

Міністерство освіти і науки України
Рівненський державний гуманітарний університет
Кафедра інформаційних технологій та моделювання

Кваліфікаційна робота

за освітнім ступенем «бакалавр»

на тему:

« Система супроводу та тренування надання першої медичної допомоги »

Виконав:

здобувач ІV курсу

групи ІІЗ-41

спеціальності 121 «Інженерія
програмного забезпечення»

Здобувач Коханевич Віталій Михайлович

Науковий керівник: к.п.н., доц. Петренко С. В.

Рівне – 2025

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ НАВЧАННЯ ПЕРШІЙ МЕДИЧНІЙ ДОПОМОЗІ З ВИКОРИСТАННЯМ ІНТЕРАКТИВНИХ ТЕХНОЛОГІЙ	6
1.1 Актуальність і значення навчання першій медичній допомозі	6
1.2 Основні принципи надання першої медичної допомоги	7
1.3 Сучасні методи навчання навичкам домедичної допомоги	8
1.4. Аналіз цифрових рішень і вебплатформ у сфері медичного навчання	11
1.5. Роль інтерактивних технологій у підготовці до екстрених ситуацій	14
РОЗДІЛ 2. ПІДХОДИ ДО РОЗРОБКИ ОСВІТНІХ ВЕБСИСТЕМ	16
2.1. Цифрова трансформація освіти: загальні підходи та виклики	16
2.2. Вимоги до освітніх систем у сфері домедичної допомоги	17
2.3. Психолого-педагогічні принципи розробки інтерактивних платформ	18
2.4. Роль структури, зручності та доступності контенту в навчанні	20
2.5. Інтерактивність і мотиваційні елементи у вебдодатках	21
РОЗДІЛ 3. РОЗРОБКА ТА РЕАЛІЗАЦІЯ НАВЧАЛЬНОГО ВЕБДОДАТКА	23
3.1 Архітектура й логіка побудови вебдодатка Rescuers Life	23
3.2 Технологічний стек клієнтської частини	24
3.3 Керування контентом через Hygraph CMS	30
3.4 Помічник у реальному часі на базі OpenAI API	33
3.5 Відеоуроки та навчальні матеріали	34
3.6 Перевірка знань (інтерактивні тести)	37
3.7 CI/CD та інфраструктура розгортання	38
3.10 Інтерактивна карта допомоги	42
3.11 Оцінка ефективності та подальший розвиток	45
ВИСНОВКИ	48
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	50
Додаток А Ключові зразки коду застосунку	53
Додаток Б Тести (Рівень складності: Середній)	72
Додаток В Посилання на Github	75
Додаток Г Динамічний UX/UI веб-додатку Rescuers.life	76

ВСТУП

Актуальність дослідження обумовлена зростаючою потребою в підвищенні рівня обізнаності та підготовки громадян до надання першої медичної допомоги. За даними Всесвітньої організації охорони здоров'я (ВООЗ), своєчасне надання допомоги може суттєво зменшити рівень смертності та полегшити відновлення постраждалих. Однак у суспільстві відчувається брак знань та навичок щодо ефективного надання першої допомоги в екстрених ситуаціях. Багато людей, не маючи необхідної підготовки, у критичні моменти відчувають невпевненість або діють неправильно, що знижує шанси на виживання постраждалих. Тому розробка системи, яка дозволить користувачам навчатися надавати першу медичну допомогу через вебдодаток, має значну суспільну важливість. Ця система буде корисною для медичних працівників, волонтерів, студентів та всіх, хто прагне розширити свої знання з першої допомоги.

Мета дослідження полягає у створенні інтерактивної вебсистеми для тренування навичок надання першої медичної допомоги. Дослідження спрямоване на розробку, тестування та впровадження вебдодатка, який дозволить користувачам навчатися та відпрацьовувати основні прийоми першої допомоги за допомогою інтерактивних вправ, відеоуроків, практичних завдань, а також включатиме інтегрованого помічника, що надає підказки в реальному часі, і карту з відображенням найближчих медичних пунктів.

Завдання дослідження

Для досягнення сформульованої мети дослідження необхідно виконати такі завдання:

1. проаналізувати сучасні підходи до надання першої медичної допомоги та існуючі цифрові методи навчання;
2. спроектувати структуру вебдодатка з урахуванням принципів зручності, доступності та ефективного засвоєння матеріалу;

3. розробити вебдодаток, що включає навчальні вправи, відеоуроки, інтерактивну карту, теоретичні матеріали та помічника для підтримки користувача.
4. реалізувати основний функціонал вебдодатка з використанням фреймворку Vue та інших бібліотек, зокрема інтерактивного помічника для дій у реальному часі та мапи з відображенням найближчих медичних пунктів;
5. провести тестування вебдодатка на предмет зручності користування та ефективності навчання;
6. обґрунтувати актуальність і значущість запропонованого рішення для підготовки користувачів до надання першої медичної допомоги.

Об'єктом дослідження є вебдодаток, призначений для навчання навичкам надання першої медичної допомоги з використанням інтерактивних технологій. Додаток включає теоретичні матеріали, відеоуроки, тести, карту з розташуванням найближчих медичних пунктів та інтегрованого помічника для надання підказок у реальному часі. Дослідження зосереджено на розробці та вдосконаленні цього інструменту з метою підвищення ефективності навчання та зручності користування..

Предметом дослідження є процес навчання та практичні методи відпрацювання навичок надання першої медичної допомоги за допомогою інтерактивного вебдодатка. Особлива увага приділяється тому, як цифрові технології, такі як віртуальний помічник, тестові модулі, відеоуроки та геолокаційні сервіси, можуть покращити підготовку користувачів до дій в екстрених ситуаціях.

Методи дослідження

Для досягнення цілей дослідження були використані такі основні методи:

Теоретичні методи: Включають аналіз наукової літератури для визначення існуючих підходів до навчання першій медичній допомозі, вивчення сучасних тенденцій у використанні інтерактивних технологій у навчанні, а також синтез наявної інформації для розробки методологічної бази дослідження.

Емпіричні методи: Проведення спостережень та експериментів із використанням вебдодатка для оцінки його ефективності, збирання та аналіз

відгуків користувачів, узагальнення отриманих результатів для покращення функціоналу додатка та впровадження його в освітній процес.

Практичне значення дослідження

Результати дослідження можуть бути використані для підвищення рівня обізнаності та практичних навичок у наданні першої медичної допомоги серед широких верств населення, зокрема студентів, медичних працівників-початківців та звичайних громадян. Запропонована система супроводу та тренування сприяє закріпленню теоретичних знань за допомогою інтерактивних вправ, відеоуроків і тестових завдань. Особливу практичну цінність становить інтегрований віртуальний помічник, який надає підказки в реальному часі, а також карта з відображенням найближчих пунктів надання медичної допомоги, що дозволяє користувачам швидко орієнтуватися в екстрених ситуаціях. Розроблена система може бути ефективно використана в навчальних закладах, центрах підготовки волонтерів, під час тренінгів для співробітників різних установ, зокрема тих, що працюють із великими групами людей (школи, аеропорти, громадські установи).

Структура роботи

Дипломна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків. Загальний обсяг роботи складає 49 сторінок (без урахування списку використаних джерел та додатків). У роботі використано 26 джерел, зокрема наукові статті, нормативні документи та електронні ресурси. Робота містить 4 додатки: ключові зразки коду застосунку, приклади тестів середнього рівня, посилання на github та динамічний UX/UI веб-додатку Rescuers.life.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ НАВЧАННЯ ПЕРШІЙ МЕДИЧНІЙ ДОПОМОЗИ З ВИКОРИСТАННЯМ ІНТЕРАКТИВНИХ ТЕХНОЛОГІЙ

1.1 Актуальність і значення навчання першій медичній допомозі

Перша медична допомога - це комплекс термінових заходів, які надаються постраждалому до прибуття кваліфікованої медичної допомоги. Вона має вирішальне значення у випадках травм, серцевих нападів, кровотеч, утоплення, втрати свідомості та інших невідкладних станів. Правильно надана допомога в перші хвилини після інциденту може суттєво підвищити шанси на виживання та зменшити ризик тяжких ускладнень.

На сьогоднішній день у багатьох країнах, зокрема в Україні, рівень підготовки населення до надання першої допомоги залишається недостатнім. За статистикою Червоного Хреста, лише близько 10 – 20% громадян мають необхідні знання та вміння для правильного реагування в надзвичайних ситуаціях. При цьому більшість випадків, де потрібна перша допомога, трапляються в побуті, на роботі або в громадських місцях, де поруч можуть бути звичайні громадяни, а не медики.

Особливої актуальності питання набуває в умовах війни, терористичних загроз, техногенних катастроф та епідемій. У таких ситуаціях швидке реагування на місці події нерідко є єдиним шансом врятувати життя до прибуття професійної медичної допомоги. Крім того, підготовка до надання першої допомоги важлива для працівників освітніх закладів, транспорту, сфери обслуговування, а також волонтерів та батьків.

Традиційні методи навчання першій допомозі (наприклад, очні курси з інструкторами) хоча й ефективні, але мають низку обмежень: доступність, вартість, гнучкість графіку. Саме тому все більшої популярності набувають цифрові інструменти — мобільні додатки, онлайн-платформи та інтерактивні вебдодатки.

Вони дозволяють людям самостійно опанувати необхідні знання та навички у зручному форматі, з використанням відео, візуалізацій, тестів і симуляцій.

Таким чином, створення сучасної вебсистеми для навчання першій медичній допомозі відповідає актуальним соціальним викликам та має важливе значення як з точки зору підвищення безпеки громадян, так і з точки зору розвитку загальної культури відповідального ставлення до життя та здоров'я.

1.2 Основні принципи надання першої медичної допомоги

Надання першої медичної допомоги ґрунтується на низці загальновизнаних принципів, які забезпечують ефективне, безпечне та своєчасне реагування в критичних ситуаціях до прибуття медичних працівників. Дотримання цих принципів дозволяє мінімізувати негативні наслідки для здоров'я постраждалого та суттєво підвищити шанси на його виживання.

Передусім, необхідно подбати про безпеку як самого рятувальника, так і постраждалого. Усі дії повинні відбуватися в умовах, що не загрожують життю — у разі небезпеки слід усунути загрозу або викликати відповідні служби. Далі важливо швидко оцінити стан потерпілого: перевірити наявність свідомості, дихання, пульсу, а також наявність видимих ушкоджень. Це дає змогу правильно визначити, яка допомога є першочерговою — наприклад, зупинка кровотечі, відновлення дихання чи переведення у стабільне положення.

Одним із першочергових кроків у будь-якій ситуації є виклик професійної медичної допомоги. Навіть якщо перша допомога надається якісно, медики повинні прибути якомога швидше, щоб забезпечити повноцінне лікування. Після цього необхідно виконати основні дії: зупинити кровотечу, забезпечити прохідність дихальних шляхів, провести серцево-легеневу реанімацію, обробити рани або іммобілізувати кінцівки. Усі дії повинні відповідати конкретній ситуації та рівню підготовки рятувальника.

Крім фізичної допомоги, надзвичайно важливо надати постраждалому психологічну підтримку — зберігати спокій, впевнено говорити, словесно підтримувати людину, яка перебуває в стані шоку. Це може допомогти знизити паніку та стабілізувати стан до прибуття швидкої. При цьому важливо дотримуватись етичних і правових норм: поважати гідність постраждалого, діяти в межах своєї компетентності та не завдавати шкоди.

Дії повинні бути простими та ефективними — не варто намагатися виконати складні медичні процедури без спеціальної підготовки. Основна мета — стабілізувати стан постраждалого до приїзду медиків.

У сучасній практиці часто застосовуються структуровані алгоритми надання допомоги. Зокрема, алгоритм MARCH, що активно використовується у військовій медицині, пропонує чітку послідовність дій: Massive bleeding (зупинка критичної кровотечі), Airway (забезпечення прохідності дихальних шляхів), Respiration (оцінка та підтримка дихання), Circulation (оцінка кровообігу), Head injury/Hypothermia (контроль стану голови, профілактика переохолодження). Цей алгоритм може бути адаптований і до цивільного середовища, особливо в умовах катастроф, воєнних дій чи інших кризових ситуацій.

Необхідно додержуватись правила - ніколи не можна відмовлятися від надання допомоги потерпілому і вважати її марною, навіть якщо у нього відсутні серцебиття, пульс, дихання. У всіх випадках, зокрема при ураженні електричним струмом, вирішувати питання про доцільність подальших заходів щодо оживлення потерпілого та давати висновок про його смерть має право тільки лікар.[3]

1.3 Сучасні методи навчання навичкам домедичної допомоги

У зв'язку зі зростанням кількості надзвичайних ситуацій в Україні та світі, підготовка населення до надання домедичної допомоги набуває виняткової актуальності. Серед причин — військові дії, техногенні катастрофи, стихійні лиха та

зростання побутових травм. Вкрай важливо, щоб якомога більше громадян володіли базовими знаннями та навичками, необхідними для порятунку життя до прибуття медиків.

Сучасні методи навчання домедичної допомоги поєднують традиційні підходи з інноваційними цифровими рішеннями, що дозволяє охопити широку аудиторію, зробити навчання гнучким, доступним та ефективним.

Лекції залишаються важливим засобом передачі знань, особливо на теоретичному етапі. Їх проводять фахівці — медики, інструктори Червоного Хреста, викладачі медичних закладів. Лекції можуть бути очними або онлайн. Вони дозволяють структурувати знання, отримати відповіді на запитання та почути практичні приклади з реального життя. Семінари часто включають обговорення кейсів, роботу в групах і симуляційні завдання, що сприяє активному засвоєнню матеріалу.

Відеоуроки та інструкційні ролики - цей формат став надзвичайно популярним завдяки доступності та можливості багаторазового перегляду. Відеоуроки можуть пояснювати складні дії покроково — як, наприклад, накласти джгут, стабілізувати потерпілого або виконати СЛР. Вони часто супроводжуються графікою, закадровими поясненнями та демонстрацією на манекенах або акторах. Серед прикладів — безкоштовний курс "Домедична допомога" на платформі Prometheus. Відеоформат особливо ефективний для візуалізації та запам'ятовування дій, а також зручний для самостійного навчання в будь-який час.

Тренінги з інструкторами та демонстрації на манекенах - це найефективніший метод формування практичних навичок. Учасники тренінгів безпосередньо відпрацьовують алгоритми дій під наглядом інструктора, отримують миттєвий зворотний зв'язок та корекцію. Сучасні манекени можуть мати сенсори тиску, положення рук, оцінювати якість СЛР та формувати статистику. Такі тренінги проводяться, зокрема, в Центрах підготовки Червоного Хреста, навчальних закладах МОЗ, університетах.

Мобільні додатки — один із найзручніших інструментів для самостійного навчання та оперативного доступу до інформації в надзвичайних ситуаціях. Наприклад, застосунок "Перша допомога" від Червоного Хреста України містить покрокові інструкції, інфографіку, тести, працює офлайн і адаптований під українські реалії. Застосунок дозволяє повторити алгоритми дій у будь-який момент, навіть без інтернету. Він також має розділи з нагадуваннями, інтерактивними сценаріями дій та функціями самоперевірки.

Вебплатформи забезпечують комплексну підготовку як для широкого загалу, так і для фахівців. Платформи на кшталт firstaid.org.ua, створені у співпраці з МОЗ та Всеукраїнською радою реанімації, містять не лише текстові протоколи, а й інтерактивні елементи — відеолекції, демонстрації, презентації, тести та графічні схеми. Курси адаптовані до міжнародних стандартів (ERC, ILCOR), містять перевірочні блоки та тематичні модулі. Особливу увагу приділено структурі навчального контенту: після кожного розділу користувач проходить тестування, а система фіксує прогрес і надає рекомендації. Окремі вебплатформи інтегрують функцію віртуального супроводу — користувач може отримати консультацію онлайн або у вигляді автоматизованого порадики. Це забезпечує постійний зв'язок і підтримку, навіть якщо людина навчається самостійно. Деякі ресурси мають розділи для викладачів з доступом до методичних матеріалів, а також можливість сертифікації після проходження курсу.

VR/AR-технології (віртуальна та доповнена реальність) Сценарії у віртуальній реальності дозволяють змодельовати дії в екстремальних умовах — ДТП, вибух, бойові дії. Таке занурення створює психологічно наближене до реальності середовище, що критично важливо для підготовки до стресових ситуацій. AR-елементи дають змогу тренуватися вдома, проектуючи інструкції на реальні об'єкти. Наприклад, додатки можуть відображати точне положення рук для СЛР або вказувати місце для накладання джгута.

Гейміфіковані онлайн-курси дозволяють заробляти бали за правильні відповіді, змагатися з іншими учасниками, проходити рівні складності. Це особливо ефективно для молоді. Адаптивні системи аналізують помилки користувача і підлаштовують матеріал — показують додаткові пояснення, відео або нові задачі. Це підвищує мотивацію до навчання та забезпечує індивідуальний підхід.

Загалом, сучасні методи навчання навичкам домедичної допомоги враховують різні стилі сприйняття інформації, створюють інтерактивне середовище та дозволяють поєднувати теорію з практикою. Саме таке комплексне навчання здатне якісно підготувати людину до дій у реальній небезпечній ситуації.

1.4. Аналіз цифрових рішень і вебплатформ у сфері медичного навчання

У сучасному освітньому середовищі цифрові рішення відіграють ключову роль у підготовці населення до надання першої допомоги. Вони дають змогу зробити навчання гнучким, персоналізованим, доступним з будь-якої точки країни — що є критично важливим в умовах війни, внутрішнього переміщення населення, обмеженого доступу до офлайн-ресурсів.

Цифрові платформи для навчання першій допомозі мають кілька очевидних переваг. По-перше, вони доступні цілодобово й із будь-якого пристрою, тож людина може повторити алгоритм навіть у дорозі. По-друге, система підлаштовує матеріал під поточний рівень знань користувача: новачкові пропонує базові інструкції, а досвідченому — поглиблені сценарії. По-третє, інтерактивний формат із тестами, анімаціями та змодельованими ситуаціями значно краще утримує увагу, ніж статичний текст. Усі рекомендації легко оновити, щойно виходять нові протоколи, тому контент завжди лишається актуальним. Додаткову мотивацію забезпечують елементи гейміфікації — бейджі, рейтинги, електронні сертифікати. Нарешті, такі курси без проблем інтегруються з іншими освітніми сервісами, наприклад

електронними щоденниками чи платформою Moodle, що спрощує впровадження в навчальний процес.

Розглянемо приклади різних платформ, які на практиці демонструють ці переваги.

firstaid.org.ua — офіційний освітній ресурс від Всеукраїнської ради реанімації у співпраці з МОЗ. Містить адаптовані до українських реалій протоколи допомоги за стандартами ERC та ILCOR. Присутні відеоуроки, графічні схеми, інтерактивні тренінги, презентації та контрольні питання. Ресурс підтримує навчання для широкої аудиторії: від цивільного населення до викладачів, волонтерів і медичних інструкторів.

