

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерних наук _____
(повна назва)

Кафедра _____ програмної інженерії _____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти _____ перший (бакалаврський) _____

Програмна система для підтримки соціальної адаптації при переїзді в іншу країну

_____ (тема)

Виконав:
здобувач _____ 4 _____ року навчання
групи _____ ПЗП-21-2 _____

_____ Владислав БАРСУКОВ _____
(Власне ім'я, ПРІЗВИЩЕ)

Спеціальність _____ 121 – Інженерія програмного _____
забезпечення _____
(код і повна назва спеціальності)

Тип програми _____ освітньо-професійна _____

Освітня програма _____ програмна інженерія _____
(повна назва освітньої програми)

Керівник _____ проф. кафедри ІІІ Володимир БОНДАРЄВ _____
(посада, Власне ім'я, ПРІЗВИЩЕ)

Допускається до захисту
Зав. кафедри _____

_____ (підпис)

_____ Кирило СМЕЛЯКОВ _____
(Власне ім'я, ПРІЗВИЩЕ)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерних наук
 Кафедра _____ програмної інженерії
 Рівень вищої освіти _____ перший (бакалаврський)
 Спеціальність _____ 121 – Інженерія програмного забезпечення
 Тип програми _____ освітньо-професійна
 Освітня програма _____ програмна інженерія
 (шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
 (підпис)
 « ____ » _____ 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Здобувачеві _____ Барсукову Владиславу Олександровичу
 (прізвище, ім'я, по батькові)

1. Тема роботи _____ Програмна система для підтримки соціальної адаптації при переїзді в іншу країну

Затверджена наказом по університету № 397 Ст від 19.05.2025

2. Термін подання здобувачем роботи до екзаменаційної комісії 10.06.2025

3. Вихідні дані до роботи Розробити застосунок для підтримки соціальної адаптації при переїзді в іншу країну, котрий міститиме в собі 7 основних модулів, доступних через головне меню, з використанням фреймворків Next.js для клієнтської частини та NestJS для серверної логіки.

4. Перелік питань, що потрібно опрацювати в роботі

Вступ, аналіз предметної галузі, формування вимог до програмної системи, архітектура та проектування програмного забезпечення, опис прийнятих програмних рішень, тестування розробленого програмного забезпечення, висновки, додатки.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі	21.04.2025	<i>виконано</i>
2	Створення специфікації ПЗ	25.04.2025	<i>виконано</i>
3	Проектування ПЗ	30.04.2025	<i>виконано</i>
4	Розробка ПЗ	15.05.2025	<i>виконано</i>
5	Тестування ПЗ	20.05.2025	<i>виконано</i>
6	Оформлення пояснювальної записки	25.05.2025	<i>виконано</i>
7	Підготовка презентації та доповіді	01.06.2025	<i>виконано</i>
8	Попередній захист	03.06.2025	<i>виконано</i>
9	Нормоконтроль, рецензування	04.06.2025	<i>виконано</i>
10	Здача роботи у електронний архів	05.06.2025	<i>виконано</i>
11	Допуск до захисту у зав. кафедри	08.06.2025	<i>виконано</i>

Дата видачі завдання « 20 » « квітня » 2025р.

Здобувач 

Керівник роботи _____ проф. кафедри ІІ Володимир БОНДАРЄВ
(підпис) (посада, Власне ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи бакалавра, 81 стор., 29 рис., 1 табл., 16 джерел.

СОЦІАЛЬНА МЕРЕЖА, ІНТЕГРАЦІЇ В СЕРЕДОВИЩІ, ВЕБ-ПЛАТФОРМА, MYSQL, NESTJS, NEXT.JS, PRISMA ORM, TYPESCRIPT

Об'єктом розробки є інноваційна веб-платформа, створена для сприяння соціальній адаптації людей, які переїхали до нової країни або місцевості.

Мета роботи – розробити ефективний та інтуїтивно зрозумілий веб-додаток, що надасть користувачам змогу швидко інтегруватися у нове середовище завдяки пошуку однодумців за інтересами, мовами, хобі, цілями, а також участі у тематичних заходах та спілкуванні в групах.

У процесі роботи використовувалися сучасні технології розробки: серверний фреймворк NestJS, клієнтська частина була реалізована з використанням Next.js, для взаємодії з базою даних MySQL застосовано ORM Prisma, а мова програмування TypeScript забезпечила високу стабільність та підтримуваність коду.

В результаті створено унікальний веб-сервіс, що підтримує функції реєстрації, авторизації, генерації персоналізованих рекомендацій друзів та подій на основі схожих інтересів, геолокації користувачів, а також надає можливість створювати власний контент у вигляді статей, заходів та тематичних груп. Додатковою перевагою платформи є продумана система захисту даних користувачів і зручний інтерфейс, який дозволяє легко освоїтись навіть людям без технічного досвіду.

ABSTRACT

SOCIAL NETWORK FOR INTEGRATION IN A NEW ENVIRONMENT, WEB PLATFORM, MYSQL NESTJS, NEXT.JS, PRISMA ORM, TYPESCRIPT

The object of development is an innovative web platform designed to facilitate social adaptation for individuals moving to a new country or area.

The purpose of the thesis is to develop an effective and user-friendly web application enabling rapid social integration by connecting users based on common interests, languages, hobbies, goals, as well as participation in themed events and group interactions.

The development involved modern web technologies: the backend was implemented using the NestJS framework, the frontend utilized Next.js, database interactions with MySQL were managed through Prisma ORM, and the TypeScript language was used to enhance code stability and maintainability.

As a result, a unique web service was created, offering user registration, authentication, personalized friend and event recommendations based on similar interests and geographic proximity, as well as the ability to create user-generated content such as articles, events, and groups. Additional advantages include robust user data protection mechanisms and an intuitive interface, accessible even to users without technical expertise.

ЗМІСТ

Вступ.....	8
1 Аналіз предметної галузі.....	10
1.1 Аналіз предметної галузі.....	10
1.1.1 Попит на цифрові інструменти для адаптації.....	10
1.1.2 Аналіз існуючих платформ.....	11
1.2 Виявлення та вирішення проблем.....	17
1.3 Постановка задачі.....	19
1.3.1 Цільова аудиторія.....	20
1.3.2 Монетизація.....	20
1.3.3 Очікуваний вплив і користь.....	20
2 Формування вимог до програмної системи.....	21
2.1 Архітектура та технологічні рішення.....	21
2.2 Структура бази даних.....	21
2.3 Основна функціональність.....	22
2.4 Нефункціональні вимоги.....	23
2.5 Перспективи розвитку.....	23
3 Архітектура та проєктування програмного забезпечення.....	26
3.1 Проєктування бази даних.....	26
3.1.1 ER – діаграма класів.....	26
3.1.2 Формування повної логічної моделі бази даних.....	28
3.1.3 Нормалізація логічної моделі бази даних.....	30
3.2 Архітектура серверної частини.....	31
3.2.1 UML-діаграма розгортання.....	32
3.2.2 UML-діаграма пакетів.....	33
3.3 Клієнтська частина.....	35
3.3.1 UML-діаграма клієнтської частини.....	36
3.3.2 UML-діаграму пакетів.....	38
3.3.3 Модулі клієнтської частини.....	39
3.4 Проєктування дизайну програмної системи.....	41

	7
4 Опис прийнятих програмних рішень	44
4.1 Структура та процес збірки.....	44
4.2 Стек технологій і середовище виконання.....	44
4.3 Головна навігація й компоненти меню	45
4.4 Сторінка профілю користувача.....	47
4.5 Сторінка друзів та механізм пошуку.....	48
4.6 Групові чати та обмін повідомленнями	49
4.7 Система управління групами	50
4.8 Сторінка користувачів і рекомендації.....	51
4.9 Організація та участь у подіях	53
4.10 Статті та інформаційний розділ.....	54
4.11 Управління станом і запити до API.....	56
4.12 Аутентифікація та захист сесії.....	57
4.13 Логування та користувацькі повідомлення	57
5 Тестування програмного забезпечення.....	60
5.1 Проведення тестування.....	60
5.2 Виявлені недоліки в процесі тестування	61
6 Впровадження програмного забезпечення	63
Висновки	66
Перелік джерел посилання	68
Додаток А	70
Додаток Б.....	72

ВСТУП

В умовах стрімкої глобалізації та зростаючої мобільності населення все більше людей стикається з необхідністю адаптуватися до життя у новій країні. Перехід до незнайомого соціокультурного середовища супроводжується низкою труднощів, серед яких особливо актуальною є проблема соціальної інтеграції: встановлення нових контактів, подолання мовного бар'єру та пошук людей зі спільними інтересами. Відсутність ефективних інструментів для швидкої соціалізації може призводити до відчуття ізоляції, зниження психологічного комфорту і навіть негативно впливати на загальну успішність інтеграції в нове середовище. Таким чином, розробка програмних засобів, які полегшують процес соціальної адаптації, є надзвичайно актуальною задачею.

Темою кваліфікаційної роботи є «Програмна система для підтримки соціальної адаптації при переїзді в іншу країну». Основним завданням створеного програмного рішення є забезпечення ефективної інтеграції користувачів у нове соціальне середовище завдяки швидкому пошуку людей зі схожими інтересами, хобі, мовними навичками, а також створення умов для комфортного спілкування в рамках тематичних заходів і спільнот.

Метою роботи є створення програмного комплексу у вигляді сучасного веб-додатку, який забезпечуватиме комплексну підтримку соціалізації користувачів після переїзду в нову країну. Веб-застосунок має реалізовувати функції реєстрації та авторизації користувачів, автоматичне формування індивідуальних рекомендацій друзів на основі заданих параметрів (інтереси, хобі, мова, місце проживання), можливість створення, перегляду та участі в тематичних заходах і групах, а також можливість публікувати й переглядати тематичний контент.

В якості технологічного стека обрано найбільш сучасні та ефективні технології, що відповідають потребам розробки комплексних веб-додатків. Серверна частина реалізована з використанням фреймворку NestJS та мови програмування TypeScript, що дозволяє ефективно та структуровано організувати логіку взаємодії з даними. Для роботи з базою даних обрано ORM Prisma та систему керування базами даних MySQL. Для клієнтської частини використовується

фреймворк Next.js, який забезпечує швидке завантаження сторінок та інтерактивну взаємодію з користувачами. Локальне розгортання, тестування і налагодження серверної частини та бази даних здійснюється з допомогою програмного комплексу MAMP. У якості єдиного середовища розробки обрано Visual Studio Code, що дозволяє комфортно працювати як з клієнтською, так і з серверною частинами додатку.

Галуззю застосування результатів даної роботи є широке коло сфер, пов'язаних з міграцією, інтеграцією іноземних студентів та фахівців, соціальною підтримкою експатів, діяльністю міжнародних організацій та соціальних служб, які займаються допомогою людям, що переїхали до іншої країни. Створений веб-застосунок може бути застосовано не тільки як самостійний сервіс, а й інтегровано в інші соціальні чи освітні платформи, значно полегшуючи адаптацію новоприбулих.

Таким чином, створення цієї програмної системи має значний практичний ефект, сприяє вирішенню важливої соціальної проблеми та покращує якість життя людей, які стикаються з необхідністю адаптації в нових умовах.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Аналіз предметної галузі

В умовах сучасного глобалізованого світу та зростання міжнародної мобільності населення, соціальна адаптація людей у новому середовищі набуває особливого значення. За даними ООН, станом на 2024 рік кількість міжнародних мігрантів перевищила 280 мільйонів осіб [1], і ця цифра продовжує зростати щороку. Основними чинниками, що стимулюють міграційні процеси, є прагнення до кращих умов життя, працевлаштування, отримання якісної освіти, а також, дедалі частіше, – вимушена міграція внаслідок політичних конфліктів та воєнних дій. Особливо гостро проблема міграції проявилася у зв'язку з повномасштабною війною в Україні, що призвела до масштабних міграційних потоків українців до країн Європи та інших регіонів світу. За даними Управління Верховного комісара ООН у справах біженців, станом на початок 2025 року майже 7 мільйонів українців були змушені покинути країну та шукати прихисток за кордоном [2].

У цьому контексті важливого значення набуває соціальна адаптація, яка є складним, багатоаспектним процесом пристосування людини до нових соціокультурних умов. Під соціальною адаптацією розуміють процес поступової інтеграції особистості в нове суспільство через засвоєння його норм, цінностей, традицій, налагодження нових соціальних контактів та побудови стабільних міжособистісних відносин. Успішність адаптації значною мірою визначає якість життя мігрантів, їхній психологічний стан, здатність до ефективної праці та самореалізації у новому середовищі.

1.1.1 Попит на цифрові інструменти для адаптації

Зважаючи на значні масштаби міжнародної міграції та гостру потребу в підтримці соціальної інтеграції новоприбулих мігрантів зростає попит на спеціалізовані інформаційні та комунікаційні ресурси, які б сприяли швидшому і комфортнішому входженню новоприбулих у суспільне життя. Упродовж останнього десятиліття у світі з'явилась низка платформ і сервісів, покликаних допомагати мігрантам знайти однодумців, ознайомитися з місцевою культурою й

отримати корисні поради щодо адаптації. Водночас детальніший аналіз таких продуктів свідчить, що вони не повною мірою задовольняють актуальні потреби мігрантів, особливо коли йдеться про користувачів з України, котрі часто потребують додаткової морально-психологічної підтримки в умовах воєнного конфлікту.

1.1.2 Аналіз існуючих платформ

Одним із найвідоміших рішень є Meetup що позиціонує себе як платформа для організації заходів і зустрічей за інтересами (див. рис. 1.1). Її базова ідея полягає в тому, щоб люди з однієї локації, які мають спільні захоплення (спорт, мистецтво, технології, вивчення мов), могли легко об'єднуватись у групи та проводити зустрічі в офлайн або онлайн. До переваг Meetup належить велика спільнота користувачів у багатьох країнах світу, а також зручний пошук заходів. Водночас платформа не пропонує розвинених можливостей для встановлення більш особистих контактів з урахуванням, наприклад, мовних бар'єрів чи потреб у психологічній підтримці. Крім того, для частини користувачів з України бар'єром може стати необхідність платного доступу до окремих функцій для організаторів.

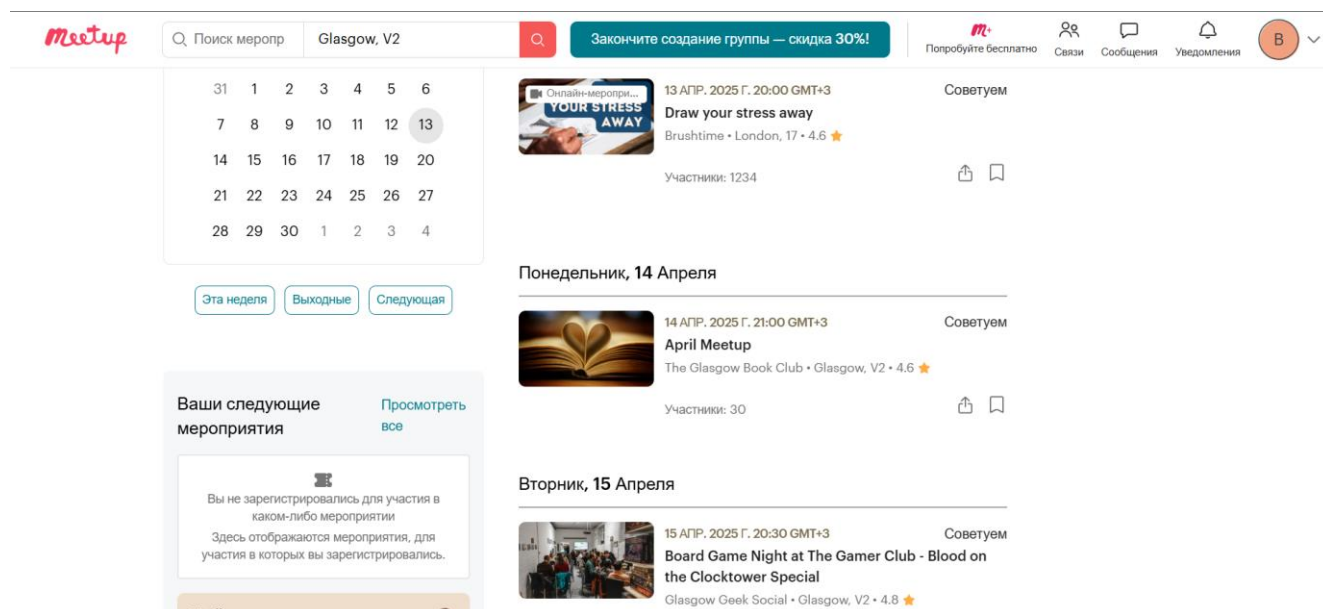


Рисунок 1.1 – Інтерфейс платформи Meetup (за даними [3])

Іншим прикладом є Internations – сайт, орієнтований на так званих «експатів», тобто людей, які переїжджають до іншої країни переважно з професійною або освітньою метою (див. рис. 1.2). Платформа пропонує широкий спектр тематичних груп і заходів, що полегшують пошук нових знайомств та обмін досвідом щодо побуту, дозвілля чи працевлаштування. Internations робить акцент на більш «преміальному» рівні обслуговування, тому певні інструменти доступні лише за платною підпискою. Це може знизити привабливість сервісу для частини аудиторії, особливо коли йдеться про біженців або людей, які не мають стабільного доходу в новій країні. Крім того, як і Meetup, Internations не завжди враховує специфіку українських мігрантів, зокрема потребу у психологічній адаптації чи пошук співвітчизників, які могли б забезпечити моральну підтримку.

