

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Інформаційна веб-система підтримки українців під час міграції з
використанням мови програмування Python»

на здобуття освітнього ступеня бакалавра
зі спеціальності 126 Інформаційні системи та технології
(код, найменування спеціальності)
освітньо-професійної програми Інформаційні системи та технології
(назва)

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

(підпис)

Олександр СЛЄПЧЕНКО
(ім'я, ПРІЗВИЩЕ здобувача)

Виконав: здобувач вищої освіти група ІСД-41

Олександр СЛЄПЧЕНКО
(ім'я, ПРІЗВИЩЕ)

Керівник Доктор технічних наук, доцент Ірина СРІБНА
(ім'я, ПРІЗВИЩЕ)

Рецензент: _____
(ім'я, ПРІЗВИЩЕ)

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

Навчально-науковий інститут інформаційних технологій

Кафедра Інформаційних систем та технологій

Ступінь вищої освіти Бакалавр

Спеціальність 126 Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедри ІСТ

_____ Каміла СТОРЧАК

“ _____ ” _____ 2026 року

**З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Слепченку Олександр Романовичу

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Інформаційна веб-система підтримки українців під час міграції з використанням мови програмування Python

керівник кваліфікаційної роботи: Ірина СРІБНА Доктор технічних наук, доцент

(ім'я, ПРИЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від “20” лютого 2026 р. №51

2. Строк подання кваліфікаційної роботи «01» червня 2026 р.

3. Вихідні дані кваліфікаційної роботи:

1. Дані про міграцію українців після 2022 року (UNHCR, IOM).
2. Науково-технічна література з веб-розробки та штучного інтелекту.
3. Документація основних програм.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

1. Аналіз предметної області та огляд існуючих рішень.
2. Обґрунтування вибору технологічного стеку та побудова RAG.
3. Практична реалізація системи та тестування AI-асистента.

4. Перелік ілюстраційного матеріалу: *презентація*
5. Дата видачі завдання «20» лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/ п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області та огляд існуючих рішень	23.02-10.03.26	Виконано
2	Обґрунтування технологічного стеку та проєктування RAG-системи	11.03-24.03.26	Виконано
3	Проектування архітектури, БД та інтерфейсу	25.03-07.04.26	Виконано
4	Розробка серверної частини та інформаційного модулю	08.04-21.04.26	Виконано
5	Реалізація RAG-пайплайну та AI-асистента	22.04-12.05.26	Виконано
6	Тестування системи та оцінка ефективності	13.05-20.05.26	Виконано
7	Оформлення роботи та розробка презентації	21.05-30.05.26	Виконано

Здобувач вищої освіти _____ Олександр СЛЄПЧЕНКО
(підпис) (ім'я, ПРІЗВИЩЕ)

Керівник кваліфікаційної роботи _____ Ірина СРІБНА
(підпис) (ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступня бакалавр: 67 стор., 18 рис., 27 джерел.

Мета роботи – розробка інформаційної веб-системи UkrWay для підтримки українських мігрантів у Польщі, Німеччині та Великій Британії з інтегрованим AI-асистентом на основі аналізу Telegram-даних.

Об'єкт дослідження – процес інформаційного забезпечення українських мігрантів засобами сучасних веб-технологій та штучного інтелекту..

Предмет дослідження – методи та засоби проєктування інформаційної веб-системи з AI-асистентом на основі технології RAG.

Короткий зміст роботи. Проаналізовано інформаційні потреби українських мігрантів та виявлено недоліки існуючих рішень. Обґрунтовано вибір технологічного стеку. Спроектовано архітектуру системи та RAG-пайплайн. Розроблено веб-платформу з інформаційним модулем, AI-асистентом на базі Claude API і картою організацій допомоги. Проведено тестування та оцінку ефективності системи.

КЛЮЧОВІ СЛОВА: ІНФОРМАЦІЙНА ВЕБ-СИСТЕМА, УКРАЇНСЬКІ МІГРАНТИ, ШТУЧНИЙ ІНТЕЛЕКТ, RAG, DJANGO, CHROMADB, PYTHON, ВЕКТОРНИЙ ПОШУК, TELEGRAM.

ЗМІСТ

ЗМІСТ.....	7
ВСТУП.....	8
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОБҐРУНТУВАННЯ НЕОБХІДНОСТІ РОЗРОБКИ.....	12
1.1. Загальна характеристика об'єкту дослідження.....	12
1.2. Соціально-економічні передумови української міграції після 2022 року....	14
1.3. Аналіз інформаційних потреб українських мігрантів.....	16
1.4. Огляд існуючих рішень у сфері інформаційної підтримки мігрантів.....	19
Висновки до розділу 1.....	23
2. АНАЛІЗ ТЕХНОЛОГІЙ ТА ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ ВЕБ-СИСТЕМИ UKRWAY.....	25
2.1. Вимоги до інформаційної системи UkrWay.....	25
2.2. Аналіз та обґрунтування вибору технологічного стеку.....	28
2.3 Методи побудови RAG-системи та вибір інструментів.....	32
2.4. Проєктування архітектури системи та бази даних.....	39
2.5. Проєктування інтерфейсу користувача.....	44
Висновки до розділу 2.....	49
3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ВЕБ-СИСТЕМИ UKRWAY....	51
3.1. Реалізація серверної частини та інформаційного модулю.....	51
3.2. Реалізація RAG-пайплайну та AI-асистента на основі Claude API.....	57
3.3. Тестування системи та оцінка ефективності AI-асистента.....	66
Висновки до розділу 3.....	71
ВИСНОВКИ.....	73
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	76
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація).....	80

ВСТУП

Актуальність теми. Починаючи з лютого 2022 року Україна переживає найбільше, вимушене переміщення населення за часів незалежності. Мільйони людей опинилися за кордоном без достатнього розуміння того, як влаштоване життя в новій країні: які документи необхідно оформити в перші тижні, де шукати житло, як легалізувати своє перебування та знайти роботу. Інформація існує – але вона розпорошена між урядовими сайтами на різних мовах, волонтерськими Telegram-каналами, форумами та соціальними мережами. Людина у стресовій ситуації переїзду витрачає години на те, щоб зібрати до купи відповідь на елементарне практичне питання, або купує приватні консультації у людей, які займаються допомогою міграції в конкретній країні.

Паралельно у десятках великих Telegram-груп українських мігрантів щодня накопичуються тисячі повідомлень із живим досвідом: хтось вже пройшов реєстрацію в польському уряді, хтось розібрався з німецьким Jobcenter, хтось знайшов безкоштовне житло через британські програми. Цей масив знань практично не використовується системно – він існує у вигляді неструктурованого тексту, доступного лише тим, хто знає де шукати і має час читати сотні повідомлень.

Саме тут виникає потреба в інтелектуальній платформі, яка здатна агрегувати розрізнену інформацію, аналізувати реальний досвід спільноти та надавати користувачу точну і своєчасну відповідь. Розвиток великих мовних моделей і технологій векторного пошуку – зокрема архітектури RAG (Retrieval-Augmented Generation) – робить таке рішення технічно реалізованим вже сьогодні [1; 2]. Це і визначає актуальність цього дослідження.

Аналіз останніх досліджень і публікацій. Питання інформаційної підтримки мігрантів та застосування штучного інтелекту для роботи з неструктурованими текстовими даними досліджували ряд авторів. Зокрема, Lewis P. та ін. [1] у 2020 р. запропонували архітектуру RAG, що стала основою для систем генерації відповідей на базі зовнішніх баз знань. Питання застосування мультилінгвальних мовних моделей для обробки текстів різними мовами, включно з українською, розглядали Conneau A. та ін. [2]. Проблематику інформаційних потреб українських мігрантів та ефективності цифрових каналів підтримки аналізували у звітах УВКБ ООН [3] і Міжнародної організації з міграції [4]. Попри значний обсяг досліджень у суміжних напрямках, поза увагою залишається задача побудови платформи, що поєднує RAG-пошук по живих даних Telegram-спільнот із багатомодульним веб-інтерфейсом для конкретної аудиторії – українських мігрантів у Польщі, Німеччині та Великій Британії. Усунення цієї прогалини й становить мету цієї роботи.

Мета роботи – проектування та розробка інформаційної веб-системи UkrWay для підтримки українських мігрантів у Польщі, Німеччині та Великій Британії з інтегрованим AI-асистентом, що формує відповіді на основі аналізу повідомлень із Telegram-спільнот.

Для досягнення мети у роботі виконано такі завдання:

- дослідження соціально-економічних передумов та масштабів української міграції після 2022 року;
- аналіз інформаційних потреб цільової аудиторії та огляд існуючих рішень у цій сфері;
- обґрунтування вибору технологічного стеку для реалізації платформи;
- проектування архітектури системи, бази даних та RAG-пайплайну для обробки повідомлень із Telegram.
- розробка веб-сайту, AI-чату та інтерактивної карти ресурсів

– тестування функціональності розробленої системи та оцінка якості відповідей AI-асистента.

Об'єкт дослідження – процес інформаційного забезпечення українських мігрантів засобами сучасних веб-технологій та впровадження штучного інтелекту.

Предмет дослідження – методи та засоби проектування інформаційної веб-системи з інтегрованим AI-асистентом на основі технології RAG та аналізу Telegram-даних.

Методи дослідження. У процесі виконання роботи застосовувались: методи системного аналізу предметної області; порівняльний аналіз технологічних рішень і фреймворків; об'єктно-орієнтоване проектування; прототипування інтерфейсів; методи векторного пошуку та генерації тексту на основі контексту (Retrieval-Augmented Generation).

Наукова новизна одержаних результатів полягає у розробці підходу до формування відповідей для мігрантів на основі автоматичного аналізу повідомлень реальних Telegram-спільнот із застосуванням векторного пошуку (ChromaDB) та великої мовної моделі Claude. Відмінність від існуючих довідкових ресурсів – у використанні живого досвіду тисяч учасників міграційних груп як динамічної бази знань, що оновлюється без ручного редагування.

Практична значущість одержаних результатів. Розроблена платформа UkrWay є готовим до розгортання веб-рішенням, яке скорочує час пошуку критично важливої інформації для українців за кордоном. Інтеграція AI-асистента, Telegram-бота та інтерактивної карти ресурсів робить платформу багатофункціональним інструментом адаптації – придатним як для індивідуального використання, так і для громадських організацій.

Апробація результатів та публікації. Основні положення бакалаврської роботи апробовано шляхом участі у науково-практичних конференціях, що проводились на базі Державного університету інформаційно-комунікаційних технологій:

III Всеукраїнська науково-технічна конференція «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу» (Київ, ДУІКТ, 18 листопада 2025 р.).

VII Міжнародна науково-технічна конференція «Сучасний стан та перспективи розвитку IoT» (Київ, ДУІКТ, 20 квітня 2026 р.).

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОБҐРУНТУВАННЯ НЕОБХІДНОСТІ РОЗРОБКИ

1.1. Загальна характеристика об'єкту дослідження

Об'єктом дослідження у цій роботі є процес інформаційного забезпечення українських мігрантів засобами сучасних веб-технологій та штучного інтелекту. Конкретним практичним втіленням цього об'єкта слугує платформа UkrWay – веб-система, розроблена для надання практичної допомоги українцям у трьох країнах: Польщі, Німеччині та Великій Британії. Платформа орієнтована на людей, які вже перебувають за кордоном або лише планують виїзд і потребують структурованої, актуальної та зрозумілої інформації про умови проживання в новій країні.

Необхідність створення подібного ресурсу зумовлена безпрецедентним масштабом української міграції після повномасштабного вторгнення Росії у лютому 2022 року. За даними Управління Верховного комісара ООН у справах біженців (УВКБ ООН), станом на початок 2026 року за межами України перебувало близько 5,9 мільйона зареєстрованих біженців, з яких 5,3 мільйона – у країнах Європи [3]. Це одна з найбільших хвиль вимушеного переміщення в Європі з часів Другої світової війни. Найбільшу кількість українських біженців прийняли Німеччина (понад 1,2 млн осіб) та Польща (близько 1 млн осіб). Велика Британія також входить до числа європейських держав, що прийняли значну кількість українців після початку повномасштабного вторгнення. [3].

Кожна з цих країн має власну правову систему, соціальні програми та бюрократичні процедури, що суттєво ускладнює адаптацію. У Польщі – це процедура отримання PESEL і дозволу на проживання; у Німеччині – реєстрація за місцем проживання (Anmeldung), оформлення Bürgergeld та отримання Aufenthaltstitel; у Великій Британії – верифікація статусу за схемою Ukraine Family Scheme або Homes for Ukraine та подальше отримання біометричного посвідчення (BRP). Ці процедури не є взаємозамінними і потребують окремого вивчення для кожної країни.

Предметом дослідження є сукупність технічних і проектних рішень, що забезпечують функціонування платформи UkrWay. Система охоплює чотири ключові компоненти. Перший – інформаційний модуль зі структурованими статтями про документи, житло та ринок праці в кожній із трьох цільових країн; контент зберігається у базі даних і керується через адміністративну панель Django Admin. Другий – AI-асистент на базі Claude API (модель claude-sonnet-4-6) з реалізованою архітектурою RAG: перед кожним викликом мовної моделі система виконує векторний пошук у ChromaDB по масиву реальних повідомлень із Telegram-груп мігрантів і підставляє знайдені фрагменти як контекст у системний промпт. Та інтерактивна карта організацій та ресурсів на основі Leaflet.js із фільтрацією за країною та категорією допомоги.

Технічну основу платформи складає стек Python 3.14 і Django 5 для серверної частини, SQLite у розробці та PostgreSQL у продакшн-середовищі для реляційного зберігання даних, ChromaDB для векторної бази знань, бібліотека sentence-transformers (модель paraphrase-multilingual-MiniLM-L12-v2) для генерації мультилінгвальних ембедингів, а також Telethon для збору повідомлень із Telegram-груп. Такий підхід дозволяє реалізувати повноцінну RAG-систему: ембединги генеруються у пайплайні `build_knowledge_base`, зберігаються у ChromaDB, а під час запиту користувача виконується семантичний пошук і знайдені фрагменти реального досвіду підставляються у контекст відповіді.

Актуальність проекту визначається не лише кількісними показниками міграції, а й якісним дефіцитом інформації. Більшість наявних ресурсів або надто загальні, або застарілі, або розпорошені між десятками сайтів і Telegram-каналів. Людина, яка щойно приїхала до Варшави чи Берліна, змушена витратити години на пошук відповіді на конкретне питання. UkrWay вирішує саме цю проблему: замість пошуку по десятках джерел користувач отримує консолідовану відповідь – підкріплену реальним досвідом спільноти – в одному місці.

Таким чином, UkrWay вирішує не лише завдання довідкового сайту: ключова відмінність – у тому, що AI-асистент спирається не на узагальнені знання

моделі, а на реальний досвід тих, хто вже пройшов конкретні процедури в Польщі, Німеччині та Великій Британії.

1.2. Соціально-економічні передумови української міграції після 2022 року

Масштабне переміщення українців за кордон, що розпочалося у лютому 2022 року, не є однорідним явищем з єдиною причиною. Воно формується під впливом взаємопов'язаних чинників – безпекових, економічних і соціальних, – кожен з яких самотійно не пояснює повної картини. Розуміння цих передумов принципово важливе для обґрунтування потреби в інформаційній підтримці мігрантів, адже саме характер причин еміграції визначає, яка інформація їм потрібна і в який момент.

Повномасштабне вторгнення Росії 24 лютого 2022 року спричинило одночасний масовий виїзд, якого Європа не бачила з часів Другої світової війни. Уже на квітень 2023 року, після початку вторгнення Україну покинуло понад 6.3 млн осіб – це рекордний темп переміщення для будь-якої гуманітарної кризи в новітній історії [5]. Введення воєнного стану з перших годин конфлікту та заборона виїзду за кордон для чоловіків віком 18–60 років визначили гендерний склад біженців: більшість відсотків тих, хто виїхав у перші місяці, – жінки та діти [5]. Це суттєво вплинуло на специфіку інформаційних потреб аудиторії: жінки, нерідко вперше в житті самотійно вирішуючи юридичні та побутові питання в чужій країні, потребували детального практичного супроводу.

Однак зводити причини виїзду виключно до фізичної небезпеки було б спрощенням. Навіть у регіонах України, де активних бойових дій не велося, масштабне руйнування інфраструктури – передусім енергетичної – зробило нормальне життя вкрай складним. Починаючи з осені 2022 року масовані удари по об'єктах електро- та теплопостачання позбавили значну частину населення стабільного доступу до електрики, опалення та води [8]. За оцінками Світового банку, загальний обсяг збитків від руйнування інфраструктури сягнув 486 млрд дол. США – 2,8 рази від ВВП країни за 2023 рік [8]. Саме ця обставина стала

вторинним поштовхом до міграції для тих, хто в перші місяці залишився або повернувся.

Повномасштабна війна завдала глибокого удару по ринку праці України. Рівень зайнятості серед працездатного населення знизився з 73% до 67% від довоєнного показника [6]. Рівень безробіття серед нещодавно переміщених осіб усередині країни сягав 24%, тоді як серед тих, хто перебуває у вимушеному переміщенні понад рік, – 13% [7]. Реальна заробітна плата суттєво скоротилася у 2022 році, і лише у 2024 році відновилася до довоєнного рівня [8]. Для значної частини населення, особливо у приватному секторі та малому бізнесі, економічна нежиттєздатність залишатися в Україні стала самостійним мотивом для виїзду, незалежно від безпекового чинника.

Разом з тим за кордоном економічна інтеграція українців відбувається нерівномірно. За даними УВКБ ООН та МОМ, станом на 2025 рік більшість українських біженців у Європі є зайнятими [3], проте вони відстають від місцевих працівників за рівнем зайнятості, багато хто ще не адаптувався, не вивчив мови [3]. Серед основних бар'єрів – мовна несумісність, невизнання українських дипломів і кваліфікацій, а також відсутність доступного догляду за дітьми для жінок, які прибули в країну без допомоги чоловіка, або з інших причин. Кожен із цих бар'єрів має конкретний інформаційний вимір: мігрант, що не знає, як підтвердити диплом у Польщі або отримати місце у дитячому садку у Франкфурті, об'єктивно потребує структурованої, практичної, актуальної інформації.

Міграційна криза 2022–2026 років суттєво загострила давні демографічні проблеми України. За даними ІОМ, через вимушену міжнародну міграцію країна втратила понад 6,7 млн осіб [7]. Більшість тих, хто виїхав, – люди працездатного і репродуктивного віку, що посилює дефіцит людського капіталу в ключових секторах. Народжуваність в Україні у 2025 році впала до історичного мінімуму – близько 0,8–0,9 дитини на жінку [8]. Показовим є й те, що 57% українських біженців, опитаних УВКБ ООН у 2024 році, висловили бажання повернутися додому [3], однак реальне повернення стримується тривалою небезпекою, відсутністю житла і браком економічних перспектив.

Для цілей цієї роботи важливий окремий аспект: навіть ті, хто планує повернення, протягом місяців або років проживають за кордоном і потребують повноцінної інтеграції в систему соціального захисту, охорони здоров'я та зайнятості країни перебування. Саме в цей перехідний період гострота інформаційної потреби є найвищою.

Особливості трьох цільових країн платформи. Польща, Німеччина та Велика Британія – три найчисельніші центри концентрації українських мігрантів поза межами колишнього СРСР – суттєво різняться за умовами прийому. Польща, як найближча географічно, прийняла мігрантів у найкоротші терміни, однак тривалий статус тимчасового захисту без чіткої перспективи легалізації створює для них правову невизначеність [5]. Німеччина запропонувала широкий соціальний пакет (Bürgergeld, безкоштовні мовні курси, інтеграційні заходи), що зумовило її привабливість для тривалого перебування, проте складна бюрократична система стає суттєвим бар'єром [5]. Велика Британія реалізувала приватні схеми прийому (Homes for Ukraine, Ukraine Family Scheme) з унікальними вимогами до документів, що не мають аналогів у ЄС [5]. Кожна з цих моделей формує специфічний набір інформаційних потреб, які не перекриваються між собою.

Підсумовуючи, причини виїзду українців після 2022 року охоплюють кілька рівнів: безпекова загроза, руйнування економіки та соціальної інфраструктури, мовні й правові бар'єри в країнах прийому. Сукупність цих чинників визначає аудиторію, яка одночасно потребує актуальної практичної інформації і є найменш захищеною в сенсі доступу до неї. Детальний аналіз інформаційних потреб цієї аудиторії здійснено у наступному підрозділі.

1.3. Аналіз інформаційних потреб українських мігрантів

Проектувати інформаційну систему без розуміння того, що саме потрібно знати користувачу і коли – означає будувати довідник для самого себе. Саме тому аналіз інформаційних потреб передусь будь-яким технічним рішенням. У цьому підрозділі систематизовано потреби українських мігрантів за тематичними

категоріями та часовими етапами, з опорою на дані міжнародних досліджень 2024–2026 років.

Часова структура потреб. Перше, що обов’язково варто зафіксувати: потреби мігранта в перший тиждень і через рік – це майже різні завдання. У перші тижні після приїзду критичними є документи і статус: без реєстрації людина не може отримати ні соціальної допомоги, ні медичної страховки, ні легального доступу до ринку праці. Саме в цей момент вона найбільш вразлива – стрес, незнання мови, відсутність соціальних зв’язків. Від одного до шести місяців акцент зміщується: на перший план виходять пошук житла, робота, запис до лікаря. А от на тривалому горизонті – від пів-року і далі – з’являються якісно інші питання: підтвердження дипломів, перехід з тимчасового захисту на довгострокове проживання, зрештою рішення про повернення чи залишення. Кожен із цих етапів формує власний набір запитів. Вони не перекриваються.

