**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА**

Факультет прикладної математики та інформатики

Кафедра інформаційних систем

**МАГІСТЕРСЬКА РОБОТА**

РОЗРОБКА ВЕБ-АПЛІКАЦІЇ ДЛЯ ДОПОМОГИ УКРАЇНСЬКІЙ АРМІЇ

Виконав(ла):студентка групи ПМіМ-22

спеціальності 122 - компʼютерні науки

  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

                                                                                             (підпис)                          (прізвище та ініціали)

Керівник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

                                                                                                          (підпис)                  (прізвище та ініціали)

Рецензент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

                                                                                                          (підпис)                  (прізвище та ініціали)

2023

# **Вступ**

Я народилася і живу у країні, в основі якої лежить столітній досвід боротьби та змагань за свою незалежність. Адже це ми вийшли на Майдан тоді, коли наш тодішній президент і уряд відмовились від європейського цивілізаційного вибору. Це ми організували повстання, аналогів якого у світовій історії не було. Це ми прийняли бій з фанерними щитами і в пластикових касках за свою свободу.

Це наша країна зберігає керованість і всі свої державні функції у найбільшій в історії війні з використанням найбільшої у світі за останні 70 років військової потужності щодо нас. Це ми не здались і успішно воюємо 10 років з найсильнішою ядерною державою світу. Це ми вперті і вільні, хто всупереч агентам впливу у власній владі, найсильнішим державам світу, які прагнуть нашого фізичного знищення, успішно знищуємо всіх, хто має нахабство підняти на нас зброю і утверджуємо своє право, і свою волю на своїй Богом даній землі.

Це Ми. І все це трапилось тільки з нами. Латиняни казали: "Historia est magistra vitae" - Історія - вчителька життя. Численні історичні сторінки свідчать про величезний духовний потенціал, гідність та силу нашого народу. Попри будь-які виклики впродовж століть ми завжди залишалися вірними ідеям свободи та незалежності.

Сьогодні серця кожного українця сповнені любові до своєї держави, віри у її перемогу та вдячності її героям за кожен прожитий день. Настали складні часи для кожного українця в тилу, а особливо на фронті. На плечах ЗСУ лежить доля країни, майбутнє її громадян. Ми розуміємо, як це важко і як це відповідально. Без них не було б у нас надії на завтрашній день, віри у мирне небо над головою кожного українця. Саме тому кожен свідомий українець повинен жити з девізом: «Або в ЗСУ, або для ЗСУ!».

Волонтерство під час війни є одним із найбільш важливих аспектів у забезпеченні допомоги тим, хто цього потребує і кому це вкрай необхідно. Волонтерство українців стало прикладом солідарності та відданості своїй країні. Це стало доказом того, що український народ здатний на великі дії та здатний об’єднуватися в складні часи. Волонтери стали прикладом того, як можна змінити світ на краще своїми діями. Ми долучаємося до різних проєктів для допомоги нашим військовим, адже прекрасно усвідомлюємо, що чим більше ЗСУ пожертвуємо та допоможемо тим швидше Переможемо. Саме це має пам’ятати кожен і робити все від нас залежне, щоб полегшити роботу нашим військовим. Памʼятаймо про ціну, яку вони кладуть на вівтар за нас, заради Перемоги.

Цитуючи слова головнокомандуючого ЗСУ генерала Валерія Залужного: «Ми боремося за вас, за ваше майбутнє, ваше право жити у власній державі та втілювати свої мрії. Ми б’ємося за нашу землю, щоб не залишити війну у спадок вам. Нам всім зараз важко, дуже важко — але вже точно ніколи не буде соромно.»[1] Тому і нам не має бути соромно, що не змогли чи не захотіли допомогти нашим захисникам.

Насправді робота в тилу не менш важлива, ніж робота на передовій. Війна не закінчилася, тому і надалі мусимо робите все можливе заради Перемоги. Волонтерство дає змогу людям не лише допомагати, а й набувати нові знання та навички, підвищувати свою самооцінку та змінювати світ навколо себе.

Чимало людей знають про наступні організації, мета яких спрямована на допомогу українській армії:

1. **“Повернись живим”:** благодійний фонд допомоги армії заснований у 2014 році. За весь час на допомогу армії було зібрано 9 мільярдів гривень. Фонд займається закупівлею різного типу озброєння: автомобілі, тепловізійна оптика, засоби захисту, зв’язку тощо.
2. **“Армія SOS”:** волонтерська організація яка займається виготовленням безпілотних літальних апаратів, постачає радіорозвідувальне обладнання та транспорту для військових. Заснована вона ще у 2014 році.
3. **KOLO:** повністю некомерційний благодійний фонд, зосереджений на збір коштів для закупівлі оптики, засобів зв’язку тощо. Приймають тільки запити офіційні від військових частин. Особливість - розіграш різних цікавих слотів за донат на ЗСУ.
4. **Турбота про військовослужбовця:** надають всю інформацію про те як отримати різні типи допомоги після поранення: як отримати виплати після поранення, що робити у випадку потреби протезування тощо. Також містять зразки документів/рапортів, які можуть бути корисними для військових. Знайдуться і корисні відеоматеріали з тактичної медицини.