Prometheus — український громадський проєкт масових відкритих онлайн-курсів, запущений 2014 року. Головною метою проєкту є безкоштовне надання онлайн-доступу до курсів університетського рівня всім бажаючим, а також надання можливості публікувати та розповсюджувати такі курси провідним викладачам, університетам та компаніям.[6]

Застосунок "Перша допомога" (Червоний Хрест України) — зручний інструмент для швидкого доступу до інструкцій у надзвичайних ситуаціях. Працює офлайн, має категорії ситуацій (опіки, утоплення, кровотеча тощо), вбудовані тести, просту навігацію. Постійно оновлюється відповідно до подій в Україні.

ДСНС України (dsns.gov.ua) — офіційний сайт має розділи з порадами щодо дій під час пожеж, вибухів, обстрілів, техногенних аварій. Хоч і не є вузькоспеціалізованим медичним ресурсом, платформа слугує доповненням до медичних алгоритмів — у ній надано інструкції щодо евакуації, комплектування аптечки, поведіння в стресових ситуаціях.

Інтегровані картографічні рішення (SmartCity) — деякі міста України запроваджують цифрові мапи, що відображають розташування аптек, пунктів домедичної допомоги, дефібриляторів. У кризовій ситуації такі сервіси допомагають швидко зорієнтуватися і знайти ресурси.

Google Classroom, Moodle, Zoom — навчальні платформи, які активно використовуються викладачами для супроводу курсів з домедичної допомоги. Зокрема, інструктори можуть проводити онлайн-заняття, надавати матеріали, перевіряти тести й контролювати прогрес.

Сучасні тенденції розвитку:

1. Інтеграція чат-ботів. Сучасні чат-боти у Telegram або Viber виконують роль миттєвих консультантів, які допомагають користувачам у режимі реального часу. Вони надають покрокові алгоритми дій у критичних ситуаціях: при кровотечі, втраті свідомості, інфаркті. Чат-боти можуть включати візуалізацію (картинки, схеми), аудіопідказки та інтеграцію з гарячими лініями. Їх особливість — миттєвий доступ, простота у використанні та персоналізовані відповіді.
2. Впровадження штучного інтелекту (ШІ). ШІ дозволяє аналізувати відповіді користувача та автоматично адаптувати подачу матеріалу. Наприклад, якщо учень часто помиляється у тестах на тему СЛР, система автоматично підбирає додаткові приклади, повторення теми, альтернативні пояснення або відео. Це створює індивідуальну траєкторію навчання, підвищуючи ефективність засвоєння знань.
3. Комбіноване навчання (blended learning). Поєднання онлайн-навчання з практичними заняттями дозволяє максимально ефективно засвоїти матеріал. Наприклад, користувач проходить онлайн-курс, а після — відвідує очний тренінг з інструктором. Це дає змогу спочатку ознайомитися з теорією, а потім закріпити її на практиці. Такий формат активно використовується в медичних вишах та на волонтерських тренінгах.
4. Віртуальні тренажери з VR/AR-технологіями. Створення віртуального середовища для тренувань дає змогу моделювати складні сценарії — дорожньо-транспортні пригоди, бойові поранення, масові ураження. Користувач у VR-окулярах повинен приймати рішення в реальному часі, виконувати дії у

правильній послідовності, отримувати зворотний зв'язок. Це забезпечує емоційне занурення та формує навички поведінки в умовах стресу. AR-технології дозволяють використовувати додатки прямо в домашніх умовах — наприклад, проектувати інструкції на реальні предмети (манекен, подушку, людину).

Ці тенденції демонструють, що цифрова трансформація освіти в сфері першої допомоги вже відбулася, і подальший її розвиток ґрунтується на гнучкості, адаптивності та орієнтації на користувача. Новітні підходи допомагають не просто навчити, а створити рефлекторну готовність діяти швидко та впевнено.

1.5. Роль інтерактивних технологій у підготовці до екстрених ситуацій

Інтерактивні технології стали невід'ємною складовою сучасного підходу до навчання, особливо у сфері підготовки до дій в екстрених ситуаціях. Їх використання дозволяє не лише засвоювати теоретичні знання, але й активно формувати практичні навички, адаптувати поведінку до стресових умов та швидко реагувати у критичних ситуаціях.

Однією з основних переваг інтерактивних технологій є можливість імітації реальних надзвичайних ситуацій. Завдяки інтерактивним симуляціям, користувач має змогу відпрацьовувати алгоритми дій при опіках, травмах, серцевих нападах, утопленні тощо. Це значно ефективніше, ніж пасивне вивчення текстових інструкцій, оскільки залучає до процесу навчання візуальне сприйняття, моторні навички та логічне мислення.

Важливим інструментом є інтерактивні тести та завдання, які дозволяють не лише перевірити рівень засвоєння матеріалу, а й закріпити правильну послідовність дій. У поєднанні з миттєвим зворотним зв'язком такі елементи формують активне навчальне середовище, де користувач самостійно визначає прогалини у знаннях і може одразу їх заповнити.

Окрему роль відіграють віртуальні помічники — це інтерактивні елементи або голосові/текстові інтерфейси, які можуть супроводжувати користувача в реальному часі. Наприклад, під час симуляції або навіть у справжній ситуації, користувач може отримувати підказки щодо послідовності дій, що допомагає зменшити стрес і уникнути помилок.

Крім того, інтерактивні карти з позначенням медичних пунктів, аптек, дефібриляторів (AED) та інших ресурсів дозволяють користувачеві швидко орієнтуватися в реальному просторі. Це надзвичайно важливо в кризовій ситуації, коли час є критичним чинником.

Інтерактивні технології також сприяють персоналізації навчального процесу. Користувач може проходити матеріал у власному темпі, повторювати складні теми, обирати ті сценарії, які є для нього найактуальнішими (наприклад, допомога дитині, літній людині, травма в побуті чи в ДТП).

Не менш важливим є фактор мотивації, який забезпечується за рахунок гейміфікації: інтерактивні балли, досягнення, рейтинги, рівні складності — все це робить навчання не лише корисним, а й захоплюючим процесом.

РОЗДІЛ 2. ПІДХОДИ ДО РОЗРОБКИ ОСВІТНІХ ВЕБСИСТЕМ

2.1. Цифрова трансформація освіти: загальні підходи та виклики

Цифрова трансформація освіти є ключовим чинником модернізації навчального процесу в усьому світі. Упровадження цифрових технологій у навчання сприяє підвищенню доступності знань, індивідуалізації навчальних траєкторій та розвитку нових форм взаємодії між учнем і викладачем. Вона стала особливо актуальною в умовах пандемії COVID-19, а також у контексті дистанційної освіти, військових загроз і потреби у швидкому перенавчанні населення.

Сучасний освітній процес дедалі частіше переноситься у цифрове середовище — це зумовлює необхідність переосмислення педагогічних стратегій, технічної інфраструктури та підготовки кадрів. Технології дозволяють не лише транслювати лекційний матеріал онлайн, а й будувати повноцінні інтерактивні навчальні середовища, що включають відео, тести, симуляції, форуми, індивідуальні кабінети та систему зворотного зв'язку.

Цифрова трансформація освіти змінює не лише інструменти, а й сам підхід до навчання. По-перше, вона дозволяє зробити навчання більш гнучким — кожен може засвоювати матеріал у зручному для себе темпі, обираючи, коли і як вчитися. Це повністю відповідає ідеї навчання впродовж життя.

Також змінюється роль викладача. Він вже не просто джерело інформації, а радше наставник, який підтримує, направляє та допомагає учням навчатися самостійно.

Традиційні підручники поступово відходять на другий план. Їх замінюють інтерактивні лекції, візуальні матеріали, реальні приклади й практичні завдання, що роблять навчання цікавішим і ближчим до реального життя.

Крім того, сучасна освіта активно використовує цифрові технології — це і хмарні сервіси, і віртуальні класи, і платформи для управління навчанням, які дозволяють відстежувати успіхи та адаптувати процес під кожного учня.

Попри численні переваги, цифрова трансформація освіти стикається і з низкою серйозних викликів. Один із головних — це цифрова нерівність: не всі мають доступ до якісного Інтернету чи сучасної техніки, що створює бар'єри для участі в навчальному процесі.

Ще однією проблемою є недостатній рівень цифрової грамотності. І учні, і викладачі не завжди володіють необхідними навичками для ефективного використання цифрових інструментів.

Також варто згадати про інформаційне перевантаження. Коли джерел занадто багато, але вони погано структуровані, це може тільки заплутати і ускладнити засвоєння матеріалу.

Окрему увагу потрібно приділяти безпеці. Онлайн-платформи повинні забезпечувати надійний захист персональних даних та навчальних результатів, адже витоки інформації можуть мати серйозні наслідки.

2.2. Вимоги до освітніх систем у сфері домедичної допомоги

Системи навчання домедичної допомоги повинні враховувати практичний характер цієї підготовки, високу відповідальність і потребу в швидкому реагуванні. Насамперед вони мають подавати матеріал так, щоб користувач одразу розумів, як діяти на місці події: у курсах обов'язково повинні бути чіткі алгоритми, покрокові інструкції та реалістичні сценарії — від дорожньо-транспортної пригоди до зупинки серця чи масивної кровотечі.

Ефективність значно зростає, коли інформація подається візуально: відео, схеми, анімації та інтерактивні симуляції допомагають відпрацювати навички у наближених до реальності умовах. Водночас інтерфейс платформи має залишатися

інтуїтивним для всіх — від школяра чи водія до досвідченого волонтера, а сам зміст повинен легко адаптуватися під різний рівень підготовки.

Оскільки вчитися доводиться «на ходу», доступ із телефона, підтримка офлайн-режиму й адаптивний дизайн стають обов'язковими. Важливе місце займає й система перевірки знань: короткі тести, ситуаційні завдання та механізми самоконтролю дають змогу одразу побачити свої прогалини й отримати корисні підказки.

Увесь контент повинен опиратися на актуальні міжнародні та національні протоколи — ERC, ILCOR, MARCH — і бути адаптованим до українських реалій. Нарешті, не варто забувати про психоемоційну складову: дружнє навчальне середовище, нагадування, поради та елементи гейміфікації підтримують мотивацію й допомагають засвоювати складний матеріал без зайвого стресу.

2.3. Психолого-педагогічні принципи розробки інтерактивних платформ

Розробка сучасних освітніх вебплатформ неможлива без урахування психолого-педагогічних особливостей навчального процесу. Якість цифрового навчання визначається не лише зручністю інтерфейсу чи технічною реалізацією функціоналу, але й тим, наскільки система відповідає закономірностям сприйняття, мислення, уваги, пам'яті та мотивації користувача.

Одним із фундаментальних принципів є принцип мультимедійності. Поєднання тексту з візуальними елементами — зображеннями, відео або анімаціями — сприяє глибшому розумінню й утриманню інформації у довготривалій пам'яті. Когнітивна теорія мультимедійного навчання Річарда Майєра підкреслює важливість одночасної роботи візуального й слухового каналів, що особливо актуально у навчанні практичним навичкам, зокрема в сфері домедичної допомоги.

Ще одним важливим принципом є доступність. Платформа має забезпечити користувачу просту й логічну навігацію, зрозумілу мову подачі матеріалу, адаптацію під різні рівні попередніх знань. Це дозволяє уникнути інформаційного

перевантаження, яке часто знижує ефективність сприйняття. У контексті домедичної допомоги особливо важливо, щоб користувач у критичній ситуації міг швидко згадати й відтворити необхідні дії.

Активність у процесі навчання також має вирішальне значення. Інтерактивна платформа повинна залучати до участі: це може бути виконання завдань, проходження тестів, вибір дій у симульованій ситуації. Така взаємодія не лише підтримує інтерес, але й стимулює глибшу когнітивну обробку інформації. Відповідно до підходів конструктивістської педагогіки, знання найкраще засвоюються тоді, коли учень сам активно їх будує.

Індивідуалізація навчального процесу є ще одним важливим елементом. Користувач повинен мати змогу обирати темп навчання, повертатися до складних тем, отримувати індивідуальні підказки або адаптивні завдання. Це дозволяє формувати персоналізовану траєкторію навчання, що особливо актуально при підготовці широкого кола користувачів — від школярів до медичних працівників.

Не менш значущим є принцип позитивного підкріплення. Мотивація підтримується через використання елементів гейміфікації: система досягнень, бали, значки, відкриття нових рівнів. Такі елементи формують відчуття успіху та просування вперед, що підвищує залучення до навчання. Дослідження показують, що використання мотиваційних індикаторів підвищує завершуваність онлайн-курсів на 15–30%.

Нарешті, необхідно враховувати принцип емоційної безпеки. Користувач має відчувати, що помилки — це частина навчання. Інтерфейс платформи повинен сприяти підтримці, а не покаранню, забезпечуючи дружню реакцію на неправильні дії. Це особливо актуально у сфері підготовки до дій в екстрених умовах, де страх перед помилкою може заблокувати подальше навчання.

Розробка інтерактивної платформи повинна бути міждисциплінарною, поєднуючи педагогіку, психологію, UX/UI-дизайн та інформаційні технології. Саме синтез цих сфер забезпечує створення ефективного й корисного навчального продукту.

2.4. Роль структури, зручності та доступності контенту в навчанні

Одним із найважливіших чинників ефективності навчання у цифровому середовищі є структура, зручність користування та доступність навчального контенту. Від того, наскільки логічно побудована платформа, наскільки легко орієнтуватися у її матеріалах і як швидко можна знайти потрібну інформацію, безпосередньо залежить успішність засвоєння знань користувачем.

У процесі проєктування освітніх систем велика увага приділяється структурованості навчального матеріалу. Контент має бути поділений на зрозумілі тематичні блоки, мікротеми та розділи, які послідовно формують цілісну картину. Такий підхід відповідає когнітивній здатності людини до поетапного сприйняття і зменшує когнітивне навантаження. Подання великої кількості інформації без внутрішньої логіки або без ієрархії ускладнює сприйняття й може знизити рівень мотивації до навчання.

Не менш важливим є зручний інтерфейс платформи. Користувач повинен мати змогу інтуїтивно орієнтуватися на сайті, швидко знаходити потрібні функції, переходити між темами та контролювати свій прогрес. Особливе значення має чітка навігація, помітність ключових елементів, зрозуміла візуальна мова. Досвід користувача (User Experience) має бути побудований таким чином, щоб не відволікати увагу від навчального процесу, а навпаки — сприяти фокусуванню на змісті.

Доступність є окремим критично важливим аспектом. Це поняття охоплює як технічну доступність (відсутність помилок, підтримка різних пристроїв і браузерів, адаптивність), так і соціальну — забезпечення можливості користування для людей із різними рівнями підготовки, когнітивними особливостями або навіть обмеженнями здоров'я. У цьому контексті важливо застосовувати принципи інклюзивного дизайну, наприклад, підтримку шрифтів, які легко читаються, контрастність елементів, доступність для екранних читачів тощо.

Окремо варто зазначити важливість швидкого доступу до ключової інформації в умовах стресу. Для навчання домедичній допомозі користувач має миттєво

знаходити алгоритми дій або інструкції — це може стати критично важливим у реальних надзвичайних ситуаціях. Саме тому платформа має містити функцію швидкого пошуку або окремі «екстрені» сторінки з короткими, лаконічними підказками.

Також до аспекту доступності належить і мовна політика. Важливо, щоб матеріали були подані державною мовою, з можливістю дублюю іншими мовами у випадку мультинаціональної аудиторії. Прості, чіткі формулювання сприяють кращому розумінню навіть серед тих, хто не має спеціальної медичної підготовки. Таким чином, продумана структура, зручність взаємодії та загальна доступність — це не лише додаткові переваги освітнього вебдодатка, а його критично необхідні складові. Їхня реалізація напряду впливає на результативність навчання, рівень залучення користувачів і здатність ефективно засвоїти та застосувати знання у реальних умовах.

2.5. Інтерактивність і мотиваційні елементи у вебдодатках

Інтерактивність є однією з найважливіших характеристик сучасних освітніх вебдодатків, яка безпосередньо впливає на глибину засвоєння матеріалу, рівень залучення користувача та тривалість уваги до навчального процесу. На відміну від пасивного споживання інформації у форматі лекцій або підручників, інтерактивне середовище забезпечує активну участь користувача у формуванні знань.

Під інтерактивністю у вебосвіті розуміють динамічну взаємодію між системою та користувачем, що виявляється у виконанні завдань, реагуванні на зміни в інтерфейсі, виборі сценаріїв, проходженні тестів або симуляцій. Цей тип взаємодії підвищує когнітивну активність, дозволяє самостійно приймати рішення, моделювати поведінку у конкретних ситуаціях, зокрема в кризових умовах — таких, як надання домедичної допомоги.

Інтерактивні елементи сприяють формуванню навичок через імітацію реального досвіду: наприклад, моделювання серцево-легеневої реанімації або зупинки кровотечі

в симульованих ситуаціях. У платформах, що спеціалізуються на медичному навчанні, активно використовуються системи з віртуальними помічниками, покроковими сценаріями, таймерами, візуальними індикаторами правильності дій. Це забезпечує негайний зворотний зв'язок, що значно пришвидшує процес навчання та коригування помилок.

Особливу роль у підтриманні навчальної активності відіграють мотиваційні механізми. Серед найпоширеніших — елементи гейміфікації: бали за правильні відповіді, система досягнень, рейтинги, відкриття нових рівнів чи значків. Вони не лише створюють відчуття прогресу, а й стимулюють користувача до продовження навчання. Дослідження свідчать, що використання гейміфікаційних елементів у вебосвіті здатне підвищити залучення користувачів до 30–40%.

Не менш важливим є персоналізований підхід до мотивації. Платформа може адаптувати повідомлення, поради та завдання відповідно до рівня користувача, його успішності або частоти взаємодії з контентом. Застосування таких механізмів забезпечує глибше занурення у навчальний процес та створює відчуття індивідуальної уваги.

Крім того, важливо враховувати психологічний комфорт користувача. Позитивний зворотний зв'язок, підказки замість покарання, доброзичливий інтерфейс — усе це підсилює внутрішню мотивацію. За теорією самодетермінації (Deci & Ryan, 1985), відчуття автономії, компетентності та залученості є основними чинниками стійкої внутрішньої мотивації. Саме ці аспекти й можуть бути реалізовані через інтерфейс та логіку взаємодії в освітньому вебдодатку.

РОЗДІЛ 3. РОЗРОБКА ТА РЕАЛІЗАЦІЯ НАВЧАЛЬНОГО ВЕБДОДАТКА

3.1 Архітектура й логіка побудови вебдодатка **Rescuers Life**

Систему ми побудували за класичною трирівневою схемою: клієнтський інтерфейс, API-шар і контур CI/CD. Така структура полегшує масштабування, тестування та подальший розвиток проєкту.

На клієнтському рівні інтерфейс працює на Vue 3, а між сторінками користувача перемикає Vue Router. За стилі відповідає Tailwind CSS, тому власних CSS-файлів майже немає, а макети можна збирати дуже швидко. Обмін даними з бекендом іде через Axios; у запитах ми використовуємо глобальні перехоплювачі помилок, а відповіді кешуємо у localStorage. Додаток працює ще й як PWA: Service Worker заздалегідь зберігає статичні ресурси, тож навіть за слабкого інтернету інтерфейс відкривається миттєво. Кожен функціональний блок ізольований у власному Vue-модулі з composables, що робить код зрозумілим і легким у підтримці.

