.type == 'ads'">
            <b-col md="4" lg="4" xl="4" xs="12" v-for="(item, i) in filterAds" :key="i">
              <AdsItem :item="item" />
            </b-col>
          </b-row>
        </b-col>
        <b-col md="3" lg="3" xl="3" xs="12" order="1" order-md="2" v-if="(data.ads.length > 0)">
          <div :class="['filter', 'mt-4', data.volunteer_user_role_id == 1 ? 'user' : '']">
            <div class="filter-grout">
              <input type="text">
            </div>
            <div class="filter-grout">
              <div class="title">Категорія</div>
              <div class="checkbox-group" v-for="(item, i) in category" :key="i">
                <input
                  type="checkbox"
                  :id="`category-${item.volunteer_ads_role_id}`"
                  v-model="filter.categories"
                >
                <label :for="`category-${item.volunteer_ads_role_id}`">{{item.volunteer_ads_role_name}}</label>
              </div>
            </div>
            <div class="filter-grout">
              <div class="title">Терміновість</div>
              <div class="checkbox-group">
```

```
                        <input type="checkbox" id=""> <label for="">Терміновість</label>
                    </div>
                </div>
                <div class="filter-grout">
                    <div class="title">Бюджет</div>
                    <div class="range-group">
                        <vue-slider v-model="value"></vue-slider>
                    </div>
                </div>
                <div class="filter-grout">
                    <button>Фільтрація</button>
                </div>
            </div>
        </b-col>
    </b-row>
</b-container>
</div>
</template>

<script>
import StarRating from "vue-star-rating";
import FundraisingItem from "../components/FundraisingItem.vue";
import AdsItem from "../components/AdsItem.vue";
import VueSlider from 'vue-slider-component'

import { mapGetters } from 'vuex';
export default {
    components: {
        StarRating,
        FundraisingItem,
        AdsItem,
        VueSlider
    },
    data() {
        return {
            complaint: "",
            message: "",
            complaintModal: false,
            messageModal: false,
            data: {
                volunteer_user_photo: "",
                role: {},
                country: {},
                active_fundraising: {
                    preview: {}
                },
                ads: []
            },
            value: [0, 5],
            category: [],
            filter: {
                type: "fundraising"
            }
        }
    },
    created() {
        this.getData();
        this.gatCategort();
    },
    computed: {
```

```
        filterFundraising() {
            return this.data.ads.filter(item => {
                return item.type == 'fundraising'
            })
        },
        filterAds() {
            return this.data.ads.filter(item => {
                return item.type != 'fundraising'
            })
        }
    },
    methods: {
        ...mapGetters(["authUser"]),
        getData() {
            axios.get('/api/users/'+this.$route.params.id, {
                params: this.filter
            })
                .then((response) => {
                    this.data = Object.assign(this.data, response.data);
                    if(this.data.volunteer_user_role_id == 1) {
                        this.filter.type = 'ads';
                    }
                });
        },
        gatCategort() {
            axios.get('/api/category')
                .then((response) => {
                    this.category = response.data;
                });
        },
        saveRate() {
            axios.post('/api/users/'+this.$route.params.id+'/evaluation', {
                evaluation: this.data.rate
            })
                .then(() => {
                    this.getData();
                });
        },
        sendComplaint() {
            axios.post('/api/users/' + this.$route.params.id + '/complaint', {
                text: this.complaint
            })
                .then(() => {
                    this.complaintModal = false;
                });
        },
        sendMessage() {
            axios.post('/api/chats', {
                users_id: this.data.volunteer_user_id,
                message: this.message
            })
                .then(() => {
                    this.messageModal = false;
                    this.message = "";
                })
        }
    }
}
</script>
```