Юридична легалізація. Це категорія, де помилка коштує найдорожче. Відсутність правового статусу блокує доступ до всіх інших послуг – без нього не буде ні роботи, ні страховки, ні банківського рахунку. За даними УВКБ ООН, у 2024–2025 роках серед найпоширеніших труднощів – проблеми з посвідченнями особи, реєстрація змін цивільного стану і втрата статусу після повернення до України навіть на короткий час [3]. Причому кожна з трьох цільових країн має повністю відмінну процедуру. У Польщі – PESEL і карта побиту. У Німеччині – Anmeldung, потім Aufenthaltstitel, окремо Bürgergeld. У Великій Британії – верифікація за Ukraine Family Scheme або Homes for Ukraine, і лише після цього BRP. Строки, пакети документів, наслідки прострочення – все різне. Знання польської процедури нічого не дає в Берліні.

Житло. Значна частина переселенців, за даними ІОМ, не здатні покрити непередбачені витрат, так як різниця між орендною ціною відрізняється в кожній країні. Це напевно найзатратніша частина переїзду, адаптації [8]. Це цифра, яка добре пояснює, чому питання житла є таким гострим. Опитування 2024 року охопило 15 979 осіб; 56% повідомили про потребу у фінансовій підтримці, 6% досі мешкали в колективних притулках [8]. Перехід з тимчасового притулку до

орендованого житла – це окремий процес, що потребує розуміння місцевого ринку, умов субсидування і вимог орендодавців. Не всі офіційні ресурси це пояснюють це зрозуміло і по-українськи.

Зайнятість. Дослідження УВКБ ООН січня 2026 року – понад 6 000 спостережень – показало: 57% біженців із України в Європі мають роботу [3]. Здається непогано. Але вони відстають від місцевих на 22 процентні пункти за рівнем зайнятості [3], і причини тут не в небажанні працювати. Мовний бар'єр, невизнання українських дипломів, брак дитячих садків для матерів-одиначок – це конкретні системні обмеження. Підтвердити диплом, знайти роботу через місцевий портал зайнятості, розібратись у трудовому договорі чужою мовою – кожен із цих кроків потребує чіткої покрокової інструкції, а не загальних порад.

Медицина. Ситуація з охороною здоров'я окремо показова. Дослідження RefuHealthAccess (2023–2024, 550 респондентів) зафіксувало: 82,9% мали потребу в стоматологічній допомозі, 81,6% – у рецептурних ліках, 78,4% лікувалися від гострих станів, 64,0% мали хронічні захворювання [9]. Чверть опитаних поверталася в Україну за послугами, яких не могли отримати за кордоном [9]. Головні бар'єри – черги (64%), гроші, і – увага – незнання того, як взагалі влаштована система медичного страхування в країні перебування. Останнє вирішується інформацією. Не грошима, не черговістю – просто знанням того, що робити.

Канали, якими мігранти шукають відповіді. Квартальний звіт УВКБ ООН за 2024 рік зафіксував топ-5 каналів: телефонна лінія (32%), Telegram (32%), особисті зустрічі (28%), офіційні сайти (26%), Viber (25%) [10]. Telegram – нарівні з гарячою лінією. Це не випадковість – у великих міграційних групах у месенджері щодня з'являються сотні реальних повідомлень із живим досвідом. Проблема в іншому: ці повідомлення неструктуровані. Актуальна порада сусідить із застарілою, перевірена інформація – з чуткою. Знайти потрібне серед тисяч повідомлень – це окреме завдання, яке більшість людей просто не виконує.

Саме цей розрив – Telegram як основний канал проти повної неструктурованості його вмісту – є прямим обґрунтуванням RAG-підходу,

реалізованого в платформі UkrWay. Систематизація реальних повідомлень мігрантських спільнот і доступ до них через семантичний пошук перетворює хаотичний досвід спільноти на структуровану базу знань.

Таким чином, виявлено чотири ключові тематичні категорії інформаційних потреб: легалізація, житло, зайнятість і медицина. Кожна з них специфічна для окремого етапу міграції і для конкретної країни перебування. Наявні канали – передусім Telegram – охоплюють потребу, але не закривають її якісно. Аналіз існуючих цифрових рішень у цій сфері здійснено у підрозділі 1.4.

1.4. Огляд існуючих рішень у сфері інформаційної підтримки мігрантів

Перш ніж проєктувати власну систему, треба зрозуміти, що вже існує і де закінчується «достатньо». У цьому підрозділі розглянуто основні доступні рішення – урядові портали, україномовні тематичні ресурси і Telegram-спільноти – та виявлено їхні обмеження для українських мігрантів.

Visit Ukraine (visitukraine.today). Серед україномовних ресурсів це, мабуть, найширше охоплення тематик серед тих, що реально використовуються. Портал створено приватною ініціативою з інформаційною підтримкою від державних органів; він містить дані про правила в'їзду до більшості країн – не лише європейських [11]. Окрім міграційного блоку, є розділи про медичне страхування, юридичні консультації та транспортні послуги [11]. Для людини на етапі підготовки до виїзду – дійсно корисна точка старту.

Та варто зайти глибше – і картина змінюється. Платформа побудована навколо платних послуг: страхування, візова підтримка, юридичний супровід. Покрокових інструкцій із реальних інтеграційних процедур у країні перебування – того ж PESEL, Bürgergeld чи BRP – тут майже немає, або вони подані так фрагментарно. Жодних елементів AI чи персоналізованого пошуку у безкоштовному доступі також не реалізовано.

Офіційні урядові портали країн перебування. gov.pl/web/ua – окремий портал Польщі для українських громадян [12]. Ресурс повністю перекладено українською, тут є юридично точна інформація: тимчасовий захист, PESEL,

дозволи на проживання, ринок праці, медицина, освіта, соціальні виплати [12]. Звучить добре. На практиці – навігація підпорядкована логіці державної бюрократії, а не людині з конкретним запитом. Три-чотири рівні вкладеності заради відповіді на елементарне питання – норма, а не виняток. Втім, це спільна риса всіх урядових порталів, не специфічна проблема польського.

На gov.uk зібрано колекцію офіційних інструкцій щодо схем Ukraine Family Scheme і Homes for Ukraine [13]. Частину матеріалів перекладено – зокрема «Welcome: a guide for Ukrainians arriving in the UK» [13]. Решта залишається англomовною і написана для тих, хто вже розуміється на британській системі – не для тих, хто щойно приїхав.

Germany for Ukraine (germany4ukraine.de). Тут ситуація помітно краща. Це офіційний портал федерального уряду Німеччини, повністю перекладений українською [14]. Охоплює все що потрібно: в'їзд, реєстрацію (Anmeldung), Aufenthaltstitel, житло, Bürgergeld, медицину, освіту і побут у Німеччині [14]. Один із найповніших офіційних ресурсів серед трьох цільових країн – і при цьому має ту саму структурну ваду, що й усі попередні. Статичний контент. Ніякого інтерактивного пошуку. Жодного живого досвіду спільноти. Нормативна база оновлюється раз на кілька місяців – а практика мігрантів змінюється щодня.

Telegram-спільноти. За даними квартального звіту УВКБ ООН, Telegram використовують 32% опитаних мігрантів – нарівні з телефонною лінією [10]. Є чому. У великих тематичних групах циркулює те, чого немає на жодному офіційному сайті: реальний досвід людей, які вже пройшли конкретну процедуру. Там, де регламент розходиться з фактичною практикою установи – у термінах розгляду, неочевидних вимогах до документів, поведінці конкретного відділку – Telegram нерідко точніший за будь-який урядовий портал.

Зворотний бік відомий, але варто показати його на особистому прикладі. Великі новинні канали – зокрема «Україна Online: Новини | Політика» (понад 1,4 млн підписників [19]) – часто публікують міграційні новини у форматі, що спотворює суть офіційних рішень, тобто використовують багато клікбейту. 27 березня 2026 р. уряд Норвегії ухвалив поправку до постанови про тимчасовий

колективний захист: чоловіки 18–60 років, за окремими винятками, більше не отримуватимуть колективний захист автоматично, натомість можуть подавати заяву на притулок у загальному порядку [17]. Норма набрала чинності 4 травня 2026 р. окремою постановою від 17 квітня (Forskrift nr. 600) [18]; стосується вона тільки нових заявників, а вже виданий захист – не скасовується. Того ж дня в українських каналах з'явилися заголовки про «скасування програми тимчасового захисту в Норвегії для 18-60 в той же день, коли вийшла новина, без переліку винятків, без згадки про відкладену дату набрання чинності. Пересічний читач отримав враження і скоріше всього, почав шукати інформацію на офіційних сайтах.

Таких прикладів чимало – навколо Bürgergeld у Німеччині, BRP у Великій Британії, продовження статусу тимчасового захисту в Польщі відбувалося те саме. Живий досвід у Telegram дуже цінний. Нюанс у тому, що він перемішаний із непідтвердженою інформацією без жодного механізму верифікації. Власне цей розрив – між масовістю каналу і відсутністю структури – і є точкою, де RAG-підхід UkrWay пропонує конкретне рішення: реальні повідомлення мігрантських спільнот, але відфільтровані через векторний пошук і подані лише у разі достатньої релевантності до запиту.

З-поміж міжнародних ініціатив Signpost від International Rescue Committee стоїть окремо – це найближчий існуючий аналог до того, що пропонує UkrWay. Платформа охоплює понад 30 країн і нараховує близько 20 млн зареєстрованих користувачів [16]. У листопаді 2024 р. IRC запустив пілот AI-асистента у чотирьох країнах; система підтримувала 11 мов і використовувала моделі від OpenAI, Anthropic і Google [15]. Шестимісячний звіт за результатами пілоту опубліковано у червні 2025 р. [16].

Ключове обмеження в самій логіці платформи: контент верифікується централізованими волонтерськими командами. Швидкозмінний досвід конкретних мігрантських спільнот за такої моделі відобразити неможливо – відповіді завжди спиратимуться на статичну базу знань, а не на те, що люди пишуть один одному прямо зараз.

Узагальнення. Наведений аналіз дозволяє систематизувати розглянуті рішення за ключовими параметрами. У таблиці 1.1 подано порівняльний огляд платформ за п'ятьма критеріями, що є найбільш значущими для цільової аудиторії.

Таблиця 1.1

Порівняльний аналіз існуючих рішень інформаційної підтримки мігрантів

Платформа / ресурс	Охоплення країн	Українська мова	AI-асистент	Досвід спільноти	Єдина точка входу
Visit Ukraine	Більшість	Так	Ні	Ні	Частково
gov.pl/web/ua (Польща)	1	Так	Ні	Ні	Ні
gov.uk (UK)	1	Частково	Ні	Ні	Ні
Germany4Ukraine (DE)	1	Так	Ні	Ні	Ні
Telegram-спільноти	Тематичні	Так	Ні	Так	Ні
Signpost / United for Ukraine	Близько 30	Так	Пілот 2024–2025	Ні (статична база)	Ні
UkrWay (пропонована)	3	Так	Так (RAG-підхід)	Так (Telegram→ChromaDB)	Так

Примітка. «Пілот 2024–2025» – шестимісячний тест (вересень 2024 – лютий 2025) без повноцінного виходу в продакшн на момент написання роботи [16].

З таблиці 1.1 видно: жодне з розглянутих рішень не закриває всі п'ять критеріїв одночасно. Урядові портали дають юридичну точність, але втрачають у доступності й охопленні мов. Telegram забезпечує живий досвід – без верифікації. Signpost найближчий до комплексного рішення, проте статична база знань і відсутність специфіки трьох обраних країн в одному інтерфейсі залишаються

невирішеними. Visit Ukraine і Germany4Ukraine орієнтовані на реального мігранта, але кожен лише у своєму тематичному або географічному фокусі.

Виявлений дефіцит і визначає проєктне рішення UkrWay: україномовна платформа, що в одному інтерфейсі поєднує структурований інформаційний модуль для трьох країн, RAG-асистента за допомогою штучного інтелекту на основі даних Telegram-чатів.

Висновки до розділу 1

Станом на початок 2026 року за межами України перебуває близько 5,9 млн зареєстрованих біженців, з яких 5,3 млн – у Європі [3]. Польща, Німеччина та Велика Британія – три основні країни концентрації – суттєво різняться за правовими системами, процедурами легалізації та соціальними пакетами. Уніфікованого інформаційного рішення для всіх трьох одразу не існує: кожна країна потребує окремого змістового наповнення. Масштаб і характер міграції після лютого 2022 року підтверджують стійкий і довготривалий попит на структуровану інформаційну підтримку.

Причини виїзду не зводяться до однієї – безпекової. Руйнування енергетичної інфраструктури (збитки – 486 млрд дол. США [8]), падіння рівня зайнятості з 73% до 67% [6], безробіття серед нещодавно переміщених на рівні 24% [7] сформували аудиторію, вразливу одночасно у фінансовому, правовому і побутовому вимірах. Близько 90% тих, хто виїхав у перші місяці – жінки та діти [5]: люди, які нерідко вперше самостійно вирішують юридичні питання в чужій країні. Звідси специфіка запитів: документи на тимчасовий захист, догляд за дітьми, медицина, пошук роботи.

Інформаційні потреби мігрантів охоплюють чотири категорії – легалізація, житло, зайнятість і медицина – і змінюються залежно від тривалості перебування. PESEL, Bürgergeld і BRP – три окремі системи без спільних елементів. Telegram залишається одним із головних каналів пошуку інформації (32% опитаних [10]), проте сам по собі проблеми не вирішує, повідомлення там неструктуровані,

достовірна інформація сусідить із хибною без жодного механізму розрізнення, витрачається час на те, щоб написати в чат, дочекатись відповіді.

Огляд існуючих платформ виявив спільний недолік: жодна не поєднує україномовний інтерфейс, охоплення трьох цільових країн, AI-асистента і доступ до реального досвіду мігрантських спільнот в одному місці. Урядові портали точні юридично, але незручні структурно. Signpost охоплює понад 30 країн і близько 20 млн користувачів [16] і все одно спирається на статичну базу знань. Telegram дає живе, але хаотичне. Цей набір незакритих потреб обґрунтовує необхідність створення платформи UkrWay.

Для вирішення виявленої проблеми обрано архітектуру RAG це генерація відповідей на основі семантично релевантних фрагментів реальних повідомлень із Telegram-спільнот мігрантів [1]. Мультилінгвальні ембединги (paraphrase-multilingual-MiniLM-L12-v2) і векторна база ChromaDB забезпечують коректну роботу з україномовним контентом [2]. Система не вигадує відповідь – вона знаходить найближче до запиту з того, що реальні люди вже написали і перевірили на практиці. Детальний розгляд технологічних засобів реалізації здійснено у Розділі 2.

2. АНАЛІЗ ТЕХНОЛОГІЙ ТА ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ ВЕБ-СИСТЕМИ UKRWAY

2.1. Вимоги до інформаційної системи UkrWay

Перед реалізацією проекту, нам потрібно розуміти, те що саме система має робити і її цільову аудиторію. Результат з аналізу, який було проведено у першому розділі, дозволяє зрозуміти опис проблеми, та як правильно її формулювати, аналізувати та вирішувати.

Функціональні вимоги випливають з виявлених у підрозділі 1.3 інформаційних потреб українців-мігрантів, чітко визначено такі функціональні параметри до сайту(табл. 2.1).

Таблиця 2.1

Функціональні вимоги до інформаційної системи UkrWay

№	Вимога	Обґрунтування
Ф1	Структурований інформаційний модуль із статтями по чотирьох категоріях (документи, житло, робота, медицина) для трьох країн	Мігранти потребують практичної інформації за конкретними темами в конкретній країні [1.3]
Ф2	AI-асистент, що відповідає українською мовою на запити щодо міграції	32% мігрантів шукають інформацію через месенджери і цифрові канали [10]; потрібна інтерактивна підтримка
Ф3	Інтеграція RAG-пошуку по реальних повідомленнях Telegram-спільнот	Живий досвід мігрантів є найціннішим джерелом [1.3], але неструктурованим – RAG вирішує це
Ф4	Інтерактивна карта організацій допомоги з фільтрацією по країні та категорії	Мігрантам потрібен доступ до верифікованих організацій підтримки в країні перебування

Продовження таблиці 2.1

№	Вимога	Обґрунтування
Ф5	Збереження історії діалогу з AI-асистентом у межах сесії	Повторне введення контексту при кожному повідомленні знижує зручність використання
Ф6	Адміністративна панель для управління контентом без програмування	Актуальність інформації потребує регулярного оновлення без технічних навичок
Ф7	Коректна робота системи за відсутності RAG-бази знань (graceful fallback)	Система повинна бути працездатною на етапі до наповнення бази Telegram-даними

Аналіз з табличних даних показує, що основні аспекти охоплюють чотири напрями це інформаційний (Ф1), інтелектуальний (Ф2, Ф3), навігаційний (Ф4) і системний (Ф5–Ф7). Кожен пункт має своє обґрунтування, використовуючи інформацію з підрозділів 1.3 і 1.4, де було проведено потрібний, детальний аналіз. В представленій таблиці нас цікавить більше основні компоненти роботи. До прикладу, без Ф7 все ламається, так як база знань постійно має бути наповненою даними, щоб було з чим працювати. Ф2 і Ф3 взагалі потребують доступ до сторонніх сервісів, такі як Telegram та Claude. Якщо один з них перестає глобально працювати, або десь API-key не проплачений, все, алгоритм роботи показує просто помилку. Це чіткі приклади, які продумувались ще перед створенням сайту.

Досліджено також нефункціональні параметри роботи. Окрім функціональних, система повинна відповідати правилам доступності, продуктивності та зручності використання для юзерів (табл. 2.2).

Таблиця 2.2

Нефункціональні вимоги до інформаційної системи UkrWay

№	Вимога	Конкретний показник
НФ 1	Мова інтерфейсу – виключно українська	100% елементів UI і контенту – українською
НФ 2	Мобільна адаптивність	Коректне відображення на екранах від 320 рх
НФ 3	Відсутність реєстрації для базового доступу	Користувач отримує інформацію без створення акаунту
НФ 4	Безкоштовний доступ до всіх функцій	Жодних платних підписок для кінцевого користувача
НФ 5	Час відповіді AI-асистента	До 10-15 секунд для стандартного запиту
НФ 6	Збереження сесії чату	30 днів (SESSION_COOKIE_AGE = 86400×30 с)
НФ 7	Масштабованість бази даних	SQLite у розробці, PostgreSQL у продакшні; векторна БД – ChromaDB (PersistentClient)
НФ 8	Стійкість до помилок API	Три рівні graceful fallback: немає RAG → немає ключа → помилка API

Проаналізуємо також нефункціональні, які можна розбити на два блоки. Перший, важливий – доступність (НФ1–НФ4): україномовний інтерфейс, відсутність особистих акаунтів і безкоштовність привертає увагу аудиторії, яка перебуває в стресовій ситуації переїзду. Причому вона навіть може використовувати це в моменті, коли їй треба терміново вирішити питання. Другий це надійність (НФ5–НФ8). Варто відмітити на НФ5 окремо: щодо порогу у 10 секунд. RAG-пошук у ChromaDB і виклик Claude API додають затримку

порівняно зі статичним контентом, причому іноді вона може бути довгою. Але саме це забезпечує найбільш точну відповідь на основі реального досвіду мігрантів, а не загальну довідку. Параметр НФ2 (від 320 рх) аргументований тим, що більшість мігрантів використовують смартфони як основний пристрій доступу до мережі інтернету за кордоном.

Результатом аналізу даного підрозділу визначено основні показники, які можна перевірити в роботі системи, які уважно аргументовані і також відображають деякі нюанси по роботі з системою, які мають зовнішній вплив. Те, які пункти ми пояснили впливає опис з використаними технологіями, детально розглянемо в підрозділі 2.2.

2.2. Аналіз та обґрунтування вибору технологічного стеку

Вибір технологій дуже важливий момент для побудови якісного проектного рішення, яке визначає можливості системи довгостроково. Кожен компонент стеку UkrWay обирався з урахуванням всіх вимог, які сформульовані в 2.1 підрозділі. В даному випадку проаналізовано аргументи щодо нашого стеку, також розглянуто альтернативи.

Для реалізації серверної частини розглядались три популярні Python-фреймворки: Django (наш вибір), Flask, FastAPI. Кожен з них відіграє важливу роль у створенні веб-додатків, розглянуто можливості кожну з них (табл 2.3).

Таблиця 2.3

Порівняльний аналіз Python фреймворків для серверної частини

Критерій	Django 5.x	Flask	FastAPI
Вбудований ORM	Так	Ні	Ні
Адміністративна панель	Вбудована	Стороння	Стороння
Сесійний механізм	Вбудований	Розширення	Розширення
Асинхронність	Часткова	Часткова	Повна (ASGI)

Продовження таблиці 2.3

Критерій	Django 5.x	Flask	FastAPI
Продуктивність (req/s)	~435 req/s	~315 req/s	~444 req/s
Крива навчання	Середня	Низька	Середня
Придатність для CMS-проектів	Висока	Середня	Низька

Дані з таблиці 2.3 наголошують на продуктивності FastAPI, судячи з статистики, але для інформаційного-веб ресурсу це не головний аспект [20]. Варто звернути увагу також на критичні параметри, такі як наприклад вбудована адмін-панель. Вона відіграє важливу роль у проєкті, так як вона дозволить адміністратору сайту, наповнювати контент без знання в програмуванні, що закриває функціональну вимогу, а саме управління контентом без програмування. Жоден із запропонованих аналогів не дає таку можливість зі старту [22]. Вбудований ORM і сесійний механізм суттєво скорочує час розробки [23]. Тому в цьому випадку обрана саме Django, не як найкращий варіант, а більш як оптимальніший до проєкту.