Усе навчальне наповнення зберігаємо в NuGraph і дістаємо через GraphQL, тому клієнт отримує тільки ті поля, які реально потрібні компоненту. Помічник у реальному часі працює через OpenAI API: клієнт формує prompt, сервер генерує відповідь, а фронтенд лише додає розмітку й посилання, перш ніж показати її користувачеві.

Код лежить у GitHub, а GitHub Actions підхоплює кожен pull request: запускає літери, юніт-тести та збірку через Vite. Якщо все проходить без помилок, артефакти одразу деплоються на Vercel, де формується preview-версія. Ми можемо перевірити зміни в «живому» середовищі й, за потреби, одним кліком повернути попередню стабільну збірку.

Такий компонентно-орієнтований дизайн з чітким поділом шарів дає швидкий клієнт, надійну доставку контенту й повну свободу для подальшого розвитку функціоналу.

3.2 Технологічний стек клієнтської частини

Vue (Рис. 3.1) — це сучасний прогресивний JavaScript-фреймворк, який дозволяє легко створювати інтерактивні та швидкі вебзастосунки. Він має низький поріг входження, що робить його ідеальним для як новачків, так і для досвідчених розробників. Завдяки реактивності даних і простій структурі компонентів, розробка в Vue стає швидкою та інтуїтивно зрозумілою.[26]

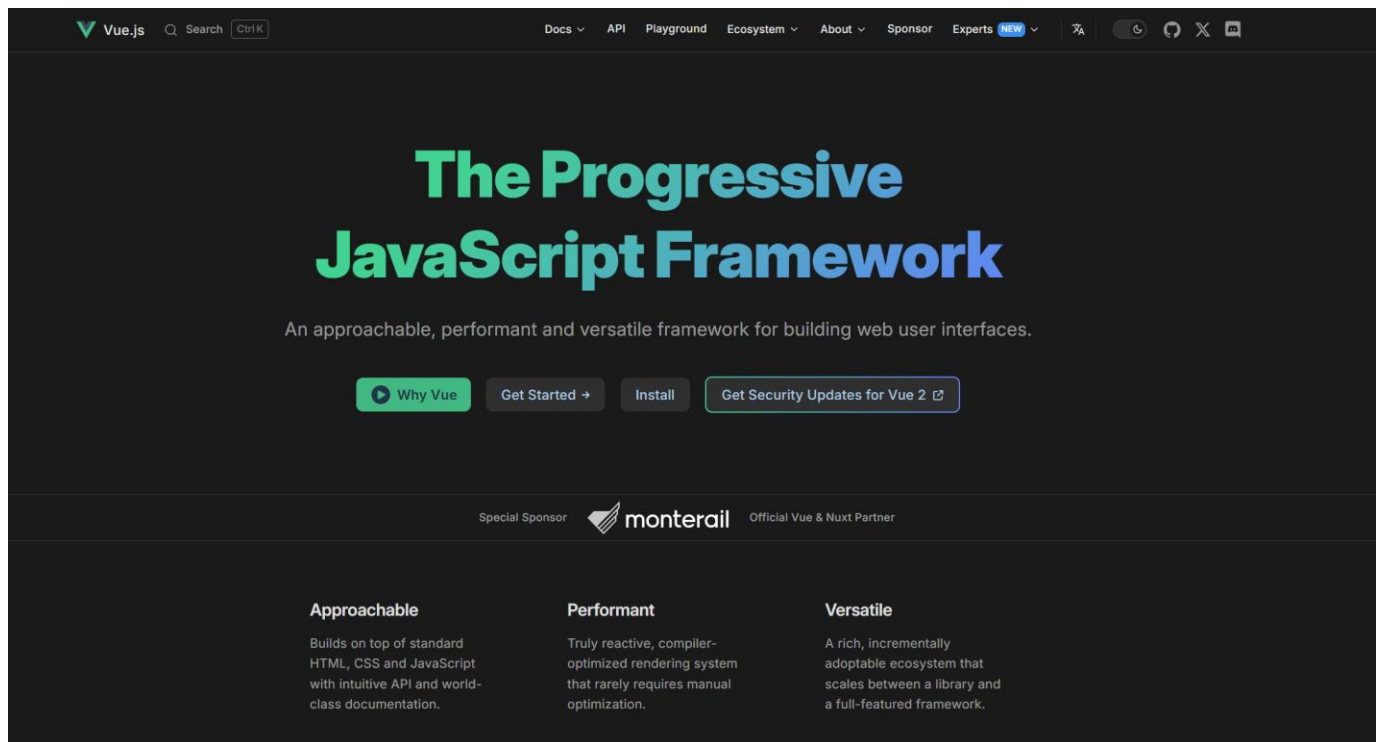


Рисунок 3.1. Головна сторінка Vue

Використовувати Vue варто тому, що він забезпечує чудову продуктивність, підтримує гнучку модульну архітектуру та має активну спільноту. З Vue легко масштабувати проєкти від маленьких сайтів до великих корпоративних систем. Також він прекрасно інтегрується з іншими бібліотеками і сучасними інструментами фронтенду.

Vue забезпечує чудовий баланс між простотою використання та потужністю функціоналу. Його синтаксис зрозумілий і легкий для вивчення, що дозволяє швидко стартувати нові проєкти. При цьому Vue підтримує передові підходи, такі як компоненти, реактивність і модульність.

Однією з головних переваг Vue є його гнучкість. Його можна використовувати як для невеликих інтерактивних елементів на сторінці, так і для створення повноцінних односторінкових застосунків. Завдяки підтримці Composition API розробники отримують більше можливостей для структурування та повторного використання коду.

Vue активно підтримується спільнотою та має велику кількість готових бібліотек і рішень. Це значно пришвидшує розробку і спрощує інтеграцію нових можливостей. Використовуючи Vue, компанії скорочують час виходу продукту на ринок і мінімізують витрати на підтримку.

Ще одна причина обрати Vue — це його висока продуктивність і невеликий розмір ядра. Завдяки оптимізованій роботі з DOM і ефективній обробці змін, застосунки на Vue працюють швидко навіть на слабких пристроях. Це дозволяє створювати зручні та доступні продукти для широкої аудиторії.

Vue Router (Рис. 3.2) - це офіційна бібліотека маршрутизації для фреймворка Vue.js. Вона забезпечує плавну навігацію між різними компонентами без перезавантаження сторінки. Завдяки Vue Router можна легко будувати односторінкові застосунки (SPA), де зміна контенту відбувається миттєво та без втрат продуктивності.

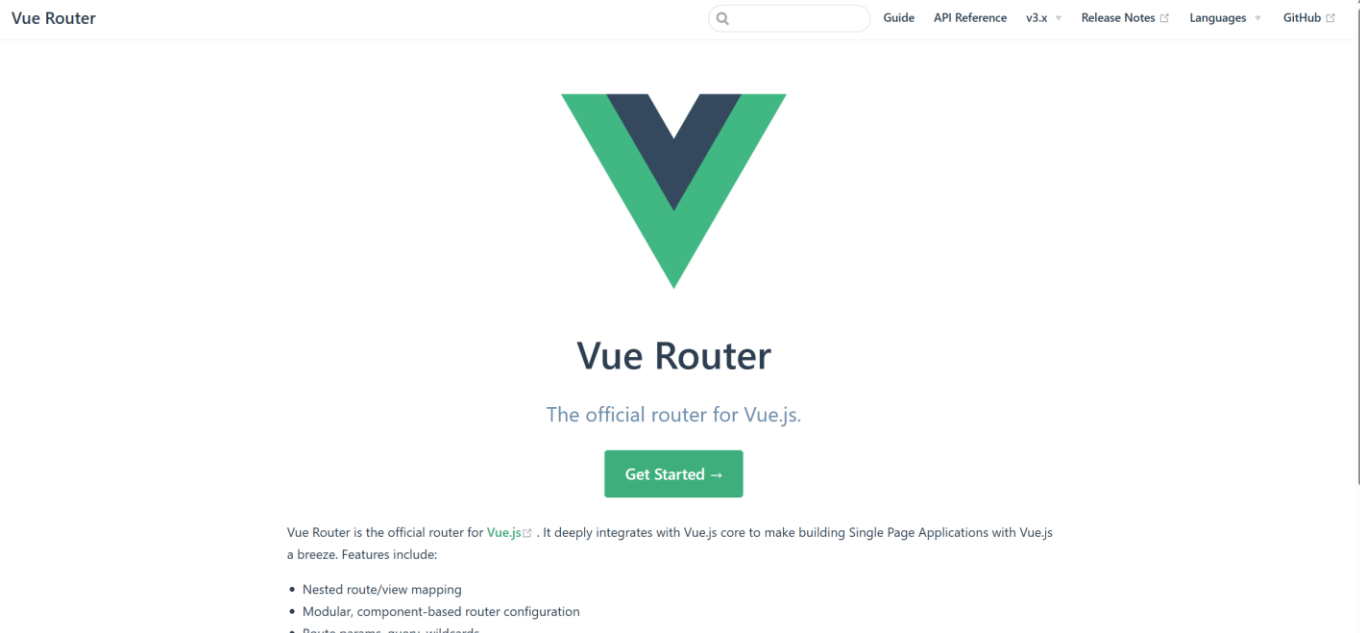


Рисунок 3.2. Головна сторінка Vue Router

Однією з головних переваг Vue Router є підтримка динамічних, вкладених і параметризованих маршрутів. Це дозволяє створювати гнучку та глибоку навігаційну структуру навіть для великих застосунків. Додатково бібліотека підтримує lazy loading, що дає можливість завантажувати компоненти лише тоді, коли вони потрібні, і тим самим зменшувати розмір початкового пакета.

Vue Router також підтримує режим історії браузера через HTML5 History API, завдяки чому URL-адреси виглядають чисто і природно без символів типу "#". Це важливо для SEO-оптимізації та підвищення зручності користувачів. Крім того, бібліотека дозволяє легко реалізувати захист маршрутів за допомогою навігаційних гвардій.

Ще одна важлива перевага Vue Router — простота інтеграції з іншими частинами екосистеми Vue. Налаштування маршрутизації інтуїтивно зрозуміле навіть для новачків, а розширені можливості дають змогу досвідченим розробникам

створювати складні логіки переходів, анімації сторінок та інші покращення взаємодії з користувачем.

Tailwind CSS (Рис. 3.3) — це сучасний утилітарний CSS-фреймворк, який дозволяє швидко створювати адаптивні та стильні інтерфейси без написання великої кількості кастомного CSS. Він надає набір готових класів для управління відступами, кольорами, розмірами, шрифтами та іншими властивостями безпосередньо в HTML-розмітці.

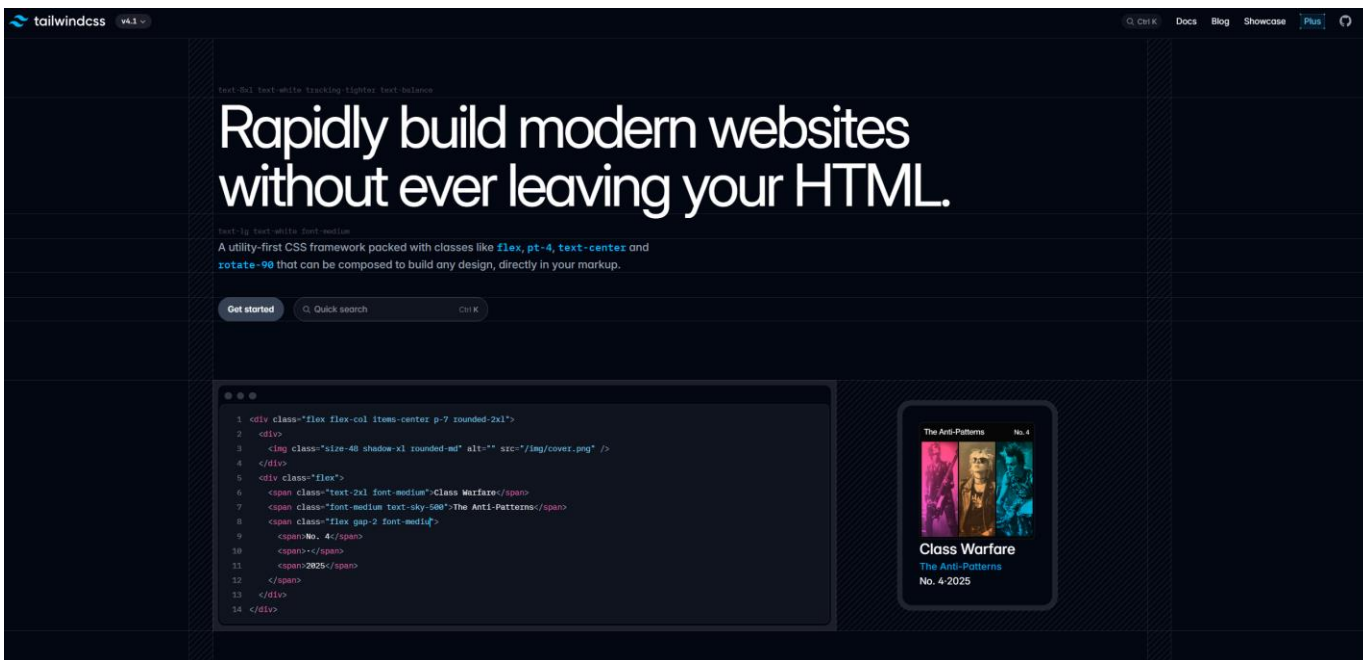


Рисунок 3.3. Головна сторінка Tailwind CSS

Однією з головних переваг Tailwind CSS є висока швидкість розробки. Завдяки системі утиліт кожен елемент можна миттєво стилізувати, не перемикаючись між файлами. Це значно скорочує час розробки інтерфейсів і дозволяє швидко бачити результат прямо у браузері.

Tailwind CSS підтримує повну адаптивність за допомогою вбудованих брейкпоінтів. Крім того, він має потужну систему налаштувань через конфігураційний файл, що дозволяє легко кастомізувати палітру кольорів, шрифти, відступи та інші параметри під потреби конкретного проєкту.

Ще однією сильною стороною Tailwind є можливість легко створювати унікальний дизайн без обмежень готових компонентів. Tailwind не нав'язує зовнішній вигляд, а натомість дає інструменти для гнучкого й точного налаштування будь-якого елемента. Це робить його відмінним вибором як для стартапів, так і для великих корпоративних проєктів.

Vite (Рис. 3.4) — це інструмент для розробки вебзастосунків, створений Еваном Ю, автором фреймворку Vue.js. Уперше Vite представили у 2020 році у відповідь на проблеми, з якими стикалися розробники, використовуючи традиційні інструменти збірки, такі як Webpack і Gulp. На той час основною проблемою, яку Vite мав вирішити, була низька швидкість розробки у великих проєктах.

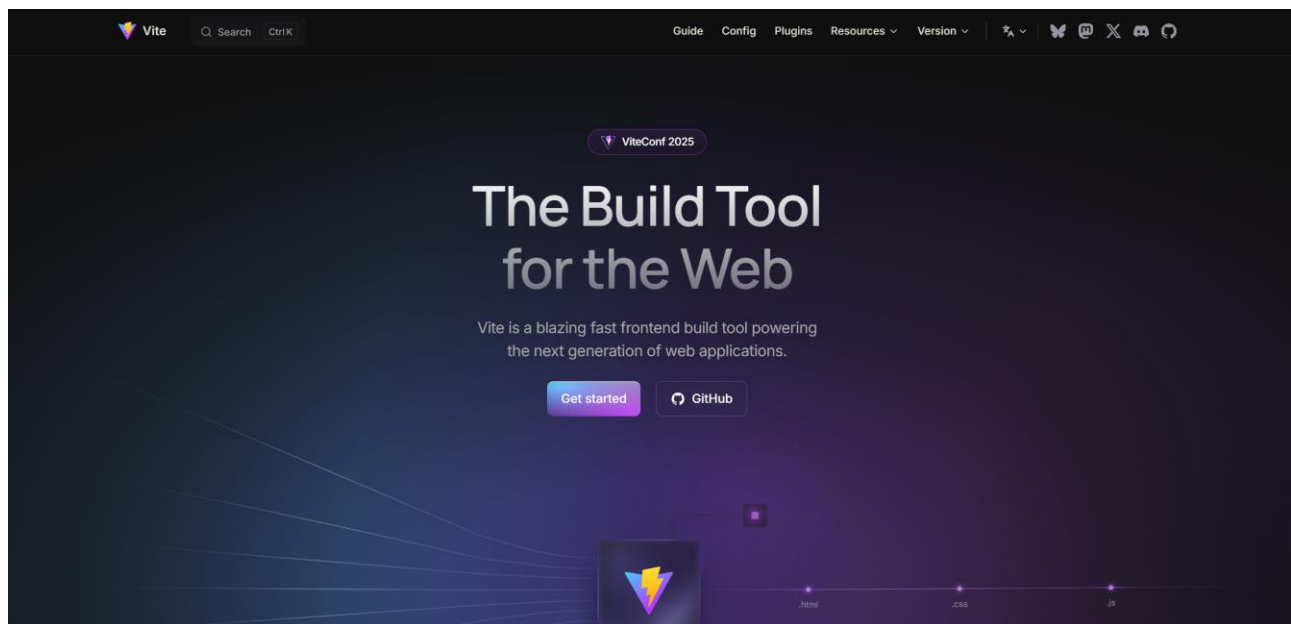


Рисунок 3.4. Головна сторінка Vite

Інструменти старого покоління базувалися на процесі бандлінгу, що передбачав об'єднання всіх модулів програми в один або кілька файлів, перш ніж їх запускали в браузері. Перезбірка проєкту при внесенні змін вимагала чимало часу, особливо якщо код складався з десятків тисяч рядків. Це сповільнює продуктивність і заважає зосередитися на написанні коду.

В основі Vite — використання можливостей ES-модулів, які дають змогу запускати код безпосередньо в браузері без попереднього бандлінгу. Відтак не треба створювати складні бандли на етапі розробки, а отже скорочується час на збірку та перезбірку проекту.[23]

Однією з головних переваг Vite є надзвичайно швидке оновлення змін завдяки технології hot module replacement (HMR). Це дозволяє розробникам моментально бачити результат своєї роботи в браузері без повного перезавантаження сторінки, що значно прискорює розробку та тестування.

Vite також має гнучку архітектуру плагінів і просту інтеграцію з популярними фреймворками, такими як Vue, React, Svelte та іншими. Завдяки цьому його легко налаштувати під будь-які потреби проекту, розширювати можливості збірки та оптимізації.

Ще одна важлива перевага Vite — це мінімальний розмір фінальної збірки та оптимізація ресурсів для продакшена. Під час виробничої збірки Vite використовує Rollup для ефективного збирання, що гарантує швидке завантаження сайту і високу оцінку продуктивності у Lighthouse та інших інструментах.

Axios (Рис. 3.5) — це популярна бібліотека для здійснення HTTP-запитів у вебзастосунках. Вона працює на базі промісів і надає зручний синтаксис для обробки запитів до серверів, що значно спрощує інтеграцію API у фронтенд-проектах. Axios підтримує роботу як у браузері, так і на сервері за допомогою Node.js.

Головною перевагою Axios є простота у використанні та висока гнучкість налаштувань. Бібліотека підтримує автоматичну обробку JSON-даних, встановлення заголовків запиту, налаштування тайм-аутів, обробку помилок і перехоплення запитів та відповідей за допомогою інтерцепторів.