```
<style lang="scss" scoped>
    .bg-red {
        background: #910000;
    }
    .bg-green {
        background: #0C6715;
    }
    .header {
        .avatar {
            width: 100%;
            height: 100%;
            object-fit: cover;
        }
        .general {
            background: #910000;
            display: flex;
            align-items: center;
            padding: 0 45px;
            .country {
                font-weight: 300;
                font-size: 18px;
                line-height: 22px;
            }
            .name {
                font-weight: 700;
                font-size: 40px;
                line-height: 49px;
            }
            .content {
                button {
                    width: 340px;
                    background: url('/images/chevron-right.png') no-repeat;
                    background-position: right 18px center;
                    margin-top: 25px;
                }
            }
        }
        .adss-item {
            margin-bottom: 0;
            display: flex;
            flex-direction: column;
            justify-content: space-between;
            height: 100%;
        }
        button {
            width: 100%;
            display: block;
            font-weight: 700;
            font-size: 16px;
            color: #fff;
            border: 1px solid #FFFFFF;
            padding: 8px;
            margin-bottom: 20px;
        }
        .head-title {
            font-weight: 700;
            font-size: 42px;
            line-height: 110%;
            text-transform: uppercase;
            transform: rotate(90deg);
```

```
            padding: 10px;
            span {
               -webkit-text-fill-color: transparent;
               -webkit-text-stroke: 1px #FFFFFF;
            }
         }
      }
      .navigation {
         button {
            width: 345px;
            border: 0;
            background: #000000;
            padding: 24px;
            display: inline-block;
            color: #FFFFFF;
            font-weight: 600;
            font-size: 18.4px;
            margin: 25px 32px 25px 0;
            border-left: 10px solid #910000;
         }
         button.active {
            background: #910000;
            border-left: 10px solid #910000;
         }
      }.user {
         button {
            border-left: 10px solid #0C6715;
         }
         button.active {
            background: #0C6715;
            border-left: 10px solid #0C6715;
         }
      }
      @media screen and (max-width: 600px) {
         .hide-block {
            display: none;
         }
         .general {
            padding: 45px !important;
         }
         .navigation {
            display: flex;
         }
      }
</style>
```

AdsItem.vue

Сторінка оголошення.

```
<template>
   <b-container fluid="xl">
      <b-modal v-model="donateModal" centered size="sm" title="Задонатити">
         <b-form-input v-model="sum" class="text-center"></b-form-input>
         <template #modal-footer="{ ok }">
            <b-button size="xs" variant="success" @click="sendDonate()">
               Надіслати
            </b-button>
         </template>
      </b-modal>
```

```
    <b-modal v-model="complaintModal" centered size="xs" title="Скарга">
      <b-form-textarea rows="3" style="height: 100px;" v-model="complaint"></b-form-textarea>
      <template #modal-footer="{ ok }">
        <b-button size="xs" variant="success" @click="sendComplaint()">
          Надіслати
        </b-button>
      </template>
    </b-modal>