Щодо клієнтської частини, то Ключове рішення тут це відмова від SPA-архітектури (Single Page Application) на користь серверного рендерингу з HTMX. Пояснення це те, що UkrWay не є складним інтерактивним додатком з багатим клієнтським станом. Проєкт переважно інформаційний ресурс з кількома динамічними елементами – чат і карта. React чи Vue вирішували б ці задачі, але ціною значного підвищення складності стеку та реалізації.

HTMX важить 14 КБ у стислому вигляді, React разом з ReactDOM понад 42КБ [21]. За даними з бенчмарків, для CRUD-додатків HTMX дозволяє завантажувати контент на 40-60% швидше при відкритті сторінки [21]. Для мігрантів важливий пункт, так як в основному вони використовують мобільний інтернет. Також, HTMX інтегрується з Django Templates, тобто розробка

залишається в єдиному середовищі. Tailwind CSS підключено через CDN, що також спрощує потребу у складній пайплайні збірки.

Детально про базу даних. Для зберігання реляційних даних (статті, сесії, ресурси) розглядалось два підходи: реляційна СУБД і документоорієнтована MongoDB (табл. 2.4).

Таблиця 2.4

Порівняльний аналіз підходів до зберігання даних

Критерій	SQLite / PostgreSQL	MongoDB
Тип даних	Реляційні, структуровані	Документи (JSON)
Підтримка Django ORM	Повна	Часткова (через djongo)
Міграції Django	Вбудовані	Не підтримуються
Складність налаштування	Низька	Середня
Придатність для UkrWay	Висока	Надмірна

З таблиці 2.4 видно, що MongoDB орієнтована на неструктуровані або слабоструктуровані дані – JSON-документи зі змінною схемою. Дані UkrWay структуровані й передбачувані: статті мають фіксовані поля, сесії – чітку схему, ресурси, координати і категорії. Застосовувати MongoDB тут надмірність без жодної практичної користі. SQLite використовується в середовищі розробки через нульову конфігурацію. PostgreSQL для майбутнього продакшн через стабільність і масштабованість. Обидва повністю підтримуються Django ORM без жодних додаткових бібліотек. Для векторного зберігання ембедингів обрано ChromaDB – окрема база, що зберігає чанки Telegram-повідомлень і забезпечує семантичний пошук, детально про цей компонент розглядається у підрозділі 2.3.

Допоміжні компоненти. Окрім ключових технологій, у стеку присутні кілька допоміжних інструментів. Whitenoise забезпечує роздачу статичних файлів безпосередньо з Django без потреби в окремому веб-сервері, це спрощує в подальшому деплой і закриває нефункціональний параметр НФ5 (час відповіді).

Python-dotenv відповідає за безпечне зберігання змінних середовища:

ANTHROPIC_API_KEY, TELEGRAM_BOT_TOKEN та інші важливі, приватні дані не потрапляють до репозиторію. Бібліотеки crispy-forms і crispy-tailwind інтегрують Django-форми з Tailwind CSS, забезпечуючи консистентний вигляд елементів інтерфейсу без ручного написання CSS-класів.

Загальну схему взаємодії всіх компонентів платформи наведено на рис. 2.1.

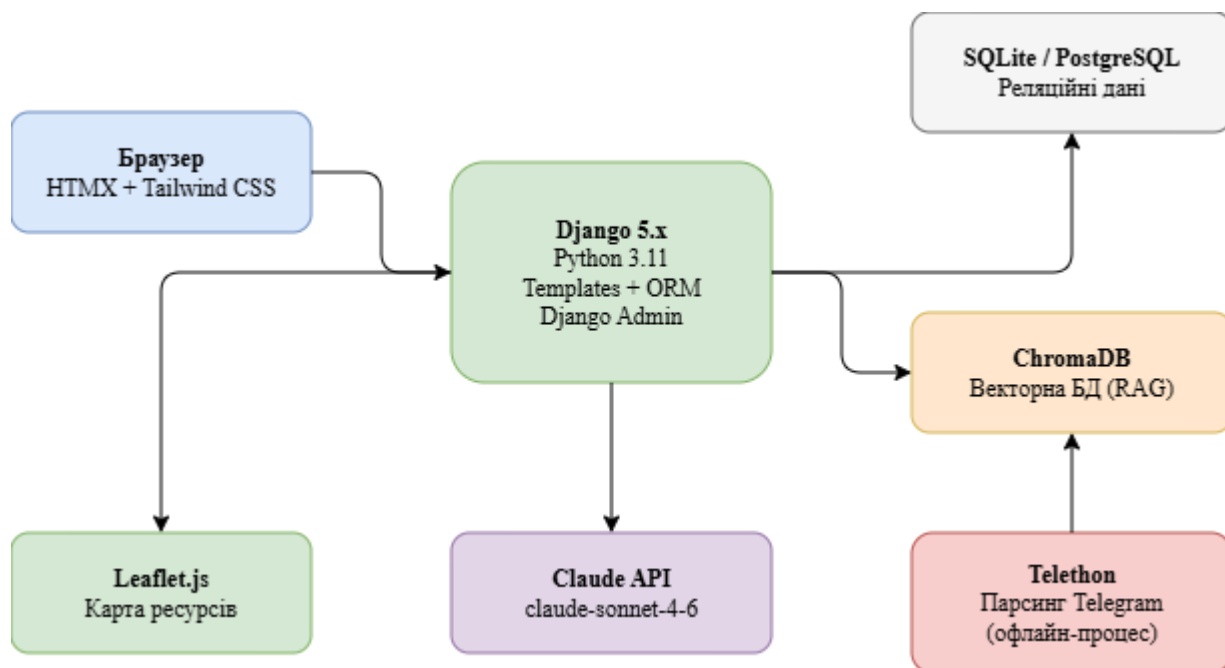


Рис. 2.1. Схема технологічного стеку UkrWay

Схема на рис. 2.1 уточнює ключову архітектурну особливість платформи: Django виступає єдиним центральним вузлом, через який проходять усі запити – від браузера до бази даних, від Claude API до карти ресурсів. Така централізована архітектура спрощує підтримку і деплой: замість кількох незалежних сервісів – один Django-процес з чітко розмежованими модулями. Окремо стоїть бібліотека Telethon, яка працює офлайн і не бере участі в обробці запитів користувача, а лише наповнює ChromaDB повідомленнями з чатів, перед запуском системи. Це і є реалізація graceful fallback: якщо векторна база ще не наповнена, Django продовжує працювати без RAG-контексту.

Зведений технологічний стек. У таблиці 2.5 наведено повний перелік технологій платформи UkrWay з прив'язкою до конкретних параметрів з підрозділу 2.1.

Таблиця 2.5

Технологічний стек платформи UkrWay та відповідність функціональним параметрам

Компонент	Технологія	Версія	Закриває параметр
Backend	Django + Python	5.x + 3.14	Ф5, Ф6, НФ7
Frontend	Tailwind CSS + HTMX	CDN / 2.x	НФ1, НФ2, НФ5
Шаблони	Django Templates	вбудовані	НФ1
Реляційна БД	SQLite → PostgreSQL	3.x / 16.x	НФ7
Векторна БД	ChromaDB	0.6.x	Ф3
Ембединги	sentence-transformers	3.x	Ф3
ІІІ-платформа	Claude API	(claude-sonnet-4-6)	Ф2
Карта	Leaflet.js	1.9.x	Ф4
Статика	Whitenoise	6.x	НФ5
Змінні середовища	python-dotenv	1.x	НФ8

Таблиця 2.5 підтверджує, що кожен компонент стеку прив'язаний до конкретного параметру з підрозділу 2.1, а не обраний довільно. Усі параметри Ф1–Ф7 і НФ1–НФ8 закриті без надлишкових технологій і без прогалин. Методи побудови RAG-пайплайну, з використанням ChromaDB, sentence-transformers і Telethon – детально розглянуто у підрозділі 2.3.

2.3 Методи побудови RAG-системи та вибір інструментів.

У Розділі 1 виявлено конкретну проблему: Telegram-спільноти мігрантів містять цінний практичний досвід, але він неструктурований і недоступний для

системного пошуку [10]. Закрити цю прогалину технічно дозволяє архітектура Retrieval-Augmented Generation (RAG) – підхід, описаний Lewis et al. у 2020 році [1] і детально досліджений у оглядовій роботі Gao et al. [26]. Ідея проста: перед кожним викликом мовної моделі система шукає в базі знань найближчі за змістом фрагменти і підставляє їх у системний промпт. Claude відповідає не з загальних знань, а з реального досвіду людей, які вже пройшли конкретні процедури.

Але перш ніж обирати інструменти – варто відповісти на питання, чому взагалі RAG, а не щось інше? Розглядалось три підхода (табл. 2.6).

Таблиця 2.6

Порівняльний аналіз підходів до збагачення відповідей мовної моделі

Критерій	Чистий LLM	RAG	Fine-tuning
Актуальність даних	Обмежена датою навчання	Динамічна, оновлюється	Статична після навчання
Потреба у власних даних	Не потрібні	База знань	Великий розмічений датасет
Вартість впровадження	Низька	Середня	Висока
Якість для вузької теми	Середня	Висока	Дуже висока
Час оновлення бази знань	—	Мінімальний	Повне перенавчання
Придатність для UkrWay	Недостатньо	Оптимально	Надмірно

Fine-tuning дав би найточніші відповіді – але потребує тисячі розмічених прикладів і повного перенавчання кожного разу як змінюється законодавство. Чистий LLM не знає специфіки польського PESEL чи британського BRP у деталях, що з'явилися після дати його навчання. RAG це баланс: база знань оновлюється незалежно від моделі, нові повідомлення з Telegram додаються без

жодного перенавчання. Для платформи де практика мігрантів змінюється швидше за будь-який регламент це розумний вибір [26].

Щодо векторної бази, вона дуже важлива у зв'язці з AI. Три варіанти розглядались серйозно: ChromaDB, pgvector і Pinecone. Почнемо з того, чому від двох відмовились. Pinecone – хмарний SaaS з відмінною продуктивністю, але при масштабуванні понад 5 млн векторів ціна сягає \$500–1500 на місяць [24]. Для освітнього проєкту, що має залишатися автономним, це неприйнятно. Pgvector додає векторний пошук до PostgreSQL – технічно привабливо, але потребує окремого налаштування індексів і при одночасних запитах поступається спеціалізованим рішенням за латентністю [24]. ChromaDB – нульова конфігурація через PersistentClient. Це перша і головна причина вибору. Далі нативна Python-інтеграція без проміжних шарів. І третє: повна автономність без зовнішніх сервісів і без витрат. За даними практичного порівняння, один VPS з 4,8 ГБ RAM комфортно обробляє мільйони векторів на ChromaDB [24] саме стільки генерують активні Telegram-групи мігрантів.

Вибір моделі ембедингів. Повідомлення в Telegram-групах мігрантів – суміш мов: українська, польська, німецька, англійська в одному чаті. Потрібна мультилінгвальна модель. Розглядались три варіанти з бібліотеки sentence-transformers. Multilingual-e5-large забезпечує вищу точність, але файл важить понад 500 МБ [27]. LaBSE підтримує 109 мов і також точна, але схожого розміру [27]. Обрана paraphrase-multilingual-MiniLM-L12-v2 підтримує майже всі мови, включно з українською, перетворює текст у 384-вимірний вектор і важить 471 МБ при 118 млн параметрів [25]. Альтернативи це multilingual-e5-large і LaBSE – забезпечують вищу точність, але значно більший розмір і вищі вимоги до пам'яті [25]. Для платформи де RAG-пошук відбувається при кожному повідомленні, компактна архітектура MiniLM є оптимальним балансом між якістю і швидкістю. Підтримка української мови зображена на Рис 2.2.

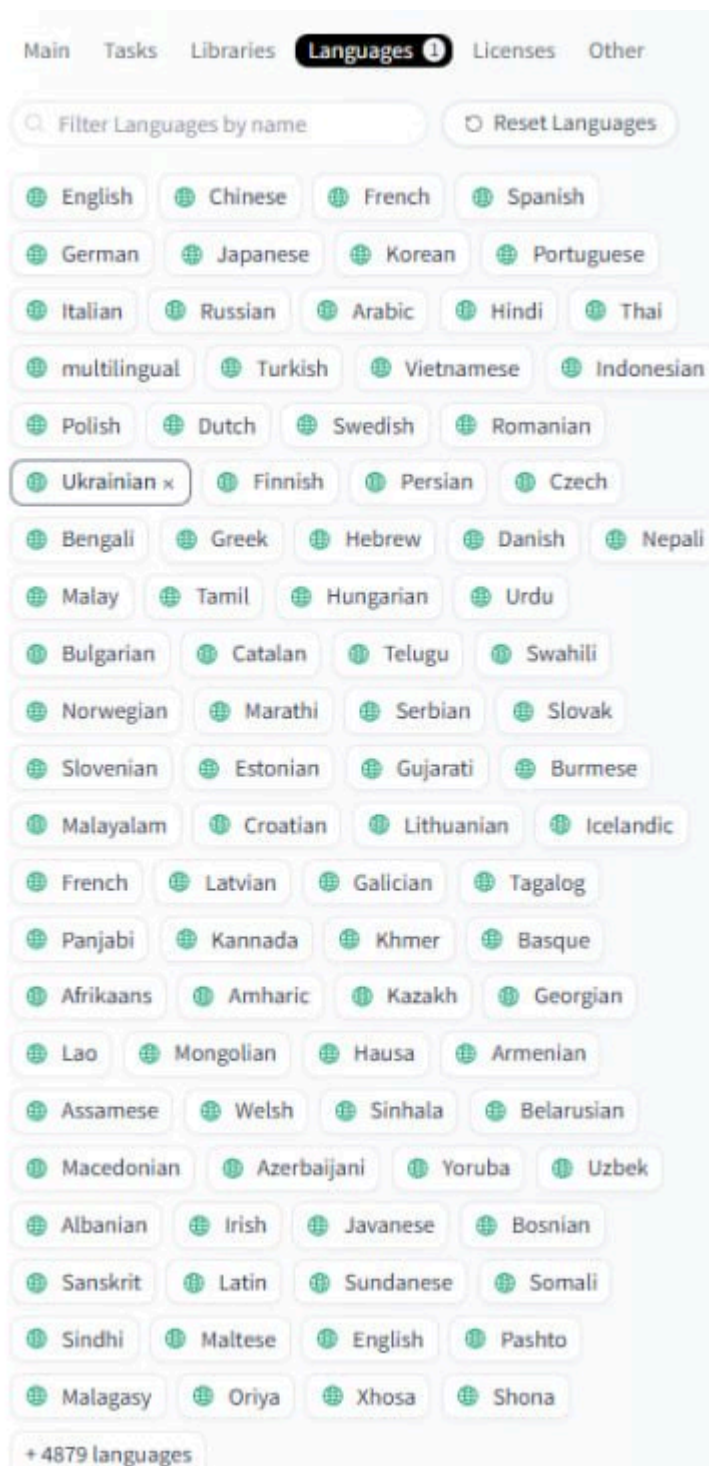


Рис. 2.2. Підтримувані мови моделі paraphrase-multilingual-MiniLM-L12-v2 [25]

Рис. 2.2 підтверджує: українська входить до переліку підтримуваних мов – це задокументована характеристика моделі [25]. Саме ця властивість стала вирішальною для коректної векторизації україномовних Telegram-повідомлень.

Виставлені, та підібрані конкретні пайплайни, які допоможуть формувати ці повідомлення з чатів телеграм. Конкретні числа визначались у процесі розробки – кожен відповідає практичній потребі і прописані в коді (табл. 2.7).

Таблиця 2.7

Параметри RAG-пайплайну платформи UkrWay

Параметр	Значення	Обґрунтування
Мінімальна довжина чанку	80 символів	Мінімальна довжина зібраного чанку після об'єднання повідомлень
Максимальна довжина чанку	500 символів	Обмежує обсяг контексту в промпті Claude
Часове вікно групування	5 хвилин	Об'єднує повідомлення однієї розмови у зв'язний фрагмент
Мінімум алфавітних символів	10 символів	Фільтрує стікери, посилання і числові повідомлення
top_k (країна визначена)	7	Більше контексту при відомій географії запиту
top_k (країна не визначена)	4	Менше, але точніше без географічного фільтру
Поріг cosine distance	< 0.7	Відкидає нерелевантні результати перед підстановкою
Метрика подібності	cosine (hnsw:space)	Стандарт для семантичного пошуку по текстових ембедингах

Цифри в таблиці 2.7 – не довільні. Поріг 0.7 означає: якщо жоден фрагмент не набирає достатньої семантичної близькості, система повертає порожній результат і Claude відповідає без RAG-контексту. Краще дати загальну відповідь, ніж підставити нерелевантний контекст і збити модель. Часове вікно 5 хвилин пояснюється природою Telegram: повідомлення часто розбиті на кілька коротких

частин і мають сенс лише разом для формування правильного контексту. Приклад базово чанку наведено на рис 2.3

```
Текст: "Підкажіть хтось – PESEL вже отримала,
        тепер треба зареєструватися у NFZ?
        Так, обов'язково, йдете в будь-який
        відділ NFZ з PESEL і паспортом"

Метадані: { country: "poland",
            date: "2024-03-15T14:32:00" }

ID: chunk_42
```

Рис. 2.3. Приклад чанку, який формується з повідомлень

Саме такі фрагменти, як показані на Рис 2.3 це реальні запитання і відповіді з Telegram-груп мігрантів – підставляються в системний промпт перед викликом Claude API. Дата публікації зберігається в метаданих і відображається у відповіді, щоб користувач розумів актуальність інформації.

Визначення країни запиту. Перед семантичним пошуком система виконує `detect_country()` функція визначає до якої країни відноситься запит. Алгоритм простий, запит переводиться в нижній регістр, для кожної країни підраховується кількість ключових слів-збігів, обирається країна з максимальним рахунком. Якщо збігів немає – пошук іде без географічного фільтру. Ключові слова підібрані під реальну лексику мігрантів: для Польщі – «pesel», «karta pobytu», «zlotih»; для Німеччини «anmeldung», «bürgergeld», «jobcenter»; для Великої Британії «brp», «nhs», «homes for ukraine». Завдяки цьому при відомій країні `top_k` зростає з 3 до 5 більше контексту там де він найточніший.

Загальну схему пайплайну наведено на рис. 2.4. Він складається з двох незалежних процесів: офлайн – збір і індексування даних командами `fetch_telegram` і `build_knowledge_base`; онлайн – обробка запиту користувача в реальному часі.

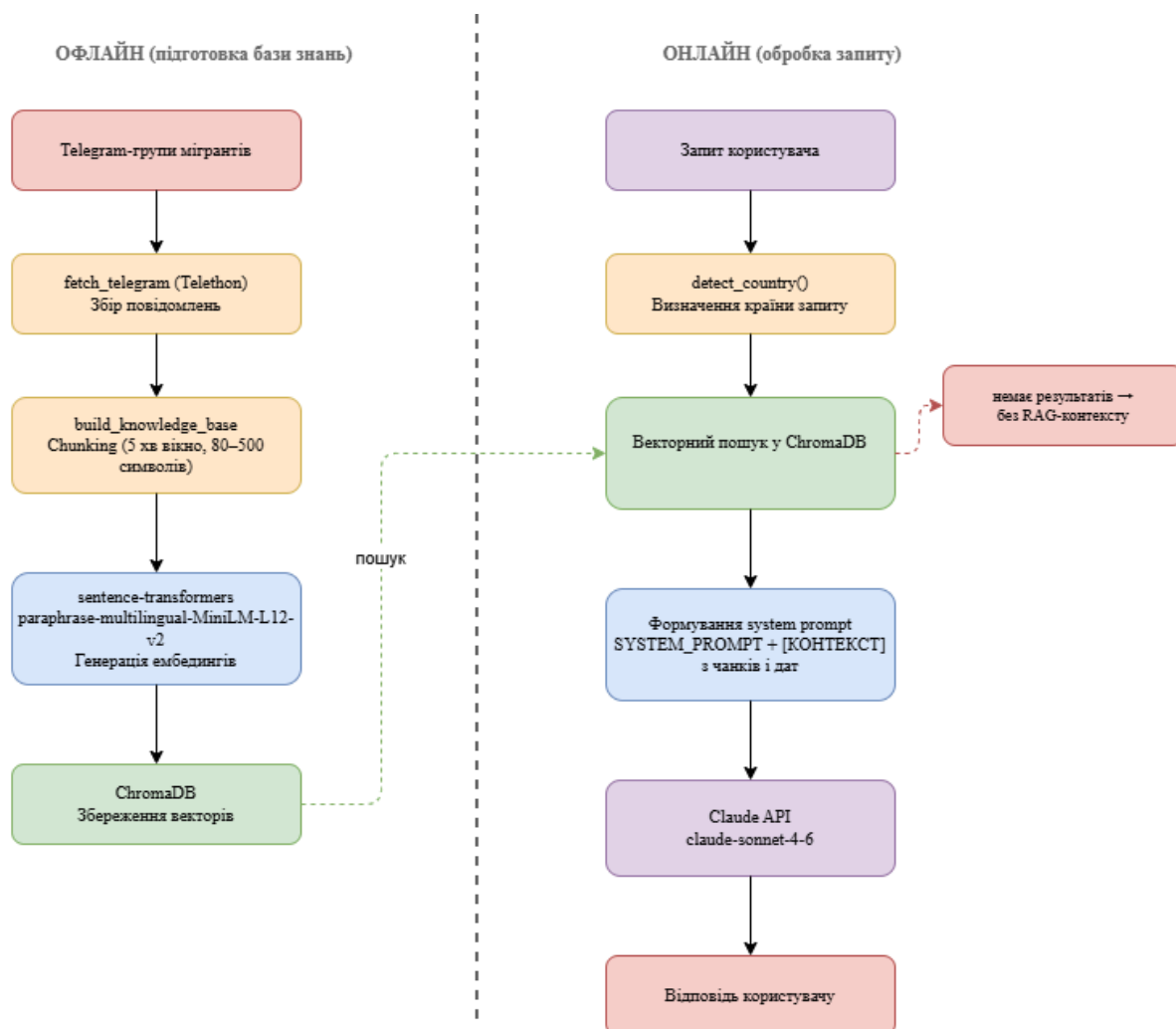


Рис. 2.4. Схема RAG-пайплайну платформи UkrWay

Схема на рис. 2.4 показує ключову перевагу такої архітектури: офлайн-частина може виконуватись незалежно від роботи сайту. Новий батч Telegram-повідомлень індексується і ChromaDB автоматично включає їх у наступні пошуки – без перезапуску системи і без перенавчання моделі.