Також такі сайти, як **Волонтерство в Україні**, **Help Ukraine**, **UA Helpers**, є відомими платформами, які займаються допомогою українській армії. Ці ресурси зібрали навколо себе активну спільноту волонтерів та людей, які бажають підтримати військових. Внесок кожної організації є безсумніву дуже важливий для нашої перемоги.

Але, серед перелічених веб аплікацій нема такої, яка б змогла структурувати та зібрати всі типи допомоги військовим в одному місці. Тим не менш, не має рішення для зборів на реабілітацію військових.

Всі ми знаємо, що кожна аплікація – це вирішення окремої проблеми. Тому, щоб вирішити питання найбільш ефективного та швидкої допомоги українським військовим я обрала тему “Розробка веб-аплікації для допомоги військовим – UA Defenders”.

# **1 Суть магістерської роботи**

Суть даного завдання полягає у створенні застосунку, який допоможе спростити складний процес допомоги волонтерів для війскових. Веб-застосунок зможе об'єднати різні видів волонтерських інінціатив в одному місці. Це в свою чергу забезпечить централізоване керування, струтурування та координацію різноманітних благодійних акцій, проєктів, фондів та інших джерел допомоги. Веб-аплікація буде корисна для тих, хто бажає швидко та легко обрати спосіб допомоги військовим та здійснити переказ.

Завдяки веб-застосунку, користувачі зможуть швидко та зручно переглянути доступні способи допомоги військовим, такі як фінансові внески, матеріальна підтримка, волонтерська робота, надання послуг та багато інших. Це дозволить різним групам людей, включаючи звичайних громадян, організації, бізнеси активно долучитись до процесу надання допомоги.

Користувач, що бажає допомогти, має змогу авторизуватись за допомогою електронної пошти та паролю. Після реєстрації можна обрати вид допомоги: благодійний про’єкт, лікування військових, запити матеріальної допомоги. Опубліковані також будуть і реквізити найбільш відомих фондів, які збирають кошти для української армії.

Люди, які розпочали роботу над власним благодійним проєктом, матимуть змогу розмістити свій збір у додатку. Для цього необхідно буде заповнити форму з відповідними полями. Також буде можливість і для розміщення оголошення про збір на лікування військового, заповнивши для цього відповідну форму.

Для користувачів, які бажатимуть розказати про власний запит матеріальної допомоги, або ж пропозицію, як вони можуть допомогти, цей додаток також стане корисним.

# **2 Постановка задачі**

## **2.1 Фізична постановка задачі**

Аплікація повинна передбачати наступні типи користувачів у системі:

1. Незареєстрований користувач
2. Зареєстрований користувач
3. Адміністратор

**Незареєстровані користувачі** розробленої інформаційної системи повинні отримати наступні можливості:

1. можливість дізнатись інформацію про даний Web застосунок;
2. можливість перегляду фондів та їхні реквізити;
3. перегляд опублікованих благодійних проєктів;
4. пошук/фільтрація опублікованих проєктів за локацією та катигорією;
5. перегляд опублікованих зборів на лікування військових;
6. пошук/фільтрація опублікованих зборів на лікування військових за Ім’ям та Прізвищем;
7. перегляд опублікованих пропозицій/прохань;
8. пошук/фільтрація пропозицій/прохань за локацією та категорією;
9. можливість зареєструватись для подальшого пошуку необхідного типу допомоги чи створення різноманітних оголошень.

Функціональні можливості для **зареєстрованих користувачів** повинні передбачати наступне:

1. створення персонального профілю: користувач вводить електронну пошту та пароль. Ці дані перевіряються на валідність;
2. можливість зворотнього зв’язку з організаторами з допомогою заповненої форми, вказавши наступні поля:
3. Електронна адреса;
4. Коментар;
5. Інші способи зв’язку;
6. створення запиту на публікацію благодійного проєкту завповнивши форму з наступними полями:
7. Електронна пошта;
8. Назва (коротко про суть);
9. Опис проєкту (детальний опис);
10. Реквізити банку;
11. Локація;
12. Документи;
13. Фотографія;
14. Як зв’язатись;
15. Категорія;
16. створення запиту на публікацію збору на лікування військового, завповнивши форму з наступними полями:
17. Прізвище, Ім’я, По-батькові;
18. Електронна пошта;
19. Локація;
20. Військова частина;
21. Військова історія та поранення;
22. Історія лікування;
23. Сума, яку необхідно зібрати;
24. Документи;
25. Фотографія;
26. Реквізити банку;
27. Як зв’язатись;
28. створення запиту на публікацію пропозиції чи прохання про допомогу, завповнивши відповідні два типи форми з наступними полями:
29. Назва;
30. Електронна пошта;
31. Локація;
32. Опис;
33. Категорія;
34. Як зв’язатись;
35. керування особистим кабінетом:
36. Редагування персональної інформації;
37. Редагування/видалення публікацій з благодійними проєктами;
38. Редагування/видалення публікацій з запитами на реабілітацію військових;
39. Редагування/видалення публікацій з запитами на реабілітацію військових;
40. Зміна паролю;
41. Вихід з особистого кабінету;
42. Видалення особистого акаунту.