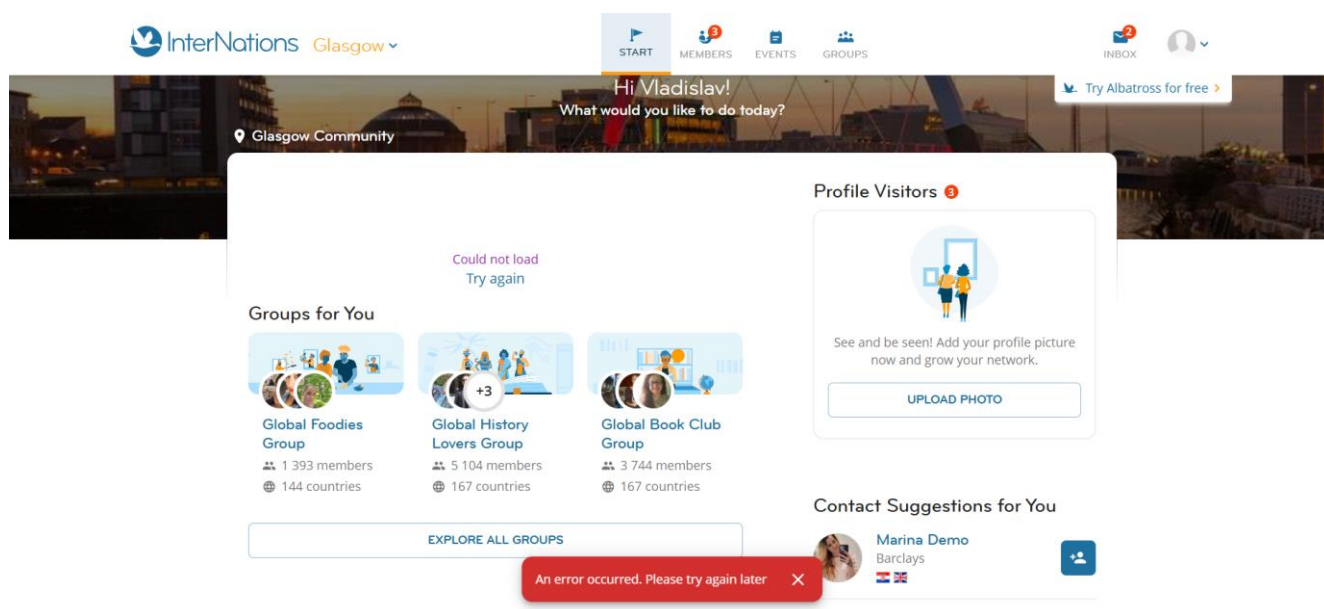


Рисунок 1.2 – Головна сторінка платформи InterNations (за даними [4])

Ще одним популярним джерелом інформації та інструментом для адаптації є Expatica, яка позиціює себе як платформу для всебічної інформаційної підтримки експатів (див. рис. 1.3). Вона пропонує статті, поради та новини стосовно життя в різних країнах, включно з інформацією про отримання віз, оренду житла, навчання тощо. Сильним боком Expatica є тематична різноманітність матеріалів, а також орієнтація на глобальний ринок. Водночас цей сервіс меншою мірою сприяє налагодженню безпосередніх контактів із місцевими жителями чи пошуку друзів

серед інших новоприбулих. Тобто він виконує більше роль енциклопедії або гіда, тоді як аспект соціальної інтеграції залишається неповно охопленим.

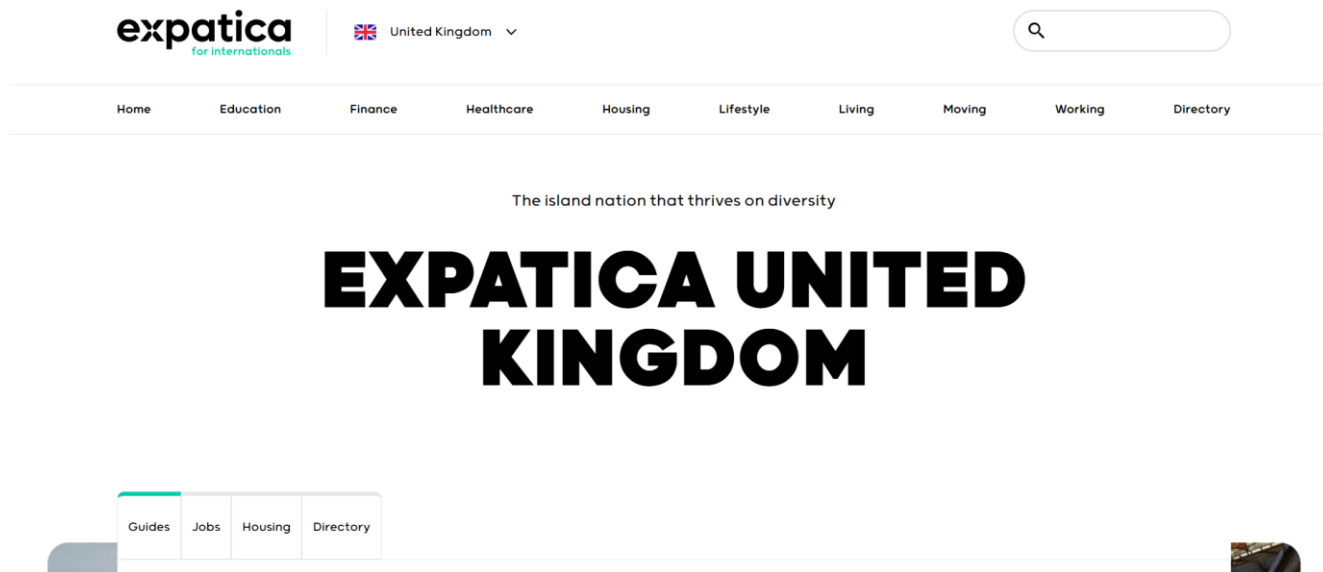


Рисунок 1.3 – Інтерфейс довідкового порталу Expatica (за даними [5])

На відміну від попередніх платформ, орієнтованих здебільшого на групові заходи й інформаційну підтримку, InterPals фокусується на листуванні та мовній практиці (див. рис. 1.4). Спочатку цей сайт створювався як майданчик для пошуку «pen friends» із різних країн, а згодом його функціонал розширився – нині користувачі можуть шукати співрозмовників для мовного обміну або навіть дружби. Така спрямованість добре підходить для вдосконалення мовних навичок, однак загальна соціальна адаптація залишається за межами основних можливостей платформи. InterPals не пропонує вбудованих механізмів для створення локальних груп чи обговорення питань, пов’язаних із побутом та культурою конкретної країни.

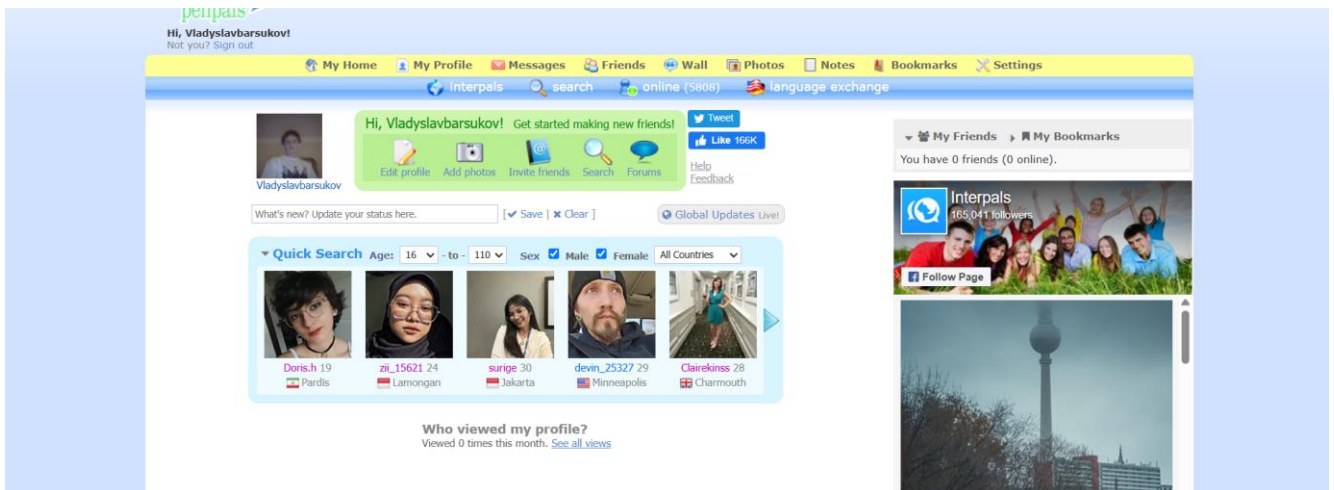


Рисунок 1.4 – Інтерфейс платформи InterPals (за даними [6])

Схожою до InterPals за основною ідеєю є Couchsurfing, що, однак, спеціалізується на пошуку місцевих мешканців, готових прийняти подорожніх у себе вдома (див. рис. 1.5). Цей проєкт має на меті налагодження міжкультурних зв'язків через безпосередню взаємодію господарів із гостями. Попри таку позитивну ініціативу, Couchsurfing не завжди забезпечує сталість зв'язків або глибоке занурення в місцеве середовище – зазвичай користувачі використовують його короткочасно під час подорожей чи переїзду. Крім того, сервіс не має розвинених засобів для пошуку саме тих, хто перебуває в схожих життєвих обставинах (наприклад, біженців), а мовна підтримка часто обмежується мінімальним набором інструментів.

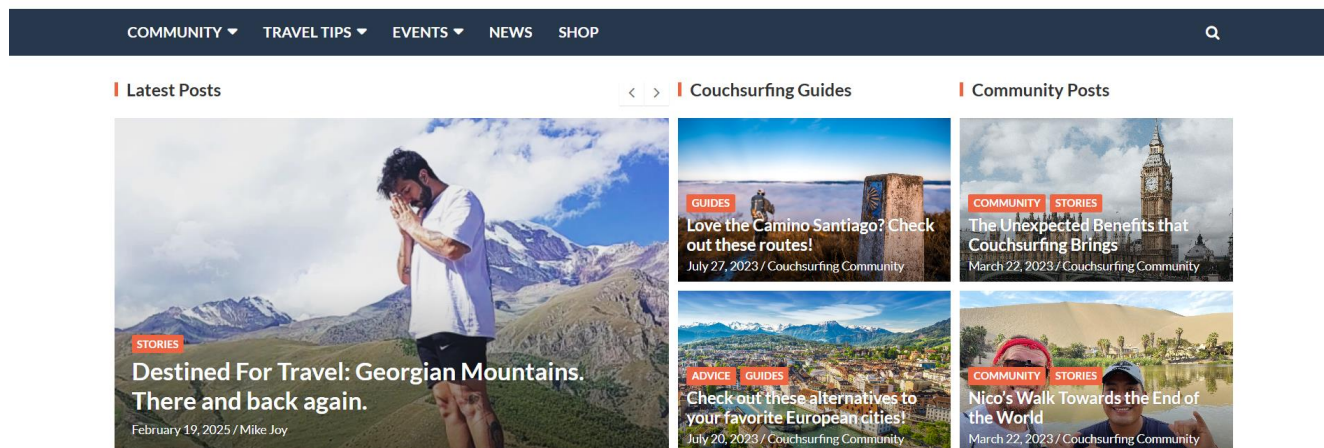


Рисунок 1.5 – Інтерфейс платформи Couchsurfing (за даними [7])

Окрім згаданих платформ, чимало людей звертаються до соціальних мереж та різноманітних ініціатив, що зароджуються у них. Зокрема, Facebook слугує популярним майданчиком для створення тематичних груп мігрантів чи біженців, де учасники діляться корисними порадами, контактами, історіями адаптації. Telegram-канали або Discord-спільноти також дедалі частіше об'єднують українців за кордоном, пропонуючи оперативне інформування про локальні події, вакансії чи можливості соціальної допомоги (див. рис. 1.6 - рис. 1.7). Сильним боком таких мережеских об'єднань є їхня гнучкість і швидкість поширення нової інформації. Водночас вони нерідко страждають на відсутність чіткої структури: інформація може губитися в нескінченних повідомленнях, а пошук потенційних друзів чи знайомих із близькими інтересами ускладнений.

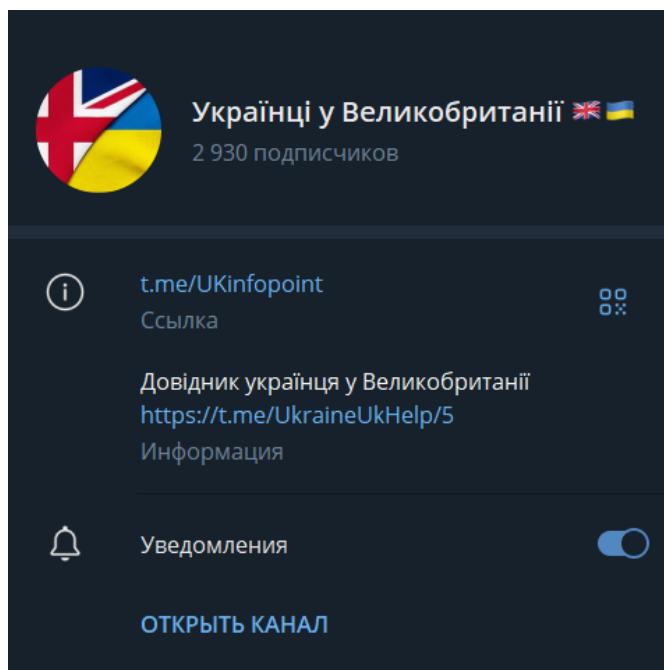


Рисунок 1.6 – Інформаційний телеграм-канал об’єднання українців (за даними [8])

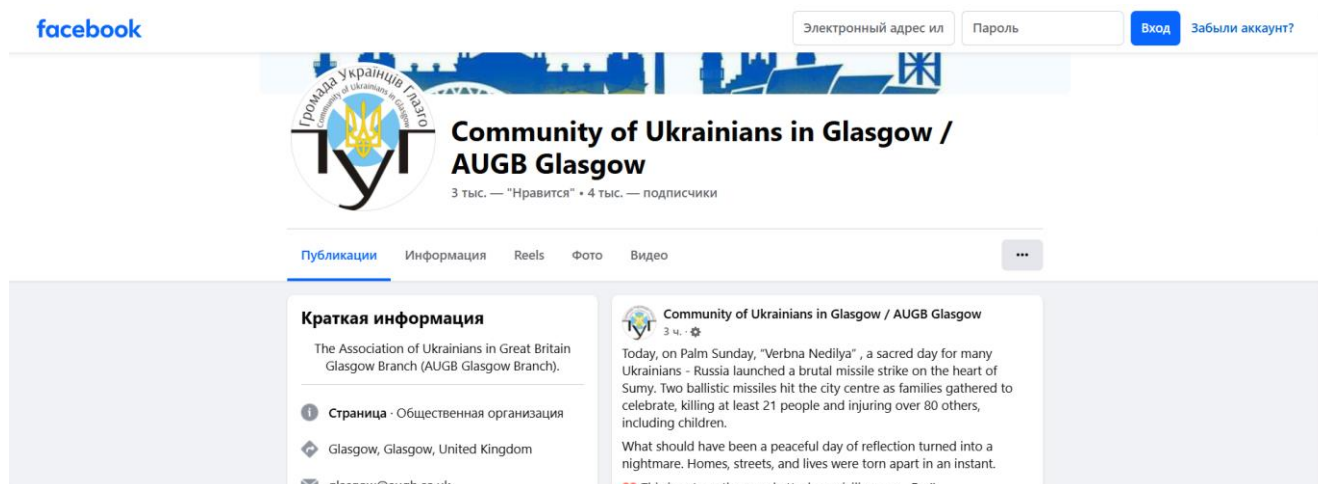


Рисунок 1.7 – Facebook-сторінка онлайн-спільноти (за даними [9])

Отже, більшість існуючих рішень, покликаних полегшити адаптацію мігрантів (зокрема тих, хто залишив Україну через війну), мають низку суттєвих обмежень. Серед ключових недоліків можна назвати відсутність розвинених механізмів персоналізації, недостатню локалізацію, а також незначну увагу до формування довготривалих соціальних зв’язків. Крім того, інформація та корисні відомості часто виявляються розділені між різними сервісами, соціальними

мережами й каналами, що ускладнює отримання комплексної підтримки в єдиному місці.

У світлі цих спостережень постає проблема розробки спеціалізованого програмного інструменту, здатного врахувати сучасний контекст зростаючих міграційних потоків, необхідність поглибленої соціальної взаємодії та високий запит на локалізований контент, орієнтований передусім на мігрантів. Такий підхід міг би відповідати сучасним стандартам зручності й безпеки, забезпечуючи інструменти для пошуку однодумців, об'єднання у групи та обміну важливою інформацією без перевантаження користувача другорядними даними. Зрештою, це створило б додаткові можливості для глибшої інтеграції мігрантів у нове середовище й водночас стимулювало б розвиток інноваційних форм онлайн-взаємодії, що задовольняють потреби глобалізованого суспільства.