Таким чином, вибір інструментів RAG-системи: ChromaDB, paraphrase-multilingual-MiniLM-L12-v2 і конкретні параметри пайплайну – визначався трьома критеріями: автономність від зовнішніх платформ, підтримка україномовного контенту і мінімальна складність розгортання. Архітектуру системи і базу даних розглянуто у підрозділі 2.4.

2.4. Проектування архітектури системи та бази даних

Платформа UkrWay побудована за модульним принципом – кожен функціональний блок реалізований як окремий Django-додаток з чітко визначеною зоною відповідальності. Такий підхід спрощує підтримку і масштабування: зміни в одному модулі не зачіпають інші.

Архітектура Django-додатків. Проект складається з п'яти власних Django-додатків (табл. 2.8). Чотири з них мають повноцінний веб-інтерфейс з URL-маршрутами і шаблонами; knowledge – виключно сервісний модуль без HTTP-обробників.

Таблиця 2.8

Модулі платформи UkrWay та їх призначення

Модуль	Призначення	URL-маршрути	Шаблони
core	Головна сторінка, сторінка «Про проект», політика конфіденційності	/, /about/, /privacy/	base.html, home.html, about.html, privacy.html
countries	Довідник по 3 країнах: статті у 4 категоріях (документи, житло, робота, медицина)	/countries/, /countries/<slug>/	list.html – список країн; detail.html – вкладки з акордеоном статей
chat	AI-чат з RAG-контекстом; зберігає сесії і повідомлення в БД	/chat/ – інтерфейс; /chat/send/ – POST-запит; /chat/clear/ – очищення	chat.html – основний інтерфейс; partials/messages.html – HTMX-partial для повідомлень
resources	Інтерактивна карта організацій допомоги з фільтрацією по країні та категорії	/resources/?country=&category= – GET-фільтри	map.html – картки організацій і Leaflet.js карта

Продовження таблиці 2.8

Модуль	Призначення	URL-маршрути	Шаблони
knowledge	RAG-сервіс (ChromaDB + sentence-transformers); management commands для збору і індексування Telegram-даних	– (без HTTP-маршрутів)	– (без шаблонів)

З таблиці 2.8 видно: knowledge не обробляє HTTP-запити напряму, але є ключовою залежністю для chat звідти chat/views.py імпортує rag_service для збагачення промпту. Залежності між модулями однонаправлені: core і resources імпортують Country з countries; chat/views.py імпортує rag_service для збагачення промпту. Зворотних залежностей немає, тобто countries і knowledge ні від кого не залежать. Це класичний принцип низького зв'язування, який дозволяє тестувати і розгортати кожен модуль незалежно.

Структуру файлів проєкту з ПЗ PyCharm наведено на рис. 2.5. Кожен модуль дотримується стандартної Django-конвенції: models.py, views.py, urls.py, admin.py плюс папки templates і migrations. Система пройшла 4 міграції – остання (0002 у countries) додала поле flag_code до моделі Country для відображення прапорів.

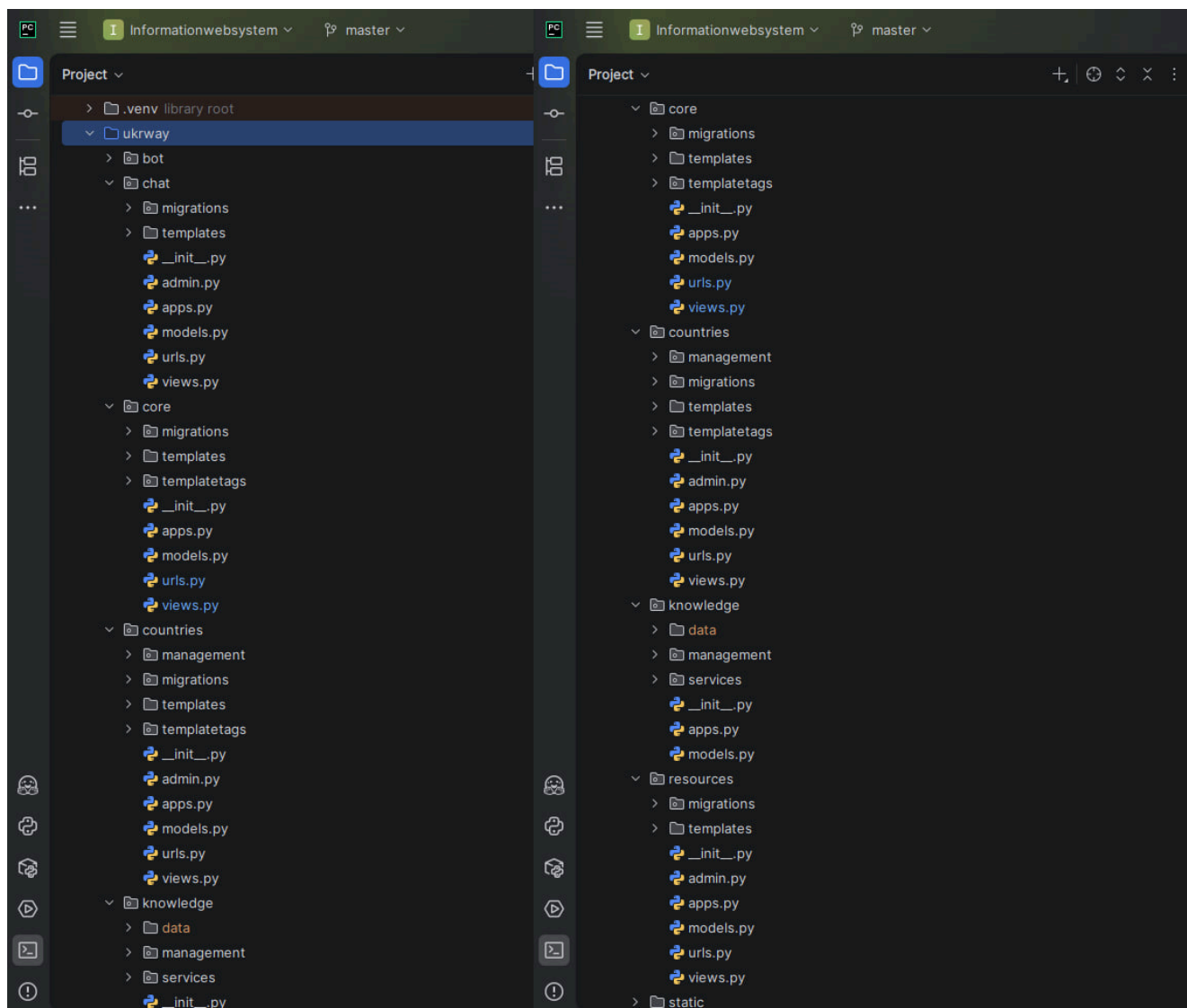


Рис. 2.5. Структура файлів проєкту UkrWay з PyCharm

Рис. 2.5 підтверджує модульність архітектури. Папка knowledge/data/ містить ChromaDB-базу і JSON-файли Telegram-повідомлень вони додані до .gitignore і не потрапляють до репозиторію через значний обсяг. Маршрутизацію запитів між модулями централізовано в головному urls.py (лістинг 2.1).



```
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('', include('core.urls')),  
    path('countries/', include('countries.urls')),  
    path('chat/', include('chat.urls')),  
    path('resources/', include('resources.urls')),  
    # наведено скорочено  
]
```

Лістинг 2.1. Головний файл маршрутизації ukrway/urls.py (скорочено)

Чотири виклики `include()` і маршрут адмін-панелі означають що кожен модуль керує власними маршрутами самостійно. Додавання нового функціонального блоку потребує одного рядка тут і окремого `urls.py` у новому модулі без змін в існуючих.

Проектування бази даних. Реляційна база містить шість моделей, згрупованих у три логічні блоки: контентний ($\text{Country} \rightarrow \text{InfoCategory} \rightarrow \text{InfoArticle}$), ресурсний ($\text{Country} \leftarrow \text{Resource}$) і сесійний ($\text{ChatSession} \leftarrow \text{ChatMessage}$). ER-діаграму наведено на рис. 2.6.

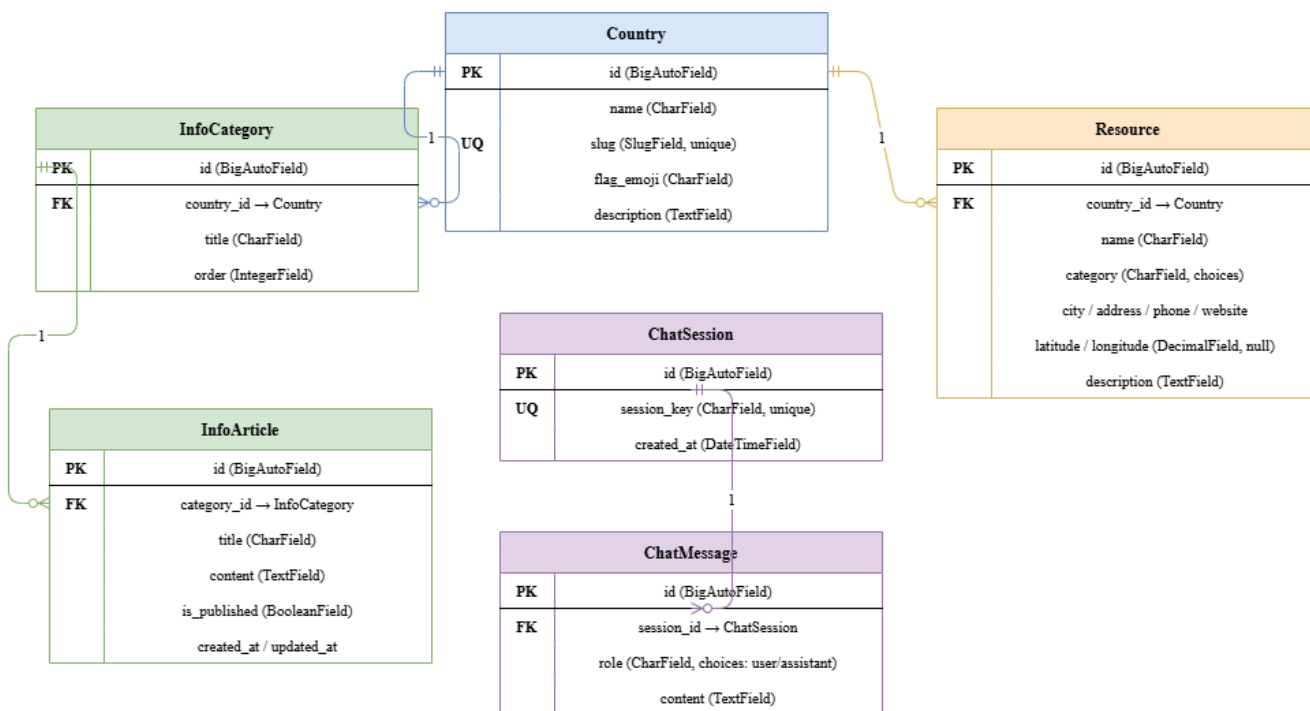


Рис. 2.6. ER-діаграма бази даних платформи UkrWay

Рис. 2.6 показує принципове рішення Country є центральною сутністю, до якої прив'язані і контентні статті (через InfoCategory), і ресурси організацій. ChatSession і ChatMessage утворюють ізольований кластер чат не залежить від жодної іншої моделі. Станом на 14.05.2026 база містить 3 країни, 12 категорій, 20 статей і 17 ресурсів.

Структура моделей відображає кілька свідомих проєктних рішень. Resource.category реалізовано як CharField з фіксованим набором 7 значень, а саме legal, housing, work, mental, education, medical, general. ChatMessage.role визначено через choices з двома значеннями («user» і «assistant»), що дозволяє передавати повну історію діалогу до Claude API без додаткової трансформації. Поля Resource.latitude і longitude мають null=True ресурси без координат відображаються у списку організацій, але не на карті Leaflet.js. Поле InfoArticle.is_published дозволяє адміністратору вмикати і вимикати публікацію статей через Django Admin без видалення запису.

Таким чином, архітектура UkrWay побудована на п'яти незалежних модулях з однонаправленими залежностями і реляційній БД з шістьма моделями. Модульність забезпечує незалежне розгортання кожного компонента, а структура

БД відображає предметну область без надлишкових сутностей. Принципи проєктування інтерфейсу користувача розглянуто у підрозділі 2.5.

2.5. Проєктування інтерфейсу користувача

Інтерфейс UkrWay проєктувався під конкретну аудиторію – людей у стресовій ситуації переїзду, які шукають термінову практичну інформацію. Це визначило чотири базові принципи, які пронизують кожне рішення: від кольору кнопки до структури сторінки.

Принципи проєктування, перший – мінімум кроків до інформації. Жодної реєстрації, жодних підписок ChatSession ідентифікує користувача через Django session_key автоматично – людина відкриває сайт і одразу може поставити питання. Другий – україномовність як базова умова, а не опція: LANGUAGE_CODE='uk' у settings.py, system prompt Claude налаштований на відповіді виключно українською. Третій мобільна адаптація: Tailwind CSS забезпечує коректне відображення від 320 px, а навігаційне меню згортається в hamburger на екранах до 768 px через vanilla JS у base.html. Четвертий це візуальна асоціація з Україною через кольорову схему, що формує довіру у цільовій аудиторії, тому були підібрані такі кольора

Кольорова схема і типографіка. Усі кольорові рішення підтверджені в base.html (табл. 2.9).

Таблиця 2.9

Організація UI/UX для платформи

Елемент інтерфейсу	HEX-код	Обґрунтування вибору
Навігація, hero-секція, заголовки	#005BBB	Синій колір українського прапора – формує миттєву асоціацію і довіру у цільовій аудиторії
СТА-кнопки, акцентні елементи	#FFD500	Жовтий колір українського прапора – забезпечує максимальну видимість кнопок на синьому фоні

Продовження таблиці 2.9

Елемент інтерфейсу	HEX-код	Обґрунтування вибору
Фон сторінки	#FFFFFF	Білий – чистота і читабельність контенту
Основний текст	#1F2937	Темно-сірий – достатній контраст без різкості чорного
Фон секцій і карток	#F9FAFB	Світло-сірий – візуальне розмежування блоків без рамок
Footer	#111827	Темний – чітке завершення сторінки, контраст з білим фоном

Таблиця 2.9 показує: колірна пара #005BBB і #FFD500 це не просто естетичне рішення. Синьо-жовта схема миттєво сигналізує цільовій аудиторії що платформа створена для українців. Для людини в стресі це важливий психологічний сигнал довіри. Щодо Типографіки, обраний шрифт Inter, який підключений через Google Fonts; він добре читається на будь-якому пристрої і підтримує кирилицю.

Навігація платформи побудована за принципом «одного рівня» – жодних вкладених меню: Головна → Країни → Ресурси → AI-чат → Про нас. Кнопка «Запитати AI» винесена окремо жовтим кольором у правий кут navbar – вона завжди помітна незалежно від поточної сторінки. На сторінці країни реалізовано два JS-компоненти без фреймворків: tabs (перемикання між категоріями Документи / Житло / Робота / Медицина) і accordion (розкриті статті з датою оновлення). AI-чат використовує HTMX: при відправці повідомлення не відбувається перезавантаження сторінки – HTMX відправляє POST-запит на /chat/send/ і замінює вміст блоку #chat-messages через partial messages.html. Це забезпечує миттєву реакцію інтерфейсу без WebSocket і без складного SPA.

Головну сторінку платформи наведено на рис. 2.7.

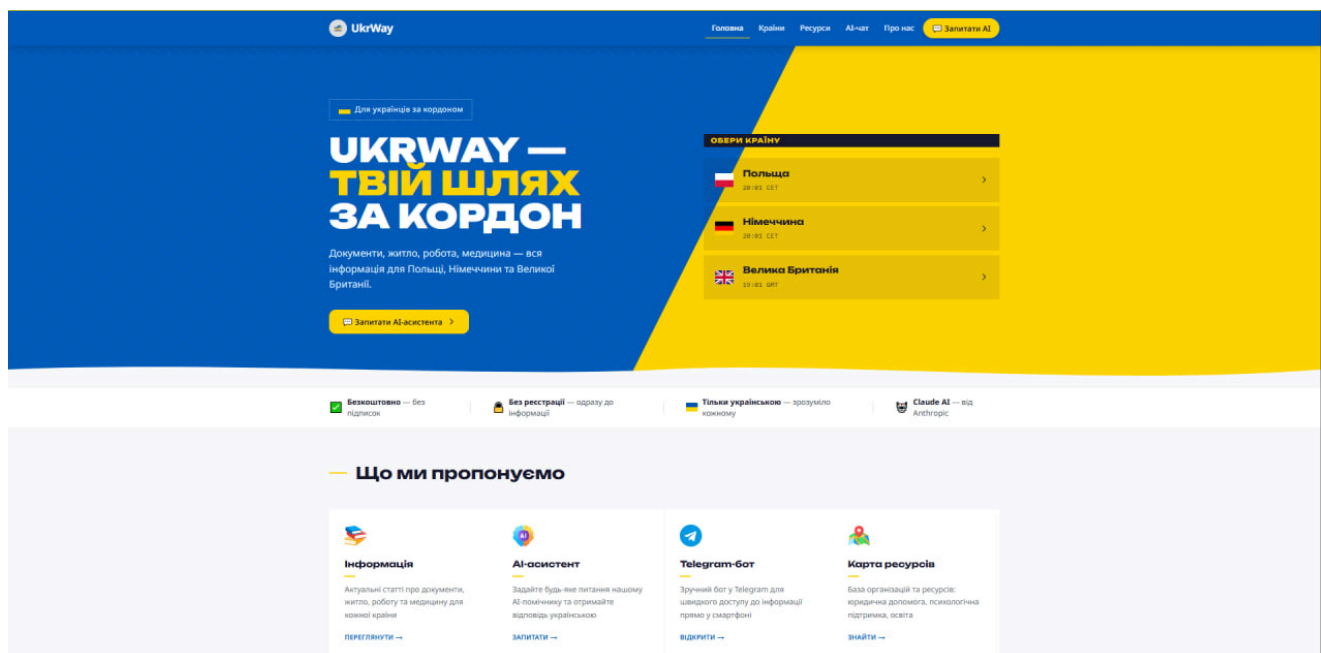


Рис. 2.7. Головна сторінка платформи UkrWay

Рис. 2.7 демонструє ключове проєктне рішення hero-секції: двоколонний layout де ліворуч заголовок і Call To Action, праворуч – панель вибору країни з поточним часом у кожному часовому поясі.. Живий годинник – не декоративний елемент: він допомагає мігранту зорієнтуватися в різниці часових зон. Нижче hero-секції чотири переваги платформи одним рядком: «Безкоштовно», «Без реєстрації», «Тільки українською», «Claude AI». Ці чотири тези відповідають нефункціональним параметрам НФ1-НФ4 з підрозділу 2.1.

Сторінку конкретної країни, а саме Польщі наведено на рис. 2.8.