Можливості **адміністратора** включатимуть:

1. Перегляд головної сторінки та сторінок з різноманітними типами допомоги;
2. Перегляд запитів публікації благодійних проєктів (включаючи детальну інформацію);
3. Перегляд запитів публікації зборів на лікування військового (включаючи детальну інформацію);
4. Перегляд оголошень пропозиції/прохання матеріальної допомоги (включаючи детальну інформацію);
5. Затвердження/відхилення публікації оголошення про благодійний проект;
6. Затвердження/відхилення публікації оголошення про збір на лікування військового;
7. Затвердження/відхилення публікації оголошення пропозиції/прохання матеріальної допомоги;
8. Перегляд запитів зворотнього зв’язку.

Результатом збору та елісітації вимог для даної інформаційної системи є діаграма прецедентів або Use Case діаграма ***(рисунок 2.1)***. Як зазначає PMBOK, Use Case діаграма – це діаграма, яка описує і досліджує як користувач взаємодіє з системою, щоб досягти конкретної мети [1].

***Рисунок 2.1*** – Діаграма прецедентів для інформаційної системи

## **2.2 Програмна постановка задачі**

### **2.2.1 Вибір технологій та їх актуальність**

В сучасному світі швидкого технологічного розвитку вибір технологій для розробки веб-додатку стає стратегічним вирішенням, яке може значно вплинути на якість, продуктивність та майбутнє проєкту.

Вибір технологічного стеку MERN для розробки веб-додатка визначає не тільки технічні аспекти, але і ключові аспекти успішного виконання проєкту. Значущість правильного вибору технології полягає у його відповідності конкретним вимогам, потребам користувачів та стратегії розвитку проєкту. Нижче наведено головні фактори важливості вибору технологічного стеку:

1. *Відповідність вимогам проєкту:* технологія повинна бути адаптована до специфічних потреб проєкту, включаючи функціональність, масштабованість та можливості розширення.
2. *Ефективність та продуктивність розробки:* вибір технологій повинен сприяти ефективній розробці, забезпечуючи продуктивність розробників та швидкість впровадження нового функціоналу.
3. *Спільні стандарти та спільнота розробників*: використання технологій, які відповідають стандартам та користуються підтримкою великої та активної спільноти розробників, забезпечує доступ до досвіду та рішень спільноти.
4. *Масштабованість та продуктивність використання ресурсів:* технології повинні бути здатні масштабуватися для обробки зростаючого обсягу даних та користувачів, забезпечуючи оптимальну продуктивність використання ресурсів.
5. *Забезпечення безпеки та захисту даних*: технології повинні відповідати вимогам безпеки, забезпечуючи ефективний захист від потенційних загроз та збереження конфіденційності даних.
6. *Гнучкість та сумісність:* гнучкість технологій дозволяє легко адаптувати додаток до нових вимог та змін у бізнес-логіці. Також важливо, щоб технології були сумісні між собою.
7. *Технічна підтримка та обслуговування:* важливо обирати технології, які мають надійну технічну підтримку та можливості обслуговування для забезпечення стабільності та ефективності системи.

Відповідно до цих аспектів, вибір MERN stack для проекту визначається його спроможністю забезпечити комплексне вирішення завдань, пов'язаних з розробкою веб-додатку, та задовольнити високі стандарти ефективності, безпеки та масштабованості.

Технологічний стек MERN (MongoDB, Express.js, React.js, Node.js) представляє собою потужний та комплексний інструментарій для розробки веб-додатків. Кожна з технологій в MERN виконує важливі функції, які взаємодіють для створення високопродуктивних та масштабованих додатків.

### **2.2.2 Архітектура MVC та її взаємозв'язок з MERN stack**

Архітектура MVC (Model-View-Controller): Архітектура MVC є архітектурним патерном, який спрощує розробку програмних додатків. Він розділяє додаток на три основні компоненти:

1. **Модель (Model)** – відповідає за управління даними та бізнес-логікою. Модель зберігає інформацію та обробляє всі маніпуляції з нею.
2. **Вид (View)** – відповідає за представлення даних користувачу та взаємодію з ним. View відображає інформацію, яку надає Model, і відслідковує події користувача.
3. **Контролер (Controller)** – обробляє вхідні події від користувача, взаємодіє з Model та View для виконання конкретних дій.