1.2 Виявлення та вирішення проблем

У ході аналізу було встановлено, що більшість існуючих рішень для соціальної адаптації орієнтується або на короткочасні зустрічі, або на обмежені за функціоналом мовні чи культурні обміни. Такий підхід лише частково допомагає людям, які опинилися в новому середовищі, оскільки вони змушені відвідувати різноманітні платформи й канали (соціальні мережі, інформаційні портали, форуми) задля збирання потрібних відомостей про можливості навчання, працевлаштування та побутову адаптацію. Унаслідок цього виникає відчутне розділення інформації: корисний контент розпорошений серед різних ресурсів, а єдиного механізму швидкого пошуку немає. Це особливо ускладнює швидке входження в нове середовище, адже користувачі не отримують централізованої підтримки чи рекомендацій.

Наступною проблемою є брак цілісної системи рекомендацій, де було б зручно отримувати підказки щодо місцевих заходів, потенційних контактів чи груп, відповідних рівню володіння мовою, професійним інтересам або типовим потребам тих, хто нещодавно переїхав. Людям доводиться витратити чимало часу на самостійне сортування заходів чи контактів. Особливо гостро це відчувають ті, хто

перебуває в новому середовищі вперше та не знає локальних особливостей, не має знайомих чи друзів, здатних надати «прискорену інтеграцію». Таким чином, відсутність продуманої рекомендаційної логіки веде до того, що користувачі залишаються один на один із величезним масивом не чіткої та не зрозумілої інформації.

Окремо слід відзначити, що, незважаючи на популярність підходу з багатомовністю, чимало сервісів усе ж таки працюють у одномовному режимі. З одного боку, якщо інтерфейс обрано лише англійською мовою, це дає змогу об'єднувати широку аудиторію, адже англійська вважається міжнародною мовою спілкування. З іншого боку, користувачам потрібен певний рівень знань цієї мови для повноцінної взаємодії, а також для споживання контенту, що може відштовхнути тих, хто не почувається впевнено в англійськом середовищі. Отже, викликом залишається поєднати компактність і зручність англійського інтерфейсу з достатньою глибиною налаштувань, аби користувачі різного рівня адаптації могли формувати стабільне соціальне оточення.

Ще одним аспектом, який часто недооцінюється, є психологічна складова адаптації. Частина користувачів після переїзду може зазнавати емоційного напруження, пов'язаного з мовним бар'єром, культурними розбіжностями чи браком підтримки з боку знайомих. Ті сервіси, які орієнтовані лише на функції пошуку друзів або проведення зустрічей, зазвичай не пропонують додаткових розділів із психологічною чи інформаційною підтримкою (наприклад, статей про подолання стресу чи контактів центрів емоційної допомоги). Як наслідок, людина може відчувати, що, окрім побіжного соціального спілкування, не отримує більш поглиблених інструментів для полегшення свого становища.

З урахуванням усіх вищезгаданих проблем постає потреба у створенні комплексного веб-застосунку, який поєднує в собі функції пошуку однодумців, рекомендаційну систему для підбору релевантних подій чи контактів, а також довідкову базу про життя в новій країні. Така концепція передбачає, що користувачі зможуть не лише знаходити нові знайомства за схожими інтересами, але й оперативно дізнаватися про локальні курси мови, найпоширеніші помилки, з

якими стикаються мігранти, чи навіть отримувати підказки щодо психологічної підтримки. Усе це повинно бути зосереджено в одному інтерфейсі, аби зменшити залежність від розрізнених платформ і розпорошеного контенту.

Для вирішення виокремлених проблем доцільно реалізувати систему персоналізації, коли користувач зазначає свій рівень володіння мовами, професійні й культурні інтереси, отримує адаптовані підказки з відбором груп, заходів та можливих контактів, використовувати цілісний інтерфейс, орієнтований на міжнародну аудиторію. При проектуванні варто дотримуватися мінімалістичного й логічного підходу: лаконічна структура, чітка термінологія, інтуїтивна навігація. У подальшому, за потреби розширення платформи, можливе додавання різних локалізацій та забезпечити інформаційну та психологічну підтримку: зокрема, розділи з порадами щодо культури, менталітету та основних правових питань. Це, вочевидь, виходить за межі звичайного пошуку друзів, проте саме такий підхід відрізнятиме комплексне рішення від уже наявних вузьких платформ.

Таким чином, можна швидко знайти однодумців для комунікації та ознайомитися з довідково-психологічним розділом, та уникнути проблем, які зазвичай постають перед новоприбулими. Людині не доведеться одночасно шукати дещо в чатах, інше – на офіційних сайтах, а ще інше – на форумах чи в спеціалізованих групах соціальних мереж. Усе найнеобхідніше зможе міститися в єдиному місці, де користувач зможе користуватися зручною навігацією та системою підказок на основі заданих інтересів і потреб.

Цей підхід забезпечить як швидшу соціалізацію, так і формування більш довготривалих зв'язків, оскільки користувачі зможуть обирати групи за інтересами або рівнем мови й залишатися в них скільки завгодно. Також, можливе розширення механізму «подій», де кожен користувач або група може анонсувати офлайнві зустрічі, вебінари, спільні екскурсії.

1.3 Постановка задачі

Веб-додаток має реалізовувати функції реєстрації, авторизації, створення профілю користувача з можливістю вказувати інтереси, рівень володіння мовами,

країну та місто проживання. На основі цієї інформації передбачається формування системи персоналізованих рекомендацій, яка допоможе користувачу швидше знайти нових знайомих, долучитися до релевантних груп або заходів і тим самим полегшити адаптацію в новій країні. Також передбачається можливість створення спільнот за інтересами та участі у тематичних групах..

1.3.1 Цільова аудиторія

Цільовою аудиторією веб-застосунку є люди, які переїхали або планують переїзд в іншу країну з метою роботи, навчання, постійного проживання або туризму на тривалий період. Основні вікові рамки аудиторії становлять 18 – 45 років, включаючи студентів, молодих спеціалістів, IT-фахівців, фрилансерів та інших активних користувачів інтернету, які шукають швидкий і комфортний спосіб адаптуватися до нового середовища.

1.3.2 Монетизація

Монетизацію застосунку можна реалізувати через модель freemium – основний функціонал доступний безкоштовно, тоді як додаткові можливості (розширений пошук, додаткові рекомендації, ексклюзивний контент або розширені функції профілю) пропонуються за підпискою. Також можлива додаткова монетизація через рекламні інтеграції з партнерами (мовні школи, місцеві заходи, туристичні компанії тощо).

1.3.3 Очікуваний вплив і користь

Реалізація запропонованого рішення має принести користь широкому колу людей, які перебувають у процесі адаптації після переїзду. Очікується, що платформа сприятиме формуванню нових соціальних зв'язків, підвищенню обізнаності про локальні події, ініціативи та ресурси, що доступні в новому середовищі. У довгостроковій перспективі це може знизити рівень соціальної ізоляції, покращити емоційний стан користувачів та посилити відчуття включеності до нової спільноти.

2 ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ

2.1 Архітектура та технологічні рішення

Розробка програмна система для підтримки соціальної адаптації при переїзді в іншу країну потребує чіткого визначення її архітектури, структури бази даних, функціональності та принципів взаємодії між клієнтською та серверною частинами. Даний розділ містить концепт-документ, що описує передбачувані технічні, функціональні та нефункціональні вимоги до веб-застосунку, який планується реалізувати з використанням сучасного технологічного стеку.

Передбачається, що програмна система буде побудована як двокomпонентна архітектура, що включатиме клієнтську частину на базі фреймворку Next.js та серверну частину на основі NestJS. Очікується, що вся логіка взаємодії користувача із системою буде реалізована через зручний веб-інтерфейс, який забезпечить швидкий доступ до основного функціоналу. Взаємодія між frontend та backend здійснюватиметься через REST API, який дозволить обмінюватися даними у форматі JSON.

З боку серверної частини NestJS буде обрано як основу для побудови масштабованого та модульного застосунку [10]. Очікується, що логіка обробки запитів буде організована через контролери, кожен з яких відповідатиме за певну ділянку функціоналу: обробку профілів користувачів, подій, груп, системи рекомендацій та статусів. Бізнес-логіку буде винесено у відповідні сервіси з метою забезпечення логічного розділення відповідальностей і спрощення подальшої підтримки.

2.2 Структура бази даних

Для зберігання даних планується використати реляційну базу MySQL у зв'язці з ORM Prisma [11]. Це забезпечить роботу із сутностями в об'єктно-орієнтованому стилі. Очікується, що структура бази даних включатиме таблиці користувачів, подій, груп, статусів, повідомлень, інтересів, мов та пов'язаних між собою сутностей. Кожен користувач матиме можливість зберігати у своєму профілі особисту інформацію, інтереси, країну, місто та мови для комунікації.

2.3 Основна функціональність

Клієнтська частина на базі Next.js дозволить реалізувати рендеринг сторінок на сервері, що сприятиме пришвидшенню завантаження контенту [12]. Застосунок буде побудований на компонентах, що забезпечуватимуть відображення стрічки подій, груп, списку користувачів, налаштувань профілю та системи рекомендацій. Передбачається, що взаємодія з сервером відбуватиметься асинхронно для забезпечення плавності користувацького досвіду.

Особливу увагу планується приділити реалізації механізму персоналізованих рекомендацій. На основі інформації з профілю (інтереси, мови, локація, статус) система буде формувати пропозиції щодо потенційних контактів, подій або груп. Такий алгоритм дозволить користувачам ефективніше адаптуватися до нового соціального середовища, швидше налагоджуючи нові зв'язки.

Також у межах системи передбачається реалізація функціоналу для створення і перегляду подій та груп. Користувачі зможуть не лише приєднуватися до наявних заходів, а й створювати власні з описом, локацією, часом та можливістю коментування. Групи будуть місцем об'єднання за інтересами або локацією, де можна публікувати контент, вести дискусії та долучати нових учасників.

Система статусів користувачів буде ще одним засобом взаємодії та ідентифікації активності. Очікується, що кожен користувач матиме змогу обрати відповідний статус, який буде сигналізувати іншим про його настрій, наміри чи поточну діяльність. Це дозволить сформувати більш контекстну взаємодію в межах спільноти.

Поведінка користувача передбачатиме поетапну взаємодію із системою: після реєстрації – заповнення профілю, після чого система надаватиме персоналізовані рекомендації. Надалі користувачі зможуть приєднуватися до подій, спільнот, встановлювати нові контакти та вести комунікацію через вбудовану систему обміну повідомленнями. Усі зміни, оновлення або запити супроводжуватимуться відповідними сповіщеннями.

2.4 Нефункціональні вимоги

Нефункціональні вимоги до системи включатимуть високу швидкість, надійність та безпечність. Передбачається, що система зможе обробляти основні запити за час до однієї секунди. Буде впроваджено засоби безпеки, включаючи JWT-аутентифікацію.

2.5 Перспективи розвитку

Враховуючи обрану архітектуру та технології, проєкт має високий потенціал до подальшого масштабування. У перспективі система зможе підтримувати додаткові функції, розширення мовного середовища, інтеграцію із зовнішніми сервісами та мобільні платформи. Застосунок розглядатиметься як гнучке та масштабоване рішення, яке може відповідати запитам широкого кола користувачів в умовах змінного глобального середовища.

При розробці цієї системи також буде враховано гнучкість архітектури з можливістю розширення функціональних можливостей без необхідності істотного переписування коду. Модульна структура дозволить додавати нові сервіси, наприклад, систему рекомендацій подій на основі машинного навчання або адаптивні опитування користувачів для збору даних про їх потреби. Це дозволить удосконалити систему відповідно до змін у поведінці користувачів або розширення аудиторії.

Планується також у майбутньому реалізувати базову аналітику використання функцій системи. Вона дозволить адміністраторам аналізувати активність користувачів, кількість взаємодій, популярність груп та подій. Це дасть змогу визначати точки зростання, вдосконалити інтерфейс і точніше планувати майбутні оновлення. Дані аналітики будуть зберігатися у відповідних таблицях бази даних і періодично оновлюватися.

Крім традиційного текстового чату, розглядається можливість реалізації функціоналу коротких аудіоповідомлень між користувачами через деякий час. Така форма спілкування є природнішою та швидшою в багатьох ситуаціях, особливо коли мова йде про людей, які не володіють швидким набором тексту іноземною

мовою. Це стане в пригоді під час адаптації, коли потрібне швидке пояснення, консультація чи моральна підтримка.

Очікується, що структура бази даних також буде розширюватися. Зокрема, можуть бути додані таблиці для збереження історії активності користувачів, їх попередніх статусів, участі в подіях, переглянутих груп та взаємодій із системою рекомендацій. Це дозволить у майбутньому створити інтелектуальну систему повторних нагадувань або історії рекомендацій, яка допоможе користувачам повертатися до раніше переглянутого контенту.

З метою залучення нових користувачів і популяризації платформи, у майбутньому буде доцільно реалізувати реферальну систему. Користувачі зможуть запрошувати друзів, отримуючи бонуси у вигляді розширеного функціоналу, доступу до преміум-груп або збільшення можливостей профілю. Такий механізм також має вплив на активність спільноти та сприятиме органічному зростанню платформи.

Інтерфейс користувача планується зробити максимально доступним і зрозумілим навіть для осіб із мінімальним досвідом користування цифровими сервісами. Згодом увагу буде приділено адаптації для людей з обмеженими можливостями: передбачатиметься сумісність з програмами для зчитування тексту з екрана, використання контрастних кольорів [13].

При проектуванні клієнтської частини буде передбачено можливість часткового кешування даних, що дозволить зменшити кількість запитів до серверу та прискорити рендеринг контенту. Це особливо важливо при використанні платформи у країнах із нестабільним або повільним інтернет-з'єднанням, де стабільна робота застосунку є критичною.

Ще одним напрямом розвитку платформи стане створення інструментів для модераторів спільнот. Їм буде надано окрему панель керування, де можна буде підтверджувати нових учасників, модерувати повідомлення, блокувати користувачів за порушення правил. Це дозволить зберегти дружню атмосферу в групах і запобігти небажаному контенту.

У майбутньому можлива інтеграція з картографічними сервісами, які дозволять користувачам переглядати події на карті міста або фільтрувати їх за географічною близькістю. Це зробить пошук релевантних заходів інтуїтивнішим і природнішим, особливо для новоприбулих користувачів, які ще не орієнтуються в місцевості.

Загалом, концепція розробки програмна система для підтримки соціальної адаптації при переїзді в іншу країну має на меті створення екосистеми, що не лише допомагатиме користувачам знаходити однодумців, але й сприятиме формуванню сталих локальних спільнот. Система має стати містком між людьми з різних культур, що перебувають у подібних життєвих обставинах, та забезпечити їм платформу для зростання, співпраці та взаємопідтримки.

3 АРХІТЕКТУРА ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Проєктування бази даних

3.1.1 ER – діаграма класів

На основі проведеного аналізу предметної області було побудовано модель бази даних у вигляді спрощеної ER-діаграми, яка візуалізує структуру основних сутностей та взаємозв'язків між ними. Дана діаграма відображає логічну модель збереження даних для програмної системи, орієнтованої на підтримку соціальної інтеграції користувачів у новому середовищі (див. рис. 3.1).

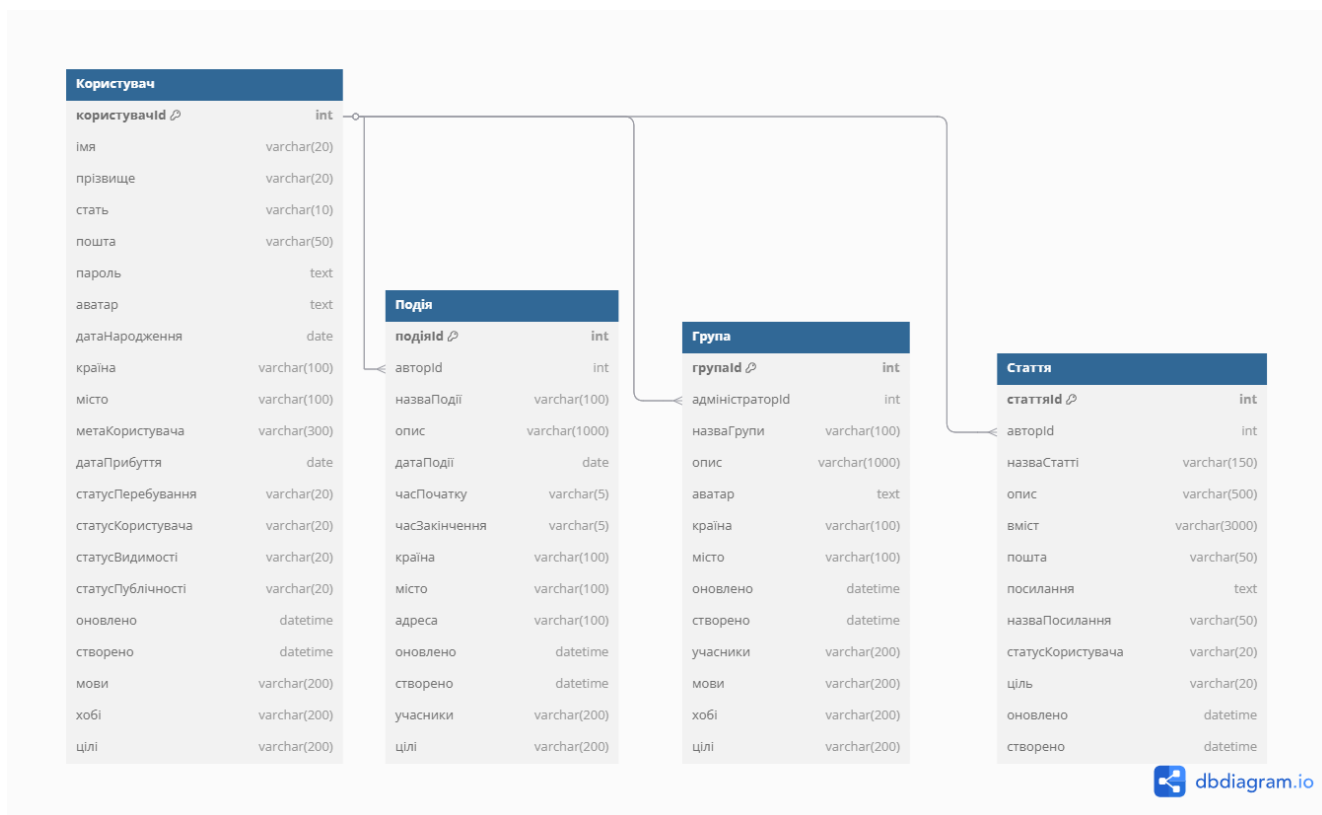


Рисунок 3.1 – Спрощена ER-діаграми (рисунок виконаний самостійно)

Користувач виступає як центральна сутність системи, що містить усю персональну інформацію про зареєстрованих осіб, включаючи ім'я, прізвище, стать, дату народження, країну та місто проживання. Додатково передбачено збереження аватара, електронної пошти, пароля (в зашифрованому вигляді), а також таких характеристик, як інтереси, мовні навички, особисті цілі та статуси. Користувачі можуть вказувати дату прибуття, мету перебування та свій поточний

емоційний стан через систему статусів. Структура також дозволяє зберігати дату створення та оновлення профілю.