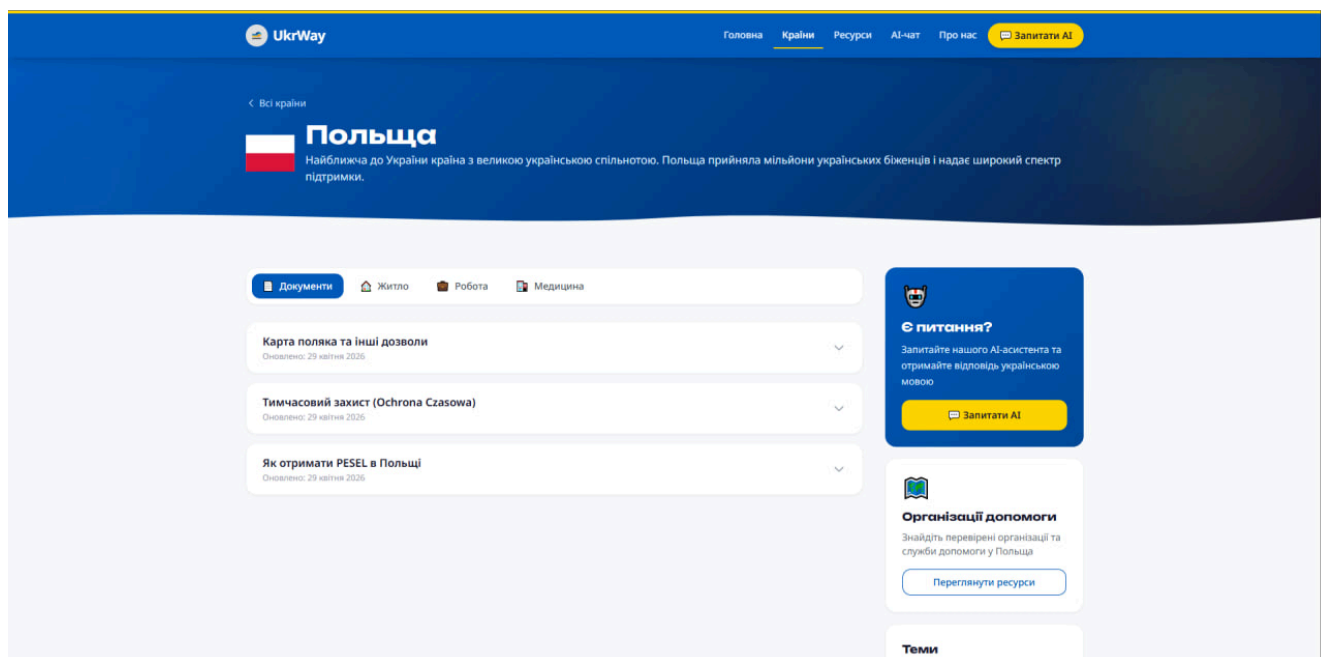


Рис. 2.8. Сторінка країни (Польща) платформи UkrWay

Рис. 2.8 показує інформаційний модуль у дії: активна вкладка «Документи» виділена синім, решта сірі. Акордеон відображає статті з назвою і датою останнього оновлення, це зручно, тому що мігрант одразу бачить наскільки актуальна інформація. Sidebar праворуч містить два блоки: CTA (Call-To-Action) до AI-чату («Є питання? Запитайте нашого AI-асистента») і перелік організацій допомоги в цій країні. Такий layout дозволяє одночасно читати статтю і одразу перейти до AI якщо стаття не дала відповіді.

Інтерфейс AI-чату наведено на рис. 2.9.

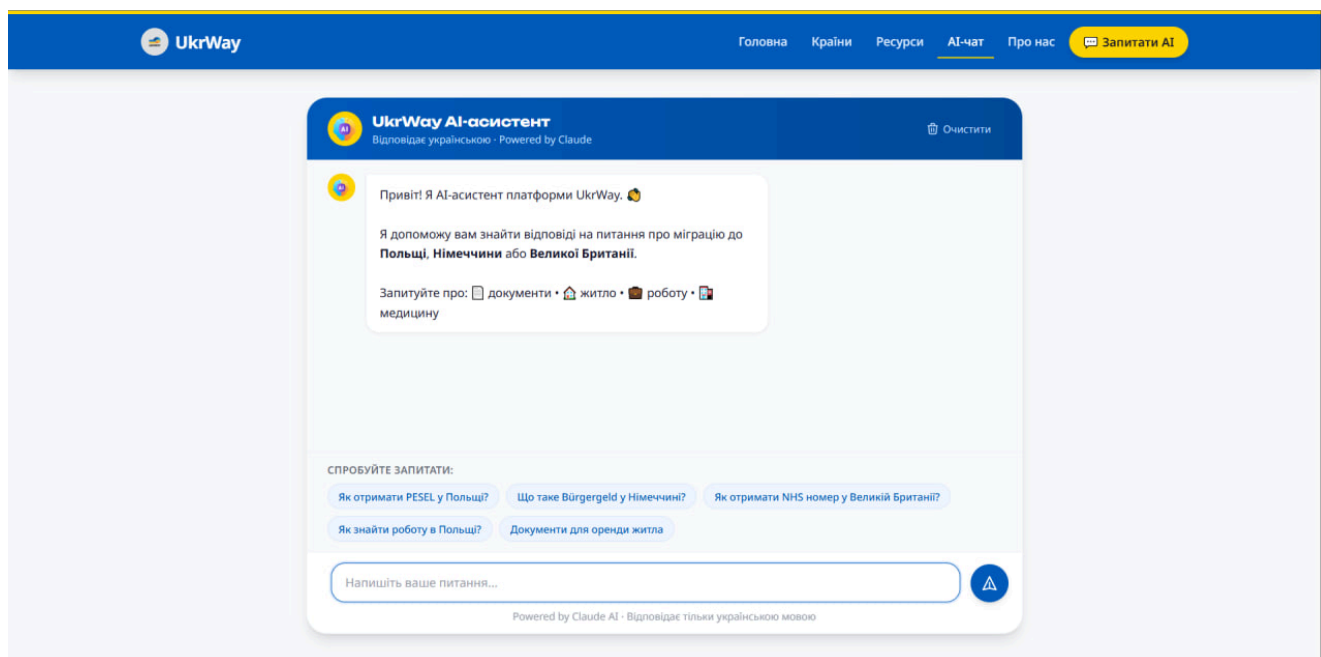


Рис. 2.9. Інтерфейс AI-асистента платформи UkrWay

Рис. 2.9 відображає мінімалістичний інтерфейс чату, централізований контейнер, заголовок «UkrWay AI-асистент · Відповідає українською · Powered by Claude», п'ять швидких запитань у вигляді кнопок-підказок («Як отримати PESEL у Польщі?», «Що таке Bürgergeld у Німеччині?» тощо). Підказки знижують поріг входу – користувач може просто натиснути на готове питання замість того щоб формулювати самостійно. Поле введення з кнопкою відправки і кнопка «Очистити» для скидання сесії – весь інтерфейс умісний на одному екрані без прокрутки.

Карту ресурсів наведено на рис. 2.10.

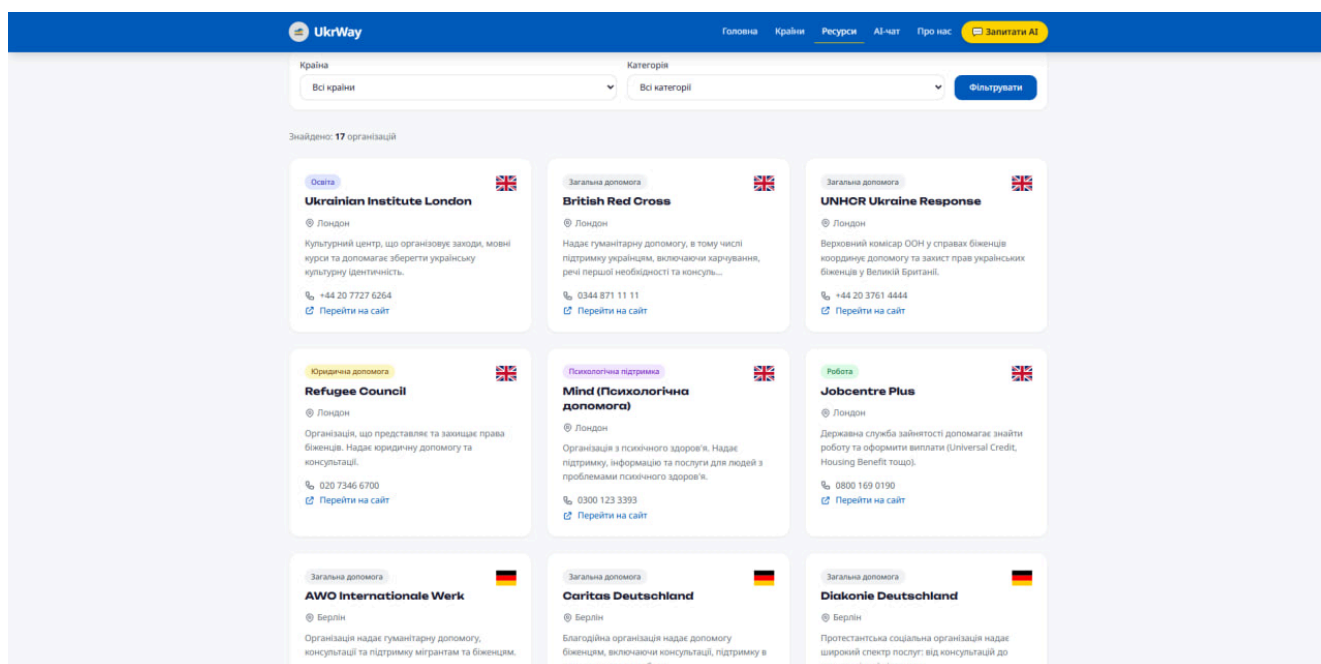


Рис. 2.10. Карта ресурсів платформи UkrWay

Рис. 2.10 демонструє два режими перегляду ресурсів: картки організацій зверху і інтерактивна карта Leaflet.js знизу. Фільтри «Країна» і «Категорія» з лічильником «Знайдено: 17 організацій» дозволяють швидко звужити пошук по потрібній країні, або сфері допомоги. Кожна картка містить назву, місто, категорію у вигляді кольорового бейджа, короткий опис, телефон і посилання на сайт, всю інформацію необхідну для першого контакту. На карті Leaflet.js маркери розташовані у трьох країнах.

Таким чином, інтерфейс UkrWay реалізує чотири основні принципи проєктування: мінімум кроків до інформації, україномовність, мобільна адаптивність і візуальна асоціація з Україною. Кожне рішення, від двоколонного hero-layout до HTMX-чату без перезавантаження, обумовлене потребами цільової аудиторії.

Висновки до розділу 2

За результатами аналізу технологій та проєктування інформаційної веб-системи UkrWay встановлено таке.

Перш за все – вимоги. З виявлених у Розділі 1 інформаційних потреб мігрантів сформульовано сім функціональних і вісім нефункціональних параметрів системи. Кожен має конкретний показник: час відповіді AI-асистента – до 10 секунд, збереження сесії – 30 днів, мінімальний екран – 320 px. Окремо закладено три рівні відмовостійкості на випадок недоступності RAG, API-ключа або зовнішнього сервісу – без цього платформа була б нестабільною при будь-якому збої.

Щодо технологій: Django 5.x обрано не як найпродуктивніший варіант, а як найоптимальніший для цього проєкту – через вбудовану адмін-панель, ORM і сесійний механізм; HTMX замість React через відсутність потреби у складному фронтенд-білді і розмір 14 КБ проти 47 КБ; SQLite/PostgreSQL замість MongoDB через структурований характер даних. Для AI-асистента обрано RAG-архітектуру, чистий LLM не знає деталей процедур що змінились після дати навчання, fine-tuning потребує великого датасету і перенавчання при кожній зміні законодавства. ChromaDB і модель paraphrase-multilingual-MiniLM-L12-v2 (471 МБ, понад 50 мов) забезпечують автономну роботу з україномовним контентом без залежності від зовнішніх сервісів. Поріг cosine distance менше 0.7 відкидає нерелевантні результати і не підставляє хибний контекст у промпт.

Стосовно архітектури: п'ять модулів з однонаправленими залежностями, шість моделей у трьох логічних групах – контентний, ресурсний і сесійний. Чат повністю ізольований від контентних моделей – це свідоме рішення, а не випадковість. Станом на 14.05.2026 база містить 3 країни, 12 категорій, 20 статей і 17 ресурсів організацій допомоги.

Нарешті одне з найважливіших, що бачить юзер, це інтерфейс. Синьо-жовта схема (#005BBB і #FFD500) не просто естетика це сигнал довіри для цільової аудиторії. Відсутність реєстрації скорочує шлях до інформації до нуля кроків. HTMX оновлює чат без перезавантаження сторінки, адаптивність від 320 px забезпечує коректну роботу на мобільних. Усі чотири унікальні екрани – головна, сторінка країни, AI-чат і карта ресурсів – реалізують параметри визначені у підрозділі 2.1. Практична реалізація цих рішень розглядається у Розділі 3.

3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ВЕБ-СИСТЕМИ UKRWAY

3.1. Реалізація серверної частини та інформаційного модулю

На основі архітектурних рішень, спроектованих у підрозділі 2.4, реалізовано серверну частину платформи UkrWay засобами Django 5.x і Python 3.14. Реалізація охоплює три складові: моделі бази даних, адміністративну панель управління контентом і логіку обробки запитів інформаційного модулю.

Реалізація моделей. Спроектowana у підрозділі 2.4 ієрархія Country → InfoCategory → InfoArticle реалізована через Django ORM. Модель Country наведено у лістингу 3.1.

```

4  class Country(models.Model): 14+ usages  ⚡ Alex
5      name = models.CharField(max_length=100, verbose_name='Назва')
6      slug = models.SlugField(unique=True)
7      flag_emoji = models.CharField(max_length=10, verbose_name='Прапор')
8      flag_code = models.CharField(max_length=2, default='', verbose_name='ISO код прапора')
9      description = models.TextField(verbose_name='Опис')
10
11     class Meta:  ⚡ Alex
12         verbose_name = 'Країна'
13         verbose_name_plural = 'Країни'
14         ordering = ['name']
15
16     def __str__(self):  ⚡ Alex
17         return f'{self.flag_emoji} {self.name}'

```

Лістинг 3.1. Модель Country (countries/[models.py](#))

З лістингу 3.1 видно кілька проєктних рішень. Поле slug з параметром unique=True містить текстовий ідентифікатор країни у URL наприклад, /countries/poland/. Параметр унікальності гарантує, що дві країни не можуть мати однакову адресу. Поле flag_code зберігає двосимвольний код країни окремо від емодзі-прапора – це дозволяє використовувати його у CSS-бібліотеках відображення прапорів. Метод __str__ визначає текстове представлення об'єкта: рядок виду «Польща» відображається в адміністративній панелі Django при виборі країни зі списку.

Модель InfoArticle, що замикає контентну ієрархію, наведено у лістингу 3.2.

```

38 class InfoArticle(models.Model): 4+ usages  ⌵ Alex
39     category = models.ForeignKey(
40         InfoCategory, on_delete=models.CASCADE,
41         related_name='articles', verbose_name='Категорія'
42     )
43     title = models.CharField(max_length=300, verbose_name='Заголовок')
44     content = models.TextField(verbose_name='Зміст')
45     is_published = models.BooleanField(default=True, verbose_name='Опубліковано')
46     created_at = models.DateTimeField(auto_now_add=True)
47     updated_at = models.DateTimeField(auto_now=True)
48
49     class Meta:  ⌵ Alex
50         verbose_name = 'Стаття'
51         verbose_name_plural = 'Статті'
52         ordering = ['title']
53
54     def __str__(self):  ⌵ Alex
55         return self.title

```

Лістинг 3.2. Модель InfoArticle (countries/models.py)

Ключове поле тут це “is_published” типу BooleanField з default=True. Воно реалізує вимогу Фб: адміністратор може вимкнути публікацію статті не видаляючи її з бази. Поля created_at і updated_at з auto_now_add і auto_now відповідно, заповнюються автоматично Django і не потребують ручного введення. Дата оновлення відображається на сторінці країни поруч з назвою статті, тобто мігрант бачить наскільки актуальна інформація в статті. Автоматично підтягується, при оновленні в Django Admin.

Реалізація адміністративної панелі. Функціональний параметр Фб – «управління контентом без програмування» реалізовано через Django Admin. Реєстрацію моделей наведено у лістингу 3.3


```

1  from django.contrib import admin
2  from .models import Country, InfoCategory, InfoArticle
3
4
5  @admin.register(Country)  ⚙ Alex
6  class CountryAdmin(admin.ModelAdmin):
7      list_display = ['name', 'slug', 'flag_emoji']
8      ⚡ prepopulated_fields = {'slug': ('name',)}
9
10
11  @admin.register(InfoCategory)  ⚙ Alex
12  class InfoCategoryAdmin(admin.ModelAdmin):
13      list_display = ['title', 'country', 'icon', 'order']
14      list_filter = ['country']
15      ordering = ['country', 'order']
16
17
18  @admin.register(InfoArticle)  ⚙ Alex
19  class InfoArticleAdmin(admin.ModelAdmin):
20      list_display = ['title', 'category', 'is_published', 'updated_at']
21      list_filter = ['category__country', 'is_published']
22      search_fields = ['title', 'content']
23      list_editable = ['is_published']

```

Лістинг 3.3. Реєстрація моделей у Django Admin (countries/admin.py)

Кілька налаштувань заслуговують окремої уваги. У `InfoArticleAdmin` параметр `list_editable = ['is_published']` дозволяє адміністратору перемикаати публікацію статей прямо зі списку одним кліком, без відкриття кожного запису. Параметри `list_filter` і `search_fields` забезпечують швидкий пошук серед наявних статей за назвою, вмістом та країною. У `CountryAdmin` параметр `prepopulated_fields = {'slug': ('name',)}` автоматично генерує `slug` при введенні назви країни. Скріншот адмінки наведено на рис. 3.1.

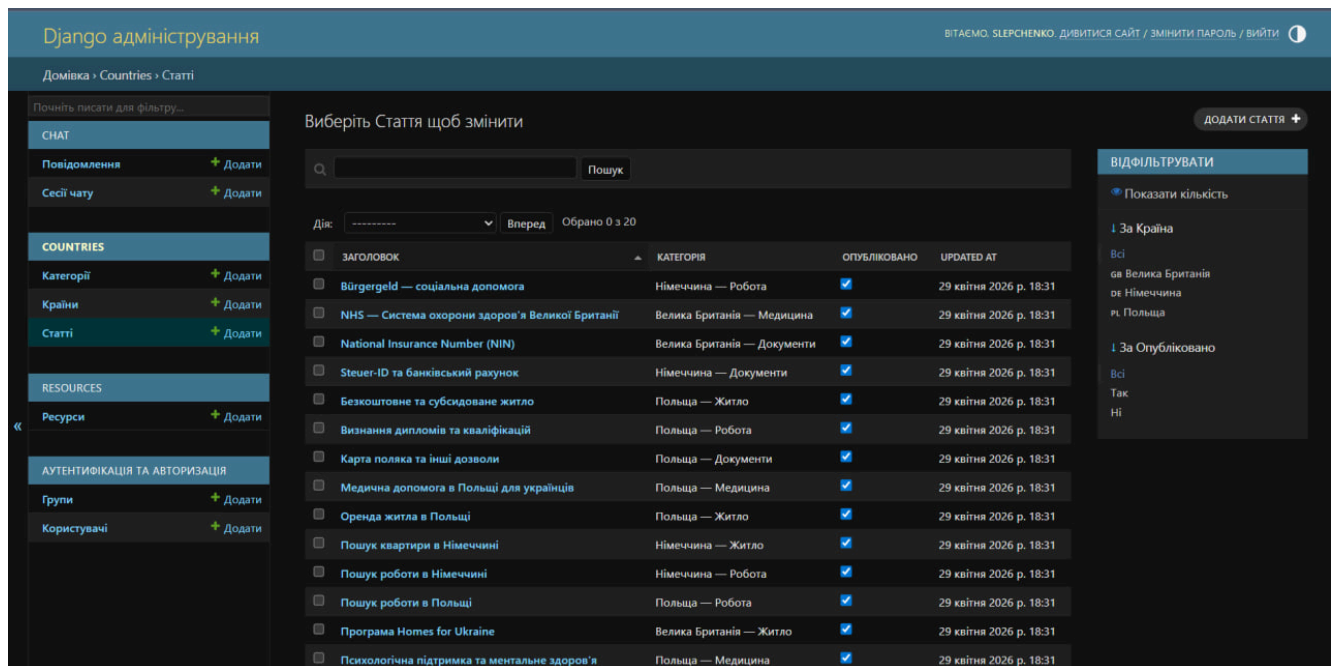


Рис. 3.1. Адміністративна панель Django Admin – список статей InfoArticle

Рисунок 3.1 підтверджує роботу адміністративної панелі. У таблиці видно колонку «Опубліковано» з чекбоксами по яким адміністратор може вмикати або вимикати публікацію статей одним кліком прямо зі списку. Праворуч розташована панель фільтрації за країною та статусом публікації. Вгорі працює пошук по заголовку та змісту статті. Весь цей функціонал створено декларативно – лише через налаштування класу InfoArticleAdmin, без написання власного JavaScript-коду. Ця розробка дуже важлива для адміністраторів сайту.

Реалізація інформаційного модулю. Центральна функція інформаційного модулю country_detail обробляє запит на сторінку конкретної країни. Її реалізацію наведено у лістингу 3.4.

```

10 def country_detail(request, slug): 1 usage  Ⓐ Alex
11     country = get_object_or_404(Country, slug=slug)
12     categories = country.categories.prefetch_related('articles').order_by('order')
13     published_categories = [
14         cat for cat in categories
15         if cat.articles.filter(is_published=True).exists()
16     ]
17     return render(request, template_name='countries/detail.html', context: {
18         'country': country,
19         'categories': published_categories,
20     })

```

Лістинг 3.4. Функція обробки запиту сторінки країни (countries/views.py)

Функція коротка – 8 рядків, але містить кілька важливих рішень. Виклик `get_object_or_404` повертає сторінку помилки 404 якщо країна з таким slug не знайдена. Виклик `prefetch_related('articles')` завантажує всі статті для всіх категорій одним SQL-запитом – без цього база даних отримувала б окремий запит для кожної категорії. Список `published_categories` містить лише ті категорії, у яких є хоча б одна опублікована стаття, порожні вкладки на сторінці країни не відображаються. Алгоритм обробки запиту наведено на рис. 3.2.