    <b-row>
      <b-col md="5" lg="5" xl="5" xs="12">
        <div class="swiper-container">
          <div class="swiper-wrapper">
            <div class="swiper-slide" v-for="(item, i) in data.photos" :key="i">
              <img :src="item.volunteer_fotos_file" class="photo">
            </div>
          </div>
          <div class="swiper-pagination"></div>
        </div>
      </b-col>
      <b-col md="7" lg="7" xl="7" xs="12">
        <div class="content">
          <div class="title">
            {{ data.volunteer_ads_name }}
            <button v-if="data.freeze == 0 && data.type == 'fundraising' && data.volunteer_cost > data.donate_count"
class="chevron"
              @click="donateModal = !donateModal">Задонатити</button>
          </div>
          <div :class="['progress', data.volunteer_cost > data.donate_count ? '' : 'complete']" v-if="data.type ==
'fundraising'">
            <div
              :class="['progress_line', data.volunteer_cost > data.donate_count ? 'bg-red' : 'bg-green']"
              :style="data.volunteer_cost > data.donate_count ? `width: ${Math.round(data.donate_count * 100 /
data.volunteer_cost)}%` : 'width: 100%'"
            >{{Math.round(data.donate_count * 100 / data.volunteer_cost)}}%</div>
          </div>
          <ul class="info">
            <li v-if="data.category">Категорія: {{ data.category.volunteer_ads_role_name }}</li>
            <li v-if="data.volunteer_cost">Бюджет: $ {{ data.volunteer_cost }}</li>
            <li v-if="data.country">Країна: {{ data.country.volunteer_country_title }}</li>
            <li v-if="data.volunteer_ads_link">Посилання: <a :href="data.volunteer_ads_link">{{
data.volunteer_ads_link }}</a></li>
          </ul>
          <p v-html="data.volunteer_ads_description"></p>
          <b-row>
            <b-col>
              <div class="user">
                <div class="avatar">
                  <router-link :to="`/users/${data.user.volunteer_user_id}`">
                    <img :src="data.user.volunteer_user_photo" alt="">
                  </router-link>
                  <div>
                    <router-link :to="`/users/${data.user.volunteer_user_id}`">
                      {{ data.user.volunteer_user_name }} {{ data.user.volunteer_user_surname }}
                    </router-link>
                    <star-rating class="rate" :rating="Math.round(data.user.rate)"
                      :star-size="10" :show-rating="false" :read-only="true" :padding="5">
                    </star-rating>
                  </div>
                </div>
```

```
                    </div>
                </b-col>
                <b-col style="text-align: right" v-if="authUser()">
                    <button @click="complaintModal = !complaintModal">Поскаржитися</button>
                </b-col>
            </b-row>
        </div>
    </b-col>
</b-row>
</b-container>
</template>

<script>

import StarRating from "vue-star-rating";
import Swiper from 'swiper/swiper-bundle.esm.js';
import 'swiper/swiper-bundle.css';

import { mapGetters } from 'vuex';

export default {
    components: {
        StarRating,
        Swiper,
    },
    data() {
        return {
            sum: "",
            complaint: "",
            donateModal: false,
            complaintModal: false,
            data: {
                photos: [],
                user: {}
            }
        }
    },
    mounted() {
        this.getData();
        setTimeout(function () {
            new Swiper('.swiper-container', {
                pagination: {
                    el: '.swiper-pagination',
                },
            });
        }, 1000);
    },
    methods: {
        ...mapGetters(["authUser"]),
        getData() {
            axios.get('/api/ads/' + this.$route.params.id)
                .then((response) => {
                    this.data = Object.assign(this.data, response.data);
                });
        },
        sendDonate() {
            axios.post('/api/ads/' + this.$route.params.id + '/donate', {
                sum: this.sum
            })
                .then(() => {
```

```
                this.donateModal = false;
                this.getData();
            });
        },
        sendComplaint() {
            axios.post('/api/ads/' + this.$route.params.id + '/complaint', {
                text: this.complaint
            })
                .then(() => {
                    this.complaintModal = false;
                });
        }
    }
}
</script>

<style lang="scss" scoped>
.photo {
    width: 100%;
    height: auto;
}

.content {
    .title {
        font-weight: 600;
        font-size: 32px;
        line-height: 20px;
        margin-bottom: 25px;
        display: flex;
        align-items: center;
        justify-content: space-between;
    }

    ul.info {
        li {
            font-weight: 400;
            font-size: 14px;
            line-height: 24px;
        }
    }

    p {
        font-weight: 400;
        font-size: 14px;
        line-height: 26px;
        margin-top: 13px;
    }

    .user {
        margin-top: 10px;
        padding-top: 10px;
        border-top: 1px solid #fff;
        display: flex;
        justify-content: space-between;

        .avatar {
            display: flex;
            align-items: center;
            font-weight: 700;
            font-size: 15px;
```

```
        line-height: 24px;

        img {
            width: 33px;
            height: 33px;
            object-fit: cover;
            margin: 0 15px 0 0;
        }