Сутність Подія пов'язується з користувачем через зовнішній ключ, що вказує на автора. Вона містить детальний опис запланованих заходів, включаючи назву, місце проведення (країна, місто, адреса), дату та часові межі. Для кожної події зберігається список учасників, визначених за критеріями, подібними до користувацьких інтересів, статусів або мов. Важливим є також поле «цілі», що дозволяє встановити тематичну спрямованість події, наприклад: нетворкінг, культурний обмін, відпочинок тощо.

У структурі Група зберігається інформація про соціальні спільноти, які створюються користувачами для об'єднання за інтересами або місцем перебування. Група має адміністратора, який несе відповідальність за її наповнення, управління учасниками. Кожна група характеризується назвою, описом, місцем прив'язки (місто, країна) та списком учасників. Як і у випадку подій, структура включає мовний, хобі-та цільовий контекст, що дозволяє забезпечити релевантність рекомендацій.

Сутність Стаття є додатковим елементом функціональності платформи, який забезпечує можливість користувачам публікувати корисний контент – це можуть бути особисті історії адаптації, новини, поради чи аналітичні матеріали. Статті містять назву, опис, основний зміст, посилання на зовнішні ресурси, а також інформацію про автора та статус користувача. Передбачена підтримка контактної інформації (електронна пошта) для зворотного зв'язку або можливості підписки у подальшому на інші публікації автора.

Всі сутності пов'язано логічними зв'язками через поля з зовнішніми ключами, що дозволяє реалізувати типові запити до бази даних, пов'язані з отриманням подій певного користувача, фільтрацією спільнот за локацією, або переглядом усіх публікацій від конкретного автора. Зв'язки «один до багатьох» встановлено між таблицями користувачів та подій, груп, а також статей, що дозволяє в одному профілі зберігати відповідні об'єкти взаємодії.

Запропонована спрощена ER-діаграма демонструє структуровану, логічно взаємопов'язану модель бази даних, яка підтримує масштабованість, модульність і можливість подальшого розширення функціоналу. Така структура слугує основою для ефективної реалізації персоналізованих рекомендацій, інтелектуального пошуку та побудови соціальних взаємозв'язків у середовищі нового проживання.

3.1.2 Формування повної логічної моделі бази даних

На основі попередньо побудованої спрощеної ER-діаграми була сформована деталізована логічна модель бази даних, що відповідає вимогам третьої нормальної форми (3NF) (див. рис. 3.2). Така модель дозволяє усунути надлишковість, підвищити узгодженість та забезпечити ефективність при виконанні типових запитів.

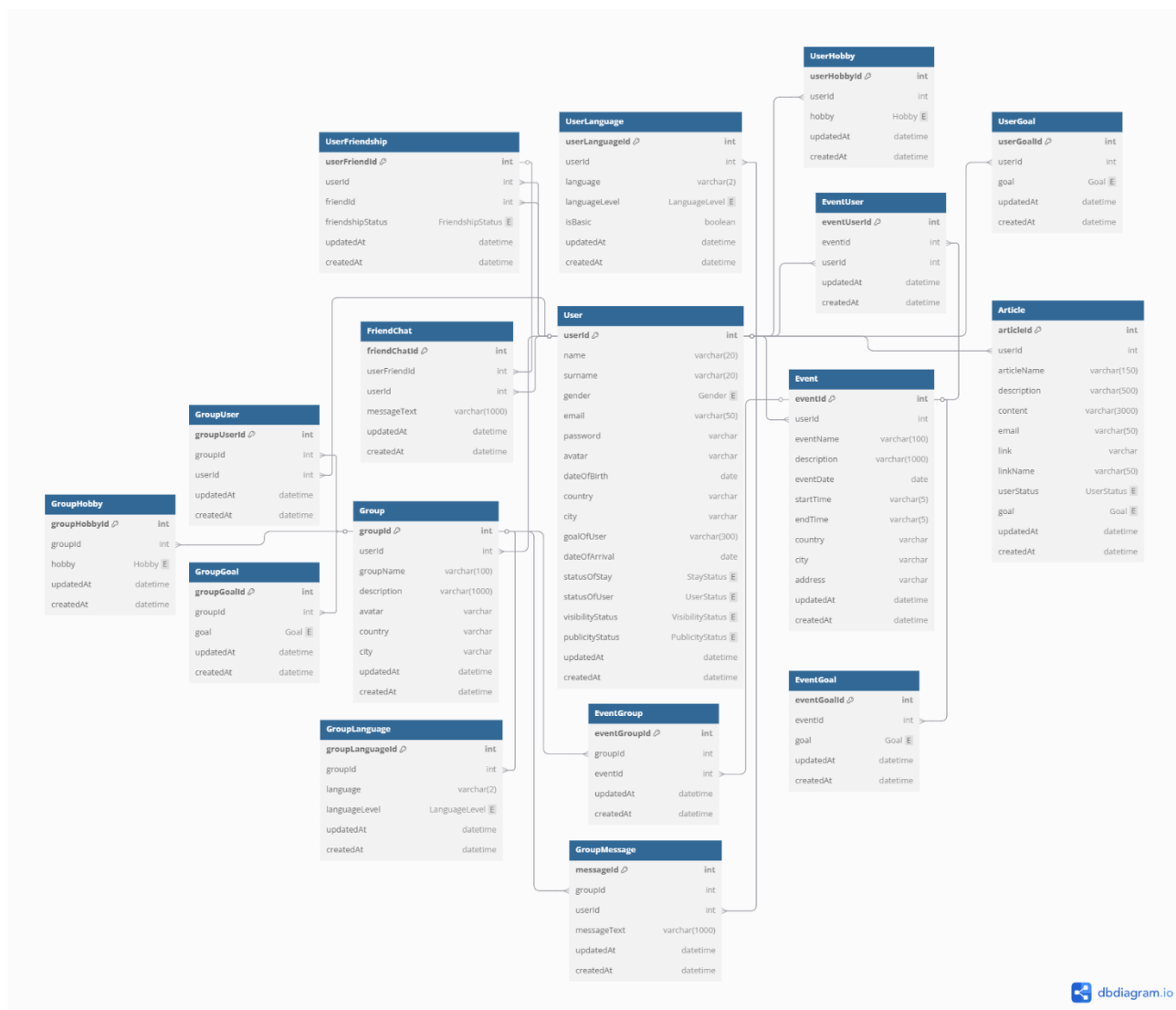


Рисунок 3.2 – Логічна модель бази даних (рисунок виконаний самостійно)

Центральною сутністю залишається таблиця користувачів (User), до якої було додано атрибути, що відображають розширену інформацію про користувача: дата народження, дата прибуття, мета перебування, статуси, мовні навички, інтереси та особисті цілі. Кожен атрибут був проаналізований на предмет функціональної залежності, що дозволило виділити окремі довідкові сутності: UserLanguage, UserGoal, UserHobby. Це дало змогу зменшити надмірність даних і забезпечити більшу гнучкість у подальшому розширенні.

Окрім цього, було розроблено структури для моделювання соціальних зв'язків між користувачами. Таблиця UserFriendship реалізує зв'язок з можливістю фіксувати статус взаємодії (очікує підтвердження, прийнято, відхилено). Додатково реалізовано структуру для збереження особистих повідомлень (FriendChat), яка дозволяє користувачам вести приватне листування у межах системи.

Сутність Event була деталізована через розширення атрибутивної частини (дата, місце, тривалість, опис, автор), а також через реалізацію таблиці EventUser, яка відповідає за реєстрацію участі користувачів у подіях. Для збереження тематики заходів та підбору релевантних подій була створена таблиця EventGoal, що дозволяє фіксувати декілька цілей для однієї події.

Групи (Group) моделюються як окремі сутності з атрибутами: назва, опис, візуальне оформлення, місце розташування, публічність тощо. Для реалізації зв'язків «багато до багатьох» між групами та користувачами використовується таблиця GroupUser, а для опису характеристик груп – таблиці GroupLanguage, GroupHobby та GroupGoal. Це дає змогу створювати складні умови фільтрації, а також автоматизувати рекомендації спільнот для нових користувачів на основі їхніх вподобань.

Сутність Article дозволяє користувачам створювати та публікувати корисні матеріали (історії, поради, новини). Кожна стаття містить назву, опис, вміст, посилання на зовнішні джерела, інформацію про автора та категоризацію за статусом користувача чи метою. Це дозволяє підвищити інформаційну підтримку новоприбулих користувачів, що є важливою частиною процесу адаптації.

Усі сутності пов'язані зовнішніми ключами, що забезпечує узгодженість між таблицями та дозволяє будувати ефективні запити. Така логічна модель забезпечує можливість масштабування та адаптації системи під зростаючу кількість користувачів без суттєвих змін у структурі бази даних.

3.1.3 Нормалізація логічної моделі бази даних

Нормалізація логічної моделі бази даних була здійснена відповідно до принципів третьої нормальної форми (3NF). Основна мета нормалізації – усунення надлишкових даних, забезпечення логічної цілісності структури та уникнення аномалій при оновленні, вставці та видаленні записів. Завдяки цьому забезпечується стабільність та ефективність функціонування всієї системи зберігання даних.

Першим кроком у процесі нормалізації стало виділення сутностей, які містили поля зі списками значень. Зокрема, атрибути, що описують інтереси, цілі користувачів та володіння мовами, були винесені в окремі таблиці – UserHobby, UserGoal, UserLanguage. Це дозволило усунути повторення в основній таблиці користувачів і забезпечити можливість прив'язки довільної кількості значень до одного об'єкта, зберігаючи при цьому просту та гнучку структуру.

Аналогічний підхід був застосований до подій та груп. Для відображення тематичних цілей подій створено таблицю EventGoal, а для груп – таблиці GroupGoal, GroupLanguage та GroupHobby. Таким чином, було реалізовано добре сформований механізм зв'язків, що дозволяє ефективно організовувати дані та забезпечувати широкий спектр можливостей для фільтрації та персоналізації вмісту.

Крім того, була впроваджена таблиця UserFriendship для реалізації зв'язків між користувачами, що дозволяє уникнути зберігання масивів ідентифікаторів у вигляді рядкових списків, які ускладнюють обробку. Всі поля, що мають взаємозалежності, були оптимізовані з урахуванням того, аби кожне значення залежало лише від первинного ключа таблиці, до якої воно належить.

В результаті нормалізації структура бази даних стала більш прозорою, гнучкою та зручною для масштабування. Усі зв'язки між сутностями були реалізовані за допомогою зовнішніх ключів, що дозволяє зберігати логічну цілісність при операціях з даними. Такий підхід також спрощує реалізацію каскадних оновлень або видалень, у разі необхідності, та дозволяє уникати типових проблем, пов'язаних з неузгодженістю записів.

Побудована модель відповідає третій нормальній формі, оскільки в жодній таблиці немає повторюваних груп даних, усі неключові атрибути мають залежність тільки від первинного ключа і не мають транзитивних залежностей. Це забезпечує високу якість зберігання інформації та гарантує стабільну роботу бази даних у процесі експлуатації програмної системи.

3.2 Архітектура серверної частини

У рамках серверної частини програмної системи, реалізованої за допомогою фреймворку NestJS, було передбачено набір функцій, що забезпечують обробку запитів користувачів та керування їхніми діями. Основу бізнес-логіки становлять сервіси, які взаємодіють з контролерами та базою даних через Prisma ORM. Кожна з функцій спрямована на обробку запитів до відповідних сутностей та забезпечення цілісності даних.

При реєстрації нового користувача система обробляє вхідні дані, створює унікальний запис у базі даних та генерує токен для подальшої авторизації. Авторизація реалізована за допомогою JWT (JSON Web Token), що забезпечує безпечний і контрольований доступ до функціоналу застосунку. Усі паролі перед збереженням проходять шифрування, що гарантує захист персональних даних.

Після входу в систему користувач отримує доступ до редагування власного профілю. Серверна логіка дозволяє оновлювати такі параметри, як місце проживання, статус, мовні навички, хобі та особисті цілі. Всі зміни синхронізуються з відповідними довідковими таблицями через зв'язки «багато до багатьох».

Однією з ключових функцій є генерація персоналізованих рекомендацій. Алгоритми на основі запитів до таблиць User, UserGoal, UserHobby та UserLanguage дозволяють підбирати релевантних потенційних друзів, групи та події. Відповідні методи реалізовано в окремих сервісах, які фільтрують користувачів за мовами, статусами або цілями перебування.

Користувач також може створювати події та групи. Для кожної дії створюється запис у таблиці з прив'язкою до автора. У випадку подій додаються відповідні записи до таблиць EventGoal та EventUser, де фіксується інформація про цілі події та учасників. Аналогічно, при створенні групи створюються записи у таблицях GroupGoal, GroupLanguage та GroupUser.

Також реалізована функція відправлення заявок у друзі. Через таблицю UserFriendship зберігаються статуси взаємодії (очікування, прийнято, відхилено), що дозволяє будувати двосторонні зв'язки між користувачами. Серверна логіка передбачає перевірку дійсності заявок, щоб уникнути дублювання або помилкових записів.

Таким чином, серверна частина системи виконує роль зв'язувального елемента між інтерфейсом та базою даних, реалізуючи всі ключові дії користувачів, пов'язані з налаштуванням профілю, соціальними зв'язками, створенням контенту та персоналізованими рекомендаціями.

3.2.1 UML-діаграма розгортання

З точки зору архітектури, програмна система реалізована на основі трірівневої моделі, яка включає клієнтську частину (на базі Next.js), серверну логіку (реалізовану через NestJS) та окремий сервер бази даних, що працює на MySQL. Комунікація між клієнтом і сервером здійснюється за захищеним протоколом HTTPS, а обмін даними між сервером застосунку та базою – через TCP/IP. Така структура дозволяє розмежувати відповідальність між рівнями системи, що позитивно впливає на продуктивність і масштабованість.

UML-діаграма розгортання ілюструє фізичну конфігурацію системи та взаємодію її основних компонентів (див. рис. 3.3). Вона демонструє, як

користувачький браузер звертається до клієнтської частини (Next.js), яка у свою чергу надсилає запити до серверної частини (NestJS). Серверна логіка обробляє запити, взаємодіє з базою даних і повертає клієнту необхідну інформацію. Кожен компонент виконує свою чітко визначену роль у загальній системі, що дозволяє легко масштабувати платформу, розподіляючи навантаження між мікросервісами чи окремими вузлами.

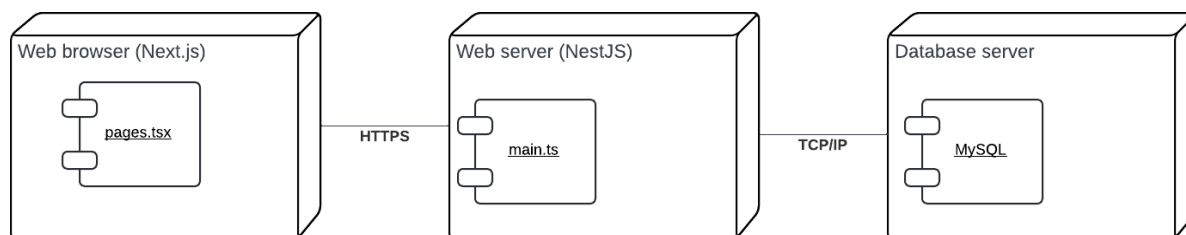


Рисунок 3.3 – UML-діаграма розгортання (рисунок виконаний самостійно)

3.2.2 UML-діаграма пакетів

UML-діаграма пакетів відображає, як компоненти системи організовані в окремі логічні блоки, які взаємодіють між собою. Це допомагає краще зрозуміти структуру коду, полегшує навігацію по проєкту і підтримку програмного забезпечення (див. рис. 3.4).