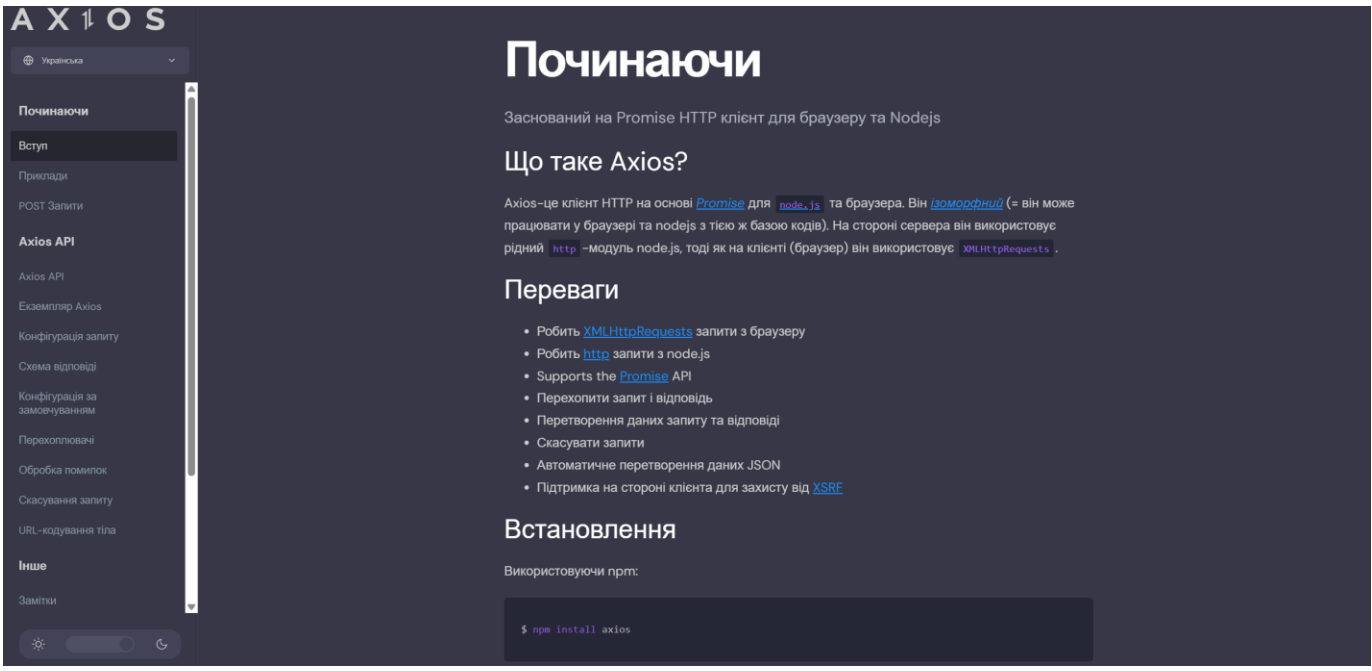


Рисунок 3.5. Головна сторінка Axios

Axios також надає зручні можливості для роботи з авторизацією, обробкою токенів і передачі даних у форматі FormData. Це робить його чудовим вибором для розробки сучасних інтерактивних застосунків, які активно взаємодіють із сервером через REST або GraphQL API.

Ще одна важлива перевага — підтримка скасування запитів і об'єднання кількох запитів в один. Завдяки цьому розробники можуть легко оптимізувати взаємодію клієнта і сервера, забезпечуючи кращу продуктивність і стабільність застосунку.

3.3 Керування контентом через Hygraph CMS

Hygraph (Рис 3.6) (раніше GraphCMS) - це потужна безголова система управління контентом (CMS), яка дозволяє безперешкодно доставляти контент по декількох каналах. Однак простого управління та розповсюдження контенту недостатньо для залучення трафіку на ваш сайт. Щоб максимізувати видимість, ваш сайт на базі Hygraph потрібно оптимізувати для пошукових систем.

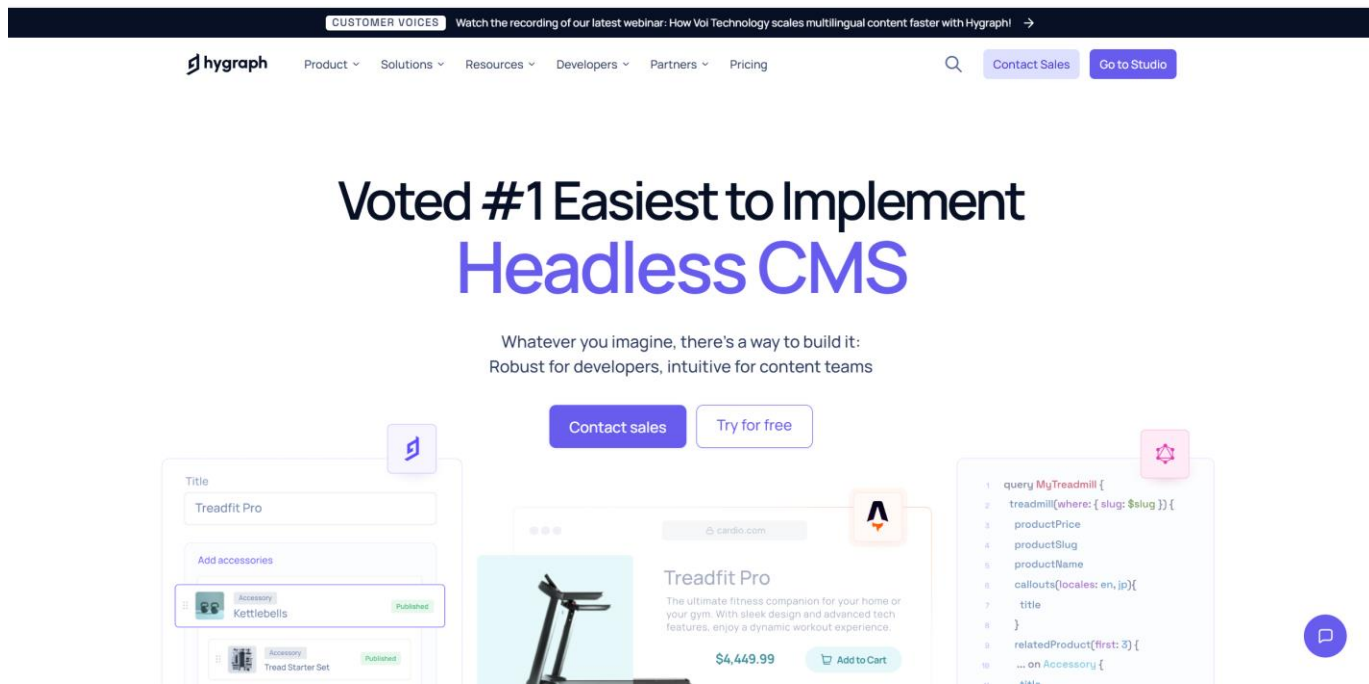


Рисунок 3.6. Головна сторінка Hygraph

Інтегрувавши Ranktracker у свою SEO-стратегію, ви зможете оптимізувати пошук ключових слів, відстежувати рейтинги та постійно покращувати свої SEO-показники. У цій статті ви дізнаєтеся про основні стратегії SEO для Hygraph і про те, як Ranktracker може допомогти вам покращити ваші зусилля.

SEO має вирішальне значення для будь-якого веб-сайту, але для тих, хто використовує безголові CMS, такі як Hygraph, це особливо важливо. Гнучкість Hygraph дозволяє вам розміщувати контент на різних платформах, але пошукові системи повинні розуміти і ранжувати ваш контент належним чином. Поєднуючи можливості Hygraph для створення структурованого контенту з SEO-інструментами Ranktracker, ви можете створити високооптимізований сайт, який буде залучати трафік і підвищувати видимість.

Порівняно з альтернативами (Strapi, Contentful, Prismic) Hygraph дає дві ключові переваги, критичні для навчальної платформи:

Гнучке моделювання даних без коду. (Рис 3.7) У drag-and-drop-редакторі можна створювати типи Lesson, Video, Quiz чи FAQ, визначати зв'язки між ними 1-N / M-N та додавати валідатори (обов'язкове поле, мінімальна довжина, унікальний slug). Завдяки цьому контентна схема змінюється за лічені хвилини без залучення розробника.

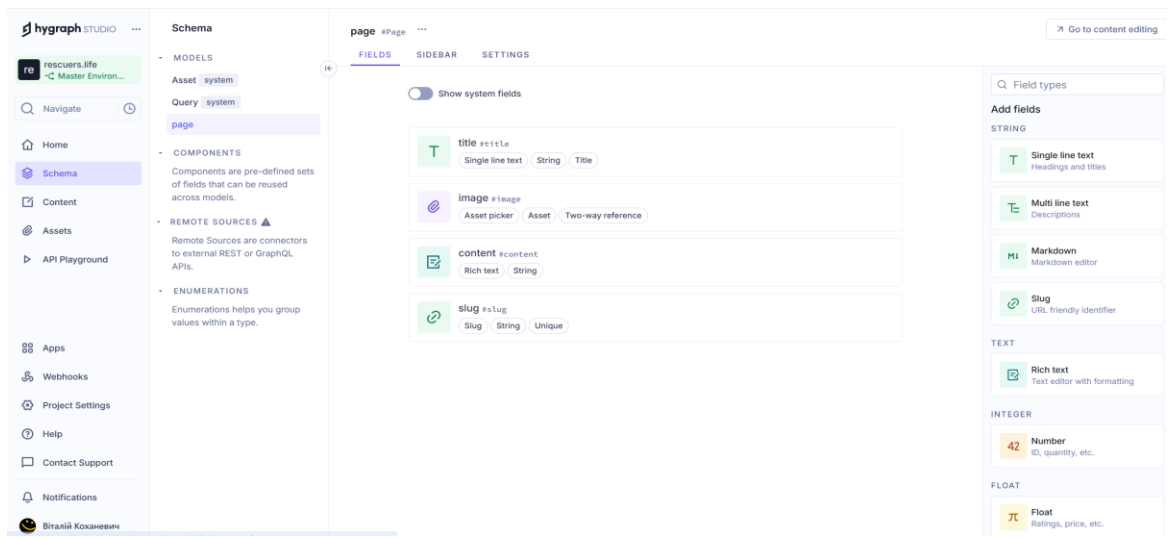


Рисунок 3.7 Редактор схем контенту в Hygraph

Webhooks і контент-попередній перегляд. Публікація або оновлення запису тригерить webhook, який через GitHub Actions запускає команду vercel deploy --prod. Таким чином новий матеріал стає доступним кінцевим користувачам автоматично, без ручного «натискання кнопки» на боці хостингу.

До додаткових плюсів належать вбудована CDN-доставка медіафайлів (AWS + Cloudfront), локалізація полів за мовами (UA, EN, RO), ролі та права доступу для редакторів, а також безкоштовний тариф, достатній для MVP-версії проекту. Під час тестів середній час відповіді GraphQL-запиту становив 80–120 мс, що цілком вкладається у вимоги Core Web Vitals.

3.4 Помічник у реальному часі на базі OpenAI API

У Rescuers Life роль «живого консультанта» виконує модель GPT-4o. Це багатомодальний алгоритм останнього покоління, який реагує майже миттєво: між надсиланням запиту й появою перших токенів минають частки секунди, тож діалог сприймається природно. У середині Vue-компонента користувач друкує своє запитання, наприклад «що робити при сильній кровотечі», а скрипт у ту ж мить формує prompt. До нього потрапляє сам текст запиту, мова інтерфейсу та JSON-масив із локального файла videos/data.json, де перелічено всі наявні відеоуроки. Запит іде на кінцеву точку /v1/chat/completions, у заголовку передається ключ API, а в тілі — модель gpt-4o-mini та режим stream, щоб відповідь приходила потоково.

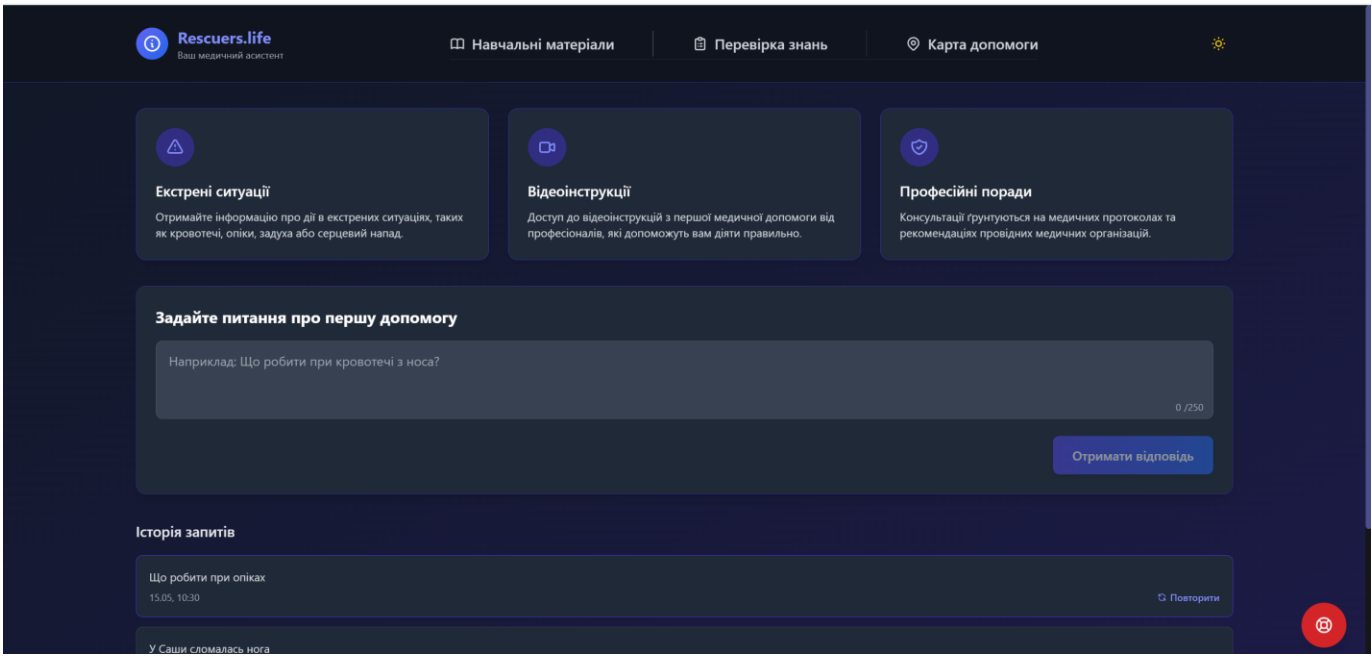


Рисунок 3.8 Сторінка помічника у реальному часі

Щойно сервер надсилає перші токени, вони безпосередньо додаються у вікно чату, тому користувач читає текст у режимі «друку на очах». Коли потік завершується, скрипт намагається розпарсити результат як JSON. Якщо всередині об'єкта є масив links, поруч із відповіддю з'являються кнопки з назвами відповідних відео;

натискання відкриває ролик у режимі Picture-in-Picture, і користувач може одразу переглянути потрібний фрагмент. Вся текстова частина проходить через DOMPurify і marked, тож Markdown перетворюється на безпечний та охайний HTML. Після кожного запиту питання додається до локальної «Історії». Якщо людина повернеться на сторінку пізніше, вона побачить свої останні десять звернень і зможе повторити будь-яке одним натисканням. При цьому в історії зберігається тільки сам текст запиту без персональних даних; відповіді та IP не логуються. Системний prompt також містить чітке правило: помічник не ставить діагнозів і не призначає ліків. Якщо користувач запитує про медикаменти, бот відповідає, що слід звернутися до лікаря, і показує номер екстреної служби. Якщо з OpenAI API стається збій, на екрані відображається сповіщення про помилку, а код автоматично підставляє запасну відповідь, збережену локально, щоб інтерфейс не залишався порожнім. Таким чином, незалежно від якості мережі, користувач усе одно отримує щонайменше базові рекомендації. Завдяки низькій латентності GPT-4o, потоковому режиму API та суворим етичним обмеженням, помічник у Rescuers Life надає швидкі й структуровані поради, не виходячи за межі першої медичної допомоги, і при цьому залишається безпечним як для користувача, так і для розробника.

3.5 Відеоуроки та навчальні матеріали

Усе навчальне наповнення Rescuers Life має відеоуроки та текстові матеріали-конспекти. Джерелом відеоконтенту слугує офіційний канал Червоного Хреста на YouTube. Ролики вбудовуються у сторінки звичайним `<iframe>` із параметрами `modestbranding=1` і `rel=0`, щоб вимкнути сторонні рекомендації та залишити користувача максимально зосередженим на темі (Рис. 3.9). Такий підхід не потребує додаткових бібліотек і водночас покладає на інфраструктуру YouTube усе кодування та глобальну доставку відео.

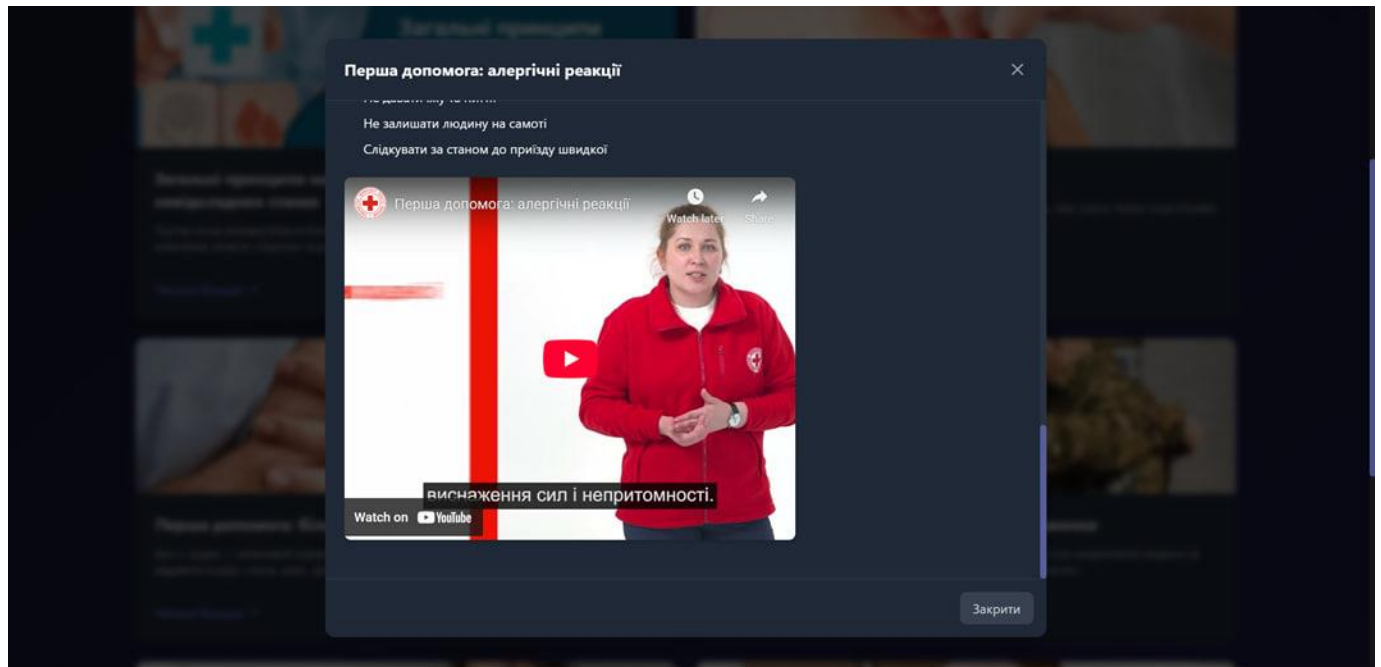


Рисунок 3.9 Вбудований відеоурок Червоного Хреста

Кожен урок на YouTube має пару в Hygraph: там зберігаються заголовок, короткий опис, мітки теми та PDF-конспект. Після публікації редактором статті Headless CMS автоматично надсилає webhook, і Vercel перевипускає тільки ту сторінку, де з'явилося нове посилання; уся процедура займає менше хвилини й не потребує втручання розробника. Конспект містить розгорнутий алгоритм дій, чек-лист і посилання на нормативні документи. Кнопка «Завантажити конспект» показується одразу під відео. На рівні інтерфейсу сторінка «Навчальні матеріали» відображає всі уроки й статті у вигляді сітки карток із пошуком і сортуванням за назвою (Рис. 3.10). При натисканні «Читати більше» відкривається модальне вікно з повним текстом HTML-статті, де користувач може прокручувати контент, не залишаючи сторінку. Пошук виконується миттєво всередині браузера, а сортування відбувається без перезавантаження завдяки реактивності Vue. Якщо статей багато, компонент автоматично додає пагінацію, щоб сторінка залишалася швидкою навіть на мобільному інтернеті.