Ця архітектура сприяє розділенню відповідальності та полегшує обслуговування та розширення додатку.



***Рисунок 2.1*** – діаграма архітектури MVC

Архітектура MVC з'явилася в середині 1970-х років із запуском програмування для графічних інтерфейсів у Xerox PARC. Спочатку вона використовувалася для реалізації додатків із складною взаємодією з користувачем, де було необхідно чітко розділити представлення даних, бізнес-логіку та обробку користувацького вводу.

Роль архітектури MVC в існуванні веб-додатку полягає в тому, що вона забезпечує:

1. *Чітке розділення відповідальності:* компоненти Model, View та Controller не залежать один від одного, що дозволяє ефективно внести зміни або розширити окремі частини додатку без змін у інших.
2. *Легкість обслуговування:* розділення логіки додатку на окремі компоненти полегшує тестування, розширення та управління кодом.
3. *Можливість паралельної розробки:* різні команди можуть одночасно працювати над Model, View та Controller без конфліктів, оскільки вони взаємодіють через визначені інтерфейси.

В сучасних веб-додатках архітектура MVC використовується для створення зручних, масштабованих та добре організованих систем. Вона підтримує розширення та підтримку коду протягом тривалого життєвого циклу додатку.

***Взаємозв'язок архітектури MVC та MERN stack:***

Архітектура MVC (Model-View-Controller) та стек MERN (MongoDB, Express, React, Node.js) взаємодіють, забезпечуючи організовану та ефективну структуру веб-додатку. Розглянемо, як вони взаємодіють між собою.

*Model та MongoDB:*

1. Model в архітектурі MVC відповідає за управління даними та бізнес-логіки.
2. MongoDB, що є частиною стеку MERN, надає постійне зберігання даних у формі NoSQL документів.
3. Model визначає структуру даних, взаємодіє з базою даних MongoDB та надає необхідні дані для інших компонентів.

*View та React:*

1. View у архітектурі MVC відповідає за представлення даних та взаємодію з користувачем.
2. React, ключовий елемент у стеку MERN, дозволяє створювати динамічні та ефективні інтерфейси користувача.
3. View у React рендерить компоненти, які відображають інформацію та реагують на події користувача.

*Controller та Express/Node.js:*

1. Controller у архітектурі MVC обробляє вхідні події від користувача та взаємодіє з Model та View для виконання конкретних дій.
2. Express, використовуючи Node.js, реалізує серверну логіку та обробку HTTP-запитів.
3. Controller визначає маршрути за допомогою Express, обробляє запити від клієнта, взаємодіє з Моделлю та оновлює View.

Таким чином, взаємодія архітектури MVC та стеку MERN сприяє створенню добре структурованих та маштабованих веб-додатків, де кожен компонент виконує свою функцію та ефективно взаємодіє з іншими.

### **2.2.3 MongoDB: кросплатформенна документно-орієнтована база даних**

MongoDB, ключова складова MERN stack, представляє сучасну базу даних, що використовується для зберігання і управління даними. Ця система відзначається своєю гнучкою схемою та використанням мови запитів на основі JSON. Популярні компанії, такі як Facebook та Google, використовують MongoDB у своєму виробництві, а старі гравці, зокрема SAP та Royal Bank of Scotland, також визнали її переваги.

Однією з ключових характеристик MongoDB є те, що вона відноситься до NoSQL баз даних. Відмінність полягає у тому, що NoSQL бази даних не мають традиційної реляційної структури, а замість цього надають можливість горизонтального масштабування. Горизонтальне масштабування в контексті баз даних означає можливість розширення системи за рахунок додавання нових серверів або вузлів у мережу. MongoDB, як NoSQL база даних, відрізняється своєю здатністю ефективно горизонтально масштабуватися, розподіляючи завдання між різними серверами або вузлами. Це надає гнучкість у роботі з великими обсягами даних або в умовах зростання потреб у ресурсах. Головна перевага MongoDB полягає в його здатності до ефективного горизонтального масштабування, що робить його популярним в сучасних системах обробки даних.

Це забезпечує високу масштабованість та гнучкість системи. Окрім того, MongoDB дозволяє програмістам працювати з даними, не замислюючись про традиційні схеми та структури таблиць, що є характерною особливістю NoSQL підходу.

Таким чином, MongoDB виступає важливою складовою стеку MERN, забезпечуючи потужний та гнучкий інструмент для роботи з даними в веб-розробці.