Система поділена на чотири основні рівні. На рівні Presentation знаходяться контролери – вони відповідають за отримання запитів від користувачів і передачу їх до бізнес-логіки. Наступний рівень – Application, він містить сервіси, які обробляють логіку запитів, звертаються до потрібних даних і формують відповіді.

Domain-рівень містить моделі та об'єкти перенесення даних (DTO), які описують структуру інформації та забезпечують її узгоджене переміщення між частинами системи. Вони слугують своєрідним мостом між логікою застосунку та даними користувачів або інших сутностей.

Інфраструктурний рівень включає Prisma Client – інструмент для взаємодії з базою даних та модуль міграцій, який відповідає за структуру таблиць і їх зміни. Цей рівень безпосередньо підключається до бази даних MySQL і забезпечує збереження та оновлення інформації в ній.

Завдяки такому поділу система є зрозумілою та простою в масштабуванні. Кожен рівень взаємодіє лише з сусідніми, що зменшує кількість зв'язків і робить систему стабільною. Розробник може легко оновити або замінити окремий модуль без потреби змінювати всю систему. Такий підхід підтримує порядок у коді та дає змогу швидко адаптуватися до нових вимог або технологій.

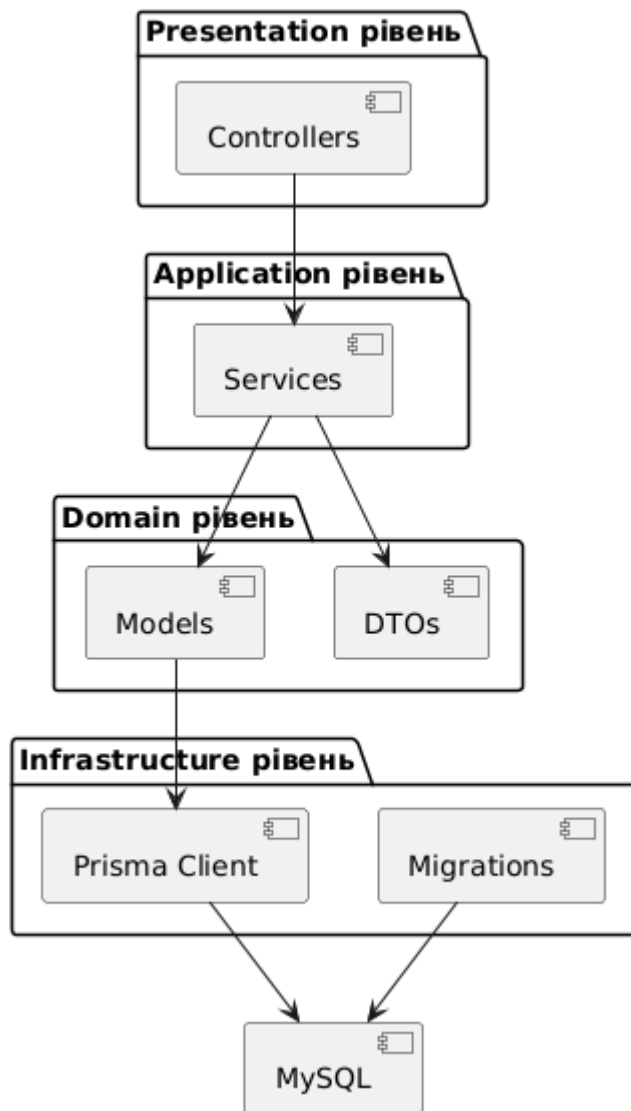


Рисунок 3.4 – UML-діаграма пакетів (рисунок виконаний самостійно)

3.3 Клієнтська частина

Клієнтська частина веб-застосунку реалізована з використанням фреймворку Next.js і відповідає за взаємодію користувача з інтерфейсом. Основна мета клієнтської частини полягає у забезпеченні зручного доступу до функціоналу системи, керування даними профілю, взаємодії з іншими користувачами, а також створення і перегляд контенту. Всі дії, що здійснюються користувачем через браузер, відправляються у вигляді HTTP-запитів на серверну частину, де обробляються відповідними контролерами NestJS.

Інтерфейс дозволяє користувачам проходити реєстрацію та авторизацію, заповнювати та редагувати профіль. При редагуванні профілю доступна зміна імені, прізвища, країни, міста, дати прибуття, мов, хобі, цілей, статусів, статусу видимості та статусу публічності. Також передбачена можливість змінити пароль.

Після авторизації користувач може переглядати інших користувачів, отримувати персоналізовані рекомендації, надсилати заявки в друзі, приймати або відхиляти запити, а також переглядати своїх друзів. Реалізовано перегляд профілів інших користувачів з можливістю переходу до чату.

Користувач має змогу створювати події з зазначенням тематики, місця, дати, часу, опису та відповідних цілей. Реалізовано також перегляд створених подій, оновлення інформації про них та можливість видалення. Для забезпечення залучення до заходів, користувач може долучатися до подій інших осіб, що відображається у відповідному розділі.

Аналогічно працює механізм груп. Користувач може створити групу, додати опис, зображення, вибрати мови спілкування, хобі та цілі. Також реалізовано функції приєднання до груп, виходу з групи, перегляду списку своїх груп та учасників інших груп.

Додатковою можливістю є створення інформаційного контенту у вигляді статей. Кожен користувач може створити статтю з темою, описом, вмістом, посиланням на зовнішні ресурси, категорією за статусом та цілями. Передбачено функції перегляду статей, редагування та видалення за потреби.

Клієнтська частина забезпечує весь необхідний функціонал для користувача, дозволяючи йому ефективно взаємодіяти з системою, адаптувати профіль до особистих уподобань та брати участь у соціальних активностях, що є важливою складовою процесу адаптації в новому середовищі.

3.3.1 UML-діаграма клієнтської частини

Для наочного відображення взаємодії користувача з функціоналом клієнтської частини була побудована UML-діаграма варіантів використання, що відображає основні дії користувача в межах системи. Ця діаграма структурує всі можливості, які має користувач на рівні інтерфейсу, та дозволяє краще уявити логіку переходів і взаємозв'язків між функціональними блоками (див. рис. 3.5 та рис. 3.6).



Рисунок 3.5 – UML-діаграма клієнтської частини: профіль, друзі, групи, користувачі (рисунок виконаний самостійно)

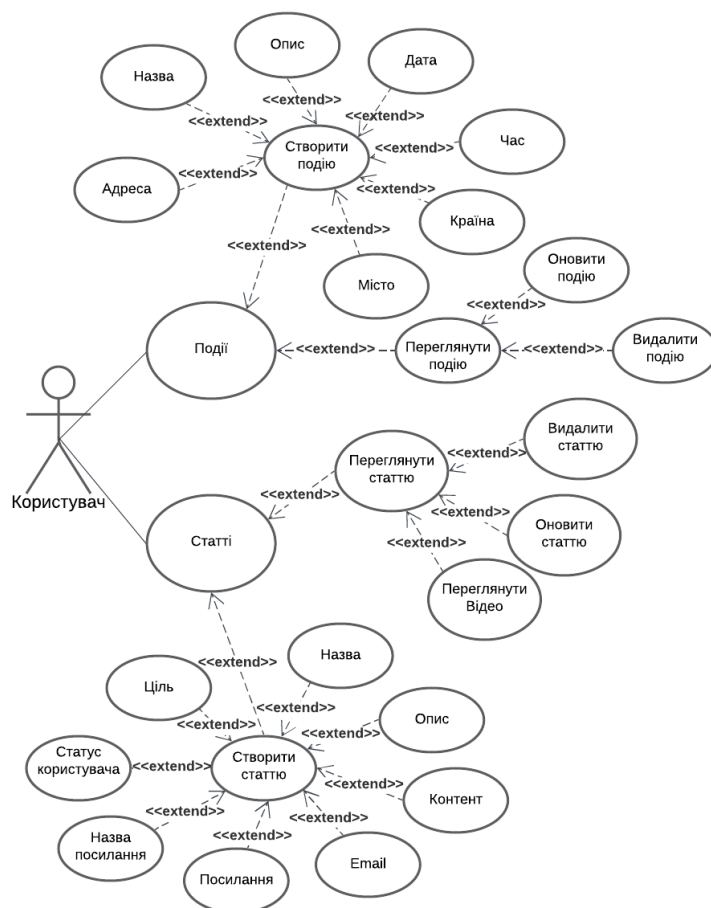


Рисунок 3.6 – UML-діаграма клієнтської частини: події, статті (рисунок виконаний самостійно)

На діаграмі видно, що користувач має можливість переглядати та редагувати свій профіль, включаючи зміну таких параметрів, як ім'я, прізвище, стать, email, країна, місто, дата прибуття, мови, хобі, цілі та статуси. Окремими розгалуженнями вказано, що користувач може змінювати пароль та налаштовувати видимість профілю.

У контексті соціальної взаємодії користувач має змогу переглядати інших користувачів, додавати їх у друзі, переглядати рекомендації, приймати або відхиляти запити на дружбу, а також переглядати профілі інших осіб. Діаграма охоплює дії, пов'язані із взаємодією з подіями: створення, перегляд, редагування та видалення подій. Передбачено також функціональність для груп: створення, перегляд, приєднання, вихід та управління складом груп.

Окремим блоком відображено функціонал, пов'язаний зі створенням та переглядом статей, включаючи заповнення ключових полів: назва, опис, контент, посилання, email та відповідні категорії. Діаграма демонструє ієрархію та логічні зв'язки між діями, що дозволить розробникам, тестувальникам та аналітикам краще розуміти структуру взаємодії користувача з системою.

UML-діаграма клієнтської частини відіграє важливу роль у процесі розробки, оскільки дає змогу формалізувати очікувану поведінку інтерфейсу з точки зору кінцевого користувача. Вона також є корисним інструментом для тестування функціональності та виявлення потенційних помилок у сценаріях використання на етапі проектування системи.

3.3.2 UML-діаграму пакетів

Для представлення архітектурної структури клієнтської частини застосунку було побудовано UML-діаграму пакетів, яка ілюструє взаємодію між основними бібліотеками, інструментами та компонентами фронтенду (див. рис. 3.7). У центрі діаграми розташовано пакет Pages, який акумулює всю бізнес-логіку інтерфейсу та забезпечує зв'язок між візуальними компонентами та даними, що надходять від серверної частини.

Цей пакет взаємодіє з основними інструментами, які використовуються у розробці інтерфейсу, зокрема React, TypeScript, Next, Tailwind CSS, NextUI, React Query, Axios, Formik, Yup та React Toastify. Такі залежності демонструють, що клієнтська частина активно використовує сучасні бібліотеки для реалізації форм, валідації, відображення повідомлень, асинхронних запитів та стилізації елементів інтерфейсу.

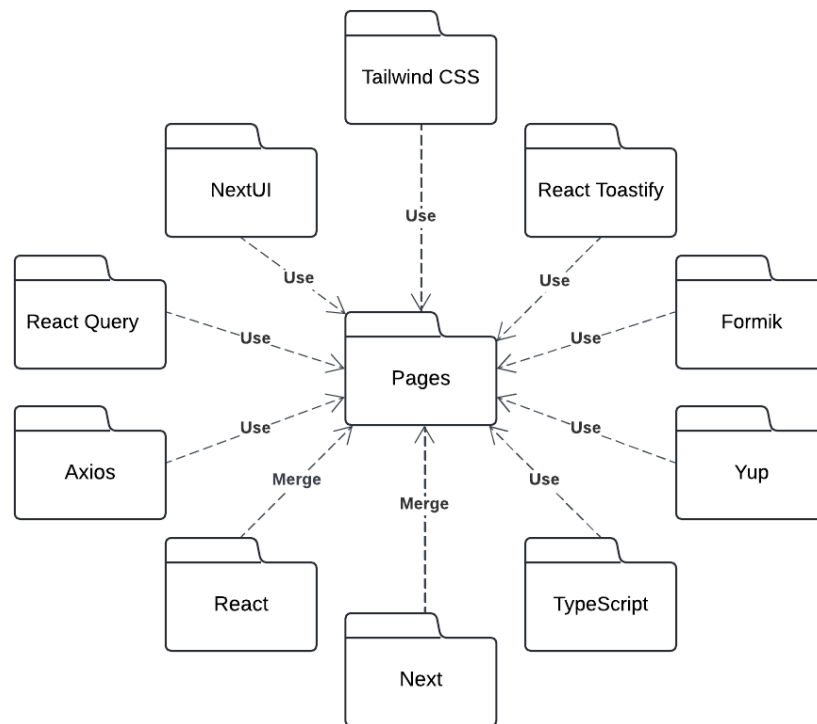


Рисунок 3.7 – UML-діаграму пакетів (рисунок виконаний самостійно)

Зв'язки між пакетами позначено як «Use», що вказує на використання функціональності одного пакета іншим. Наприклад, Pages активно використовує компоненти бібліотеки NextUI для побудови адаптивного інтерфейсу, бібліотеку Axios для HTTP-запитів, а Tailwind CSS забезпечує швидку стилізацію елементів. Залежність «Merge» між React і Next демонструє інтеграцію цих технологій у структурі клієнтської частини застосунку.

UML-діаграма пакетів дозволяє наочно побачити, як саме організована структура клієнтської частини на рівні взаємодії між технологіями. Це сприяє кращому розумінню архітектури, а також полегшує підтримку та розширення функціональності застосунку в майбутньому.

3.3.3 Модулі клієнтської частини

Клієнтська частина веб-застосунку, реалізована з використанням фреймворку Next.js, організована за принципами модульності та розподілу відповідальностей. Така структура дозволяє зручно масштабувати застосунок, підтримувати чисту архітектуру та забезпечувати повторне використання коду в межах різних компонентів системи. Кожен модуль виконує певну роль у

забезпеченні функціональності інтерфейсу користувача та логіки його взаємодії із серверною частиною.

Модуль `pages`: містить основні сторінки веб-застосунку, включаючи головну, сторінку входу, реєстрації, перегляду подій, груп, користувачів та профілю. Він також організований у вигляді маршрутизованої структури, де кожна сторінка відповідає за певну функціональність інтерфейсу, що полегшує навігацію та логіку рендерингу на стороні клієнта.

Модуль `components`: включає багаторазово використовувані інтерфейсні компоненти, такі як форми введення, кнопки, списки, картки подій, попапи, повідомлення про успіх чи помилку. Це дозволяє забезпечити узгодженість зовнішнього вигляду та поведінки у різних частинах застосунку.

Модуль `services`: відповідає за організацію взаємодії клієнтської частини із сервером через HTTP-запити. Тут реалізовані обгортки для бібліотеки `Axios`, що дозволяють отримувати, створювати, оновлювати або видаляти дані, взаємодіючи з API серверної частини.

Модуль `interfaces`: зберігає типи та інтерфейси, написані з використанням `TypeScript`. Вони забезпечують строгість типізації, зменшують кількість помилок на етапі компіляції та допомагають у документуванні структури даних, які передаються між компонентами.

Модуль `hooks`: включає кастомні хуки `React`, які використовуються для спрощення логіки керування станом, побудови ефектів або повторюваної взаємодії з API. Наприклад, хук для отримання рекомендацій або завантаження профілю користувача.

Модуль `constants`: містить фіксовані значення, такі як списки статусів користувача, попередньо визначені мовні опції, або шаблони для повідомлень. Це дозволяє централізовано змінювати статичні ресурси, не впливаючи на логіку компонентів.

Модуль `routes`: забезпечує конфігурацію маршрутизації в межах клієнтського застосунку. Включає визначення шляхів до сторінок, а також структуру

динамічних маршрутів, що забезпечує правильну навігацію у разі переходу між профілями, подіями або групами.

Модуль `styles`: містить файли стилізації, реалізовані за допомогою Tailwind CSS. Вони інтегровані безпосередньо у компоненти через класи, що дозволяє швидко адаптувати зовнішній вигляд інтерфейсу до потреб дизайну, хоча наразі реалізовано лише для десктопної версії.

Модуль `utils`: включає допоміжні функції, які реалізують спільну логіку, наприклад, форматування дати, перевірку вхідних даних або трансформацію відповідей з API. Їх використання сприяє підвищенню чистоти коду та його повторного застосування.

Усі ці модулі разом створюють цілісну архітектуру клієнтської частини, яка підтримує принципи інкапсуляції, перевикористання та відповідальності. Такий підхід забезпечує стабільну та зрозумілу основу для розширення застосунку в майбутньому.

3.4 Проектування дизайну програмної системи

Проектування дизайну програмної системи є важливим етапом розробки, оскільки визначає подальше сприйняття користувачем інтерфейсу та забезпечує зручність роботи з веб-застосунком. На етапі планування було вирішено створити інтуїтивно зрозумілий, сучасний та мінімалістичний дизайн [14], що відповідає актуальним тенденціям UI/UX-дизайну. Головна увага приділяється простоті та логічності навігації, яка дозволить користувачам легко знаходити необхідні функції та інформацію.

Передбачено використання темних тонів як базового стилю, зокрема чорний та темно-сірий кольори у поєднанні з контрастними акцентами синього кольору. Така кольорова гамма обрана з метою створення приємного для очей інтерфейсу, що дозволяє зменшити втому під час тривалого користування додатком, а також забезпечує чітке виділення інтерактивних елементів [15].