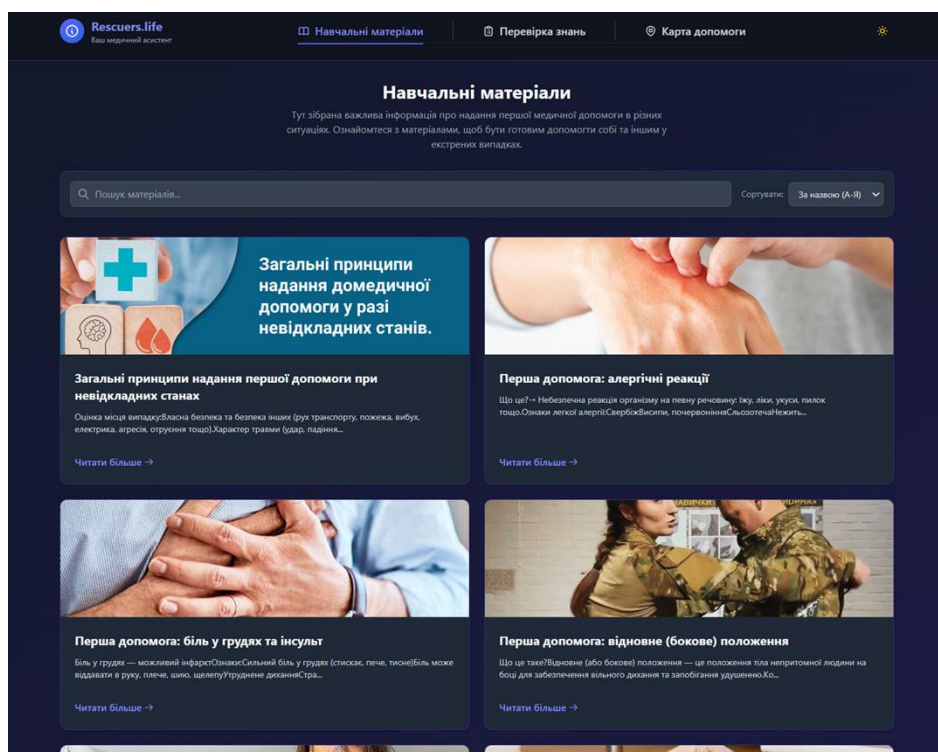


Рисунок 3.10 Список навчальних матеріалів (сітка карток)

Використання контенту Червоного Хреста дає одразу кілька переваг: матеріал створено сертифікованими інструкторами, тому він відповідає міжнародним протоколам; більшість роликів містить українські субтитри, а платформі легко додати власний переклад WebVTT; зрештою, відпадає потреба у власному відеопродакшені, і команда може зосередитись на розробці інтерактивних інструментів. Якщо Червоний Хрест оновлює ролик, адміністратору достатньо змінити в NuGraph лише ID відео — нова версія миттєво з’являється на сайті без додаткових кроків. Вбудовані інструменти YouTube автоматично підлаштовують якість потоку під швидкість мережі, тому відтворення не «фризить» навіть на 3G. У перспективі планується розширити базу матеріалів короткими анімованими GIF-інструкціями та інтегрувати таймкоди: після завершення відео помічник одразу пропонуватиме тест із питань, що покривають ключові етапи ролика.

3.6 Перевірка знань (інтерактивні тести)

Тестовий модуль реалізовано цілком власним кодом на Vue без сторонніх бібліотек—усі запитання й варіанти відповідей зберігаються просто у масиві JavaScript-об'єктів. Спершу користувач бачить стартовий екран із короткою інструкцією, обов'язковим полем для імені та селектором рівня складності: «Легкий», «Середній» або «Складний». Після вибору рівня компонент формує підмножину запитань: чотири для легкого, дев'ять для середнього і чотирнадцять для складного. Натиснувши «Розпочати тест», користувач переходить до послідовного режиму питання-за-питанням (Рис. 3.11).

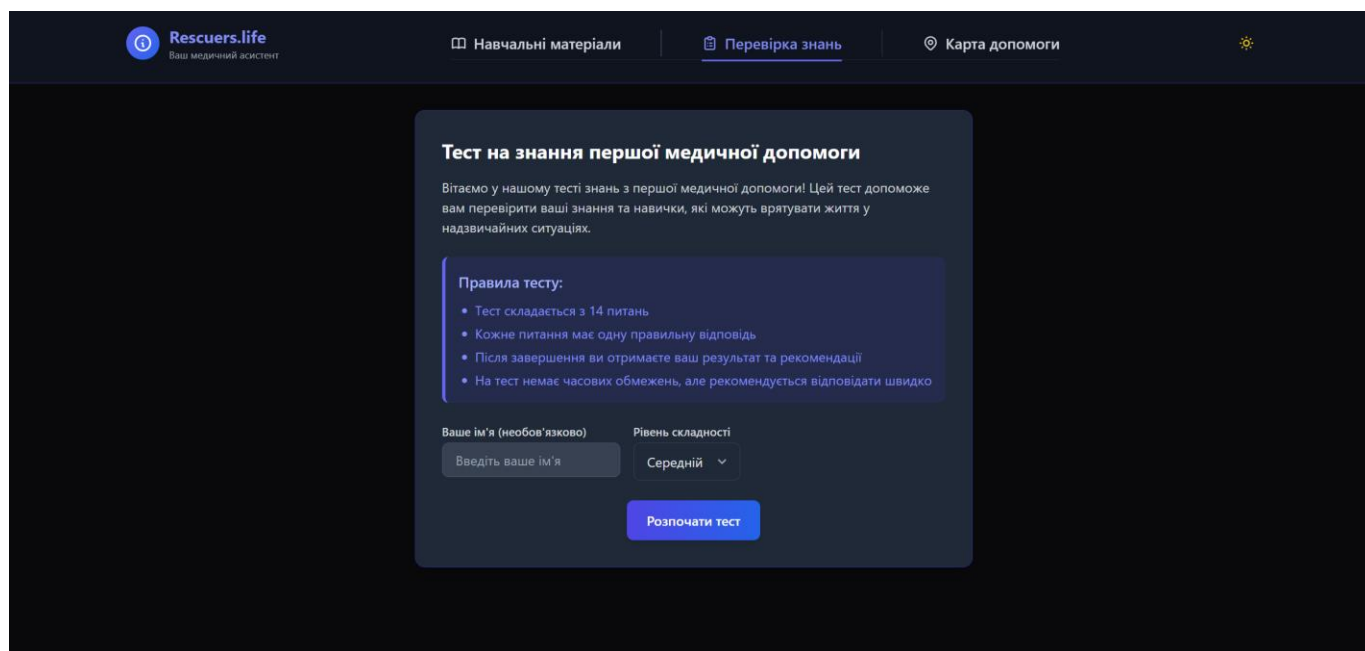


Рисунок 3.11 Повний матеріал у модальному вікні

Кожне запитання відображається разом із варіантами відповідей у вигляді радіокнопок; після вибору варіанта кнопка «Наступне» стає активною. Угорі показано прогрес-бар, що плавно наповнюється, та відсоток виконання. Компонент зберігає відповіді у масив `userAnswers`, тож можна повернутися до попереднього питання, не

втрачаючи вибору. На останньому питанні кнопка змінюється на «Завершити тест», а після натискання з'являється екран результатів: кількість правильних відповідей, процент успіху, кольоровий індикатор і текстова рекомендація, сформована залежно від відсотка. Нижче подано розгорнутий огляд усіх питань із позначкою «Правильно /Неправильно» та поясненнями.

Уся логіка навігації—перехід уперед, назад, фіксація відповіді—працює локально, без звернень до сервера. Це гарантує миттєву реакцію навіть у режимі слабкого інтернету. Коли тест завершено, користувач може натиснути «Пройти тест ще раз», і компонент повністю скидає стан до початкового, дозволяючи одразу спробувати інший рівень.

3.7 CI/CD та інфраструктура розгортання

CI/CD (Continuous Integration / Continuous Deployment) — це сучасна практика автоматизації процесів розробки, тестування та розгортання програмного забезпечення. Головна ідея полягає в тому, щоб кожен змінюваний код швидко інтегрувати в основну гілку розробки, автоматично перевіряти її якість за допомогою тестів, створювати білд і, у разі успішного проходження всіх етапів, розгорнути застосунок у продакшн середовищі.

Continuous Integration (CI) означає, що кожен розробник часто зливає свої зміни у спільну репозиторію, після чого запускаються автоматичні збірки та тести. Це допомагає швидко виявляти помилки, конфлікти та проблеми з сумісністю ще на ранніх етапах розробки. Завдяки цьому якість коду суттєво підвищується, а час на пошук і виправлення багів скорочується.

Continuous Deployment (CD) продовжує цей процес, автоматизуючи доставку перевіреного коду на сервери для користувачів. При успішному проходженні всіх тестів нова версія продукту автоматично розгортається у продакшн, без необхідності

ручного втручання. У разі виявлення критичних помилок система дозволяє швидко виконати "відкат" до стабільної версії, що мінімізує ризики для бізнесу.

Застосування CI/CD дає кілька стратегічних переваг. По-перше, зменшується вплив людського фактора: система автоматично перевіряє якість змін, незалежно від уважності чи досвіду окремого розробника. По-друге, компанія може набагато швидше випускати оновлення, нові функції та виправлення помилок, що покращує користувацький досвід і підвищує конкурентоспроможність. По-третє, завдяки відпрацьованим процесам розгортання з'являється можливість миттєвого повернення до попередньої версії у випадку збоїв, що забезпечує стабільність сервісу.

Впровадження CI/CD потребує налаштування інструментів, таких як GitHub Actions, GitLab CI, Jenkins, CircleCI або інших систем автоматизації. Також важливо мати покриття коду юніт- та інтеграційними тестами, щоб автоматичні перевірки дійсно гарантували якість продукту. У результаті добре побудована CI/CD-платформа є основою сучасної DevOps-культури й важливим кроком до гнучкої, безпечної та ефективної розробки.

GitHub (Рис. 3.12) — це популярна платформа для зберігання та управління вихідним кодом, яка використовує систему контролю версій Git. Вона забезпечує зручну співпрацю між розробниками, дозволяє відслідковувати зміни в коді, створювати гілки для розробки нових функцій, робити Pull Request-и для перевірки змін і вести обговорення безпосередньо в межах проєкту.

Усі вихідні файли проєкту зберігаються у приватному репозиторії на платформі GitHub. Використання GitHub забезпечує повний розподілений контроль версій за допомогою системи Git, що дозволяє відслідковувати історію змін, працювати у команді над одними й тими ж файлами та зручно організувати процес розробки. Крім того, GitHub надає інтуїтивно зрозумілий веб-інтерфейс для перегляду коду, створення Pull Request'ів, ведення документації та обговорення змін.

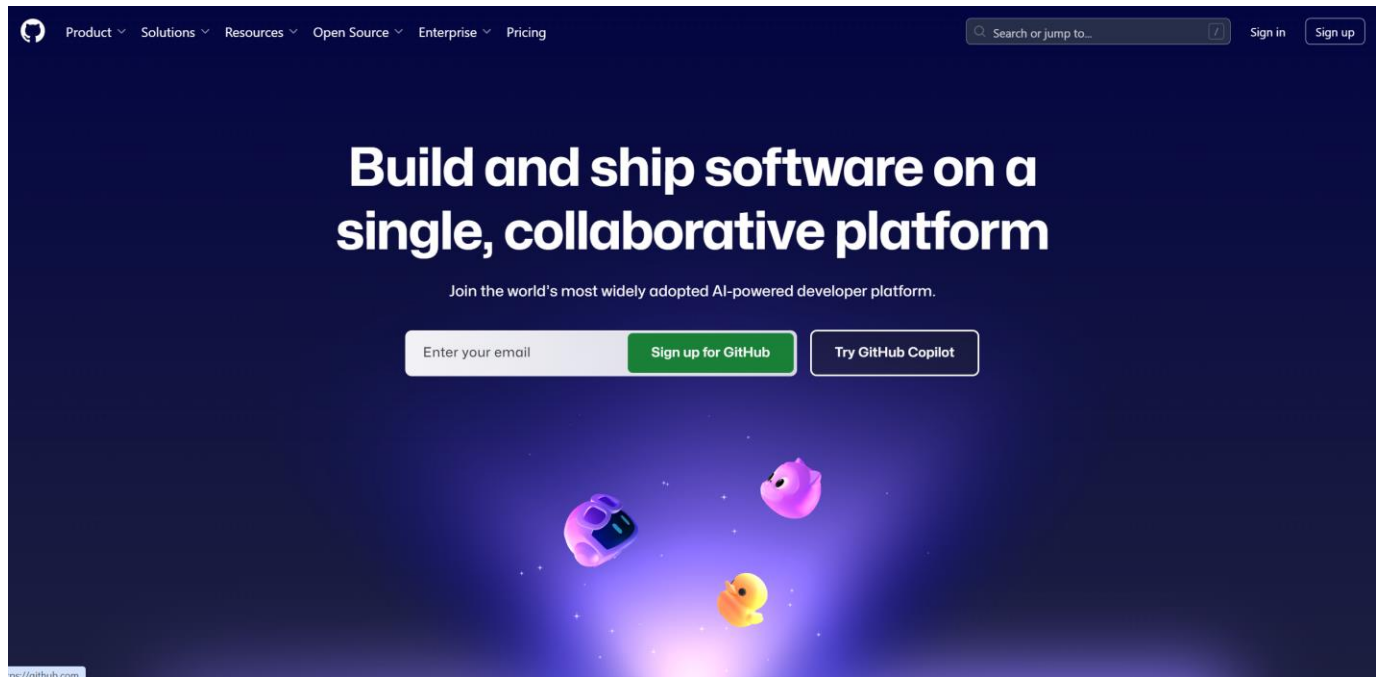


Рисунок 3.12 Головна сторінка Github

Окремою перевагою є інтеграція GitHub із системою перевірок: результати автоматичних збірок та тестувань відображаються безпосередньо у картці Pull Request (PR). Це дозволяє ревіюерам миттєво бачити, чи білд успішний ("зелений"), не переходячи у сторонні CI/CD сервіси. Така зручність значно прискорює процес перевірки та прийняття змін у проєкт.

Для оптимізації процесу збирання проєкту між прогонками використовується кешування npm-пакетів та артефактів Vite. Завдяки цьому повторні білди виконуються вдвічі швидше, що особливо важливо при активній розробці та частому внесенні змін. Використання кешу дозволяє заощаджувати обчислювальні ресурси та час розробників.

Захист конфіденційної інформації також враховано: усі секретні токени, такі як ключі доступу до OpenAI або токен деплою Vercel, зберігаються у спеціальному сховищі GitHub Secrets. Це гарантує, що важливі дані не потраплять у відкриті логи

або кодову базу. При виконанні CI/CD-процесів секрети передаються у середовище виконання без явного виведення, забезпечуючи високий рівень безпеки.

Vercel (Рис. 3.13).— це сучасна хмарна платформа, спеціалізована на хостингу статичних сайтів і serverless-застосунків. Вона ідеально підходить для розгортання Single-Page Application (SPA) завдяки простоті налаштування, високій швидкості завантаження сторінок і глибокій інтеграції з популярними інструментами розробки. Після завершення процесу білду за допомогою GitHub Actions артефакт автоматично публікується на Vercel через CLI-команду `vercel --prod`, що забезпечує швидкий і безперебійний випуск нової версії сайту.

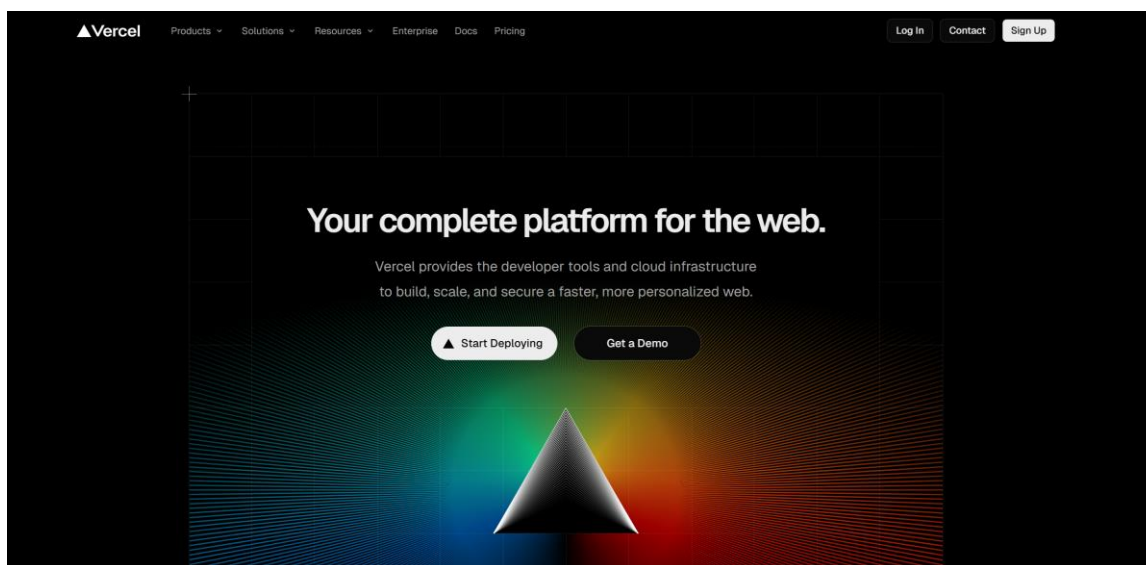


Рисунок 3.13 Головна сторінка Vercel

Однією з важливих функцій Vercel є система preview-деплоїв. Для кожного відкритого pull request автоматично створюється унікальне посилання на зразок `https://pr-42--назва-проєкту.vercel.app`, де можна переглянути, як виглядають зміни ще до їхнього об'єднання в основну гілку. Це значно спрощує процес перевірки для

дизайнерів, кураторів або інших членів команди, зменшуючи ризик випадкових помилок на продакшені.