### **2.2.4 Express: Back-End Framework**

Express - це фреймворк для Node.js, який спрощує розробку серверної частини веб-додатків. Заснований на Node.js, він надає зручний та ефективний спосіб створення веб-серверів і обробки HTTP-запитів.

Основні характеристики та концепції Express включають:

1. *Маршрутизація:* Express дозволяє визначати маршрути, що визначають, як обробляти різні види HTTP-запитів (GET, POST, тощо) для конкретних URL-шляхів.
2. *Проміжне програмне забезпечення (Middleware):* Express підтримує використання проміжного програмного забезпечення, що є функціями, які викликаються перед тим, як оброблюється сам запит. Це дозволяє виконувати різноманітні дії, такі як обробка журналів, автентифікація, обробка помилок тощо.
3. *Шаблонізатори:* Хоча Express не має вбудованого шаблонізатора, він легко інтегрується з різними шаблонізаторами, такими як Pug або EJS, які використовуються для генерації HTML.
4. *Статичні файли:* Express може обслуговувати статичні файли, такі як зображення, таблиці стилів та JavaScript, для покращення продуктивності та завантаження веб-сторінок.
5. *Робота з HTTP-відповідями та запитами:* Використовуючи Express, можна легко встановлювати статус коди, встановлювати заголовки, робити перенаправлення та інші дії з HTTP-відповідями.
6. *API розширення:* Завдяки великій кількості розширень та плагінів, Express може бути розширено для різних завдань і вимог.

Express забезпечує зручний і лаконічний спосіб створення веб-серверів, особливо в контексті створення односторінкових додатків (SPA), де генерація динамічного контенту відбувається на стороні клієнта.

### **2.2.5 React: Front-End Framework**

React - це бібліотека для створення інтерфейсів користувача, яка була розроблена компанією Facebook. Однією з ключових концепцій в React є компонентна структура. Все в React є компонентом - це відокремлений, самодостатній блок інтерфейсу, який може бути використаний для побудови більших і складніших інтерфейсів.

Одним з визначних аспектів React є використання віртуального DOM (Document Object Model). React використовує внутрішню модель віртуального DOM для ефективного відображення змін в реальному DOM. Це призводить до покращеної продуктивності додатків, оскільки зменшується кількість операцій маніпулювання DOM.

Ще однією важливою концепцією є JSX (JavaScript XML), розширення синтаксису JavaScript, яке дозволяє писати HTML-подібний код прямо в JavaScript. Це полегшує створення компонентів та розуміння їх структури.

React також часто використовується для створення односторінкових додатків (SPA), де весь контент завантажується один раз, а зміни відбуваються без перезавантаження сторінки.

Один з ключових аспектів роботи з React - використання стану (state) та властивостей (props). Компоненти можуть мати власний стан для зберігання внутрішніх даних та приймати властивості від батьківських компонентів.

React пропагує концепцію реактивного програмування, де зміни в стані або властивостях компонентів призводять до автоматичного оновлення відповідної частини інтерфейсу.

Композиція та ієрархія компонентів є ще однією важливою рисою React. Компоненти можна комбінувати та вкладати один в одного, щоб утворювати складніші структури інтерфейсу.

У цілому, React є потужним інструментом для розробки веб-додатків, що відзначається своєю ефективністю та легкістю використання.

### **2.2.6 Node.js: середовище виконання JS**

Node.js - це вільна, відкрита платформа на основі JavaScript, яка дозволяє виконувати код на стороні сервера. Основною ідеєю Node.js є можливість використання JavaScript для програмування на сервері, а не тільки на клієнтському боці, як це було традиційно.

Однією з ключових особливостей Node.js є його асинхронний та спрямований на події характер. Він використовує однопоточну модель подій, що дозволяє обробляти багато запитів одночасно без створення багатьох потоків або процесів. Це дозволяє Node.js використовуватися для створення ефективних та масштабованих додатків, особливо тих, які вимагають великої кількості одночасних підключень, таких як чати чи стрімінгові сервіси.

Node.js базується на рушію V8, який розроблений Google для використання у браузері Chrome. Використання V8 дозволяє Node.js виконувати високоефективний JavaScript-код.

Ще однією позитивною рисою Node.js є широкий набір модулів, які зазвичай розповсюджуються за допомогою npm, менеджера пакетів Node.js. Це спрощує розробку, оскільки розробники можуть використовувати готові модулі для реалізації різних функцій.

З Node.js зазвичай використовують веб-сервери, API сервери, мікросервіси, а також для реалізації інструментів командного рядка та серверних скриптів. Він дозволяє розробникам використовувати єдину мову програмування (JavaScript) для реалізації інтерфейсу користувача, серверної логіки та взаємодії з базою даних, що полегшує розробку та підтримку коду.