Всі основні розділи веб-застосунку (профіль, друзі, групи, користувачі, події та статті) заплановано розташувати в єдиній навігаційній панелі ліворуч, яка буде

доступна з будь-якої сторінки додатку. Це забезпечить швидкий доступ до всіх необхідних функцій, а також зручне переміщення між сторінками.

На сторінці профілю користувача передбачається блок з основною інформацією, яка включатиме ім'я, місто та актуальний статус користувача. Також передбачені блоки, що відображатимуть цілі користувача, інтереси, мови, якими він володіє, та список майбутніх подій, на які він зареєструвався. Кожен з цих блоків матиме можливість редагування інформації з метою забезпечення актуальності даних. Ця сторінка є основною в структурі інтерфейсу програмної системи. Саме для неї було створено базовий прототип (див. рис. 3.8), що слугує відправною точкою для подальшого проектування решти інтерфейсу.

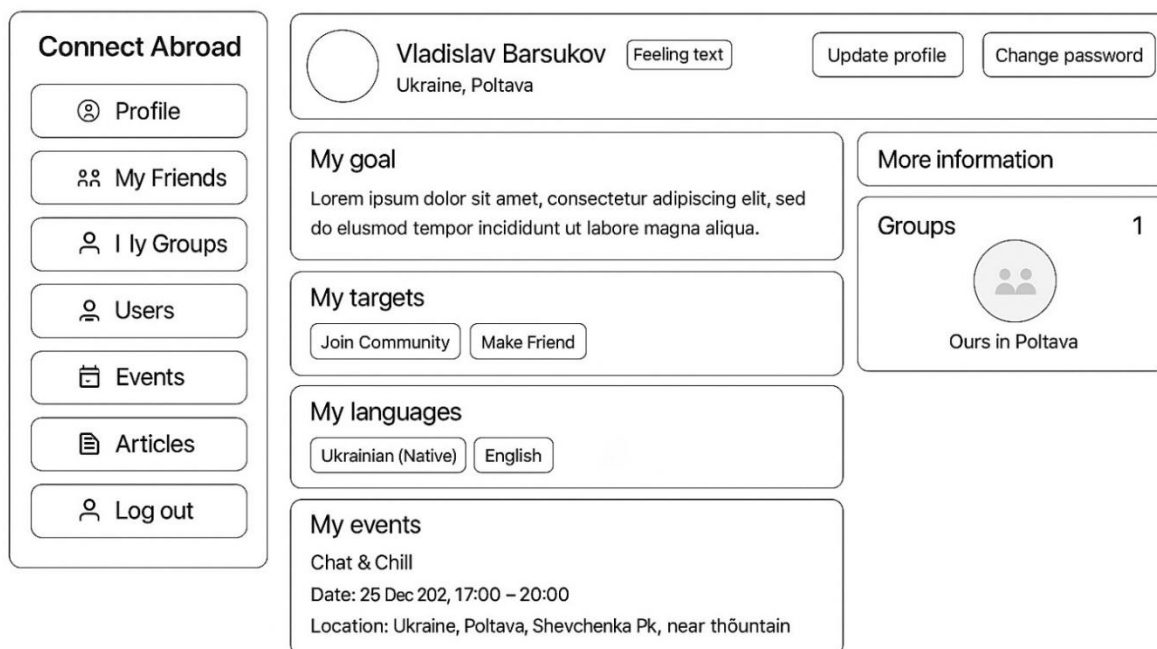


Рисунок 3.8 – Прототип інтерфейсу сторінки профілю користувача (рисунок виконаний самостійно)

Сторінка друзів буде оформлена у вигляді зручного списку з переліком друзів та блоків запитів запрошення у друзі. На цій сторінці користувач матиме змогу перейти до профілю друга та відкрити чат з ним.

Для сторінки з групами передбачено візуальне відображення у вигляді списку, де після переходу в одну з них буде відображатися назва групи, короткий опис і кнопка для швидкого приєднання. Сторінка міститиме поля з інтересами та локацією, що дозволить легко знаходити актуальні спільноти.

Сторінка «Користувачі» буде організована за аналогічним принципом, з використанням рекомендацій за мовами, статусами і місцезнаходженням. Передбачено також детальний перегляд профілів з можливістю відправлення запитів на додавання у друзі.

Розділ з подіями отримає структуру у вигляді календаря та списку найближчих заходів, що допоможе користувачам ефективно планувати свій час. Кожна подія матиме детальний опис, дату, час, місце проведення, список учасників та цілі заходу, що робитиме її максимально інформативною.

Сторінка з статтями буде реалізована зручним блоковим макетом, де кожна стаття супроводжуватиметься заголовком, коротким анонсом та посиланням на додатковий матеріал. Інтерфейс передбачає можливість фільтрації за тематикою і актуальністю матеріалів.

Для інтерактивних елементів планується використання зрозумілих іконок та тегів, що спрощують швидке розпізнавання дій, які може виконати користувач, наприклад, «долучитися до спільноти», «знайти друзів» або «взяти участь у дискусіях». Такий підхід зробить інтерфейс більш інформативним та зменшить необхідність додаткових пояснень.

Таким чином, планується створити простий, але функціональний дизайн, який дозволить забезпечити зручність і легкість взаємодії з додатком.

4 ОПИС ПРИЙНЯТИХ ПРОГРАМНИХ РІШЕНЬ

4.1 Структура та процес збірки

Програмна система для підтримки соціальної адаптації при переїзді в іншу країну розроблялася локально в середовищі Visual Studio Code, із використанням чітко структурованої файлової організації. Основна мета такої структури – забезпечити логічне розділення клієнтської та серверної частин, утиліт, стилів і конфігурацій, що значно спрощує навігацію проектом і його масштабування.

Усі вихідні коди фронтенду зберігаються в директорії `abroad-frontend`, що містить такі ключові підрозділи:

- `components` – базові UI-компоненти (`input`, `button`, меню);
- `layouts` – шаблони сторінок;
- `services` – логіка запитів до API;
- `hooks` – користувацькі React-хуки;
- `styles` – глобальні та модульні стилі Tailwind CSS;
- `utils` – утилітарні функції, регулярні вирази, константи;
- `routes` – маршрутизація;
- `dto`, `enum`, `interfaces` – типізація та опис структур даних.

Код бекенду розміщено в `abroad-backend` і містить модулі NestJS, організовані за класичною модульною структурою: `controller.ts`, `service.ts`, `module.ts`.

Для запуску клієнта використовуються стандартні скрипти з `package.json`, такі як `npm run dev` та `npm run build`. Завдяки використанню Next.js, клієнтська частина підтримує як серверний рендеринг, так і статичну генерацію сторінок.

4.2 Стек технологій і середовище виконання

Для реалізації програмної системи для підтримки соціальної адаптації було використано сучасний стек веб-технологій, який забезпечує стабільну роботу, гнучкість у розробці та зручний користувацький інтерфейс. Усі вибрані

інструменти мають відкритий код і активно підтримуються спільнотою, що полегшує розробку та подальше масштабування проєкту.

Клієнтська частина побудована на базі фреймворку Next.js (версія 14), який поєднує можливості React і серверного рендерингу. Використано нову модель маршрутизації – App Router, яка дозволяє ефективно працювати з динамічними сторінками, server/client components і автоматичним кешуванням. Це підвищує продуктивність інтерфейсу та зменшує затримки при взаємодії з користувачем.

Для стилізації застосовано Tailwind CSS, що забезпечує швидке створення адаптивного дизайну з підтримкою темної теми.

Серверна частина реалізована з використанням фреймворку NestJS, який побудований на базі Node.js і підтримує модульну архітектуру. NestJS дозволяє логічно структурувати контролери, сервіси та DTO (Data Transfer Object) в окремі модулі, що значно спрощує підтримку та розширення проєкту.

Для взаємодії з базою даних застосовується Prisma ORM – сучасний інструмент, що забезпечує типізований доступ до бази даних MySQL та підтримує міграції, які зберігаються у відповідній директорії prisma/migrations. Основна модель даних описана у файлі schema.prisma.

Уся розробка ведеться локально в середовищі Visual Studio Code, із використанням Node.js, npm (менеджер пакетів) та MySQL як системи управління базами даних. Завдяки цьому створено повністю ізольоване середовище розробки, яке не залежить від хмарних рішень.

Таким чином, використаний стек технологій дозволив реалізувати гнучкий, швидкий і масштабований застосунок, який легко адаптувати під потреби користувачів та розвиток у майбутньому.

4.3 Головна навігація й компоненти меню

Зручна та інтуїтивно зрозуміла навігація є одним із ключових факторів успішної взаємодії користувача із застосунком. У програмній системі Connect Abroad головна навігація реалізована у вигляді вертикального меню,

розташованого збоку, яке забезпечує швидкий доступ до основних розділів (див. рис. 4.1).

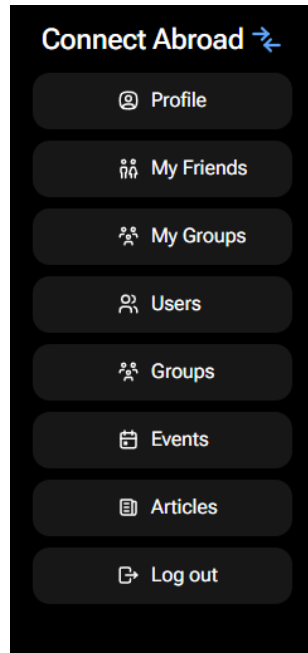


Рисунок 4.1 – Інтерфейс навігаційного меню (рисунок виконаний самостійно)

Основна структура меню описана у спеціальному конфігураційному файлі `Menu.config.ts`, де зібрані всі пункти у вигляді масиву об'єктів. Кожен елемент має ключ, текст для відображення, іконку та відповідне посилання. Завдяки цьому меню формується динамічно, без жорстко закодованих компонентів, що спрощує оновлення або розширення навігації в майбутньому.

Компонент `Menu.tsx` відповідає за візуалізацію меню. Він перебирає всі конфігуровані пункти та створює кнопки переходу з використанням бібліотеки `next/link`. Додатково в меню відображається логотип системи й індикатор активного розділу, що покращує орієнтацію користувача в інтерфейсі.

Важливою деталлю є інтеграція з авторизацією. Компонент меню автоматично підставляє `userId` до посилання профілю, зчитуючи його з токена за допомогою `jwtDecode`. Також у нижній частині меню реалізована кнопка виходу, яка викликає функцію `logout()` та очищує авторизаційні дані.

Завдяки гнучкому налаштуванню, меню можна легко адаптувати під різні ролі користувачів або мовні налаштування, використовуючи локалізовані назви пунктів.

4.4 Сторінка профілю користувача

Сторінка профілю є однією з найважливіших частин системи Connect Abroad, оскільки саме тут користувач може переглянути, редагувати або оновити свою персональну інформацію (див. рис. 4.2). Візуально інтерфейс побудовано в стилі «картки користувача», що містить загальні дані, опис статусу, а також розділи, які відображають хобі, цілі, поточний настрій і місце перебування.

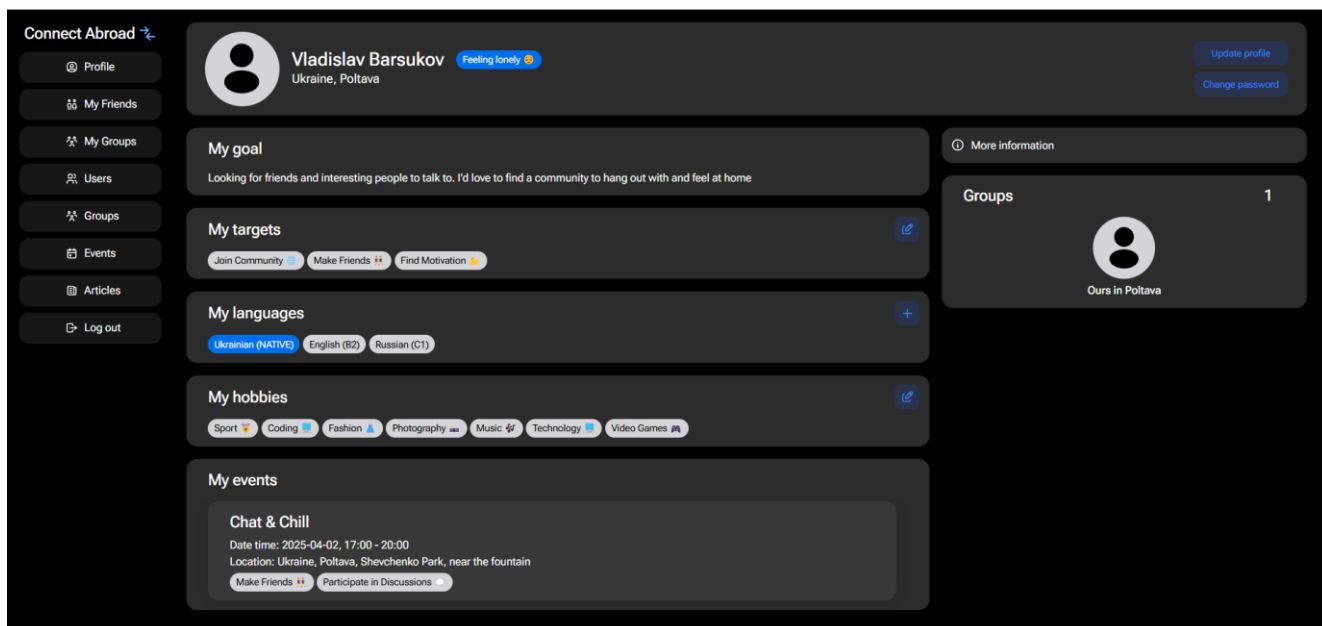


Рисунок 4.2 – Інтерфейс сторінки профілю (рисунок виконаний самостійно)

У верхній частині сторінки розміщується аватар користувача, його ім'я та країна перебування. Нижче відображаються дані, пов'язані з настроєм користувача, що дозволяє іншим користувачам швидко орієнтуватися в соціальному статусі людини.

Окремими блоками представлені хобі та цілі користувача. Дані про них отримуються через асинхронні запити до API та кешуються для зменшення навантаження. Кожен блок містить картки з іконками, назвами й коротким описом вибраного інтересу або цілі (наприклад: «Sport», «Fashion», «Music»).

Редагування профілю реалізовано у вигляді окремої кнопки, яка відкриває користувачу меню налаштувань, де можна змінити свої хобі, цілі, мову спілкування, стать, електронну пошту та пароль.

Сторінка профілю в системі не лише відображає особисті дані, але й виконує важливу соціальну функцію, допомагаючи іншим користувачам знайомитися та знаходити спільні інтереси.

4.5 Сторінка друзів та механізм пошуку

Сторінка друзів у застосунку реалізована як окремий функціональний модуль, що дозволяє користувачеві переглядати поточний список друзів, шукати нових людей за інтересами та надсилати запити на додавання до списку контактів (див. рис. 4.3).

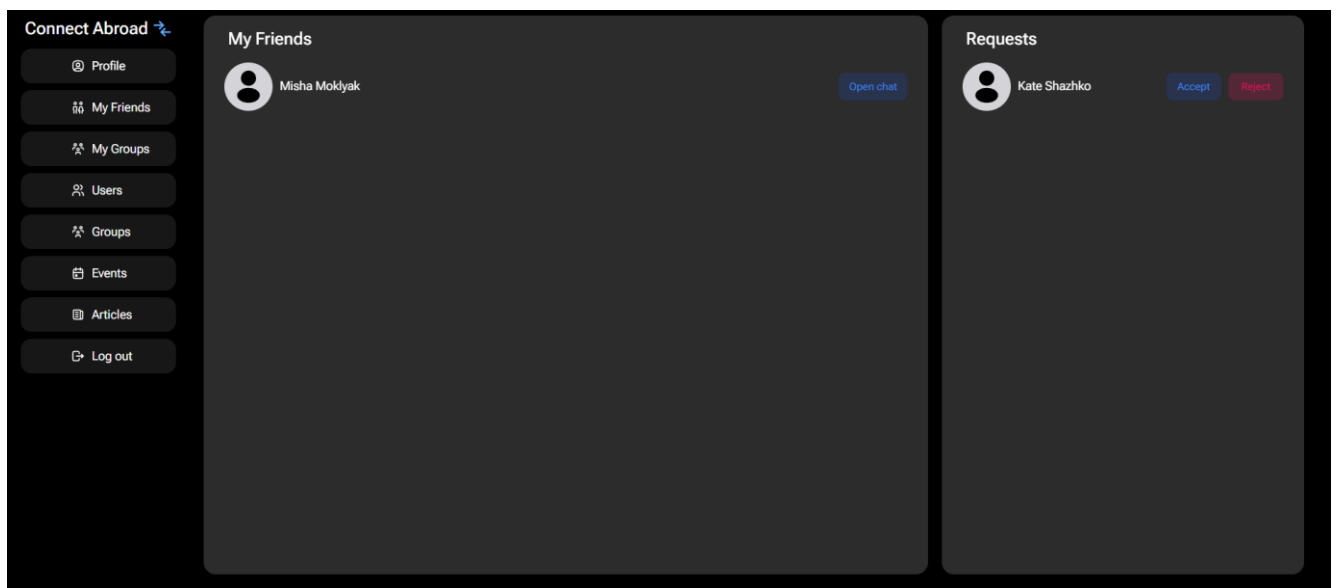


Рисунок 4.3 – Інтерфейс сторінки друзів (рисунок виконаний самостійно)

У секції «My Friends», яка показує тих користувачів, із якими вже встановлено зв'язок. Друзі відображаються у вигляді сітки або списку, залежно від розміру екрану. Кожна картка має кнопку для перегляду профілю або видалення з контактів.