Vercel використовує глобальну edge-мережу для кешування статичних ресурсів, таких як HTML, JavaScript, CSS та PDF-файли. Завдяки CDN-нодам, розташованим по всьому світу, сторінки доставляються користувачам з мінімальними затримками навіть у регіонах із повільним інтернетом. Це забезпечує блискавичне завантаження контенту незалежно від географічного розташування користувача.

Ще однією ключовою перевагою є підтримка автоматичного rollback. Якщо після публікації нової версії буде виявлено критичну помилку, достатньо натиснути кнопку «Restore previous» в інтерфейсі Vercel, і попередня стабільна версія сайту миттєво відновиться без простою сервісу. Це підвищує надійність системи і мінімізує можливі ризики для користувачів.

Коли редактор публікує новий урок у Hygraph, CMS надсилає webhook-запит на адресу <https://api.vercel.com/v1/integrations/deploy>, який запускає повторний білд та деплой. Таким чином усі зміни контенту з'являються на сайті без залучення розробника й без затримок.

Поєднання GitHub Actions і Vercel формує повний цикл CI/CD: код комітиться, автоматично перевіряється, збирається й одразу стає доступним кінцевим користувачам, причому кожен крок фіксується та може бути відкочений у разі потреби. Це зменшує час між написанням функції і її появою на сайті, забезпечує надійність оновлень і робить процес розробки максимально прозорим для всієї команди.

3.10 Інтерактивна карта допомоги

Leaflet — JavaScript бібліотека з відкритим сирцевим кодом для відображення мап на html-сторінках. Самодостатня, сучасна (HTML5, CSS3), не велика за обсягом, з широким колом ліцензійної угоди, проста у використанні.

Бібліотека реалізує підтримку шарів мап, які побудовані за технологією: WMS, GeoJSON, Тайли або векторного відображення поверхні. Інші типи проєкцій мап підтримуються за допомогою додатків. Головне, щоб кожний формат мав однозначну функцію перетворення локальних координат в географічні координати відповідно до картографічної проєкції, в якій ці локальні координати задані.[24]

OpenStreetMap постачає безкоштовні картографічні тайли, які регулярно оновлюються спільнотою, тож платформа не залежить від комерційних ліцензій Google або HERE і може кастомізувати зовнішній вигляд шарів через власний стиль. **Firebase** була розроблена компанією Google як платформа для Backend-as-a-Service (BaaS). Вона пропонує розробникам інструменти та сервіси, необхідні для створення та підтримки бекенду їхніх застосунків. Послуги включають хостинг, автентифікацію користувачів, базу даних в режимі реального часу та інші функції.

База даних Firebase (Firestore або Realtime Database) — це база даних NoSQL, завдяки чому вона дуже зручна для користувача. На відміну від SQL, де таблиці і рядки є нормою, бази даних NoSQL на кшталт Firebase використовують колекції зі списком документів без таблиць.

Firebase можна використовувати для обробки авторизації, сповіщень, зберігання файлів та інших речей, які можуть знадобитися вашим застосункам. А оскільки Google часто оновлює Firebase, нові функції завжди доступні. Крім того, консоль Firebase має гарний дизайн і робить взаємодію між Firebase Console та застосунком дуже простою.[25]

Firebase Realtime Database забезпечує хмарне зберігання координат у форматі JSON, автоматичне масштабування та миттєву синхронізацію: будь-яка зміна даних транслюється всім клієнтам через вбудований **WebSocket**-канал, що усуває необхідність у власному сервері та складній логіці опитування REST-запитами. У зв'язці Leaflet відображає карту, OSM надає безоплатні тайли, а Firebase через WebSocket миттєво «пушить» нові чи змінені точки, завдяки чому користувачі

практично в реальному часі бачать актуальне розташування медичних об'єктів і SOS-запитів.

Модуль карти базується на **Leaflet**, який завантажує тайли **OpenStreetMap** і дає мінімалістичний API для маркерів. Координати медичних пунктів та автоматичних зовнішніх дефібриляторів зберігаються у **Firestore Realtime Database**. Коли редактор додає або змінює точку, Firestore миттєво транслює оновлення через WebSocket-канал, і новий маркер з'являється у всіх користувачів без перезавантаження сторінки.

Після надання дозволу на геолокацію карта центрується на позиції користувача, позначеній синього кола . У межах приблизно двох кілометрів показуються сині маркери медичних пунктів і жовті маркери дефібриляторів; за великої їх кількості Leaflet автоматично об'єднує маркери в кластери, щоби вони не перекривалися. Натиснувши на будь-яку точку, користувач бачить назву закладу, контактний номер і дві кнопки.

Перша кнопка, «Допомогти», відкриває Google Maps у новій вкладці вже з прокладеним пішохідним маршрутом від поточного місцезнаходження до вибраного пункту . Друга кнопка, «Мені потрібна допомога», ставить на карту самого застосунку яскравий пульсуючий маркер SOS. Його координати одразу записуються у Firestore, тож інші користувачі в радіусі дії бачать сигнал і можуть натиснути «Допомогти», щоби відкрити маршрут (Рис. 3.15). SOS-маркер зникає після підтвердження допомоги або автоматично через 30 хвилин, щоби карта не засмічувалася застарілими викликами.

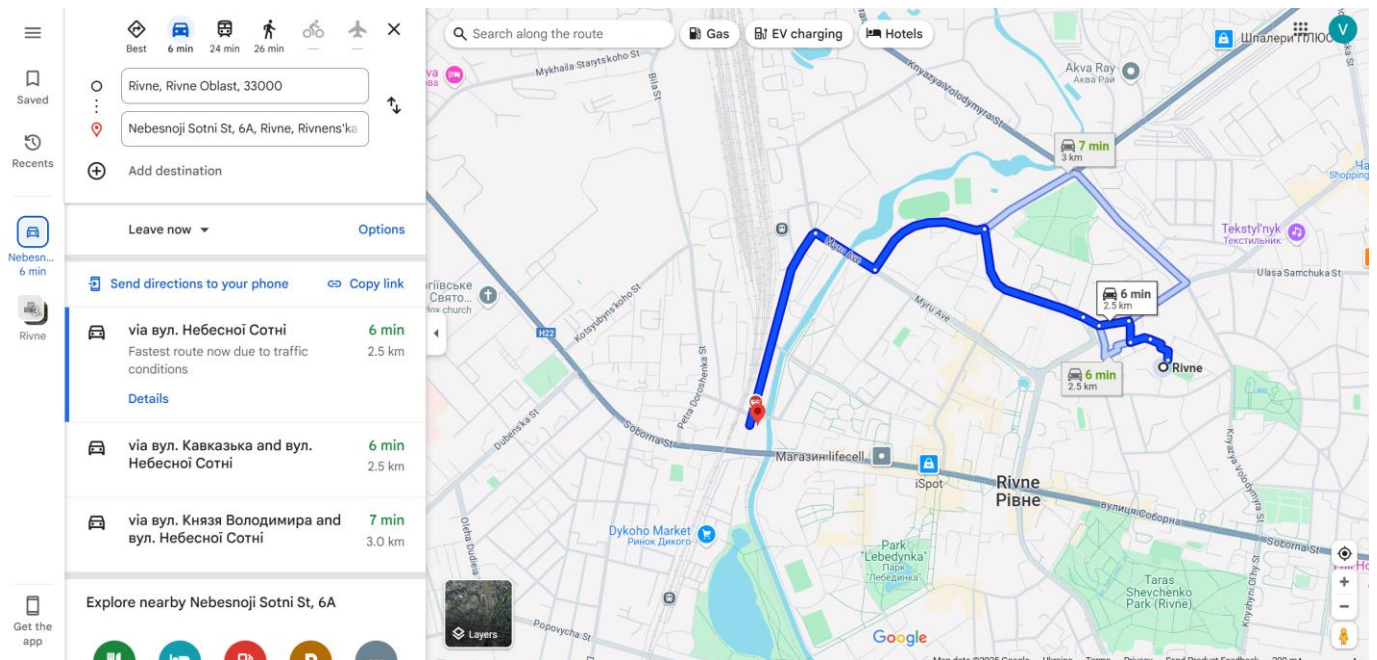


Рисунок 3.15 Маршрут до користувача, який потребує допомоги

Якщо зв'язок обривається, карта переходить у офлайн-режим: раніше завантажені плитки OSM і останній набір маркерів залишаються доступними локально, тож користувач бачить бодай попередню версію схеми. Далі планується кешувати окремий GeoJSON-файл, щоб навіть нові відвідувачі могли працювати офлайн, а також інтегрувати OpenRouteService для побудови маршруту безпосередньо всередині застосунку.

Такий зв'язок Leaflet, OSM та Firebase гарантує постійну актуальність даних, мінімальний час оновлення й інтуїтивний сценарій «побачив пункт → відкрив маршрут» або «попросив допомоги → іншим одразу видно, куди йти».

3.11 Оцінка ефективності та подальший розвиток

Після першої внутрішньої ітерації *Rescuers Life* пройшов комплекс технічних і користувачьких перевірок, результати яких підтвердили стійкість обраної архітектури й доцільність педагогічної моделі. Під час кросбраузерного тестування додаток

коректно працював у сучасних версіях Chrome, Firefox, Edge і Safari, а також у вбудованому WebView на смартфонах з Android 10 і вище. Навіть за ввімкненого режиму енергозбереження всі інтерактивні елементи — чат-помічник, відеоуроки та тести — відтворювалися без збоїв.

Заміряна продуктивність показала, що завдяки малому початковому пакету JavaScript і кешуванню через Service Worker інтерфейс з'являється на екрані менш ніж за секунду, коли користувач заходить у мережу 4G, а подальша навігація між сторінками відбувається миттєво, оскільки дані вже лежать у локальному кеші. Навіть у штучно змодельованих умовах втрати зв'язку додаток переходив у офлайн-режим без видимих затримок: відкриті раніше сторінки залишалися доступними, а результати тестів зберігалися локально й синхронізувалися автоматично після відновлення мережі.

У процесі користувацького тестування добровольці швидко опановували логіку проходження курсу: від вибору уроку через перегляд відео й читання короткого конспекту до виконання перевірного тесту. Чітко сформульовані кнопки заклику до дії скорочували до мінімуму кількість кліків, а послідовне розташування матеріалів допомагало зосередитися на змісті. Додаткову довіру викликала вказівка на те, що всі відео та текстові матеріали взяті з ресурсів Червоного Хреста; крім того, наявність субтитрів робила уроки зручними навіть у шумному середовищі або для людей із порушеннями слуху. Чат-помічник на базі OpenAI користувачі охрестили «швидкою шпаргалкою»: достатньо було поставити запитання, наприклад «що робити при опіку кип'ятком», щоб одразу отримати структурований алгоритм дій, не перемотуючи відео. Система тестів із трьома рівнями складності теж одержала схвальні відгуки: легкі запитання дозволяли швидко перевірити базові поняття, а складні — закріпити повний алгоритм дій.

Найближчий етап розвитку проєкту передбачає суттєве розширення навчальної бази. До кінця року планується додати щонайменше тридцять нових відеоуроків, що охоплять теми опіків, травм кінцівок і невідкладної допомоги дітям та немовлятам.

Паралельно готуються короткі міні-уроки з біобезпеки та алгоритмів поведінки під час стихійних лих, а також анімовані GIF-інструкції, які показуватимуть один критично важливий крок за кілька секунд. Тестовий модуль поповниться щоденними мікротестами «одне питання — одна хвилина» і можливістю отримати електронний сертифікат після завершення повного курсу й фінального іспиту.

Карта буде модернізована: крім медичних пунктів, користувач зможе бачити розташування автоматичних зовнішніх дефібриляторів і найближчих аптек, а в офлайн-режимі карта базуватиметься на кешованих шарах OpenStreetMap, щоб маршрут залишався доступним навіть без мережі. Чат-помічник отримає підтримку голосового введення й синтезу мовлення для ситуацій «вільні руки», а в перспективі — можливість аналізувати фотографії травм і генерувати детальніші рекомендації щодо перев'язування.

Додаткову мотивацію забезпечить система бейджів за навчальні дні поспіль і тематичні щотижневі виклики, а інтерфейс одержить висококонтрастну тему й підтримку масштабування шрифтів для користувачів із порушеннями зору. Контент буде повністю локалізовано румунською й англійською мовами, між якими можна буде перемикатися одним кліком. Нарешті, команда досліджує інтеграцію зі смарт-годинниками: у разі виявлення падіння чи різкого зниження пульсу пристрій надсилатиме сигнал у додаток, після чого користувач отримає миттєвий алгоритм CPR прямо на екрані телефону.

Таким чином платформа *Rescuers Life* уже продемонструвала стабільну роботу й позитивну реакцію перших користувачів, а заплановані оновлення допоможуть перетворити її на комплексну систему, що об'єднає навчання, практику та оперативну підтримку в реальних ситуаціях.

Висновки

Робота над Rescuers Life дала змогу переконатися, що сучасні веб-технології здатні перетворити суху теорію з першої допомоги на динамічний, практично орієнтований курс. Порівняння з існуючими ресурсами показало: більшості з них бракує інтерактиву, адаптації під рівень учня та швидкого оновлення контенту. Тому головним завданням стало створити платформу, що поєднує перевірену медичну інформацію з максимально живим навчальним досвідом.

Для клієнтської частини обрано Vue 3 у поєднанні з Tailwind CSS та збіркою на Vite. Такий стек забезпечив швидке завантаження навіть на мобільному зв'язку й одночасно дав змогу гнучко компоувати інтерфейс. Контент зберігається у Hygraph, а завдяки GraphQL-запитам у додаток надходить лише те, що потрібно в конкретний момент, без «зайвих» кілобайтів. Чат-помічник працює на OpenAI GPT-4o Realtime API й видає миттєві поради українською мовою, підтягує релевантні відео та формує короткі чек-листи дій. Для інтерактивної карти використано зв'язку Leaflet + OpenStreetMap з оновленням координат через Firebase Realtime Database: будь-який сигнал SOS одразу бачать усі користувачі, а маршрут до постраждалого будується одним кліком.

Окремий акцент зроблено на системі перевірки знань: тести різної складності працюють офлайн і дають миттєвий зворотний зв'язок, а результати супроводжуються порадами щодо подальшого навчання. Усе це розгорнуто на Vercel з безперервним CI/CD-циклом; отже, нове відео Червоного Хреста або оновлений протокол ERC потрапляють на сайт за лічені хвилини без ручних деплоїв.

Під час кросбраузерного тестування платформа стабільно працювала у Chrome, Firefox, Edge, Safari та мобільних WebView, що підтвердило правильність вибраного технологічного підходу. Найважливіше ж те, що перші користувачі відзначили швидкодію помічника, зручність карти і зрозумілу структуру уроків.

Таким чином, Rescuers Life демонструє, як поєднання Vue-екосистеми, headless-CMS, хмарних API та відкритих карт здатне зробити навчання першій допомозі доступним,

інтерактивним і головне дієвим. Сподіваюся, платформа допоможе користувачам засвоїти критичні навички й упевнено діяти в реальних надзвичайних ситуаціях.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Перша медична допомога. Вікіпедія. URL: uk.wikipedia.org/wiki/Перша_медична_допомога (дата звернення: 10.02.2025).
2. Всесвітня організація охорони здоров'я. Оцінювання потреб дорослого населення України у сфері домедичної допомоги. — Копенгаген : WHO Regional Office for Europe, 2023.: <https://iris.who.int/bitstream/handle/10665/368732/WHO-EURO-2023-6904-46670-68751-ukr.pdf> (дата звернення: 2025-05-28).
3. Принципи надання першої долікарської допомоги URL: https://www.wunu.edu.ua/pdf/stud_live/safety/Instruktsiya_z_nadannya_domed_dopomohy.pdf (дата звернення: 15.02.2025).
4. Червоний Хрест України. URL: <https://redcross.org.ua> (дата звернення: 17.02.2025).
5. Vue.js на платформі Medium. URL: medium.com/@jstify.community/vue-js-навіщо-він-нам-981b5e17af39#:~:text=y%202014%20році,-,Vue.,пристроїв%20за%20допомогою%20фреймворку%20Electron (дата звернення: 20.02.2025).
6. Prometheus. URL: <https://uk.wikipedia.org/wiki/Prometheus> (дата звернення: 25.02.2025).
7. Material Design – офіційний ресурс Google про адаптивний дизайн. URL: <https://material.io/design> (дата звернення: 01.03.2025).
8. Наукові статті про UX/UI у навчальних додатках, Google Scholar (запити типу "adaptive design educational apps" або "UX for educational applications"). (дата звернення: 05.03.2025).
9. Платформа з домедичної допомоги – Всеукраїнська Рада Реанімації. <https://firstaid.org.ua> (дата звернення: 08.03.2025).

10. Онлайн-курс «Домедична допомога» – Prometheus.
<https://prometheus.org.ua> (дата звернення: 12.03.2025).
11. Національна служба здоров'я України – розділ рекомендацій.
<https://nszu.gov.ua> (дата звернення: 15.03.2025).
12. Мобільний застосунок «Перша допомога» – Товариство Червоного Хреста України (дата звернення: 20.03.2025).
13. <https://redcross.org.ua> (дата звернення: 23.03.2025).
14. Український центр навчання першої допомоги (ГО)
<https://firstaidtraining.com.ua> (дата звернення: 26.03.2025).
15. Prometheus. Онлайн-курс «Домедична допомога» [Електронний ресурс]. — Режим доступу: <https://prometheus.org.ua/course/domedychna-dopomoha/> (дата звернення: 20.03.2025).
16. Платформа з домедичної допомоги / Всеукраїнська рада реанімації. — [Електронний ресурс]. — Режим доступу: <https://firstaid.org.ua/> (дата звернення: 02.04.2025).
17. Товариство Червоного Хреста України. Мобільний додаток «Перша допомога» [Електронний ресурс]. <https://redcross.org.ua/first-aid-mobile-app/> (дата звернення: 06.04.2025).
18. Державна служба України з надзвичайних ситуацій. Поради населенню [Електронний ресурс]. — Режим доступу: <https://dsns.gov.ua/uk/poradi-naselennyu> (дата звернення: 09.04.2025).
19. Кухаренко В.М., Сверчкова Л.І. Інформаційні технології навчання: методика та практика. – Харків: НТУ «ХП», 2020. (дата звернення: 12.04.2025).
20. Хоружа Н.Л. Психолого-педагогічні умови ефективного електронного навчання // Наукові записки. Педагогічні науки, 2019. (дата звернення: 20.04.2025).