# **3 Структура бази даних**

Щоб розв’язати поставлену мною задачу, необхідно проаналізувати, який вигляд буде мати база даних. В даній роботі я використовуватиму таку систему керування базами даних, як MongoDB. MongoDB відноситься до NoSQL баз даних.

NoSQL – база даних, яка має відмінний підхід зберігання даних від механізму таблиць-відношень в реляційних базах даних.

NoSQL бази даних поділяють на такі типи:

1. *Документно-орієнтовані* - тип бази даних, який призначений для зберігання JSON, BSON та документів інших форматів, де дані представлені рядком або записом і мають формат ключ-значення [3].
2. *Ключ-значення* – найпростіший тип NoSQL баз даних, да кожен елемент зберігається в базі у вигляді пари ключ-значення [3]. Найкраще такий тип використовувати для тимчасового зберігання якихось простих об’єктів (наприклад кеш).
3. Стовпчик – нереляційна база даних, де дані зберігаються у таблицях, рядках, динамічних стовпцях[2]. Вона може стати в нагоді, коли система вимагає зберігання великого обсягу інформації.
4. Граф – мережева база даних, де вузли представляють елемент даних, а ребра являє собою зв’язок між двома вузлами [3].

MongoDB відноситься до документно-орієнтованих систем керування базами даних (СКБД). Тут складовими елементами є колекцій, в той час як реляційні бази даних складаються з таблиць.

В межах даної веб-аплікації було створено наступні колекції:

* 1. User;
  2. UserInfo;
  3. Feedback;
  4. HelpRequest;
  5. CharityProject;
  6. WarriorRehabilitation.

Колекція *User* містить інформацію про користувача: імейл користувача, пароль.

Колекція *UserInfo* містить інформацію про ім’я та прізвище користувача, його номер телефону, соціальні мережі та тип.

Колекція *Feedback* містить інформацію про зворотній зв’язок: самий коментар та інші типи зв’язку з користувачем.

Колекція *CharityProject* містить інформацію про благодійний проект: назва, детальний опис, категорія, локація, інші типи зв’язку з користувачем.

Колекція *WarriorRehabilitation* містить інформацію про оголошення щодо лікування військових: ім’я, військова частина, локація, військова історія, історія лікування, сума, яку необхідно зібрати, реквізити банку та контакти.

Колекція *HelpRequest* містить інформацію про прохання/пропозиції матеріальної допомоги: назва, опис, локація, категорія, тип, інші типи зв’язку з користувачем.

ER-діаграма дасть можливість краще зрозуміти, як саме виглядає база даних для нашої веб-аплікації ***(рисунок 3.1).***

A screenshot of a computer

Description automatically generated

***Рисунок 3.1*** – ER Діаграма

# **4 Розробка**

## **4.1 Авторизація**

Для даної системи важлива точка входу. Тому для початку було розроблено вікно автентифікації та реєстрації, щоб зберігати дані користувача та його діяльність підчас користування веб-додатком.

Для всіх незареєстрованих користувачів при вході на сайт доступна  можливість дізнатись інформацію про даний Web застосунок, a також можливий перегляд запропонованих варіантів допомоги нашій армії.

     Навігаційна панель включає кнопку Вхід, за допомогою яких користувач зможе увійти на сайт або зареєструватись відповідно.

     Якщо ж ви вже зареєстровані, слід лише ввести свою електронну адресу та пароль ***(рисунок 4.1)***. В іншому ж випадку, ви повинні зареєструватись – за допомогою імені, електронної адреси, номеру телефону та паролю ***(рисунок 4.2)***. Важливим моментом є те, що паролі шифруються.

A screenshot of a login form

Description automatically generated with medium confidence

***Рисунок 4.1*** – Форма для  логування

**A screenshot of a computer

Description automatically generated**

***Рисунок 4.2*** – Форма для реєстрації

Важливо також згадати, що авторизація базується на токенах. Авторизація на основі токена – це протокол, який дозволяє користувачам підтвердити свою особу, а натомість отримати унікальний токен доступу, протягом терміну дії якого користувачі отримують доступ до веб-сайту або додатку, замість того, щоб повторно вводити облікові дані щоразу, коли вони повертаються на ту саму веб-сторінку[4]. Якщо реєстрація пройшла вдало, то користувач отримує можливість створювати різні запити, а також зявляється доступ до особистого кабінету.