Взаємодія з API реалізована через окремий модуль `user-friend.service.ts`, який забезпечує надсилання, прийняття або відхилення запитів у друзі. На

інтерфейсному рівні відображається поточний статус кожного користувача, що покращує прозорість взаємодії.

4.6 Групові чати та обмін повідомленнями

Функціональність групових чатів у програмній системі спрямована на створення умов для спілкування між користувачами, які мають спільні інтереси, цілі або перебувають в одному місті. Групові чати – це ключовий інструмент соціалізації всередині платформи (див. рис. 4.4).

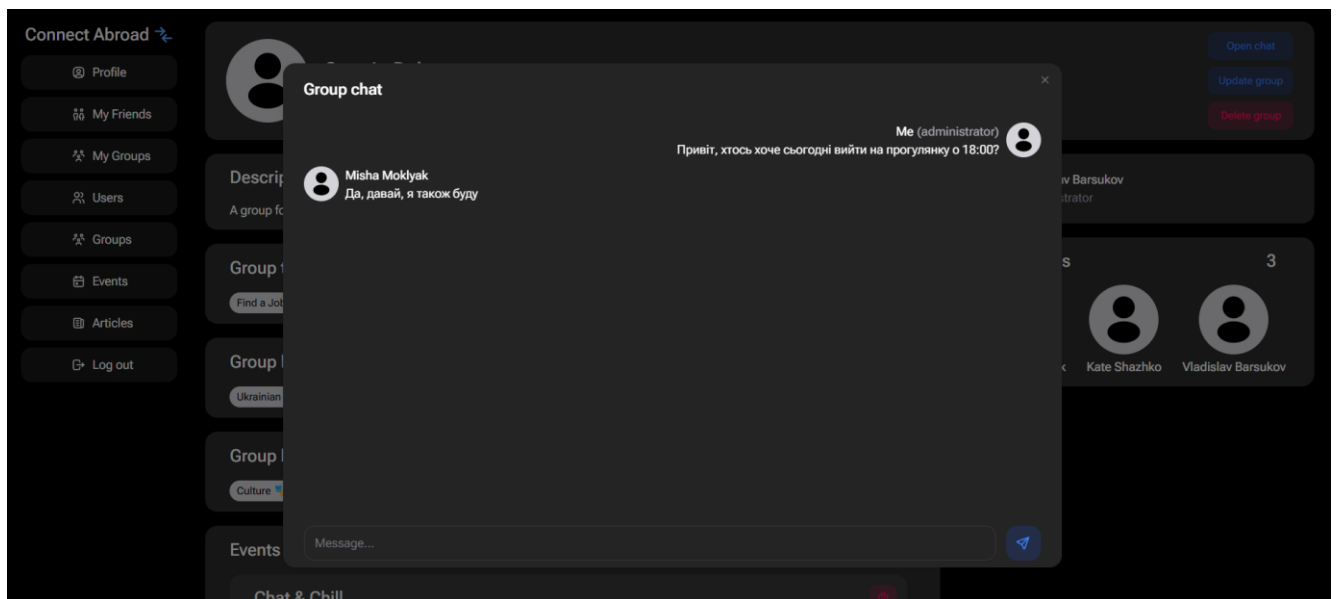


Рисунок 4.4 – Інтерфейс сторінки чату (рисунок виконаний самостійно)

Інтерфейс чатів побудований за класичним принципом месенджера. У лівій частині екрана користувач бачить список доступних груп, до яких він приєднаний. При виборі конкретної групи відкривається головне чат-вікно, де відображаються всі повідомлення, відправлені учасниками. Кожне повідомлення містить ім'я і текст.

У нижній частині розташоване поле введення з кнопкою надсилання. Повідомлення надсилаються асинхронно, оновлюються без перезавантаження сторінки, що створює відчуття «живого» спілкування.

Дизайн вікна чату виконано в мінімалістичному стилі: використані кольорові акценти для різних учасників, прокрутка повідомлень реалізована плавно, а нові повідомлення автоматично з'являються внизу при надходженні.

Отже, групові чати забезпечують неформальне спілкування, швидкий обмін досвідом і підтримку серед користувачів, які тільки адаптуються до нової країни чи середовища.

4.7 Система управління групами

У системі – групи виконують важливу роль у процесі соціалізації користувачів, вони дозволяють людям об'єднуватися за спільними інтересами, місцем проживання або цілями (див. рис. 4.5 та рис. 4.6).

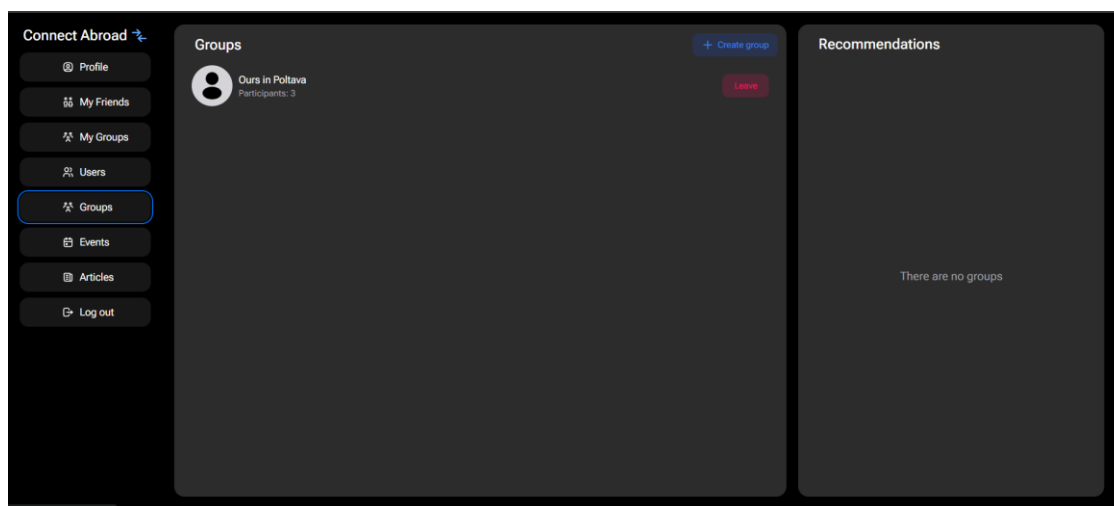


Рисунок 4.5 – Інтерфейс сторінки з групами (рисунок виконаний самостійно)

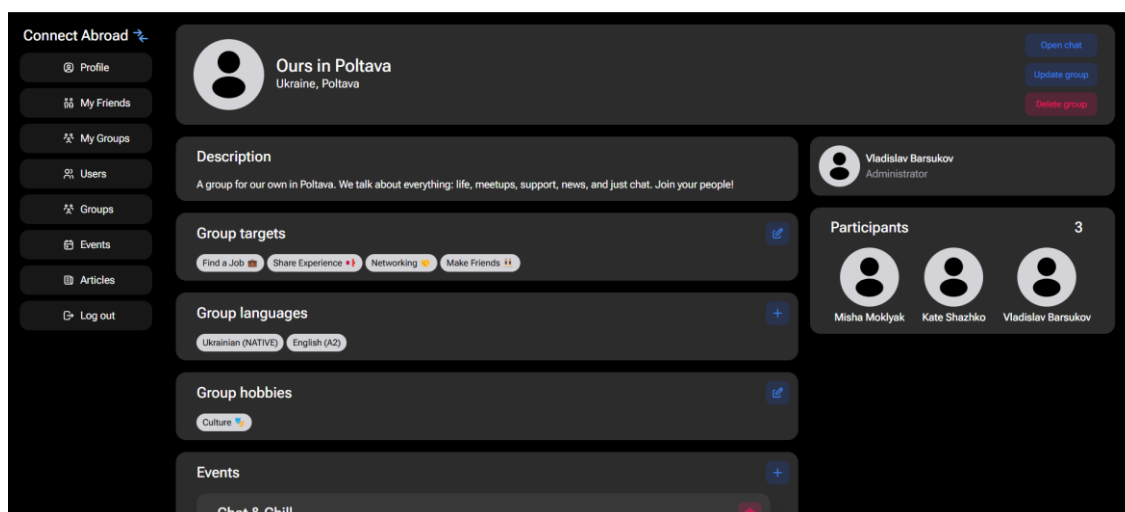


Рисунок 4.6 – Інтерфейс групи (рисунок виконаний самостійно)

Для реалізації цієї функціональності було впроваджено повноцінний модуль управління групами, який включає створення, редагування, підписку та перегляд інформації про групи.

Користувачі можуть:

- переглядати список усіх доступних груп;
- підписуватись або відписуватись від груп;
- створювати нові групи (модальне вікно для введення назви, опису, цілей і тематики);
- переглядати події, пов'язані з обраною групою.

При відкритті сторінки групи користувач бачить назву, опис, цілі групи, кількість учасників та список подій, які пов'язані з цією групою. Якщо користувач ще не є учасником, відображається кнопка про приєднання до групи. У разі підписки користувач отримує доступ до групових чатів і анонсів подій.

Функціонал підписки реалізується через окремі запити API: при натисканні кнопки система надсилає запит до відповідного ендпоінту (`/group/subscribe`) і оновлює локальний стан через TanStack Query.

Модуль управління групами дозволяє користувачам ефективно об'єднуватися у спільноти за інтересами, що значно полегшує процес адаптації після переїзду.

4.8 Сторінка користувачів і рекомендації

Однією з ключових функцій застосунку Connect Abroad є сторінка зі списком користувачів, яка дозволяє знаходити інших учасників платформи та швидко додавати їх до списку друзів (див. рис. 4.7). Ця функція реалізована у вигляді інтерактивного інтерфейсу з двома основними зонами: перелік користувачів і рекомендовані профілі.

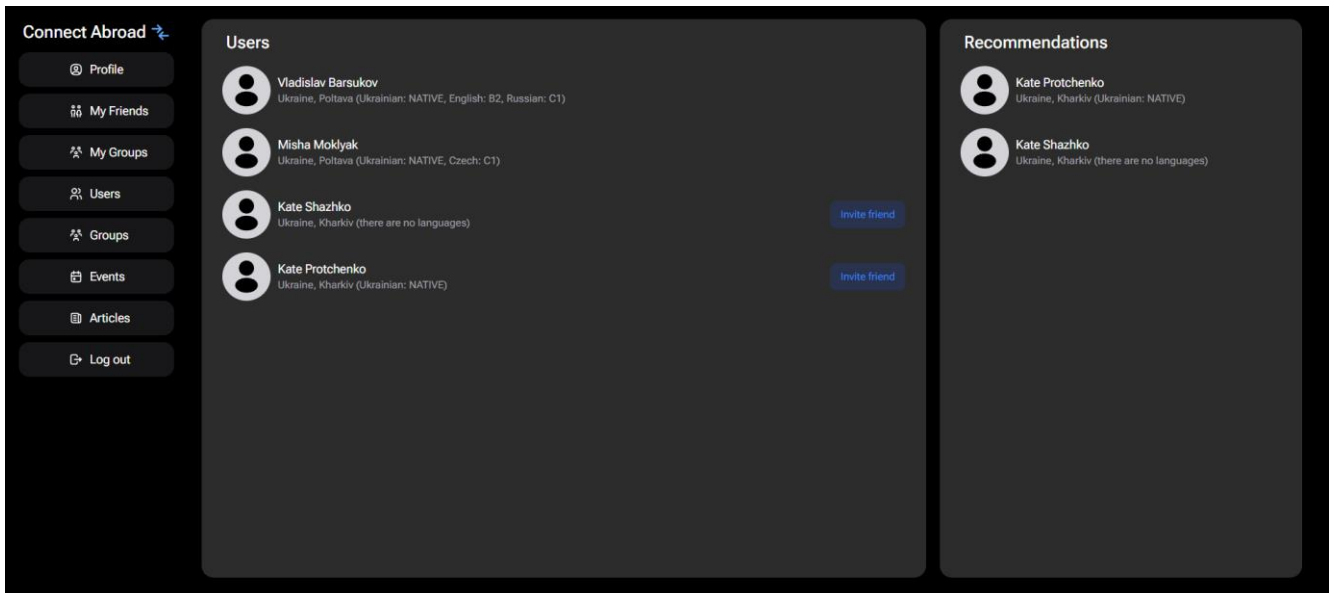


Рисунок 4.7 – Інтерфейс списку всіх користувачів (рисунок виконаний самостійно)

У центральній частині сторінки виводиться список усіх користувачів із базовою інформацією – ім'я, місцезнаходження, мови спілкування. Поруч із кожним користувачем розміщена кнопка «Invite friend», яка дозволяє швидко надіслати запит у друзі. При натисканні система виконує запит до API, і статус користувача оновлюється без перезавантаження сторінки.

У правій частині сторінки виводиться блок «Recommendations». У ньому автоматично пропонуються користувачі, з якими поточний користувач потенційно може мати спільні інтереси або мови спілкування. Такий підхід стимулює активне знайомство між учасниками платформи, особливо в перші дні після реєстрації.

Дизайн інтерфейсу побудований у мінімалістичному стилі – імена користувачів, країна та мови спілкування представлені стисло, щоб забезпечити зручне візуальне сприйняття.

Отже, ця сторінка користувачів виконує важливу роль для соціальної взаємодії, дозволяючи легко знайомитися з іншими учасниками та розширювати мережу контактів.

4.9 Організація та участь у подіях

Одним із важливих функціональних елементів системи є модуль подій, який дає можливість користувачам організовувати та брати участь у заходах, пов'язаних із соціалізацією, навчанням, культурними подіями або відпочинком. Завдяки цьому користувачі мають більше можливостей для реальної інтеграції в нове середовище.

На сторінці подій відображаються всі доступні заходи, які були створені адміністраторами груп або самими користувачами. Кожен захід містить основну інформацію: назву, опис, дату та місце проведення. Інтерфейс дозволяє швидко переглянути всі доступні події (див. рис. 4.8 та рис. 4.9).

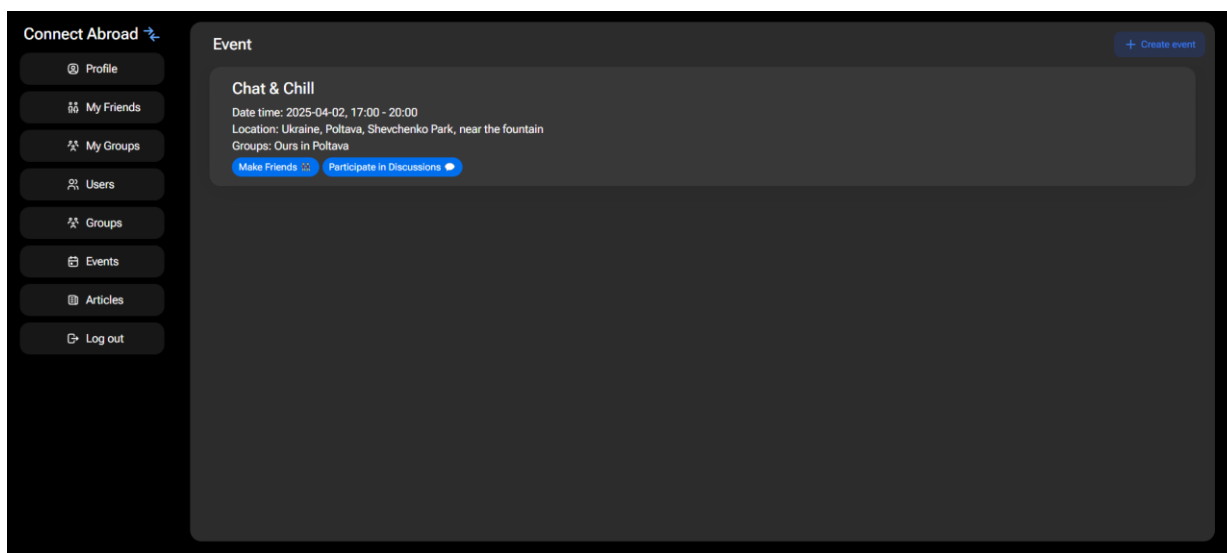


Рисунок 4.8 – Інтерфейс списку всіх подій (рисунок виконаний самостійно)

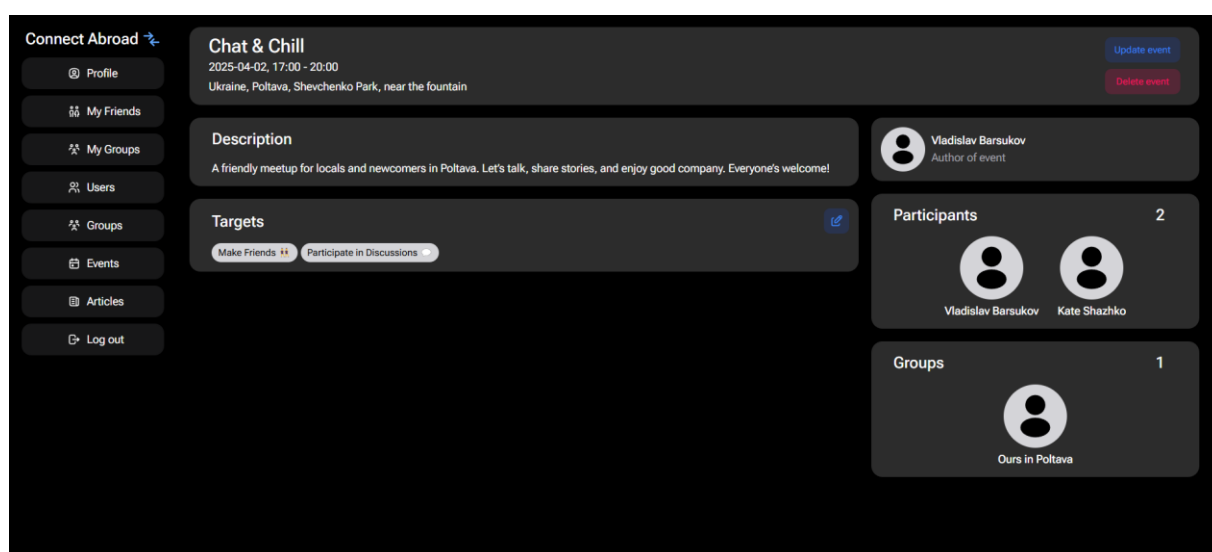


Рисунок 4.9 – Інтерфейс події (рисунок виконаний самостійно)

Користувач може обрати будь-яку подію, ознайомитись із її деталями та приєднатися до участі одним натисканням кнопки. Після реєстрації в події користувач отримує доступ до списку інших учасників, що сприяє попередньому знайомству перед заходом. Також реалізовано механізм відміни участі у випадку зміни планів.