21. Deci, E. L., & Ryan, R. M. (1985). *Intrinsic motivation and self-determination in human behavior*. Springer. (дата звернення: 23.04.2025).
22. Кухаренко В.М. (2020). *Сучасні технології електронного навчання*. Харків: НТУ «ХПІ» (дата звернення: 25.04.2025).
23. Тарабанов, О. (2025, 9 січня). Ти — фронтенд-розробник? Дізнайся, як інструмент Vite спростить тобі роботу. Dev.ua.
<https://dev.ua/blogs/posts/tarabanov-1736410707>(дата звернення: 26.04.2025)
24. Leaflet – стаття у Вікіпедії: <https://uk.wikipedia.org/wiki/Leaflet> (дата звернення: 27.04.2025).
25. Обговорення бібліотеки Leaflet на форумі DOU:
<https://dou.ua/forums/topic/44058/> (дата звернення: 29.04.2025).
26. Vue – офіційни посібник. Введення: <https://vuejs.org/guide/introduction.html>
(дата звернення: 29.04.2025)

Додатки
Додаток А
Ключові зразки коду застосунку

```

<template>
  <div>
    <header
      class="sticky top-0 z-[9999] backdrop-blur-md bg-white/70 dark:bg-gray-900/70
border-b border-indigo-100 dark:border-indigo-900 shadow-sm"
    >
      <div class="container mx-auto px-4 py-4">
        <div class="flex items-center justify-between">
          <router-link to="/">
            <div class="flex items-center gap-3">
              <div
                class="flex items-center justify-center w-10 h-10 rounded-full bg-
gradient-to-br from-indigo-500 to-blue-600 text-white"
              >
                <svg
                  xmlns="http://www.w3.org/2000/svg"
                  class="h-6 w-6"
                  fill="none"
                  viewBox="0 0 24 24"
                  stroke="currentColor"
                >
                  <path
                    stroke-linecap="round"
                    stroke-linejoin="round"
                    stroke-width="2"
                    d="M13 16h-1v-4h-1m1-4h.01M21 12a9 9 0 11-18 0 9 9 0 0118 0z"
                  />
                </svg>
              </div>
            </div>
          <div class="hidden sm:block">
            <h1 class="text-xl font-bold text-gray-900 dark:text-white">
              <span class="text-indigo-600 dark:text-indigo-400"
                >Rescuers.life</span>
            </h1>
            <p class="text-xs text-gray-500 dark:text-gray-400">
              Ваш медичний асистент
            </p>
          </div>
          <div class="sm:hidden">
            <h1 class="text-lg font-bold text-gray-900 dark:text-white">

```

```

        <span class="text-indigo-600 dark:text-indigo-400"
            >Rescuers</span>
        >
    </h1>
</div>
</div>
</router-link>

<nav class="hidden lg:block">
    <div class="flex justify-center items-center gap-8 xl:gap-12">
        <router-link
            class="relative flex flex-col items-center group text-base xl:text-lg
font-medium transition-all duration-300"
            to="/guide"
            v-slot="{ isActive }"
        >
            <div class="flex items-center gap-2 z-10">
                <svg
                    xmlns="http://www.w3.org/2000/svg"
                    class="w-5 h-5 transition-all duration-300 group-hover:text-indigo-
600 dark:group-hover:text-indigo-400"
                    :class="
                        isActive
                            ? 'text-indigo-600 dark:text-indigo-400'
                            : 'text-gray-700 dark:text-gray-300'
                    "
                    fill="none"
                    viewBox="0 0 24 24"
                    stroke="currentColor"
                >
                    <path
                        stroke-linecap="round"
                        stroke-linejoin="round"
                        stroke-width="2"
                        d="M12 6.253v13m0-13C10.832 5.477 9.246 5 7.5 5S4.168 5.477 3
6.253v13C4.168 18.477 5.754 18 7.5 18s3.332.477 4.5 1.253m0-13C13.168 5.477 14.754 5
16.5 5c1.747 0 3.332.477 4.5 1.253v13C19.832 18.477 18.247 18 16.5 18c-1.746 0-
3.332.477-4.5 1.253"
                    />
                </svg>
                <span
                    :class="
                        isActive
                            ? 'text-indigo-600 dark:text-indigo-400'
                            : 'text-gray-700 dark:text-gray-300'
                    "
                >

```

```

        class="group-hover:text-indigo-600 dark:group-hover:text-indigo-400
transition-all duration-300 whitespace-nowrap"
        >Навчальні матеріали</span
    >
</div>
<div
    :class="
    isActive
    ? 'w-full bg-indigo-600 dark:bg-indigo-500'
    : 'w-0 bg-transparent group-hover:w-full group-hover:bg-indigo-600
dark:group-hover:bg-indigo-500'
    "
    class="h-0.5 rounded-full absolute -bottom-2 transition-all duration-
300 ease-out"
    ></div>
</router-link>

<div class="w-px h-8 bg-gray-300 dark:bg-gray-700"></div>

<router-link
    class="relative flex flex-col items-center group text-base xl:text-lg
font-medium transition-all duration-300"
    to="/test"
    v-slot="{ isActive }"
    >
    <div class="flex items-center gap-2 z-10">
        <svg
            xmlns="http://www.w3.org/2000/svg"
            class="w-5 h-5 transition-all duration-300 group-hover:text-indigo-
600 dark:group-hover:text-indigo-400"
            :class="
                isActive
                ? 'text-indigo-600 dark:text-indigo-400'
                : 'text-gray-700 dark:text-gray-300'
            "
            fill="none"
            viewBox="0 0 24 24"
            stroke="currentColor"
        >
            <path
                stroke-linecap="round"
                stroke-linejoin="round"
                stroke-width="2"
                d="M9 5H7a2 2 0 0-2 2v12a2 2 0 02 2h10a2 2 0 02-2V7a2 2 0 0-2-
2h-2M9 5a2 2 0 02 2h2a2 2 0 02-2M9 5a2 2 0 012-2h2a2 2 0 012 2m-3 7h3m-3 4h3m-6-
4h.01M9 16h.01"
            />

```

```

</svg>
<span
  :class="
    isActive
      ? 'text-indigo-600 dark:text-indigo-400'
      : 'text-gray-700 dark:text-gray-300'
    "
    class="group-hover:text-indigo-600 dark:group-hover:text-indigo-400
transition-all duration-300 whitespace-nowrap"
  >Перевірка знань</span>
</div>
<div
  :class="
    isActive
      ? 'w-full bg-indigo-600 dark:bg-indigo-500'
      : 'w-0 bg-transparent group-hover:w-full group-hover:bg-indigo-600
dark:group-hover:bg-indigo-500'
    "
    class="h-0.5 rounded-full absolute -bottom-2 transition-all duration-
300 ease-out"
  ></div>
</router-link>

<div class="w-px h-8 bg-gray-300 dark:bg-gray-700"></div>

<router-link
  class="relative flex flex-col items-center group text-base xl:text-lg
font-medium transition-all duration-300"
  to="/map"
  v-slot="{ isActive }"
  >
  <div class="flex items-center gap-2 z-10">
    <svg
      xmlns="http://www.w3.org/2000/svg"
      class="w-5 h-5 transition-all duration-300 group-hover:text-indigo-
600 dark:group-hover:text-indigo-400"
      :class="
        isActive
          ? 'text-indigo-600 dark:text-indigo-400'
          : 'text-gray-700 dark:text-gray-300'
        "
      fill="none"
      viewBox="0 0 24 24"
      stroke="currentColor"
    >
    <path

```



```

        stroke-linecap="round"
        stroke-linejoin="round"
        stroke-width="2"
        d="M17.657 16.657L13.414 20.9a1.998 1.998 0 01-2.827 0l-4.244-
4.243a8 8 0 1111.314 0z"
      />
      <path
        stroke-linecap="round"
        stroke-linejoin="round"
        stroke-width="2"
        d="M15 11a3 3 0 11-6 0 3 3 0 016 0z"
      />
    </svg>
    <span
      :class="
        isActive
          ? 'text-indigo-600 dark:text-indigo-400'
          : 'text-gray-700 dark:text-gray-300'
        "
      class="group-hover:text-indigo-600 dark:group-hover:text-indigo-400
transition-all duration-300 whitespace-nowrap"
    >Карта допомоги</span>
  >
</div>
<div
  :class="
    isActive
      ? 'w-full bg-indigo-600 dark:bg-indigo-500'
      : 'w-0 bg-transparent group-hover:w-full group-hover:bg-indigo-600
dark:group-hover:bg-indigo-500'
    "
  class="h-0.5 rounded-full absolute -bottom-2 transition-all duration-
300 ease-out"
  ></div>
</router-link>
</div>
</nav>

<div class="flex items-center gap-2">
  <button
    @click="toggleTheme"
    class="p-2 rounded-full hover:bg-gray-100 dark: hover:bg-gray-800
transition-colors"
  >
  <svg
    v-if="isDarkMode"
    xmlns="http://www.w3.org/2000/svg"

```

```

class="h-5 w-5 text-yellow-400"
fill="none"
viewBox="0 0 24 24"
stroke="currentColor"
>
<path
  stroke-linecap="round"
  stroke-linejoin="round"
  stroke-width="2"
  d="M12 3v1m0 16v1m9-9h-1M4 12H3m15.364 6.364l-.707-.707M6.343 6.343l-
.707-.707m12.728 0l-.707.707M6.343 17.657l-.707.707M16 12a4 4 0 11-8 0 4 4 0 018 0z"
/>
</svg>
<svg
  v-else
  xmlns="http://www.w3.org/2000/svg"
  class="h-5 w-5 text-gray-700"
  fill="none"
  viewBox="0 0 24 24"
  stroke="currentColor"
>
<path
  stroke-linecap="round"
  stroke-linejoin="round"
  stroke-width="2"
  d="M20.354 15.354A9 9 0 018.646 3.646 9.003 9.003 0 0012 21a9.003
9.003 0 008.354-5.646z"
/>
</svg>
</button>

<button
  @click.stop="toggleMobileMenu"
  class="lg:hidden p-2 rounded-full hover:bg-gray-100 dark:hover:bg-gray-800
transition-colors"
>
<svg
  v-if="!isMobileMenuOpen"
  xmlns="http://www.w3.org/2000/svg"
  class="h-6 w-6 text-gray-700 dark:text-gray-300"
  fill="none"
  viewBox="0 0 24 24"
  stroke="currentColor"
>
<path
  stroke-linecap="round"
  stroke-linejoin="round"

```

```

        stroke-width="2"
        d="M4 6h16M4 12h16M4 18h16"
    />
</svg>
<svg
  v-else
  xmlns="http://www.w3.org/2000/svg"
  class="h-6 w-6 text-gray-700 dark:text-gray-300"
  fill="none"
  viewBox="0 0 24 24"
  stroke="currentColor"
>
  <path
    stroke-linecap="round"
    stroke-linejoin="round"
    stroke-width="2"
    d="M6 18L18 6M6 6l12 12"
  />
</svg>
</button>
</div>
</div>

<nav
  :class="
    isMobileMenuOpen ? 'max-h-96 opacity-100' : 'max-h-0 opacity-0'
  "
  class="lg:hidden overflow-hidden transition-all duration-300 ease-in-out"
>
  <div class="pt-4 pb-2 space-y-1">
    <router-link
      to="/guide"
      @click="closeMobileMenu"
      class="flex items-center gap-3 px-3 py-3 rounded-lg text-base font-medium
      transition-all duration-200 hover:bg-gray-100 dark:hover:bg-gray-800"
      v-slot="{ isActive }"
    >
      <svg
        xmlns="http://www.w3.org/2000/svg"
        class="w-5 h-5"
        :class="
          isActive
            ? 'text-indigo-600 dark:text-indigo-400'
            : 'text-gray-500 dark:text-gray-400'
        "
        fill="none"
        viewBox="0 0 24 24"

```

```

        stroke="currentColor"
      >
      <path
        stroke-linecap="round"
        stroke-linejoin="round"
        stroke-width="2"
        d="M12 6.253v13m0-13C10.832 5.477 9.246 5 7.5 5S4.168 5.477 3
6.253v13C4.168 18.477 5.754 18 7.5 18s3.332.477 4.5 1.253m0-13C13.168 5.477 14.754 5
16.5 5c1.747 0 3.332.477 4.5 1.253v13C19.832 18.477 18.247 18 16.5 18c-1.746 0-
3.332.477-4.5 1.253"
      />
    </svg>
    <span
      :class="
        isActive
        ? 'text-indigo-600 dark:text-indigo-400'
        : 'text-gray-700 dark:text-gray-300'
      "
    >
    Навчальні матеріали
  </span>
  <div
    v-if="isActive"
    class="ml-auto w-2 h-2 rounded-full bg-indigo-600 dark:bg-indigo-400"
  ></div>
</router-link>

<router-link
  to="/test"
  @click="closeMobileMenu"
  class="flex items-center gap-3 px-3 py-3 rounded-lg text-base font-medium
transition-all duration-200 hover:bg-gray-100 dark:hover:bg-gray-800"
  v-slot="{ isActive }"
>
  <svg
    xmlns="http://www.w3.org/2000/svg"
    class="w-5 h-5"
    :class="
      isActive
      ? 'text-indigo-600 dark:text-indigo-400'
      : 'text-gray-500 dark:text-gray-400'
    "
    fill="none"
    viewBox="0 0 24 24"
    stroke="currentColor"
  >
  <path

```

```

        stroke-linecap="round"
        stroke-linejoin="round"
        stroke-width="2"
        d="M9 5h7a2 2 0 00-2 2v12a2 2 0 002 2h10a2 2 0 002-2V7a2 2 0 00-2-2h-
2M9 5a2 2 0 002 2h2a2 2 0 002-2M9 5a2 2 0 012-2h2a2 2 0 012 2m-3 7h3m-3 4h3m-6-4h.01M9
16h.01"
    />
</svg>
<span
  :class="
    isActive
    ? 'text-indigo-600 dark:text-indigo-400'
    : 'text-gray-700 dark:text-gray-300'
  "
>
  Перевірка знань
</span>
<div
  v-if="isActive"
  class="ml-auto w-2 h-2 rounded-full bg-indigo-600 dark:bg-indigo-400"
></div>
</router-link>
<router-link
  to="/map"
  @click="closeMobileMenu"
  class="flex items-center gap-3 px-3 py-3 rounded-lg text-base font-medium
transition-all duration-200 hover:bg-gray-100 dark: hover:bg-gray-800"
  v-slot="{ isActive }"
>
  <svg
    xmlns="http://www.w3.org/2000/svg"
    class="w-5 h-5"
    :class="
      isActive
      ? 'text-indigo-600 dark:text-indigo-400'
      : 'text-gray-500 dark:text-gray-400'
    "
    fill="none"
    viewBox="0 0 24 24"
    stroke="currentColor"
  >
    <path
      stroke-linecap="round"
      stroke-linejoin="round"
      stroke-width="2"
      d="M17.657 16.657L13.414 20.9a1.998 1.998 0 01-2.827 0l-4.244-4.243a8
8 0 1111.314 0z"

```

```

    />
    <path
      stroke-linecap="round"
      stroke-linejoin="round"
      stroke-width="2"
      d="M15 11a3 3 0 11-6 0 3 3 0 016 0z"
    />
  </svg>
  <span
    :class="
      isActive
      ? 'text-indigo-600 dark:text-indigo-400'
      : 'text-gray-700 dark:text-gray-300'
    "
  >
    Карта допомоги
  </span>
  <div
    v-if="isActive"
    class="ml-auto w-2 h-2 rounded-full bg-indigo-600 dark:bg-indigo-400"
  ></div>
</router-link>
</div>
</nav>
</div>
</header>
<router-view />
<teleport to="body">
  <div
    v-if="activeRequest"
    class="fixed inset-0 flex items-center justify-center z-[9999] bg-black/50 px-4"
    @click="dismissNotification"
  >
    <div
      class="bg-white dark:bg-gray-800 rounded-xl shadow-xl max-w-md w-full p-6"
      @click.stop
    >
      <div class="flex items-center justify-between mb-4">
        <h3
          class="text-xl font-bold text-gray-900 dark:text-white flex items-center"
        >
          <span
            class="w-8 h-8 rounded-full bg-red-100 dark:bg-red-900 flex items-center
            justify-center text-red-600 dark:text-red-400 mr-2"
          >
            <svg
              xmlns="http://www.w3.org/2000/svg"

```

```

        class="h-5 w-5"
        fill="none"
        viewBox="0 0 24 24"
        stroke="currentColor"
      >
        <path
          stroke-linecap="round"
          stroke-linejoin="round"
          stroke-width="2"
          d="M12 9v2m0 4h.01M5 19h14a2 2 0 02-2V7a2 2 0 00-2-2H5a2 2 0 00-2
2v10a2 2 0 002 2z"
        />
      </svg>
    </span>
    Потрібна допомога!
  </h3>
  <button
    @click="dismissNotification"
    class="text-gray-500 hover:text-gray-700 dark:text-gray-400
dark: hover: text-gray-200"
  >
    <svg
      xmlns="http://www.w3.org/2000/svg"
      class="h-6 w-6"
      fill="none"
      viewBox="0 0 24 24"
      stroke="currentColor"
    >
      <path
        stroke-linecap="round"
        stroke-linejoin="round"
        stroke-width="2"
        d="M6 18L18 6M6 6l12 12"
      />
    </svg>
  </button>
</div>

<p class="text-gray-700 dark:text-gray-300 mb-4">
  {{
    activeRequest.message ||
    "Хтось потребує першої медичної допомоги поблизу!"
  }}
</p>

<div class="mb-4 h-40 rounded-lg overflow-hidden">
  <div ref="notificationMapContainer" class="w-full h-full"></div>

```



```

    limitToLast,
  } from "firebase/database";

const isMobileMenuOpen = ref(false);

function toggleMobileMenu() {
  isMobileMenuOpen.value = !isMobileMenuOpen.value;
}

function closeMobileMenu() {
  isMobileMenuOpen.value = false;
}

function handleClickOutside(event) {
  if (!event.target.closest("nav") && !event.target.closest("button")) {
    isMobileMenuOpen.value = false;
  }
}

onMounted(() => {
  document.addEventListener("click", handleClickOutside);
});

onUnmounted(() => {
  document.removeEventListener("click", handleClickOutside);
});

const isDarkMode = ref(
  localStorage.getItem("darkMode") === "true" ||
  window.matchMedia("(prefers-color-scheme: dark)").matches
);

function toggleTheme() {
  isDarkMode.value = !isDarkMode.value;
  localStorage.setItem("darkMode", isDarkMode.value);

  if (isDarkMode.value) {
    document.documentElement.classList.add("dark");
  } else {
    document.documentElement.classList.remove("dark");
  }
}

const notificationMapContainer = ref(null);
const notificationMap = ref(null);
const activeRequest = ref(null);
const userPosition = ref(null);
let L = null;
function getUserId() {
  let userId = localStorage.getItem("userId");

```

```

if (!userId) {
    userId = "user_" + Math.random().toString(36).substr(2, 9);
    localStorage.setItem("userId", userId);
}
return userId;
}

function loadHelpRequests() {
    if (!L) {

        return;
    }

    try {

        const helpRequestsRef = query(
            dbRef(database, "helpRequests"),
            orderByChild("status"),
            limitToLast(20)
        );

        onValue(helpRequestsRef, (snapshot) => {
            if (!snapshot.exists()) return;

            snapshot.forEach((childSnapshot) => {
                const requestId = childSnapshot.key;
                const requestData = childSnapshot.val();

                if (requestData && requestData.status === "active") {
                    const now = new Date().getTime();
                    const requestTime = new Date(requestData.createdAt).getTime();

                    if (now - requestTime < 30000 && requestData.userId !== getUserId()) {
                        showNotification(requestData);
                    }
                }
            });
        });
    } catch (error) {
        console.error("Помилка при завантаженні запитів:", error);
    }
}

function showNotification(requestData) {
    activeRequest.value = requestData;
}