## **4.2 Головна сторінка**

Для усіх типів користувачів є доступною головна сторінка, де розміщена інформація про запропоновані типи допомоги ***(рисунок 4.3).*** Користувач може перейти на сторінку з інформацією про фонди та реквізити, або ж ознайомитись ближче з додатком та як він працює, скориставшись кнопкою “Про нас”. Та обрати тип допомоги, який його цікавить:

1. Благодійні проєкти;
2. Реабілітація військових;
3. Запити матеріальної допомоги;

A screenshot of a website

Description automatically generated with low confidence

***Рисунок 4.3*** – Головна сторінка

## **4.3 Про нас**

Щоб зробити процес роботи даного застосунку прозорим, було розроблено сторінку “Хто ми”, яка спростить користувачу навігацію по застосунку, а також розкаже принципи його роботи та як тим скористатись ***(рисунок 4.4).***

A screenshot of a website

Description automatically generated with medium confidence

***Рисунок 4.4*** – Хто ми

## **4.4 Зворотній зв’язок**

Сторінка зі зворотнім зв'язком є важливою складовою веб-застосунку для допомоги військовим. Вона надає користувачам можливість зв'язатися з адміністрацією чи відповідальними особами та надіслати свої запити, коментарі, пропозиції або будь-яку іншу інформацію, що стосується веб-застосунку чи допомоги військовим. Користувачу необхідно заповнити свій коментар і вказати засоби зв'язку, такі як номер телефону чи соціальні мережі, за допомогою яких адміністратор зможе сконтактуватись для подальшої перевірки інформації ***(рисунок 4.5).***

Створення такої сторінки дозволяє забезпечити відкритий канал спілкування та взаємодії між користувачами та адміністрацією веб-застосунку, що підвищує рівень задоволеності.

***A screenshot of a computer

Description automatically generated***

***Рисунок 4.5*** – Зворотній зв’язок

## **4.5 Фонди та реквізити**

У користувача буде зручний доступ до інформації про різні відомі українські фонди, які займаються наданням допомоги українській армії. Основна мета полягає у забезпеченні необхідної інформації про фонди та їх реквізити для здійснення пожертв. Кожен фонд представлений з вказанням його назви, опису мети та завдань фонду, а також надані необхідні контактні дані та реквізитів для пожертвувань ***(рисунок 4.6)***.

Список представлених фондів:

1. Фонд “Повернись живим”
2. Фонд “АРМІЯ SOS”
3. ДК “Укрспецпорт”
4. Благодійний фонд Сергія Притули
5. Фонд допомоги “Крила Фенікса”
6. Рахунок Національного банку України

A screenshot of a computer

Description automatically generated with medium confidence

***Рисунок 4.6*** – Фонди та реквізити

## **4.6 Благодійні проєкти**

### **4.6.1 Перегляд благодійних проєктів**

Для користувачів, яких цікавлять відкриті благодійні проєкти, буде змога їх переглянути та ознайомитись з детальним описом. Для кожного проєкту надано банківські реквізити, необхідні для здійснення пожертви ***(рисунок 4.7)***. Щоб зробити перегляд та пошук інформації зручнішим, користувач може відфільтрувати інформацію за локацією та категорією.

*Опції можливих категорій:*

1. Автомобілі;
2. Бронежилети;
3. Рації;
4. Тепловізори;
5. Медикаменти;
6. Військовий одяг;
7. Дрони;
8. Інше.

A screenshot of a website

Description automatically generated with medium confidence

***Рисунок 4.7*** – Благодійні проєкти

A screenshot of a phone

Description automatically generated with low confidence

***Рисунок 4.8*** – Благодійні проєкти/Детальна інформація

### **4.6.2 Подати заявку**

Якщо ж користувач бажає подати заявку, щоб опублікувати свій благодійний проєкт, йому необхідно заповнити форму з наступними полями ***(рисунок 4.9)***:

1. Назва (коротко про суть);
2. Опис проєкту (детальний опис);
3. Реквізити банку;
4. Категорія;
5. Документи;
6. Фотографія;
7. Інші способи зв’язку

**A screenshot of a computer

Description automatically generated**

***Рисунок 4.9*** – Форма “Благодійний проєкт”

## **4.7 Реабілітація військових**

### **4.7.1 Перегляд зборів коштів на реабілітацію військових**

Одним з найважливіших функціоналів даного веб застосунку, є збір коштів на допомогу військовим. На цій сторінці користувач зможе ознайомитись з військовим та його історією поранення ***(рисунок 4.10)***. А також отримати реквізити, на які можна здійснити переказ. Щоб пошук інформації був зручніший, на сторінці є поле пошуку, де слід лише вказати імя та прізвище військового.