        .rate {
            height: 10px;
        }
    }

    button {
        color: #fff;
        font-weight: 700;
        font-size: 10px;
    }
    a {
        color: #fff;
    }
}

button {
    background: none;
    width: 270px;
    display: inline-block;
    height: 44px;
    font-weight: 700;
    font-size: 16px;
    color: #fff;
    border: 1px solid #fff;
}

.chevron {
    background: url('/images/chevron-right.png') no-repeat;
    background-position: right 16px center;
}

.progress {
    border-bottom: 2px solid #910000;
    background: none;
    border-radius: 0;
    height: 60px;
    margin-top: 10px;
    margin-bottom: 25px;

    &_line {
        font-weight: 400;
        font-size: 18px;
        padding: 7px 0 5px 0;
        overflow: hidden;
        margin-bottom: -1px;
        display: flex;
        align-items: center;
        justify-content: center;
    }
}.complete {
    border-bottom: 2px solid #0C6715;
```

```
    }
    @media screen and (max-width: 600px) {
        .title {
            margin-top: 20px;
            flex-direction: column;
            button {
                margin-top: 20px;
            }
        }
    }
}
</style>
```

Header.vue

Компонент шапки сайту.

```
<template>
    <div class="header">
        <b-container fluid="xl">
            <nav>
                <b-row>
                    <b-col md="9" lg="9" xl="9" xs="12">
                        <div class="navigation">
                            <div class="logo-block">
                                <a href="/">
                                    <img src="/images/logo.svg" class="logo">
                                </a>
                                <div class="menu-burger">
                                    <input id="menu__toggle" v-model="menu" type="checkbox" />
                                    <label class="menu__btn" for="menu__toggle">
                                        <span></span>
                                    </label>
                                </div>
                            </div>
                            <ul :class="authUser() && authUser().authUser.volunteer_user_role_id == 1 ? 'user' : "" v-if="menu">
                                <li :class="$route.name == 'home' ? 'active' : """>
                                    <router-link to="/">Головна</router-link>
                                </li>
                                <li :class="$route.name == 'accounts' && $route.params.id == 2 ? 'active' : """>
                                    <router-link to="/accounts/2">Волонтери</router-link>
                                </li>
                                <li :class="$route.name == 'accounts' && $route.params.id == 1 ? 'active' : """>
                                    <router-link to="/accounts/1">Користувачі</router-link>
                                </li>
                                <li :class="$route.name == 'ads' ? 'active' : """>
                                    <router-link to="/ads">Оголошення</router-link>
                                </li>
                                <li :class="$route.name == 'catalog' ? 'active' : """>
                                    <router-link to="/catalog">Каталог</router-link>
                                </li>
                                <li :class="$route.name == 'fundraising' ? 'active' : """>
                                    <router-link to="/fundraising">Збір коштів</router-link>
                                </li>
                                <li :class="$route.name == 'about' ? 'active' : """>
                                    <router-link to="/about">Про нас</router-link>
                                </li>
                            </ul>
                        </div>
                    </b-col>
                    <b-col md="3" lg="3" xl="3" xs="12">
```

```
                <div :class="authUser() ? '' : 'auth'">
                  <router-link v-if="!authUser()" to="/login" class="btn-default active">Авторизуватися</router-link>
                  <div v-else class="accounts">
                    <router-link to="/create-ads" class="btn-default active">Додати</router-link>
                    <div class="status">
                      <span>Статус:</span>
                      {{ authUser().authUser.volunteer_user_status_id == 1 ? 'Вільний' : 'Зайнятий' }}
                      <div class="logout" @click="logout()">Вихід</div>
                    </div>
                    <div class="avatars">
                      <router-link to="/profile">
                        <img
                          :src="authUser().authUser.volunteer_user_photo"
                          v-if="authUser().authUser"
                          :class="authUser().authUser.volunteer_user_role_id == 1 ? 'active-green' : 'active-red'"
                          :title="authUser().authUser.volunteer_user_id"
                        >
                      </router-link>
                      <img
                        :title="authUser().anotherRole.volunteer_user_id"
                        :src="authUser().anotherRole.volunteer_user_photo"
                        v-if="authUser().anotherRole"
                        @click="loginAnotherRole(authUser().anotherRole.volunteer_user_id)"
                      >
                      <router-link to="/register">
                        <img
                          src="/images/plus.png"
                          v-if="!authUser().anotherRole"
                        >
                      </router-link>
                    </div>
                  </div>
                </div>
              </b-col>
            </b-row>
          </nav>
        </b-container>
      </div>
    </template>