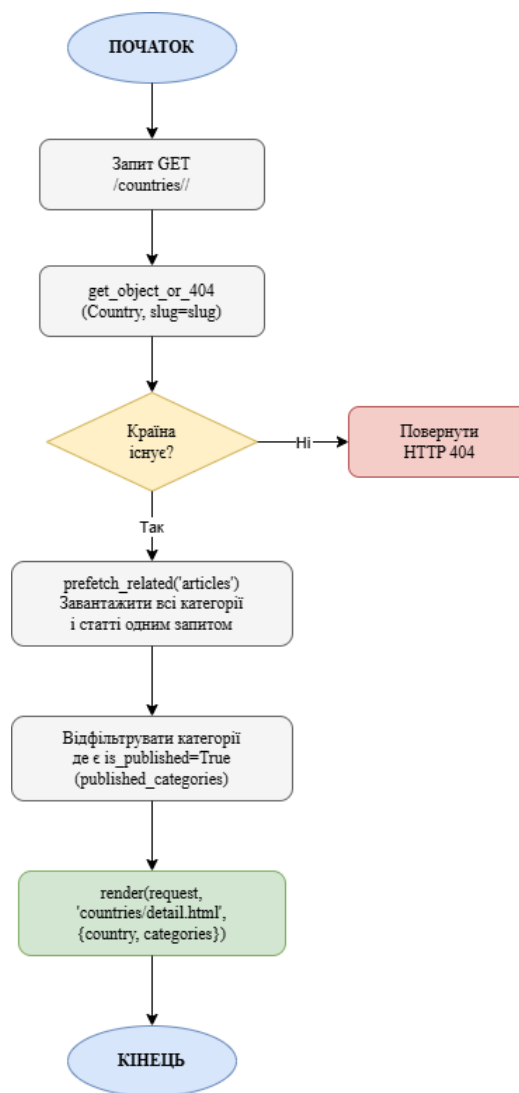


Рис. 3.2. Блок-схема алгоритму обробки запиту інформаційного модулю

Блок-схема на рис. 3.2 відображає повний цикл обробки запиту від моменту коли користувач натискає на картку країни до отримання сторінки з інформацією. Після кліку браузер надсилає GET-запит на адресу `/countries/<slug>/` – наприклад `/countries/poland/` для Польщі. Django передає запит функції `country_detail`, яка насамперед перевіряє чи існує країна з таким slug у базі даних через `get_object_or_404`. Якщо країни немає юзер отримує сторінку з помилкою HTTP 404. Якщо існує, то виконується `prefetch_related`, що завантажує всі категорії і статті для цієї країни одним SQL-запитом замість окремого запиту для кожної категорії. Далі з отриманого списку відфільтровуються лише категорії де є хоча б одна опублікована стаття, порожні вкладки на сторінці не з'являються. Зрештою

Django рендерить шаблон `detail.html` і повертає користувачу готову сторінку з вкладками (Документи, Житло, Робота, Медицина) і акордеоном статей.

Таким чином, серверна частина UkrWay реалізована за модульним принципом моделі, адміністративна панель і представлення мають чітко визначену зону відповідальності. Структура моделей відповідає спроектованій ER-діаграмі (підрозд. 2.4), Django Admin забезпечує управління контентом без програмування (вимога Ф6), а функція `country_detail` завантажує всі пов'язані дані одним запитом до бази замість багаторазових звернень це скорочує час формування сторінки. Найцікавішу частину, а саме реалізацію RAG-пайплайну та AI-асистента розглянуто у підрозділі 3.2.

3.2. Реалізація RAG-пайплайну та AI-асистента на основі Claude API

На основі методів і засобів, обґрунтованих у підрозділі 2.3, реалізовано RAG-пайплайн платформи UkrWay мовою Python 3.14. Пайплайн складається з двох незалежних процесів: офлайн-частина збирає і індексує повідомлення з Telegram-груп мігрантів, онлайн-частина обробляє кожен запит користувача в реальному часі. Обидва процеси реалізовані у модулі `knowledge`.

Збір даних через бібліотеку `Telethon`. Офлайн-процес починається з `management`-команди `fetch_telegram`, яка підключається до Telegram API через бібліотеку `Telethon` і збирає повідомлення з трьох чатів – для Польщі, Німеччини і Великої Британії. Ініціалізацію клієнта наведено у лістингу 3.5.

```

128     async with TelegramClient(session_path, int(api_id), api_hash) as client:
129         for country in countries:
130             chat = getattr(settings, CHATS[country])
131             self.stdout.write(f'\n📄 Скачую чат [{country}]: {chat}')
132             if since_date:
133                 self.stdout.write(f' 📅 Тільки з {since_date.date()} (нові → старі, зупинка при досягненні дати)')
134             if limit:
135                 self.stdout.write(f' ⚠️ Ліміт: {limit} повідомлень')
136             count = 0
137             try:

```

Лістинг 3.5. Ініціалізація Telethon-клієнта

`TelegramClient` приймає три параметри: шлях до файлу сесії, числовий `api_id` і рядковий `api_hash`, які обидва зчитуються з `.env` через `python-dotenv`. Конструкція

async with використовується як асинхронний контекстний менеджер, з'єднання автоматично закривається після завершення збору. При першому запуску Telethon інтерактивно запитує номер телефону і код підтвердження у терміналі; результат авторизації зберігається у файл session.session – при повторних запусках введення коду не потрібне. Реальний процес авторизації наведено на рис. 3.3.

```
(.venv) PS C:\Users\Admin\PycharmProjects\Informationwebsystem\ukrway> python manage.py fetch_telegram --country=germany --since=2025-05-01
System check identified some issues:

WARNINGS:
<django.template.backends.django.DjangoTemplates object at 0x000002E12D491550>: (templates.W003) 'markdown_extras' is used for multiple temp
extras'
Файл C:\Users\Admin\PycharmProjects\Informationwebsystem\ukrway\knowledge\data\telegram_raw.json вже існує. Перезаписати? [y/N] y
Please enter your phone (or bot token): +380952431685
Please enter the code you received: 72176
```

Рис. 3.3. Авторизація Telethon-клієнта при першому запуску fetch_telegram

Рис. 3.3 показує реальний процес першого запуску: після введення команди `python manage.py fetch\_telegram --country=germany --since=2025-05-01` система запросила номер телефону і код підтвердження з Telegram і успішно ініціалізувалась з нашим телеграм профілем. Параметр --since=2025-05-01 обмежує збір повідомленнями після вказаної дати при досягненні цієї дати ітерація зупиняється без завантаження зайвих даних. Цикл збору повідомлень наведено у лістингу 3.6.


```

137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
...
try:
    # з since_date – з нових до старих, зупиняємось на потрібній даті
    # без since_date – з старих до нових (стандартна поведінка)
    iter_kwargs = {} if since_date else {'reverse': True}
    async for message in client.iter_messages(chat, **iter_kwargs):
        if not message.text:
            continue
        if since_date:
            msg_date = message.date.replace(tzinfo=None)
            if msg_date < since_date:
                self.stdout.write(f' ■ Досягнуто дату {since_date.date()}, зупиняємось')
                break
        if (country, message.id) in existing_ids:
            continue
        all_messages.append({
            'id': message.id,
            'date': message.date.isoformat(),
            'text': message.text,
            'from_id': message.sender_id,
            'country': country,
            'reply_to_id': message.reply_to_msg_id,
        })
        existing_ids.add((country, message.id))
        count += 1
        if count % 1000 == 0:
            save(f'{count} повідомлень [{country}]')
        if limit and count >= limit:
            self.stdout.write(f' ■ Досягнуто ліміт {limit}')
            break

```

Лістинг 3.6. Цикл збору повідомлень через `iter_messages` (`fetch_telegram.py`)

Метод `iter_messages` обходить повідомлення від нових до старих, при використанні параметра `--since` це дозволяє зупинитись одразу при досягненні вказаної дати без завантаження зайвих даних. Для режиму без фільтра дати використовується `reverse=True` – від старих до нових. Telethon автоматично виконує пагінацію батчами по 100 повідомлень. Для кожного повідомлення зберігається `reply_to_id` – ідентифікатор повідомлення на яке воно відповідає; це дозволяє у наступному кроці відновлювати Q&A пари. Дедублікація реалізована через множину `existing_ids` з парами `(country, message.id)` – повторне завантаження тих самих повідомлень при повторному запуску неможливе. Автозбереження кожні 1000 повідомлень захищає від втрати даних при обриві з'єднання. Результат збору для Німеччини наведено на рис. 3.4.

```

32000 повідомлень [germany] збережено (32500 всього)
33000 повідомлень [germany] збережено (33500 всього)
34000 повідомлень [germany] збережено (34500 всього)
35000 повідомлень [germany] збережено (35500 всього)
36000 повідомлень [germany] збережено (36500 всього)
37000 повідомлень [germany] збережено (37500 всього)
38000 повідомлень [germany] збережено (38500 всього)
39000 повідомлень [germany] збережено (39500 всього)
40000 повідомлень [germany] збережено (40500 всього)
41000 повідомлень [germany] збережено (41500 всього)
42000 повідомлень [germany] збережено (42500 всього)
43000 повідомлень [germany] збережено (43500 всього)
44000 повідомлень [germany] збережено (44500 всього)
45000 повідомлень [germany] збережено (45500 всього)
■ Досягнуто дату 2025-05-01, зупиняємось
✓ germany: 45916 нових повідомлень

Збережено 46416 повідомлень у C:\Users\Admin\PycharmProjects\Informationwebsystem\ukrway\knowledge\data\telegram_raw.json

```

Рис. 3.4. Результат збору повідомлень командою `fetch_telegram`

З рис. 3.4 видно, що прогрес виводиться кожні 1000 повідомлень, збір зупинився після досягнення дати 2025-05-01 з результатом 45916 нових повідомлень. Це було взято з телеграм чату “УКРАЇНЦІ В НІМЕЧЧИНІ”. Загалом збережено 46416 повідомлень у файл `knowledge/data/telegram_raw.json`.

Chunking і індексування. Після збору повідомлень виконується команда `build_knowledge_base`, яка перетворює сирі повідомлення на векторні представлення і зберігає їх у ChromaDB. Ключовий алгоритм – функція `_build_chunks` – реалізує два проходи обробки. Її наведено у лістингу 3.7.

```

71 def _build_chunks(messages: list[dict]) -> list[dict]: 1 usage  Ⓜ Alex *
72     msg_by_key = {}
73     (m['country'], m['id']): m
74     for m in messages
75     if m.get('id') is not None
76 }
77
78 chunks = []
79 reply_msg_ids = set()

```

Лістинг 3.7. Функція групування повідомлень у чанки (`_build_chunks`, `build_knowledge_base.py`)

Перший прохід знаходить `reply`-ланцюжки: якщо повідомлення має `reply_to_id` – воно об'єднується з оригінальним у Q&A пару формату «Питання: ...Відповідь: ...». Такі пари є найціннішим типом контенту – вони містять і запит і

відповідь у структурованому вигляді. Другий прохід обробляє решту повідомлень за часовим вікном: повідомлення об'єднуються в один чанк якщо між ними не більше 5 хвилин, вони з однієї країни і загальна довжина не перевищує 500 символів. Мінімальна довжина чанку – 80 символів (MIN_CHUNK_LEN); для первинної фільтрації сирих повідомлень використовується окрема константа MIN_MSG_LEN = 60 символів. Процес результату індексування наведено на рис. 3.5.

```
25600/26210 чанків проіндексовано
25700/26210 чанків проіндексовано
25800/26210 чанків проіндексовано
25900/26210 чанків проіндексовано
26000/26210 чанків проіндексовано
26100/26210 чанків проіндексовано
26200/26210 чанків проіндексовано
26210/26210 чанків проіндексовано
```

■ База знань побудована: 26210 чанків у C:\Users\Admin\PycharmProjects\Informationwebsystem\ukrway\knowledge\data\chroma_db

Рис. 3.5. Завершення індексування: 26210 чанків збережено у ChromaDB

З рис. 3.5 видно, що всі 26210 чанків проіндексовано і збережено у knowledge/data/chroma_db. З 45916 вхідних повідомлень Німеччини утворилось 26210 векторних представлень, коефіцієнт стиснення близько 1.75. Це пояснюється об'єднанням Q&A пар і часовим групуванням: кілька коротких повідомлень однієї розмови перетворюються на один змістовний чанк.

RAGService: визначення країни і семантичний пошук. Онлайн-частина пайплайну реалізована у класі RAGService за патерном Singleton єдиний екземпляр ініціалізується при першому запиті і залишається в пам'яті на весь час роботи сервера (lazy initialization). Перший крок онлайн-обробки – визначення країни запиту через метод detect_country, наведений у лістингу 3.8.

```

75     def detect_country(self, query: str) -> str | None: 1 usage  ⤵ Alex
76         text = query.lower()
77         scores = {country: 0 for country in COUNTRY_KEYWORDS}
78         for country, keywords in COUNTRY_KEYWORDS.items():
79             for kw in keywords:
80                 if kw in text:
81                     scores[country] += 1
82         best = max(scores, key=lambda c: scores[c])
83         return best if scores[best] > 0 else None

```

Лістинг 3.8. Метод визначення країни запиту (knowledge/services/rag.py)

Алгоритм `detect_country` перетворює запит у нижній регістр і підраховує кількість збігів з ключовими словами для кожної країни. Для Польщі – «pesel», «karta pobytu», «zlotih»; для Німеччини – «anmeldung», «bürgergeld», «jobcenter»; для Великої Британії – «brp», «nhs», «homes for ukraine». Обирається країна з найбільшим рахунком; якщо збігів немає – повертається `None` і пошук виконується без географічного фільтру. Словниковий підхід обрано свідомо: для предметної області міграційної термінології він точніший і прозоріший ніж ML-класифікатор, тому що кожне рішення можна перевірити і пояснити.

Семантичний пошук реалізовано у методі `search()` класу `RAGService`. Метод виконує чотири послідовні операції. Спочатку запит користувача векторизується тією самою моделлю `paraphrase-multilingual-MiniLM-L12-v2`, що використовувалась при індексуванні – семантична сумісність гарантована. Потім `ChromaDB` виконує пошук по векторній базі і повертає `top_k=7` найближчих чанків за косинусною відстанню. Якщо країну вдалось визначити через `detect_country` – до запиту додається фільтр по полю `country` у метаданих. Далі результати фільтруються: чанки з $\text{cosine distance} \geq 0.7$ відкидаються як нерелевантні і не потрапляють у промпт. Нарешті чанки що пройшли фільтр ре-ранжуються за формулою: $\text{релевантність} \times 0.7 + \text{свіжість} \times 0.3$, свіжіші повідомлення отримують перевагу при однаковій семантичній близькості. Це важливо для міграційних даних де законодавство змінюється швидко і повідомлення дворічної давнини може бути вже неактуальним. Якщо після фільтрації не залишилось жодного

чанку – метод повертає порожній результат і система переходить до graceful fallback, Claude відповідає з власних знань або використовує інтегрований web_search (інтернет пошук).

Формування промпту і виклик Claude API. Результати пошуку передаються у функцію _build_system_prompt, яка формує системний промпт з RAG-контекстом. Її реалізацію наведено у лістингу 3.9.

```

23 def _build_system_prompt(user_text: str) -> str: 1 usage  Ⓐ Alex *
24     rag_result = rag_service.search(user_text)
25     chunks = rag_result.get('chunks', [])
26     country = rag_result.get('country')
27
28     if not chunks:
29         logger.info(f'[RAG] Чанків не знайдено | країна: {country}')
30         return SYSTEM_PROMPT
31
32     dates = rag_result.get('dates', [])
33     logger.info(f'[RAG] Знайдено {len(chunks)} чанків | країна: {country} | дати: {dates}')
34
35     country_label = COUNTRY_NAMES.get(country, default: '') if country else ''
36     if country_label:
37         context_header = f'Контекст з реального досвіду мігрантів у {country_label} (з Telegram-спільноти):'
38     else:
39         context_header = 'Контекст з реального досвіду українських мігрантів (з Telegram-спільнот):'
40
41     context_lines = []
42     for i, chunk in enumerate(chunks):
43         date_str = f' [{dates[i]}]' if i < len(dates) and dates[i] else ''
44         context_lines.append(f'- {chunk}{date_str}')
45
46     context_block = '\n'.join(context_lines)
47
48     return (
49         f'{SYSTEM_PROMPT}\n\n'
50         f'[КОНТЕКСТ]\n'
51         f'{context_header}\n'
52         f'{context_block}\n\n'
53         f'Використовуй цей контекст як першоджерело – це реальний досвід людей. '
54         f'Якщо контекст суперечить загальним знанням – довіряй контексту.'

```

Лістинг 3.9. Функція формування системного промпту з RAG-контекстом
(chat/views.py)

Функція самостійно звертається до RAGService, передаючи текст запиту: rag_service.search(user_text). З результату витягуються чанки, дати публікації та визначена країна. Якщо RAGService повернув чанки, то базовий системний промпт доповнюється блоком [КОНТЕКСТ] з реальними повідомленнями та датами публікації. Заголовок блоку формується динамічно: якщо вдалося

визначити країну – вказується конкретна («мігрантів у Польщі»), інакше – загальне формулювання. Інструкція «Використовуй цей контекст як першоджерело – це реальний досвід людей. Якщо контекст суперечить загальним знанням – довіряй контексту» підвищує пріоритет реального досвіду мігрантів над загальними знаннями моделі. Якщо чанків немає, то повертається базовий промпт і Claude відповідає з власних знань. Базовий системний промпт прописаний такий: «Ти – асистент платформи UkrWay. Допомогаєш українським мігрантам у Польщі, Німеччині та Великій Британії. Відповідай тільки українською мовою. Давай практичні поради щодо документів, житла, роботи та медицини. Якщо не знаєш точної відповіді – чесно скажи про це і порадь де шукати інформацію. Будь дружнім та лаконічним. Відповідай коротко і по суті – без зайвих заголовків, таблиць та емодзі. Використовуй звичайний текст з короткими абзацами.» Додатково до кожного запиту підключено вбудований інструмент `web_search_20250305` це потужний інструмент, при використанні якого Claude сам вирішує коли звернутися до пошуку, якщо ні RAG-контекст ні власні знання не дають достатньої відповіді. Алгоритм повного онлайн-циклу від запиту до відповіді наведено на рис. 3.6.

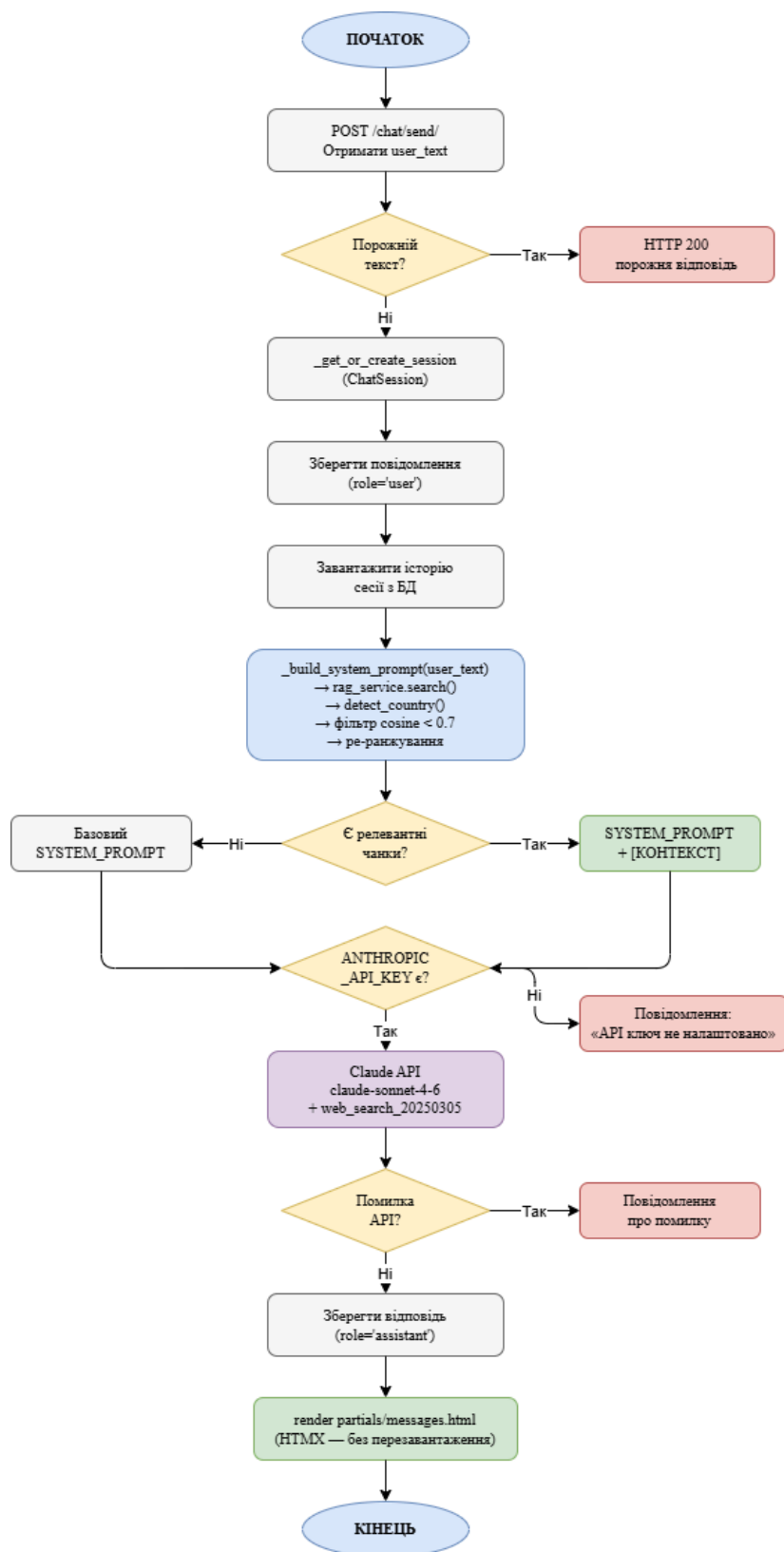


Рис. 3.6. Блок-схема алгоритму обробки запиту AI-асистента (send_message)

Блок-схема на рис. 3.6 відображає повний алгоритм функції send_message – від моменту коли користувач натискає кнопку відправки до появи відповіді в чаті.