```

```

if (navigator.geolocation && !userPosition.value) {
  navigator.geolocation.getCurrentPosition(
    (position) => {
      userPosition.value = {
        lat: position.coords.latitude,
        lng: position.coords.longitude,
      };

      createNotificationMap(requestData);
    },
    (error) => {
      console.error("Помилка геолокації:", error);

      createNotificationMap(requestData);
    }
  );
} else {

  createNotificationMap(requestData);
}

try {
  const audio = new Audio("/sounds/notification.mp3");
  audio.play();
} catch (e) {
  console.log("Неможливо відтворити звук сповіщення");
}

if ("Notification" in window && Notification.permission === "granted") {
  const notification = new Notification("Потрібна медична допомога!", {
    body:
      requestData.message ||
      "Хтось потребує першої медичної допомоги поблизу!",
    icon: "/img/help-icon.png",
  });

  notification.onclick = function () {
    window.focus();
    this.close();
  };
}
}

function createNotificationMap(requestData) {

```

```

setTimeout(() => {
  if (notificationMapContainer.value && L) {
    try {
      if (notificationMap.value) {
        notificationMap.value.remove();
        notificationMap.value = null;
      }

      notificationMap.value = L.map(notificationMapContainer.value).setView(
        [requestData.latitude, requestData.longitude],
        14
      );

      L.tileLayer("https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png", {
        attribution: "&copy; OpenStreetMap",
      }).addTo(notificationMap.value);

      const helpIcon = L.divIcon({
        className: "custom-marker",
        html: `<div class="help-marker-icon"></div>`,
        iconSize: [30, 30],
        iconAnchor: [15, 15],
      });

      L.marker([requestData.latitude, requestData.longitude], {
        icon: helpIcon,
      }).addTo(notificationMap.value);

      if (userPosition.value) {
        const userIcon = L.divIcon({
          className: "custom-marker",
          html: `<div class="user-marker-icon"></div>`,
          iconSize: [20, 20],
          iconAnchor: [10, 10],
        });

        L.marker([userPosition.value.lat, userPosition.value.lng], {
          icon: userIcon,
        }).addTo(notificationMap.value);

        L.polyline(
          [
            [userPosition.value.lat, userPosition.value.lng],
            [requestData.latitude, requestData.longitude],
          ],
          {
            color: "#4f46e5",
          }
        );
      }
    }
  }
});

```

```

        weight: 3,
        opacity: 0.7,
        dashArray: "5, 10",
    }
    ).addTo(notificationMap.value);
}
} catch (error) {
    console.error("Помилка при створенні міні-карти:", error);
}
}
}, 100);
}

```

```

function dismissNotification() {
    if (notificationMap.value) {
        notificationMap.value.remove();
        notificationMap.value = null;
    }
    activeRequest.value = null;
}

```

```

function navigateToRequest(requestData) {

```

```

    dismissNotification();

```

```

    if (!userPosition.value) {
        alert("Неможливо визначити ваше місцезнаходження для побудови маршруту.");
        return;
    }

```

```

    const destination = `${requestData.latitude},${requestData.longitude}`;
    window.open(
        `https://www.google.com/maps/dir/?api=1&origin=${userPosition.value.lat},${userPosition.value.lng}&destination=${destination}&travelmode=driving`,
        "_blank"
    );
}

```

```

provide("showNotification", showNotification);
provide("userPosition", userPosition);

```

```

onMounted(() => {

```

```

console.log("App component mounted");

if (isDarkMode.value) {
  document.documentElement.classList.add("dark");
}

if ("Notification" in window && Notification.permission !== "granted") {
  Notification.requestPermission();
}

import("leaflet")
  .then((module) => {
    console.log("Leaflet loaded in App.vue");
    L = module.default;

    if (navigator.geolocation) {
      navigator.geolocation.getCurrentPosition(
        (position) => {
          userPosition.value = {
            lat: position.coords.latitude,
            lng: position.coords.longitude,
          };
        },
        (error) => {
          console.error("Помилка геолокації:", error);
        }
      );
    }

    loadHelpRequests();
  })
  .catch((error) => {
    console.error("Error loading Leaflet in App.vue:", error);
  });
});

onUnmounted(() => {
  console.log("App component unmounted");
  if (notificationMap.value) {
    notificationMap.value.remove();
    notificationMap.value = null;
  }
}

```

```
});  
</script>  
  
<style>  
  
.user-marker-icon {  
  width: 100%;  
  height: 100%;  
  border-radius: 50%;  
  background-color: #4f46e5;  
  border: 2px solid white;  
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);  
}  
  
.help-marker-icon {  
  width: 100%;  
  height: 100%;  
  border-radius: 50%;  
  background-color: #ef4444;  
  border: 2px solid white;  
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  color: white;  
  font-size: 10px;  
  font-weight: bold;  
}  
  
.leaflet-container {  
  height: 100%;  
  width: 100%;  
}  
</style>
```

Додаток Б
Тести (Рівень складності: Середній)

Що першочергово треба зробити, прибувши на місце події з потерпілим?

Негайно надати допомогу потерпілому

Викликати швидку допомогу

Переконатися у власній безпеці

Знайти свідків події

Який номер екстреної медичної допомоги в Україні?

101

102

103

104

Яка правильна позиція рук при натисканнях на грудну клітку під час серцево-легеневої реанімації у дорослого?

Одна рука на середині грудної клітки

Дві руки на нижній частині грудної клітки

Дві руки на середині грудної клітки (на грудині)

Одна рука на верхній частині грудної клітки

Які ознаки артеріальної кровотечі?

- Темно-червона кров, яка повільно витікає з рани
- Яскраво-червона кров, яка витікає пульсуючим струменем
- Кров бордового кольору, яка витікає повільно та рівномірно
- Прозора рідина, яка сочиться з рани

Як довго слід охолоджувати опікову поверхню холодною водою?

- 1-2 хвилини
- 5-6 хвилин
- 10-15 хвилин
- До приїзду швидкої

Яка правильна послідовність дій при наданні першої допомоги постраждалому без свідомості?

- Перевірити дихання, відкрити дихальні шляхи, викликати швидку
- Викликати швидку, перевірити дихання, відкрити дихальні шляхи
- Переконався у безпеці, перевірити реакцію, відкрити дихальні шляхи, перевірити дихання, викликати швидку
- Переконався у безпеці, викликати швидку, почати СЛР

При якій глибині натискань на грудну клітку дорослого під час СЛР кровообіг найбільш ефективний?

- 2-3 см
- 3-4 см
- 5-6 см
- 7-8 см

Який основний принцип іммобілізації переломів?

- Фіксація тільки місця перелому
- Фіксація пошкодженої кістки та суглобів вище і нижче місця перелому
- Фіксація всієї кінцівки в витягнутому положенні
- Фіксація в положенні максимального розгинання

Який метод є найефективнішим для зупинки сильної кровотечі з кінцівки при наявності відповідних засобів?

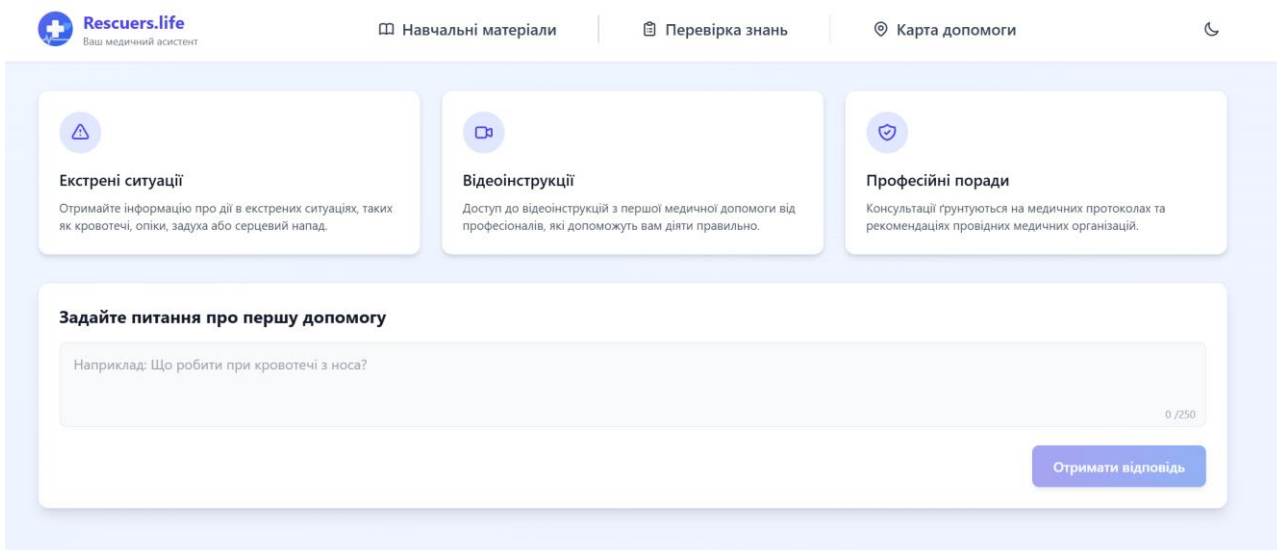
- Накладання джгута
- Пряме натискання на рану
- Накладання тиснучої пов'язки
- Застосування кровоспинного порошку

Додаток В
Посилання на Github

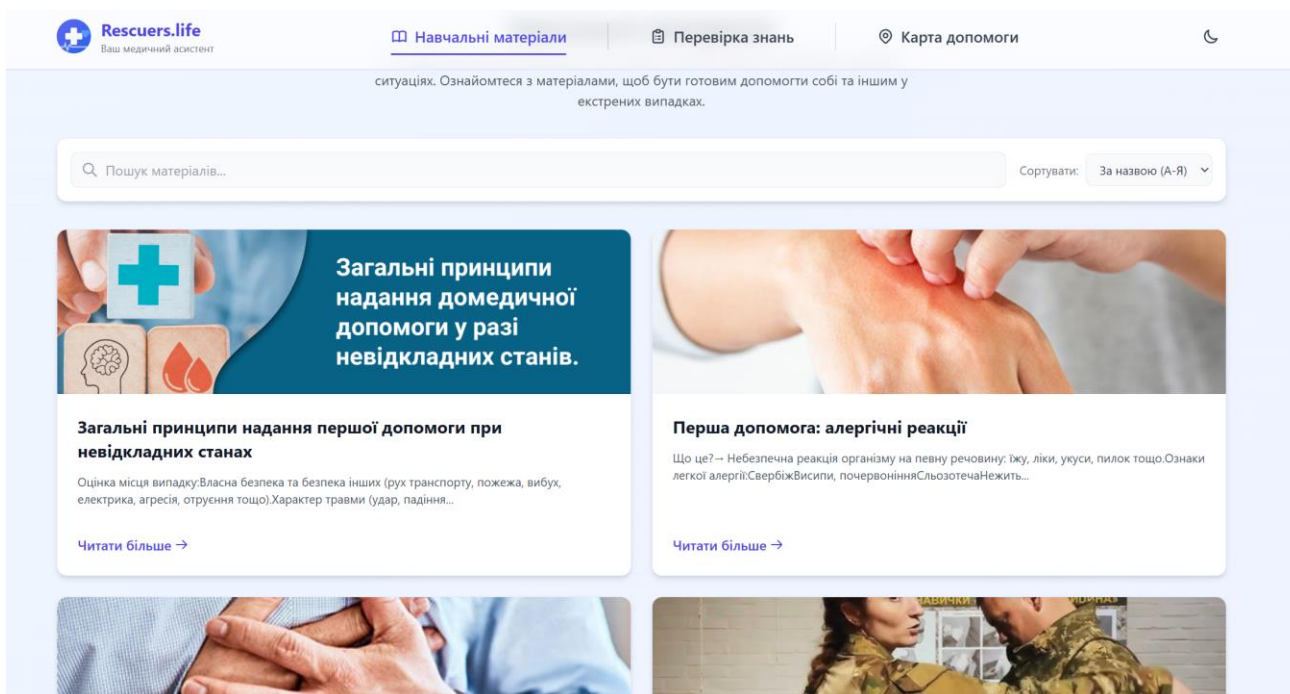


Додаток Г

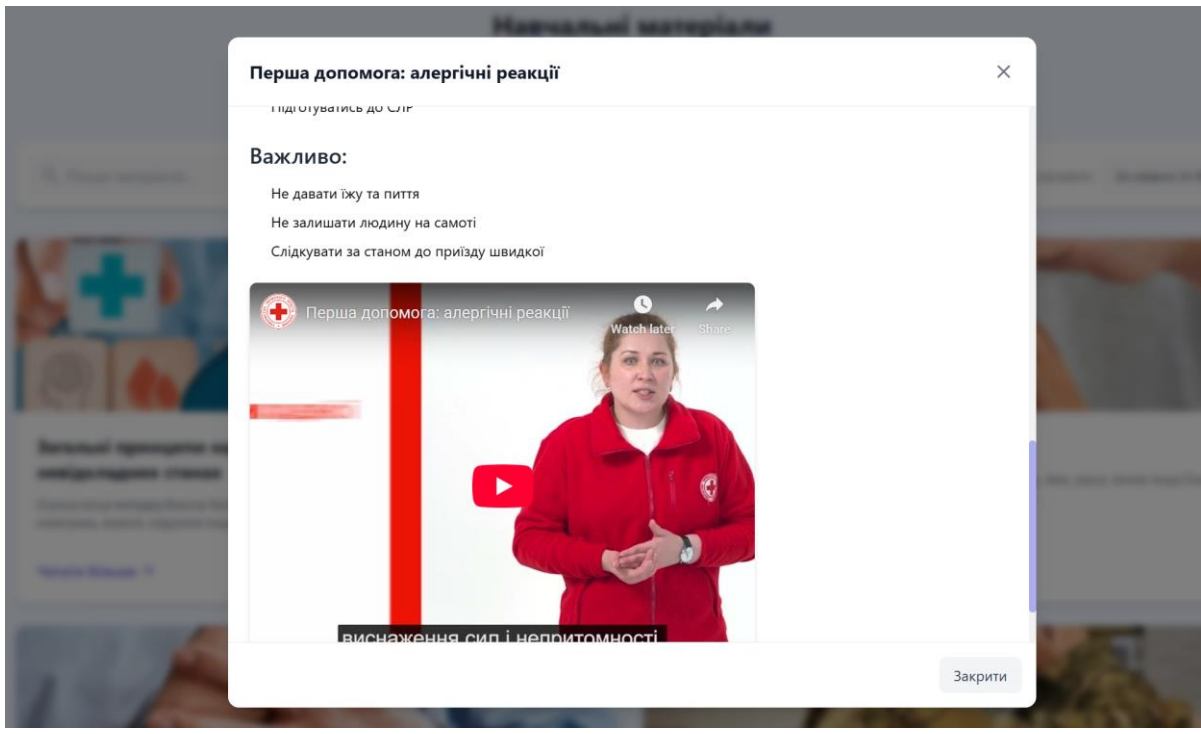
Динамічний UX/UI веб-додатку Rescuers.life



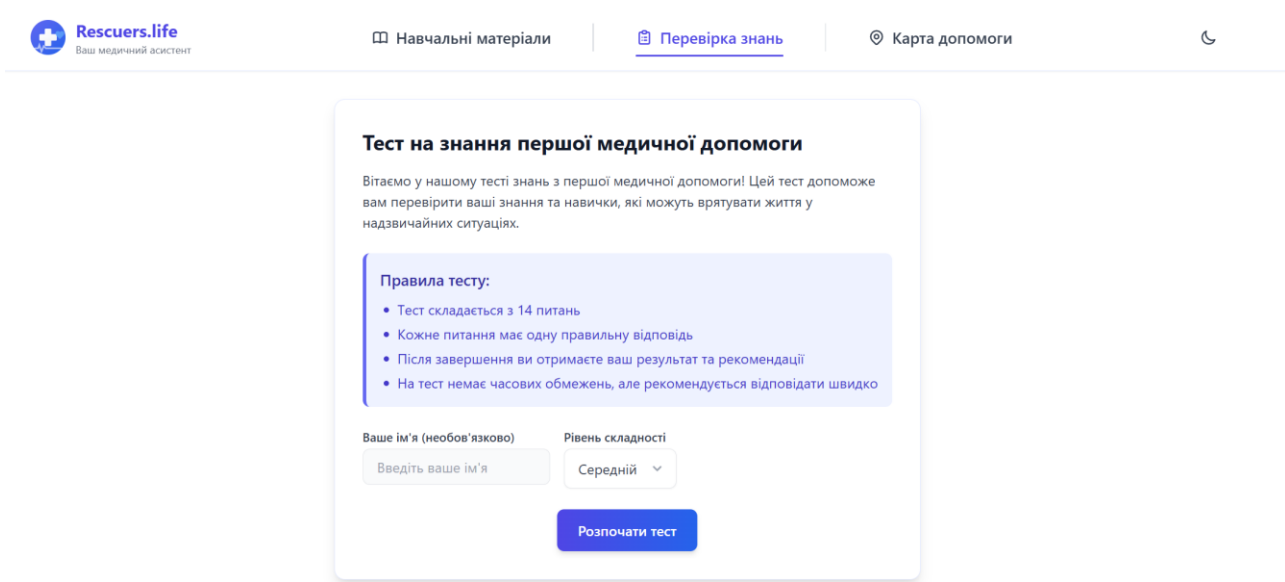
Головна сторінка веб додатку з онлайн помічником



Сторінка “Навчальні матеріали” веб-додатку Rescuers.life: каталог статей із пошуком та сортуванням



Приклад модального вікна з навчальним матеріалом та з відеоінструкцією



Сторінка “Перевірка знань”: Тест на знання першої медичної допомоги

Питання 1 з 9 11%

Який номер екстреної медичної допомоги в Україні?

101

102

103

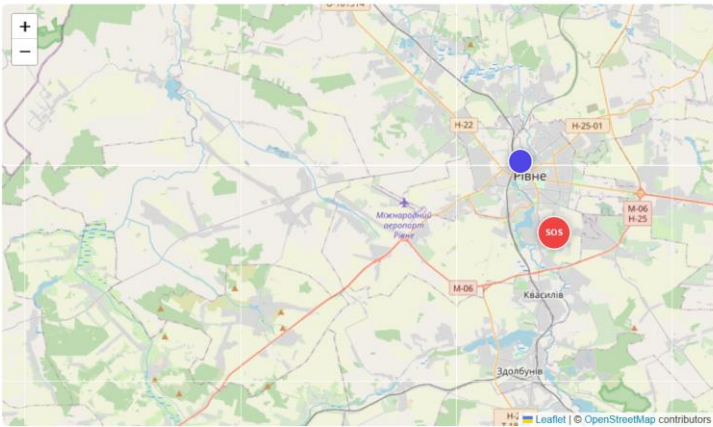
104

[Наступне >](#)

Вибір варіанту відповіді у тесті з першої медичної допомоги

Карта екстреної допомоги

Тут ви можете попросити про допомогу або допомогти іншим людям, які потребують першої медичної допомоги.



Leaflet | © OpenStreetMap contributors

[Мені потрібна допомога](#)

Сторінка “Карта екстреної допомоги” веб-додатку