    <script>
    import { mapGetters } from 'vuex';
    export default {
      data() {
        return {
          menu: true,
          ifScroll: false
        }
      },
      mounted() {
        window.addEventListener('scroll', function(event) {
          this.ifScroll = window.scrollY > 50
          if(this.ifScroll) {
            document.querySelector('.header').classList.add('scroll')
          } else {
            document.querySelector('.header').classList.remove('scroll')
          }
        })

      },
```

```
    methods: {
        ...mapGetters(["authUser"]),
        loginAnotherRole(id) {
            axios.post('/api/login-with-id/' + id)
                .then((response) => {
                    localStorage.setItem('token', response.data.token)
                    localStorage.setItem('user', JSON.stringify(response.data.user))
                    window.location.reload();
                })
        },
        logout() {
            this.$store.dispatch('logout').then(() => {
                this.$router.push('/')
            })
        },
    }
}
</script>

<style lang="scss" scoped>
.auth {
    padding-left: 50px;
}
.logout {
    cursor: pointer;
}
.accounts {
    display: flex;
    align-items: center;
    .status {
        font-weight: 500;
        font-size: 14px;
        line-height: 17px;
        margin-right: 10px;
        margin-left: 50px;
        span {
            margin-top: 3px;
            color: rgba(217, 217, 217, 0.5);
            font-size: 9px;
            display: block;
        }
    }
    .avatars {
        display: flex;
        align-items: flex-end;
        img {
            border: 1.5px solid #ABABAB;
            width: 23px;
            height: 23px;
            border-radius: 50%;
            margin-left: 5px;
            cursor: pointer;
            object-fit: cover;
        }
        img.active-red {
            border: 5px solid #910000;
            width: 44px;
            height: 44px;
        }
        img.active-green {
```

```
            border: 5px solid #0C6715;
            width: 44px;
            height: 44px;
        }
    }
    .btn-default {
        width: 200px;
        height: 44px;
        display: flex;
        align-items: center;
        justify-content: center;
        background-position: right 18px center;
    }
}
.header {
    padding: 60px 0;
    position: fixed;
    width: 100%;
    top: 0;
    z-index: 9;
    background: #000;
    transition: 0.3s;
}.scroll {
    padding: 15px 0;
}
.navigation {
    display: flex;
    align-items: center;
    height: 100%;
    img.logo {
        width: 40px;
        margin-right: 90px;
    }
    ul {
        display: flex;
        li {
            margin-right: 30px;
            border-bottom: 1px solid #000;
            a {
                font-weight: 500;
                font-size: 15px;
                line-height: 20px;
                color: #D9D9D9;
                position: relative;
            }
        }
        li:hover, li.active {
            border: 0;
            border-bottom: 1px solid #910000;
        }
    }.user {
        li:hover, li.active {
            border: 0;
            border-bottom: 1px solid #0C6715;
        }
    }
}

@media screen and (max-width: 600px) {
    .header {
```

```
        position: relative;
        padding: 20px 0;
    }
    nav, .navigation, ul {
        flex-direction: column;
    }
    .navigation {
        width: 100%;
        align-items: flex-start;
    }
    ul {
        width: 100%;
        li {
            margin: 5px 0 !important;
            text-align: center;
            width: auto;
        }
    }
    .btn-default {
        margin-top: 20px;
    }
    .logo-block {
        display: flex;
        justify-content: space-between;
        width: 100%;
    }
}
</style>
```