Алгоритм має чотири точки прийняття рішень. Спочатку перевірка на порожній текст: якщо поле введення порожнє функція повертає HTTP 200 з порожньою відповіддю без жодних звернень до бази чи API. Далі наявність релевантних чанків після RAG-пошуку: якщо ChromaDB повернула результати з $\text{cosine distance} < 0.7$ і ре-ранжування відібрало найсвіжіші, то системний промпт доповнюється блоком [КОНТЕКСТ] з реальними повідомленнями мігрантів; якщо ні – Claude отримує лише базовий промпт і відповідає з власних знань або через `web_search`. Третя наявність API-ключа в файлі `.env`: без нього користувач бачить повідомлення про проблеми на боці серверу додатку. Четверта – успішність виклику Claude API: при збої повертається зрозуміле повідомлення про помилку, а не технічний стек-трейс. Якщо всі перевірки пройдено, відповідь асистента зберігається у `ChatMessage` і повертається через `HTMX-partial` `partials/messages.html`, що оновлює тільки блок повідомлень без перезавантаження всієї сторінки.

Таким чином, RAG-пайплайн UkrWay реалізовано як два незалежних процеси з чітким розподілом відповідальності. Офлайн-частина зібрала 45916 повідомлень і побудувала бази з 26210 чанків, що виконується один раз і оновлюється за потреби. Онлайн-частина це визначення країни, семантичний пошук з ре-ранжуванням, формування промпту і виклик Claude API з `web_search` – виконується при кожному повідомленні користувача без помітної затримки. Оцінку ефективності реалізованого рішення здійснено у підрозділі 3.3.

3.3. Тестування системи та оцінка ефективності AI-асистента

Після реалізації всіх модулів платформи проведено комплексне тестування функціональне і якісне. Мета функціонального тестування – переконатись що кожна сторінка, кнопка і форма працюють відповідно до визначених у підрозділі 2.1 параметрів. Мета якісного – оцінити реальну корисність AI-асистента для цільової аудиторії.

Функціональне тестування. Перевірено всі ключові сценарії взаємодії користувача з платформою (табл. 3.1).

Таблиця 3.1

Результати функціонального тестування платформи UkrWay

№	Модуль	Тест-кейс	Очікуваний результат	Статус
1	core	Перехід на головну сторінку /	Відображення hero-секції з вибором країни	Пройдено
2	countries	Перехід на /countries/poland/	Сторінка з 4 вкладками і статтями в акордеоні	Пройдено
3	countries	Перемикання вкладок Документи / Житло / Робота / Медицина	Відображення відповідних статей без перезавантаження	Пройдено
4	chat	Надсилання повідомлення в AI-чат	Відповідь асистента без перезавантаження сторінки (HTMX)	Пройдено
5	chat	Надсилання порожнього повідомлення	HTTP 200 без відповіді, чат не змінюється	Пройдено
6	chat	Кнопка «Очистити» сесію	Історія діалогу видаляється, асистент вітається знову	Пройдено
7	resources	Фільтрація по країні «Польща»	Відображення 6 організацій з прапором Польщі	Пройдено
8	resources	Фільтрація по категоріям, країнам	Відображення ресурсів в кожній окремій країні	Пройдено

Продовження таблиці 3.1

№	Модуль	Тест-кейс	Очікуваний результат	Статус
9	resources	Карта Leaflet.js – перехід на сторінку	Маркери організацій у трьох країнах на карті	Пройдено
10	chat	Недоступний ANTHROPIC_API_KEY	Повідомлення «API ключ не налаштовано» замість помилки сервера	Пройдено
11	knowledge	RAG-база не ініціалізована (ChromaDB порожня)	Graceful fallback – Claude відповідає без контексту	Пройдено

Усі 11 тест-кейсів пройдено успішно. Особливо важливими є тести 10 і 11 вони підтверджують що система коректно обробляє нештатні ситуації: відсутність API-ключа і порожня база знань не призводять до падіння сервера, а повертають зрозумілі повідомлення користувачу. Саме ці сценарії закривають функціональний параметр Ф7 (graceful fallback), визначений у підрозділі 2.1.

Оцінка ефективності AI-асистента. Тестування проводилось з реально наповненою базою знань – 26210 чанків з Telegram-груп мігрантів у Німеччині. До системи надсилались запити що відповідають реальним потребам цільової аудиторії, описаним у підрозділі 1.3. Тестування проводилось на балансі Claude API у 5 доларів США; сукупні витрати на всі тестові запити склали близько 0.40 USD.

Для оцінки якості відповідей використано запит «Привіт, чи видає Німеччина тимчасовий захист українцям у яких вже був тимчасовий захист в іншій країні ЄС?» реальне питання яке часто виникає у мігрантів при переїзді між країнами. Це нетривіальний запит: правильна відповідь потребує знання і офіційного правила ЄС (один захист одночасно), і реального досвіду конкретної країни. Відповідь AI-асистента наведено на рис. 3.7.

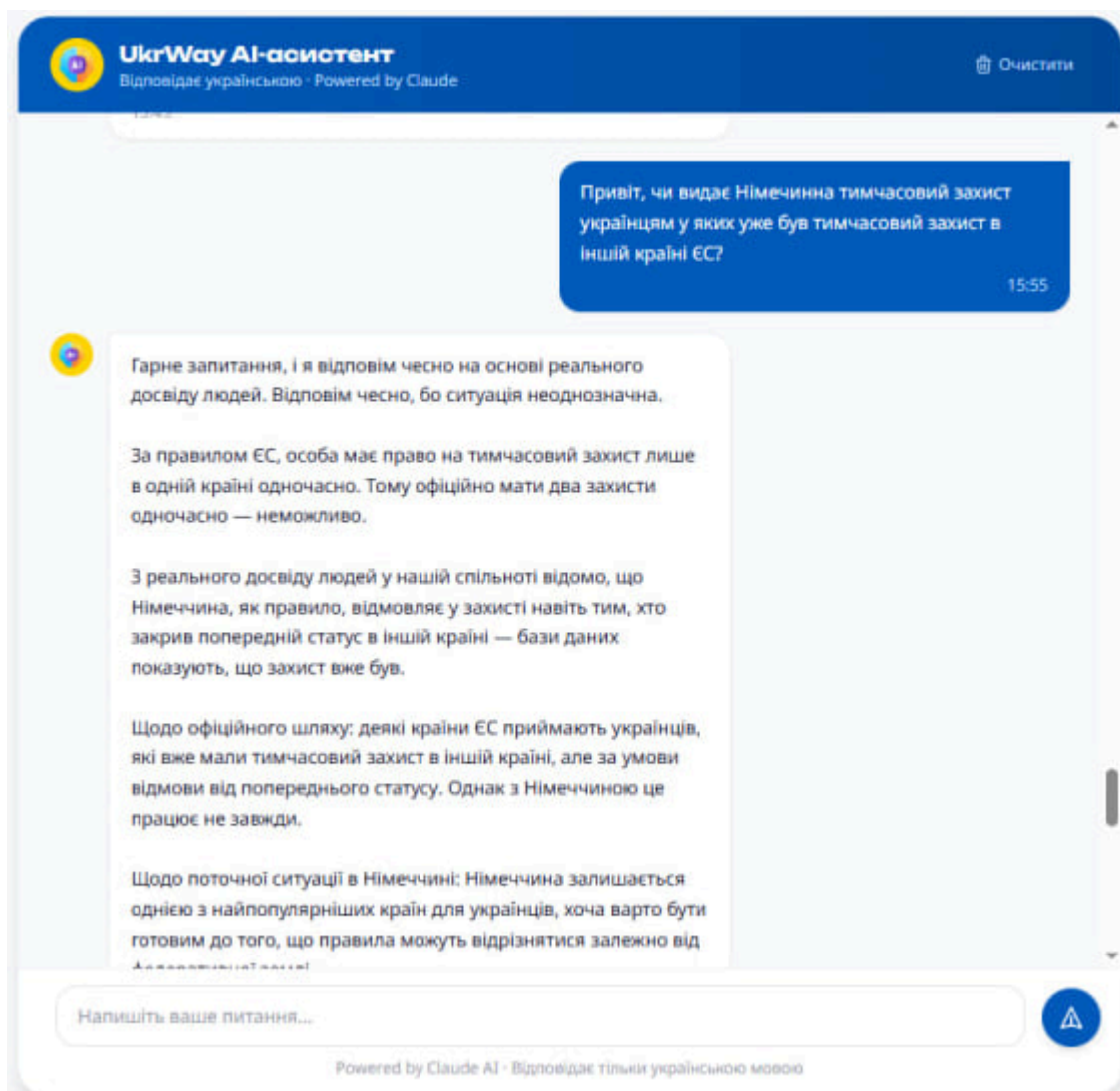


Рис. 3.7. Відповідь AI-асистента UkrWay на запит щодо тимчасового захисту в Німеччині

З рис. 3.7 видно що асистент надав структуровану відповідь з трьох частин: офіційне правило ЄС, реальний досвід з Telegram-спільноти («бази даних показують що захист вже був»), і поточна ситуація в Німеччині з застереженням що правила можуть відрізнятись. Відповідь сформована українською мовою, без зайвих заголовків і таблиць – відповідно до налаштованого системного промпту. Посилання на реальний досвід людей – саме той результат який закладався при проєктуванні RAG-підходу у підрозділі 2.3.

Логи Claude API наведено на рис. 3.8.

Logs							
Last refresh time: May 16, 2026 at 4:06 PM GMT+3							
TIME (GMT+3)	ID	MODEL	INPUT TOKENS	OUTPUT TOKENS	TYPE	SERVICE TIER	REQUEST
2026-05-16 15:59:50	req_011Cb6Hha83xLXAXCg1oQNhw	claude-sonnet-4-6	28611	646	HTTP	Standard	
2026-05-16 15:59:35	req_011Cb6Hha83xLXAXCg1oQNhw	claude-sonnet-4-6	7045	150	HTTP	Standard	
2026-05-16 15:55:43	req_011Cb6HPScbt3oKvB9ep2MSW	claude-sonnet-4-6	16140	581	HTTP	Standard	
2026-05-16 15:55:29	req_011Cb6HPScbt3oKvB9ep2MSW	claude-sonnet-4-6	6533	111	HTTP	Standard	
2026-05-16 15:45:30	req_011Cb6GcMjQrXjq4XZKXnXP7	claude-sonnet-4-6	21102	606	HTTP	Standard	
2026-05-16 15:45:16	req_011Cb6GcMjQrXjq4XZKXnXP7	claude-sonnet-4-6	6232	93	HTTP	Standard	
2026-05-16 14:43:44	req_011Cb6BuZG6MaRvAS4Jrt46u	claude-sonnet-4-6	3181	470	HTTP	Standard	
2026-05-16 14:39:28	req_011Cb6Ba8kg6mRmhr9VKUMsz	claude-sonnet-4-6	2212	903	HTTP	Standard	
2026-05-16 14:31:17	req_011Cb6Ax8SZq1eKR3s6njrtz	claude-sonnet-4-6	1397	871	HTTP	Standard	
2026-05-16 14:26:16	req_011Cb6Aa3R5rKLmi13HXPSFL	claude-sonnet-4-6	209	627	HTTP	Standard	

Рис. 3.8. Логи викликів Claude API у Claude Console (platform.claude.com)

Рис. 3.8 відображає реальні виклики API під час тестування – модель `claude-sonnet-4-6`, тип HTTP, рівень Standard. Кількість вхідних токенів коливається залежно від конфігурації запиту: прості запити без RAG-контексту споживають мінімум токенів, запити з повним RAG-контекстом і підключеним `web_search` значно більше. Це пряма залежність: чим більше можливостей задіяно, тим якіснішою і точнішою є відповідь, і тим більше API-кредитів витрачається. У ході тестування підібрано баланс між якістю відповіді і вартістю виклику: підключення `web_search` помітно підвищило релевантність відповідей на запити про актуальні зміни в законодавстві, хоча і збільшило витрати на такі запити. Загальна сума витрат на тестування близько 0.40 USD з початкового балансу 5 USD підтверджує що платформа є економічно прийнятною для некомерційного освітнього проєкту.

Результати тестування підтверджують що реалізована система відповідає поставленим функціональним і нефункціональним параметрам. Функціональне тестування довело коректну роботу всіх 11 перевірених сценаріїв, включно з

нештатними. AI-асистент з RAG-контекстом надає відповіді що спираються на реальний досвід мігрантів, головна ціль це саме те що було визначено як ключова відмінність UkrWay від існуючих рішень у підрозділі 1.4. Є напрями для подальшого вдосконалення: розширення бази знань повідомленнями з польських і британських Telegram-груп, додаткове налаштування системного промπτу і оптимізація порогу cosine distance для підвищення точності пошуку.

Висновки до розділу 3

За результатами практичної реалізації інформаційної веб-системи UkrWay встановлено таке.

Серверну частину і інформаційний модуль реалізовано засобами Django 5.x і Python 3.14 відповідно до спроектованої у підрозділі 2.4 архітектури. Моделі Country, InfoCategory, InfoArticle і Resource реалізують трирівневу ієрархію даних через Django ORM без написання SQL-запитів вручну. Адміністративна панель Django Admin з налаштованими list_editable, list_filter і prepopulated_fields забезпечує управління контентом без програмування – функціональний параметр Фб закрито. Функція country_detail завантажує всі пов'язані дані одним запитом через prefetch_related, що унеможливлює проблему багаторазових звернень до бази при відображенні сторінки країни.

RAG-пайплайн реалізовано як два незалежних процеси. Офлайн-частина збирила 45916 повідомлень з Telegram-груп мігрантів Німеччини командою fetch_telegram з параметром --since=2025-05-01; алгоритм _build_chunks у два проходи – Q&A пари через reply_to_id і часове вікно 5 хвилин – перетворив їх на 26210 семантичних чанків у ChromaDB. Онлайн-частина при кожному запиті виконує визначення країни (detect_country), семантичний пошук з фільтром cosine distance < 0.7 і ре-ранжуванням за формулою релевантність × 0.7 + свіжість × 0.3, формує системний промπτ з блоком [КОНТЕКСТ] і викликає Claude API з підключеним інструментом web_search_20250305. Три рівні graceful fallback – відсутність RAG-контексту, відсутність API-ключа, помилка API – забезпечують коректну обробку нештатних ситуацій у модулі чату: замість технічної помилки

сервера користувач отримує зрозуміле повідомлення, тоді як решта платформи продовжує працювати повноцінно.

Функціональне тестування підтвердило коректну роботу всіх 11 перевірених сценаріїв. Нештатні ситуації – порожня база знань і відсутній API-ключ – обробляються коректно без падіння сайту. Якісне тестування на реальних міграційних запитах показало: AI-асистент надає структуровані україномовні відповіді з посиланням на досвід Telegram-спільнот, коректно обробляє нетривіальні запити і при потребі доповнює RAG-контекст актуальними даними через `web_search`. Тестування проводилось на балансі 5 USD; сукупні витрати склали близько 0.40 USD – платформа є економічно прийнятною для некомерційного освітнього проєкту. Серед напрямів подальшого розвитку – розширення бази знань повідомленнями з польських і британських спільнот та оптимізація порогу `cosine distance` для підвищення точності пошуку.

ВИСНОВКИ

У кваліфікаційній роботі вирішено завдання проектування та розробки інформаційної веб-системи UkrWay для підтримки українських мігрантів у Польщі, Німеччині та Великій Британії з інтегрованим AI-асистентом на основі технології RAG. За результатами виконаної роботи зроблено такі висновки.

Досліджено соціально-економічні передумови та масштаби української міграції після лютого 2022 року. Встановлено, що станом на початок 2026 року за межами України перебуває близько 5,9 млн зареєстрованих біженців, з яких 5,3 млн – у Європі [3]. Рівень зайнятості серед переміщених осіб впав з 73% до 67% [6], безробіття серед нещодавно переміщених сягало 24% [7], збитки від зруйнованої інфраструктури оцінюються у 486 млрд дол. США [8]. Ці показники визначили специфіку цільової аудиторії: люди у стресовій ситуації переїзду, які одночасно вразливі у фінансовому, правовому і побутовому вимірах і потребують практичної інформаційної підтримки.

Проаналізовано інформаційні потреби мігрантів та виявлено недоліки існуючих рішень. Встановлено чотири ключові тематичні категорії потреб: легалізація, житло, зайнятість і медицина – кожна специфічна для конкретної країни і етапу перебування. Telegram є одним з основних каналів пошуку інформації для 32% мігрантів [10], проте повідомлення там неструктуровані і не мають механізмів верифікації. Порівняльний аналіз шести існуючих платформ показав: жодна не поєднує україномовний інтерфейс, охоплення трьох цільових країн, AI-асистента і доступ до реального досвіду спільнот в одному місці.

Обґрунтовано вибір технологічного стеку з-поміж альтернатив. Django 5.x обрано над Flask і FastAPI через вбудовану адміністративну панель, ORM і сесійний механізм; HTMX – над React і Vue через відсутність потреби у складному фронтенд-білді і розмір 14 КБ проти 47 КБ [21]; RAG-архітектуру –

над чистим LLM і fine-tuning через динамічне оновлення бази знань без перенавчання моделі [26]; ChromaDB – через нульову конфігурацію і повну автономність без зовнішніх сервісів [24]; модель paraphrase-multilingual-MiniLM-L12-v2 – через підтримку понад 50 мов включно з українською [25].

Спроектовано архітектуру системи, базу даних і RAG-пайплайн. Платформа побудована на п'яти модулях з однонаправленими залежностями; база даних містить шість моделей у трьох логічних групах – контентній, ресурсній і сесійній. RAG-пайплайн спроектовано з двома проходами chunking: Q&A пари через reply_to_id і часове вікно 5 хвилин; пошук здійснюється з фільтром cosine distance < 0.7 і re-ранжуванням за формулою релевантність $\times 0.7$ + свіжість $\times 0.3$. Інтерфейс спроектовано під аудиторію в стресовій ситуації: синьо-жовта кольорова схема (#005BBB, #FFD500), відсутність реєстрації, мобільна адаптивність від 320 px.

Розроблено веб-систему UkrWay з чотирма функціональними модулями. Інформаційний модуль містить структуровані статті по чотирьох категоріях для трьох країн з управлінням через Django Admin. AI-асистент інтегрує RAG-пошук по базі з 26210 чанків реальних Telegram-повідомлень, виклик Claude API (claude-sonnet-4-6) і вбудований інструмент web_search_20250305 для доповнення відповідей актуальними даними. Карта ресурсів на основі Leaflet.js відображає 17 верифікованих організацій допомоги у трьох країнах з фільтрацією по семи категоріях. Усі модулі реалізовано мовою Python 3.14 у межах єдиного Django-проєкту.

Проведено комплексне тестування системи. Функціональне тестування підтвердило коректну роботу всіх 11 перевірених сценаріїв включно з нештатними – відсутність API-ключа і порожня база знань обробляються без падіння сервера. Якісне тестування показало: AI-асистент надає структуровані україномовні відповіді з посиланням на реальний досвід мігрантів і при потребі доповнює RAG-контекст через web_search. Тестування проводилось на балансі 5 USD; сукупні витрати склали близько 0.40 USD, що підтверджує економічну

прийнятність платформи для некомерційного проєкту. Серед перспектив подальшого розвитку – розширення бази знань повідомленнями з польських і британських Telegram-груп та оптимізація порогу cosine distance для підвищення точності семантичного пошуку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Lewis P., Perez E., Piktus A. та ін. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. 2020. arXiv:2005.11401. URL: <https://arxiv.org/abs/2005.11401> (дата звернення: 22.03.2026).
2. Conneau A., Khandelwal K., Goyal N. та ін. Unsupervised Cross-lingual Representation Learning at Scale. 2019. arXiv:1911.02116. URL: <https://arxiv.org/abs/1911.02116> (дата звернення: 22.03.2026).
3. Ukraine Situation: UNHCR's 2026 Plans and Financial Requirements. UNHCR – The UN Refugee Agency : офіц. сайт. 2025. URL: <https://data.unhcr.org/en/documents/download/120716> (дата звернення: 26.03.2026).
4. DTM Ukraine – Internal Displacement Report – General Population Survey Round 18 (October 2024). International Organization for Migration (IOM). 2024. URL: <https://dtm.iom.int/reports/ukraine-internal-displacement-report-general-population-survey-round-18-october-2024> (дата звернення: 01.04.2026).
5. Ukraine Situation: Regional Refugee Response Plan – January–December 2024. UNHCR – The UN Refugee Agency : офіц. сайт. 2024. URL: <https://data.unhcr.org/en/documents/details/105903> (дата звернення: 03.04.2026).
6. Employment, Mobility and Labour Market Dynamics in Ukraine (November 2024). International Organization for Migration (IOM). 2024. URL: <https://crisisresponse.iom.int/sites/g/files/tmzbd11481/files/appeal/documents/Ukraine%20-%20Employment,%20Mobility%20and%20Labour%20Market%20Dynamics%20-%20Nov%202024.pdf> (дата звернення: 03.04.2026).
7. Displaced Ukrainians Face Disproportionate Unemployment Rates, IOM Report Reveals. IOM Ukraine : сайт. 22.11.2024. URL: <https://ukraine.iom.int/news/displaced-ukrainians-face-disproportionate-unemployment-rates-iom-report-reveals> (дата звернення: 11.04.2026).