A screenshot of a web page

Description automatically generated with medium confidence

***Рисунок 4.10*** – Реабілітація військового

A screenshot of a computer

Description automatically generated with medium confidence

***Рисунок 4.11*** – Реабілітація військового/Детальна інформація

### **4.7.2 Подати заявку, якщо Ви військовий**

Також користувач може опублікувати власний збір на допомогу військовому. Для цього необхідно заповнити форму з наступними полями ***(рисунок 4.12)***:

1. Прізвище, Ім’я, По-батькові;
2. Локація;
3. Військова частина;
4. Військова історія та поранення;
5. Історія лікування;
6. Сума, яку потрібно зібрати;
7. Реквізити банку;
8. Документи;
9. Фотографія;
10. Інші способи зв’язку

**A screenshot of a computer

Description automatically generated**

***Рисунок 4.12*** – Форма “Реабілітації військового”

## **4.8 Запити матеріальної допомоги**

### **4.8.1 Перегляд запитів/пропозицій матеріальної допомоги**

Якщо хтось з користувачів має бажання запропонувати свій спосіб допомоги, або ж це людина, якій необхідна допомога, варто скористатись категорією “Запити матеріальної допомоги” ***(рисунок 4.13, 4.14)***. Для перегляду запитів про допомогу чи пропозицій допомоги потрібно скористатись відповідними кнопками “Пропозиції” та “Прохання”. На сторінці також можлива фільтрація за категорією чи локацією.

A screenshot of a website

Description automatically generated with low confidence

***Рисунок 4.13*** – Запити матеріальної допомоги/Пропозиції

A screenshot of a computer

Description automatically generated with low confidence

***Рисунок 4.14*** – Запити матеріальної допомоги/Прохання

### **4.8.2 Надаю допомогу/ Прошу допомоги**

Щоб опублікувати своє оголошення, необхідно заповнити форму, скориставшись кнопками “Надаю допомогу” чи “Потрібна допомога”. З формами для заповнення можна ознайомитись на ***рисунку 4.15*** та ***рисунку 4.16.*** ***A screenshot of a computer

Description automatically generated***

***Рисунок 4.15*** – Форма “Надаю допомогу”

A screenshot of a computer

Description automatically generated

***Рисунок 4.16*** – Форма “Прошу допомоги”

## **4.9 Особистий кабінет**

### **4.9.1 Персональна інформація**

Кожний зарежстрований користувач буде мати створений особистий кабінет, де міститиметься його персональна інформація:

1. Ім’я та прізвище;
2. Електронна пошта;
3. Номер телефону;
4. Соціальні мережі.

За потреби, цю інформацію користувач зможе її відредагувати ***(рисунок 4.17)***. А також вийти чи видалити акаунт зовсім.

A screenshot of a computer

Description automatically generated

***Рисунок 4.17***– Оновлення персональних даних

Також користувач має можливіть змінити пароль ***(рисунок 4.18)***.

**A computer screen shot of a computer screen

Description automatically generated**

***Рисунок 4.18*** – Зміна паролю

### **4.9.2 Створені запити**

Усі створені користувачем запити зберігаються в його особистому кабінеті. Запити поділені за наступними категоріями:

1. Благодійні проєкти;
2. Реабілітація військових;
3. Запити матеріальної допомоги;

Користувач має можливість змінювати текст, фотографію та видаляти/додавати документи.

**A screenshot of a computer

Description automatically generated**

***Рисунок 4.19*** – Створені запити матеріальної допомоги

## **4.10 Адмін портал**

Усі створені користувачами запити будуть проходити детальну перевірку адміністратором і тільки після прийняття заявки будуть опубліковані в додатку ***(рисунок 4.20)***.

Саме тому на адмін порталі було створено чотири категорії запитів:

1. Благодійні проєкти;
2. Реабілітація військових;
3. Запити матеріальної допомоги;
4. Зворотній зв’язок.

Адміністратор має можливість дізнатись детальну інформацію стосовно кожного запиту і прийняти чи відхилити його відповідно.

Для зручності, адміністратор має зразу доступ і до всього функціоналу, який доступний звичайному користувачу.

A screenshot of a website

Description automatically generated with low confidence

***Рисунок 4.20*** – Адмін портал

# **Висновок**

Дописати: - 11.27

* Юз Кейc
* UI частина: фото нового функціоналу (1)

Висновки – 11.27

Зміст + Форматування – 11/28

# **Список використаних джерел**

1. Project Management Institute, The standard for project management and guide to the project management body of knowledge (PMBOK guide). Seventh Edition ANSI-PMI 99-001-2021 190p
2. What Is NoSQL? NoSQL Databases Explained. MongoDB. URL: [What Is NoSQL? NoSQL Databases Explained | MongoDB](https://www.mongodb.com/nosql-explained) (date of access: 22.11.2023).
3. Keita Z. NoSQL Databases - Types of NoSQL Databases and How to Use Them. *Learn Data Science and AI Online | DataCamp*. URL: <https://www.datacamp.com/blog/nosql-databases-what-every-data-scientist-needs-to-know> (date of access: 22.11.2023).
4. [What Is Token-Based Authentication? | Okta](https://www.okta.com/identity-101/what-is-token-based-authentication/)

1. [Листи військовим - Ukraїner (ukrainer.net)](https://ukrainer.net/lysty-viyskovym/)