8. Ukraine and Neighbouring Countries Crisis Response Plan 2025. International Organization for Migration (IOM). 2025. URL: <https://crisisresponse.iom.int/response/ukraine-and-neighbouring-countries-crisis-response-plan-2025> (дата звернення: 11.04.2026).
9. Kardas P., Mogilevkina I., Aksoy N. та ін. Barriers to Healthcare Access and Continuity of Care Among Ukrainian War Refugees in Europe: Findings from the RefuHealthAccess Study. *Frontiers in Public Health*. 2025. Vol. 13. DOI: 10.3389/fpubh.2025.1516161.
10. Ukraine Situation: Quarterly Operational Update Q2 2024. UNHCR – The UN Refugee Agency : офіц. сайт. 2024. URL: <https://data.unhcr.org/en/documents/download/110278> (дата звернення: 20.04.2026).
11. Visit Ukraine – сервісний портал для українців. Visit Ukraine : сайт. 2024. URL: <https://visitukraine.today/> (дата звернення: 20.04.2026).
12. Dla obywateli Ukrainy / Для громадян України. Уряд Республіки Польща : офіц. сайт. 2024. URL: <https://www.gov.pl/web/ua> (дата звернення: 21.04.2026).
13. Move to the UK if you're coming from Ukraine. Government of the United Kingdom : офіц. сайт. 2024. URL: <https://www.gov.uk/guidance/move-to-the-uk-if-youre-from-ukraine> (дата звернення: 21.04.2026).
14. Germany for Ukraine – Das Hilfeportal. Bundesregierung Deutschland : офіц. сайт. 2024. URL: https://www.germany4ukraine.de/UA/startseite_node.html (дата звернення: 21.04.2026).
15. Hadero H. AI could help scale humanitarian responses. But it could also have big downsides. *The Associated Press* : сайт. 14.11.2024. URL: <https://apnews.com/article/international-rescue-committee-ai-refugees-chatbots-9c5fae949c429c38019feabaeec5fafa> (дата звернення: 26.04.2026).
16. Pilot Report: Signpost AI Information Assistant. Signpost AI. 2025. URL: <https://www.signpostai.org/research/pilot-report-signpost-ai-information-assistant> (дата звернення: 28.04.2026).

17. Ap-regjeringen strammer inn reglene for fordrevne fra Ukraina. Regjeringen.no : офіц. сайт. 27.03.2026. URL: <https://www.regjeringen.no/no/aktuelt/ap-regjeringen-strammer-inn-reglene-for-fordrevne-fra-ukraina-menn-mellom-18-og-60-ar-far-ikke-lenger-kollektiv-beskyttelse/id3155103/> (дата звернення: 01.05.2026).
18. Men between the ages of 18 and 60 can no longer be granted temporary collective protection in Norway. UDI – Utlendingsdirektoratet : офіц. сайт. 2026. URL: <https://www.udi.no/en/important-messages/men-between-the-ages-of-18-and-60-can-no-longer-be-granted-temporary-collective-protection-in-norway/> (дата звернення: 02.05.2026).
19. Україна Online: Новини | Політика. Telegram : соц. мережа. 2024. URL: <https://t.me/UaOnlii> (дата звернення: 04.05.2026).
20. Afolabi Ifeoluwa James. API Framework Performance Benchmark: FastAPI vs Flask vs Django. Medium : сайт. 13.11.2025. URL: <https://medium.com/@afolabiiifeoluwa06/api-framework-performance-benchmark-fastapi-vs-flask-vs-django-e767ad51574c> (дата звернення: 09.05.2026).
21. HTMX vs React 2026: 14KB vs 47KB – When Each Wins. PkgPulse : сайт. 28.02.2026. URL: <https://www.pkgpulse.com/guides/htmx-vs-react-2026> (дата звернення: 13.05.2026).
22. Which Is the Best Python Web Framework: Django, Flask, or FastAPI? The PyCharm Blog. 18.02.2026. URL: <https://blog.jetbrains.com/pycharm/2025/02/django-flask-fastapi/> (дата звернення: 13.05.2026).
23. Django Documentation. Version 5.2. Django Software Foundation. 2025. URL: <https://docs.djangoproject.com/en/5.2/> (дата звернення: 13.05.2026).
24. Vector Database Comparison 2026: ChromaDB vs. Qdrant vs. pgvector vs. Pinecone vs. LanceDB for Production RAG. 4xxi : сайт. 18.03.2026. URL: <https://4xxi.com/articles/vector-database-comparison/> (дата звернення: 13.05.2026).

25. paraphrase-multilingual-MiniLM-L12-v2. Hugging Face : репозиторій моделей. 2023. URL:
<https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2>
(дата звернення: 13.05.2026).
26. Gao Y., Xiong Y., Gao X. та ін. Retrieval-Augmented Generation for Large Language Models: A Survey. 2023. arXiv:2312.10997. URL:
<https://arxiv.org/abs/2312.10997> (дата звернення: 13.05.2026).
27. Pretrained Models. Sentence Transformers : офіц. документація. 2024. URL:
https://www.sbert.net/docs/sentence_transformer/pretrained_models.html (дата звернення: 13.05.2026).

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)



Державний університет
інформаційно-комунікаційних технологій
Кафедра інформаційних систем та технологій

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Інформаційна веб-система підтримки українців під час міграції з використанням мови програмування Python

ВИКОНАВ
Слепченко Олександр
Романович

ГРУПА
ІСД-41

НАУКОВИЙ КЕРІВНИК
Ірина СРІБНА

РІК ЗАХИСТУ
Київ · 2026

Захист кваліфікаційної роботи

01 / 12



● UKRWAY · ЗАХИСТ

02 / 13

РОЗДІЛ 1

Актуальність, мета й завдання

5,9 млн біженців · 3 країни · 4 категорії потреб

АКТУАЛЬНІСТЬ

5,9 млн українських біженців у Європі станом на 2026 рік

32% мігрантів шукають інформацію через Telegram

100+ сайтів і каналів — інформація розпорошена

0 платформ, що поєднують укр. інтерфейс + AI + досвід спільноти

ЗАВДАННЯ — 6 ПУНКТІВ

1 Дослідити соціально-економічні передумови міграції після 2022 року

2 Проаналізувати інформаційні потреби мігрантів та існуючі рішення

3 Обґрунтувати вибір технологічного стеку

4 Спроекувати архітектуру системи, БД та RAG-пайплайн

5 Розробити веб-систему з AI-чатом і картою ресурсів

6 Протестувати систему та оцінити якість AI-асистента

МЕТА РОБОТИ

Розробка веб-системи **UkrWay** для підтримки українських мігрантів у Польщі, Німеччині та Великій Британії з AI-асистентом на основі RAG

РОЗДІЛ 1 · АНАЛІЗ

Аналіз предметної області

5,9+ млн

біженців усього
5,3 млн — у Європі

32%

мігрантів використовують Telegram як канал інформації

4

категорії потреб: документи, житло, робота, медицина

3

цільові країни:
Польща · Німеччина · Велика Британія

ПЛАТФОРМА	УКР. МОВА	AI	ДОСВІД СПІЛЬНОТИ	ЄДИНА ТОЧКА
gov.pl / gov.uk / Germany4Ukraine	Так / частково	— Ні	— Ні	— Ні
Telegram-спільноти	✓ Так	— Ні	✓ Так	— Ні
Signpost	✓ Так	~ Пілот	— Ні	— Ні
UkrWay	✓ Так	✓ Так (RAG)	✓ Так	✓ Так

ВИСНОВОК

Жодне існуюче рішення не закриває всі чотири критерії одночасно — це формує нішу для UkrWay.

РОЗДІЛ 2 · ПРОЄКТУВАННЯ

Технологічний стек

BACKEND

Django 5.x · Python 3.14

Вбудована адмінка, ORM, керування сесіями з коробки

FRONTEND

Tailwind CSS + HTMX

14 КБ проти 42 КБ React, без фронтенд-білду

БАЗА ДАНИХ

SQLite

Структуровані дані, повна інтеграція з Django ORM

ВЕКТОРНА БД

ChromaDB

Автономність, нульова конфігурація, локальний запуск

ЕМБЕДИНГИ

paraphrase-multilingual-MiniLM-L12-v2

50+ мов, підтримка української «з коробки»

LLM

Claude API · claude-sonnet-4-6

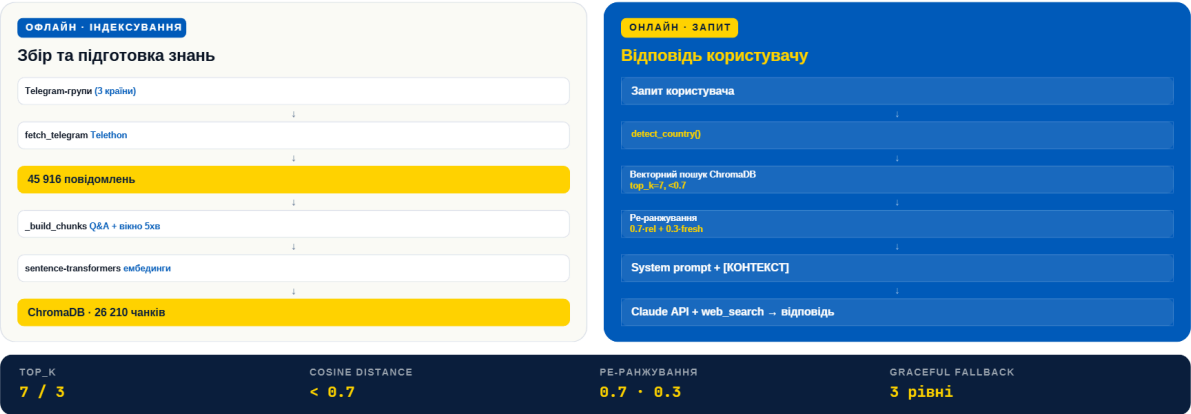
Якість відповідей українською + інструмент web_search

КАРТА

Leaflet.js

Інтерактивна карта 17 організацій-партнерів допомоги; фільтри по країні та категорії

RAG-архітектура — два потоки



5 Django-модулів

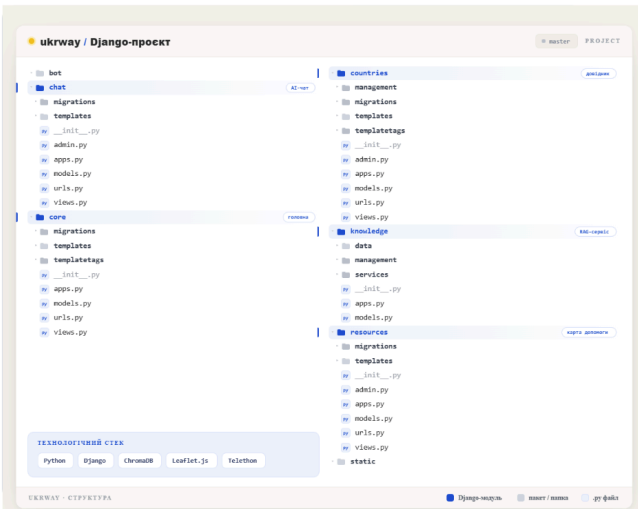
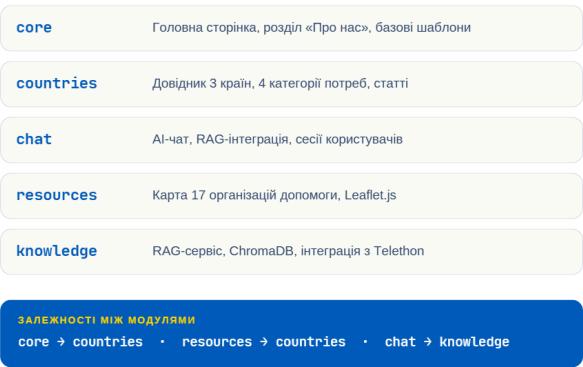


Рис. 2.4 — Структура файлів проєкту в PyCharm

Інтерфейс системи

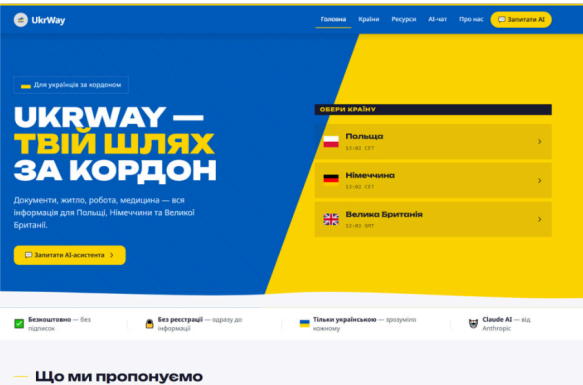


Рис. 1 — Головна сторінка платформи UkrWay

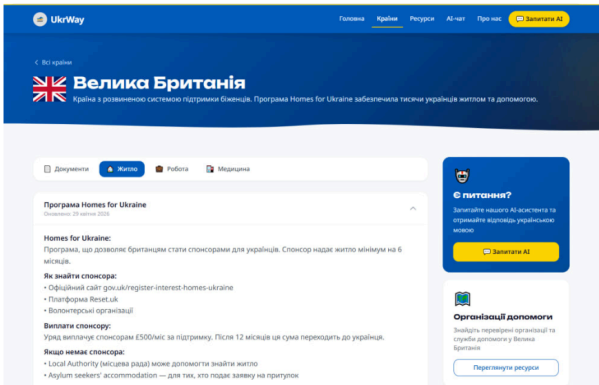


Рис. 2 — Сторінка країни з інформаційним модулем

AI-асистент і карта ресурсів

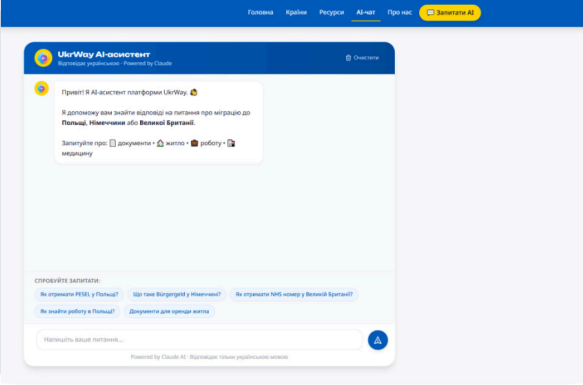


Рис. 3 — AI-асистент UkrWay

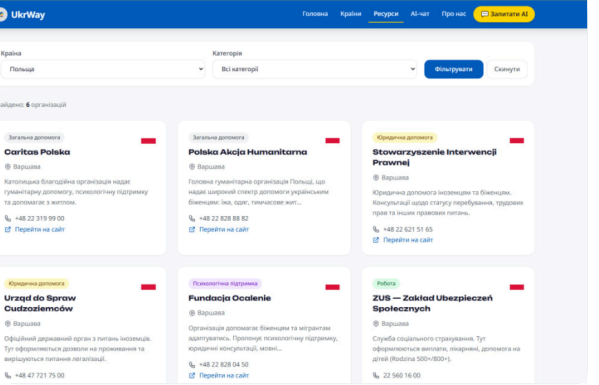


Рис. 4 — Карта організацій допомоги

РОЗДІЛ 3 · РЕАЛІЗАЦІЯ

Реалізація RAG-пайплайну

```
32000 повідомлень [germany] збережено (32500 всього)
33000 повідомлень [germany] збережено (33500 всього)
34000 повідомлень [germany] збережено (34500 всього)
35000 повідомлень [germany] збережено (35500 всього)
36000 повідомлень [germany] збережено (36500 всього)
37000 повідомлень [germany] збережено (37500 всього)
38000 повідомлень [germany] збережено (38500 всього)
39000 повідомлень [germany] збережено (39500 всього)
40000 повідомлень [germany] збережено (40500 всього)
41000 повідомлень [germany] збережено (41500 всього)
42000 повідомлень [germany] збережено (42500 всього)
43000 повідомлень [germany] збережено (43500 всього)
44000 повідомлень [germany] збережено (44500 всього)
45000 повідомлень [germany] збережено (45500 всього)
■ Досягнуто дату 2025-05-01, зупиняюсь
✓ germany: 45916 нових повідомлень

Збережено 46416 повідомлень у C:\Users\Admin\PycharmProjects\Informationweb
```

Етап 1 — Збір даних з Telegram (Telethon)

```
25600/26210 чанків проіндексовано
25700/26210 чанків проіндексовано
25800/26210 чанків проіндексовано
25900/26210 чанків проіндексовано
26000/26210 чанків проіндексовано
26100/26210 чанків проіндексовано
26200/26210 чанків проіндексовано
26210/26210 чанків проіндексовано

База знань побудована: 26210 чанків у C:\U
```

Етап 2 — Індексуння у ChromaDB

ЗІБРАНО ПОВІДОМЛЕНЬ
45 916

ЧАНКІВ У CHROMADB
26 210

TOP_K · COSINE
7 · <0.7

РЕ-РАНЖУВАННЯ
0.7 / 0.3

РОЗДІЛ 3 · ТЕСТУВАННЯ

Результати тестування

11/11 тест-кейсів — усі пройдено

Перевірено якість на 1 пробному запиті, про тимчасовий захист питання про тимчасовий захист у Німеччині

\$0.40 витрачено на тестування
тестовий баланс — \$5.00

claude-sonnet-4-6 модель LLM, що використовується
вхідні / вихідні токени фіксуються

Logs

TIME (GMT+3)	ID	MODEL	INPUT TOKENS	OUTPUT TOKENS	TYPE	SERVICE TIER	REQUEST
2026-05-16 15:59:50	req_011C0bH8A3xLXAXG1g0Nw	claude-sonnet-4-6	28611	646	HTTP	Standard	
2026-05-16 15:59:35	req_011C0bH8A3xLXAXG1g0Nw	claude-sonnet-4-6	7045	150	HTTP	Standard	
2026-05-16 15:55:43	req_011C0bH8A3xLXAXG1g0Nw	claude-sonnet-4-6	16140	581	HTTP	Standard	
2026-05-16 15:55:29	req_011C0bH8A3xLXAXG1g0Nw	claude-sonnet-4-6	6533	111	HTTP	Standard	
2026-05-16 15:45:30	req_011C0bH8A3xLXAXG1g0Nw	claude-sonnet-4-6	21102	606	HTTP	Standard	
2026-05-16 15:45:16	req_011C0bH8A3xLXAXG1g0Nw	claude-sonnet-4-6	6232	93	HTTP	Standard	
2026-05-16 14:43:44	req_011C0bH8A3xLXAXG1g0Nw	claude-sonnet-4-6	3181	470	HTTP	Standard	
2026-05-16 14:39:28	req_011C0bH8A3xLXAXG1g0Nw	claude-sonnet-4-6	2212	903	HTTP	Standard	
2026-05-16 14:31:17	req_011C0bH8A3xLXAXG1g0Nw	claude-sonnet-4-6	1397	871	HTTP	Standard	
2026-05-16 14:26:56	req_011C0bH8A3xLXAXG1g0Nw	claude-sonnet-4-6	309	637	HTTP	Standard	

Рис. 5 — Журнал API-викликів Claude Console

ВИСНОВОК AI-асистент надає структуровані україномовні відповіді з посиланням на реальний досвід мігрантів зі спільноти Telegram.

РОЗДІЛ 3 · ТЕСТУВАННЯ

Результати тестування на прикладі реального запиту

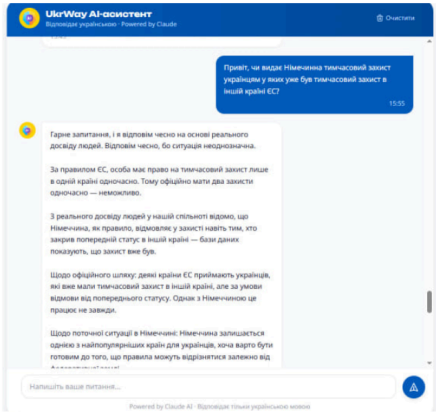


Рис. 6 — Питання, відповідь

ВИСНОВОК Отримали відповідь, яка базується на базі реального досвіду людей, інформацію яку зловив Claude вірна та надана коректна відповідь

АПРОБАЦІЯ

Апробація результатів та публікації.

Основні положення бакалаврської роботи апробовано шляхом участі у науково-практичних конференціях, що проводились на базі Державного університету інформаційно-комунікаційних технологій:

- 1 III Всеукраїнська науково-технічна конференція «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу» (Київ, ДУІКТ, 18 листопада 2025 р.).
- 2 VII Міжнародна науково-технічна конференція «Сучасний стан та перспективи розвитку IoT» (Київ, ДУІКТ, 20 квітня 2026 р.).

ПІДСУМОК

Висновки — по одному пункту на кожне завдання

- 1 Досліджено: 5,9 млн біженців, 3 цільові країни, 4 категорії потреб — жодна існуюча платформа не закриває їх одночасно
- 2 Проаналізовано: 32% мігрантів використовують Telegram, але дані неструктуровані — обґрунтовано RAG-підхід
- 3 Обґрунтовано стек: Django + HTMX + ChromaDB + Claude API — кожен компонент прив'язаний до конкретного параметра
- 4 Спроектровано: 5 модулів, 6 моделей БД, RAG-пайплайн з ранжуванням і graceful fallback
- 5 Розроблено: веб-платформа UkrWay — 26 210 чанків, 17 організацій, AI-чат з web_search
- 6 Протестовано: 11 / 11 тест-кейсів пройдено, витрати на тестування — \$0.40 з \$5.00 балансу