Функціонал участі у подіях розширює можливості користувачів для живої взаємодії, знайомства з іншими людьми, обміну досвідом і побудови міцніших соціальних зв'язків у новій країні.

4.10 Статті та інформаційний розділ

Одним із додаткових, але важливих функціональних модулів програмної система для підтримки соціальної адаптації при переїзді є розділ статей. Цей розділ дозволяє користувачам створювати, переглядати та читати публікації, що містять практичні поради, інструкції, емоційну підтримку або інші матеріали, корисні для адаптації після переїзду в нову країну.

Головна сторінка розділу «Articles» містить перелік доступних статей (див. рис. 4.10). Кожна стаття представлена у вигляді інтерактивної картки, що включає заголовок, короткий опис, теги настрою та ім'я автора. Така структура дозволяє користувачам швидко зорієнтуватися в тематиці матеріалів і обрати актуальний для себе допис.

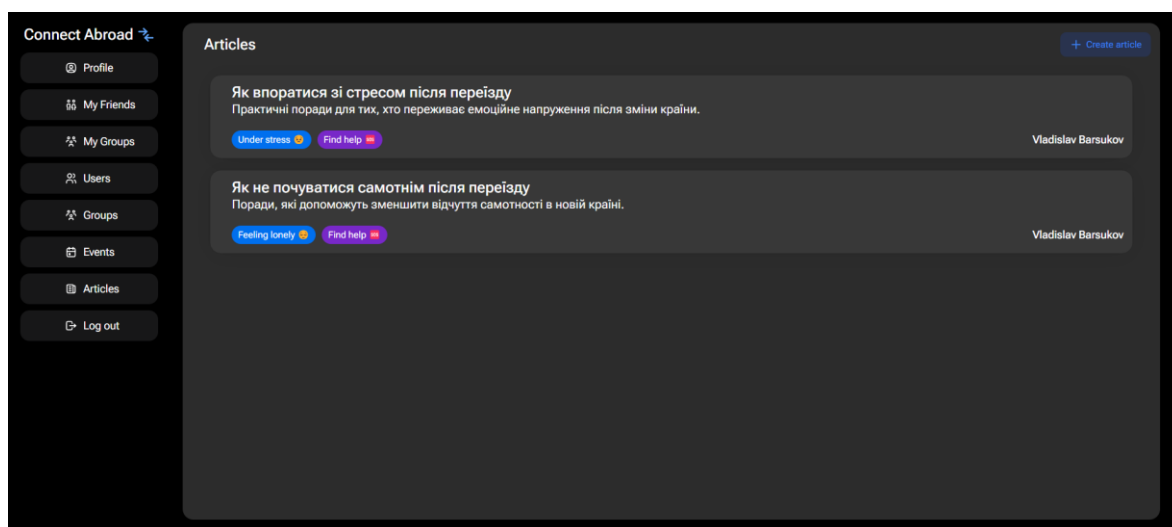


Рисунок 4.10 – Інтерфейс списку статей (рисунок виконаний самостійно)

Фрагмент показує, як список динамічно формується за допомогою `useGetArticlesQuery`, що отримує дані з API. Компонент `ArticleCard` візуалізує кожну публікацію.

При переході до певної статті відкривається повна версія тексту з розширеним описом, авторством, датою створення, повним змістом, а також кнопками для оновлення або видалення (див. рис. 4.11). У деяких статтях можливе також додавання мультимедійних матеріалів, наприклад, відео-інструкцій або посилань на ресурси.

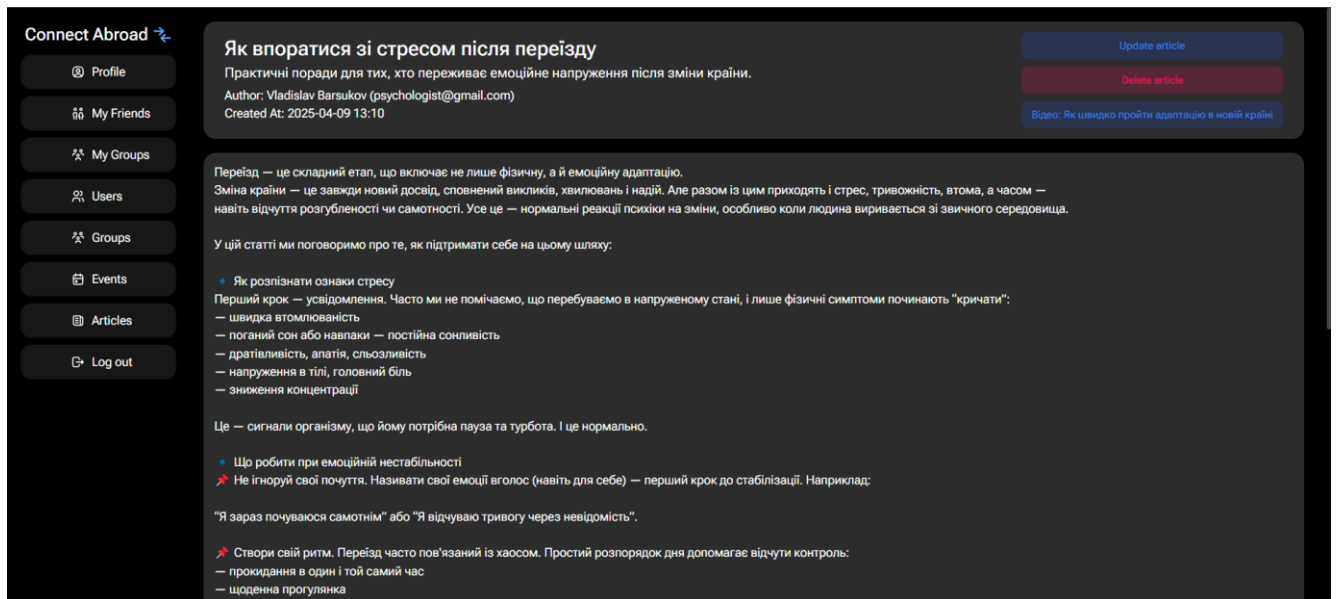


Рисунок 4.11 – Інтерфейс статті (рисунок виконаний самостійно)

Користувач може не лише читати статті, а й створювати власні через кнопку «Create Article». Інтерфейс створення реалізовано за принципом форми з полями для заголовка, опису, контенту та тегів настрою (див. рис. 4.12).

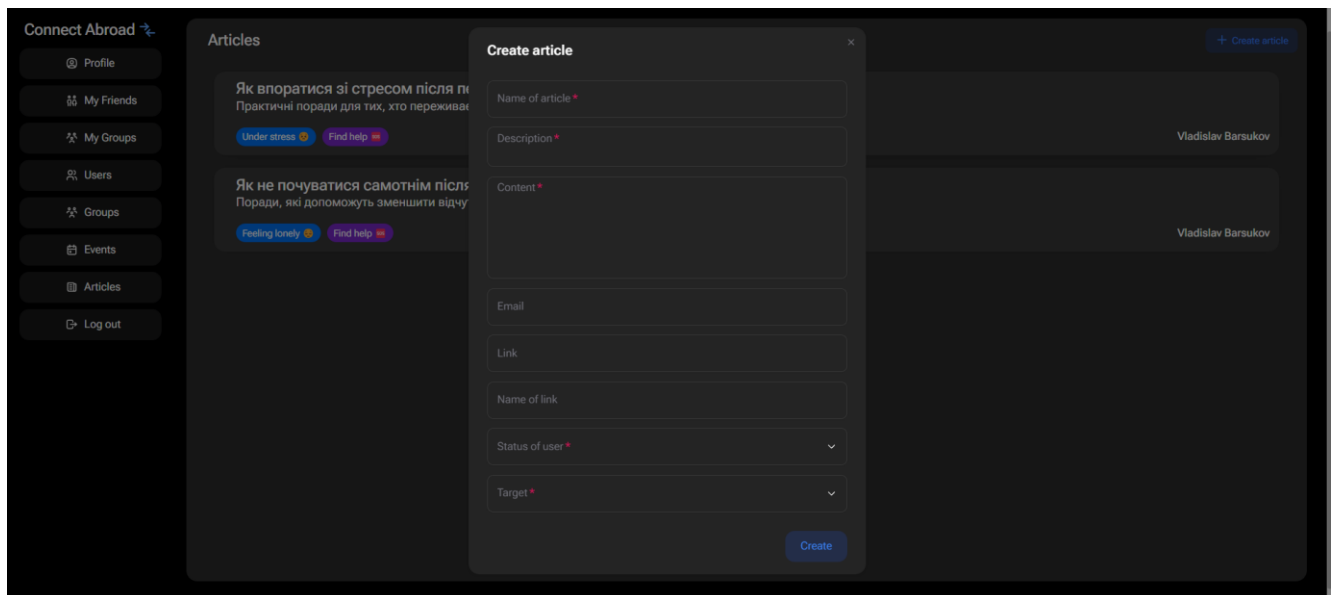


Рисунок 4.12 – Форма додавання нової статті (рисунок виконаний самостійно)

Модуль статей виконує роль інформаційного блоку платформи, сприяє обміну знаннями та підвищенню емоційної підтримки.

4.11 Управління станом і запити до API

У системі Connect Abroad важливим аспектом функціональності є взаємодія з API та збереження даних у локальному стані. Для зручного керування асинхронними запитами, кешування та оновлення даних у клієнтській частині застосунку використовується бібліотека TanStack Query, яка дозволяє автоматично опрацьовувати запити до серверу та уникати дублювання даних.

Більшість сторінок – зокрема профіль користувача, список друзів, повідомлення, групи – взаємодіють із сервером через окремі сервіси, розміщені в директорії `services/`. Ці сервіси реалізують функції надсилання HTTP-запитів за допомогою бібліотеки `Axios`.

Для кожного запиту використовується хук `useQuery` або `useMutation`, залежно від того, чи потрібно отримати, чи змінити дані. Наприклад, у компоненті профілю застосовується запит `useProfile`, який отримує дані поточного користувача з API та кешує їх на певний час. Якщо користувач переходить на іншу сторінку та

повертається назад, дані не запитуються повторно, що позитивно впливає на швидкодію.

Крім того, TanStack Query дозволяє автоматично оновлювати інформацію після успішного редагування – наприклад, після зміни хобі або статусу користувача. Оновлення стану відбувається без перезавантаження сторінки, що створює ефект живого інтерфейсу.

Таким чином, застосунок підтримує зручну й продуктивну роботу з даними, забезпечуючи швидкий відгук інтерфейсу та стабільну синхронізацію з сервером.

4.12 Аутентифікація та захист сесії

Програмна система передбачає наявність особистих облікових записів користувачів, тому одним із основних функціональних елементів є механізм аутентифікації. Авторизація здійснюється через електронну пошту та пароль, після чого користувач отримує доступ до свого профілю та функцій застосунку.

Інтерфейс входу реалізовано як стандартну форму з двома полями – email та пароль. Після успішної перевірки на стороні API `/auth/login`, користувач отримує JWT-токен, який зберігається у cookie з прапорцем `httpOnly`. Це забезпечує додатковий рівень захисту, запобігаючи доступу до токена зі сторінки через JavaScript.

Також реалізовано механізм оновлення сесії. У разі завершення строку дії токена, клієнт автоматично виконує запит до `/auth/refresh`, отримує новий токен і повторює попередню дію без втручання користувача.

Кнопка виходу у меню викликає функцію `logout()`, яка очищує cookie та перенаправляє користувача на головну сторінку. Додаткові заходи захисту, такі як прив'язка токена до IP-адреси або пристрою, на даному етапі не реалізовані, але можуть бути додані в майбутньому.

4.13 Логування та користувацькі повідомлення

У застосунку реалізовано базовий механізм інформування користувача про помилки або результати дій через toast-повідомлення, що відображаються на

екрані. Ці повідомлення активуються в разі неправильного запиту, помилки авторизації, успішного надсилання запиту чи змін у профілі (див. рис. 4.13 та рис. 4.14).

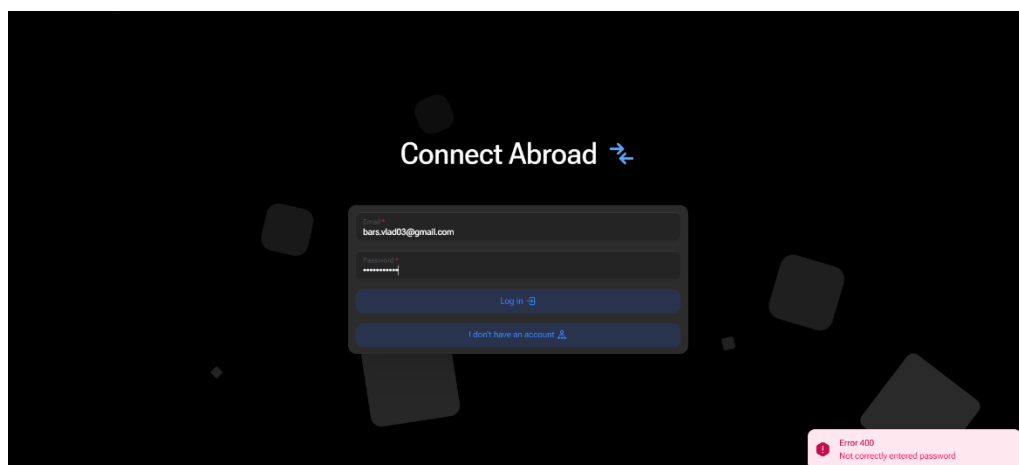


Рисунок 4.13 – Відображення повідомлення «Error» при помилці авторизації (рисунок виконаний самостійно)

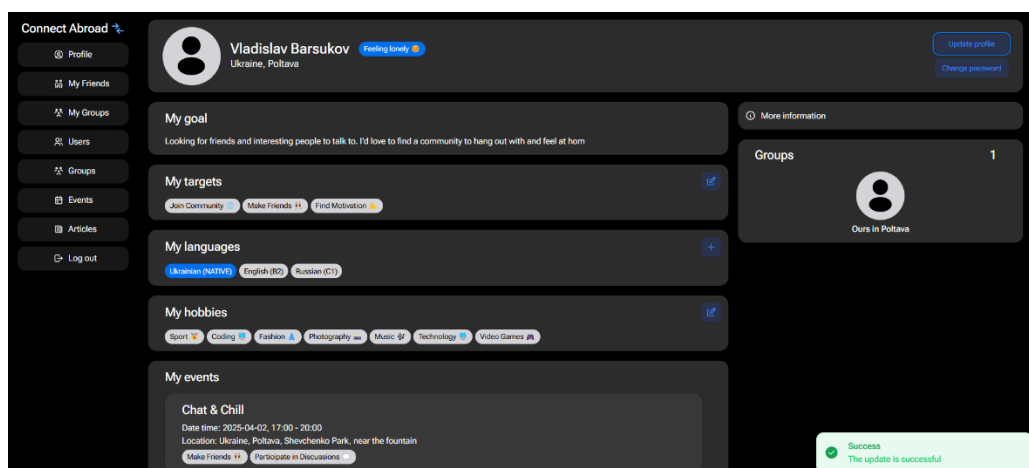


Рисунок 4.14 – Відображення повідомлення «Success» після успішних змін у профілі (рисунок виконаний самостійно)

Для реалізації цього функціоналу використовується утиліта з `alert.util.ts`, яка викликає повідомлення у відповідному стилі (успіх або помилка). Це дозволяє користувачеві швидко орієнтуватись у результаті своїх дій без потреби в перезавантаженні сторінки.

Отже, на поточному етапі система забезпечує мінімально достатній рівень реакції на події, а також має потенціал до впровадження повноцінного логування та аналітики.

5 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

5.1 Проведення тестування

У процесі розробки програмної системи для підтримки соціальної адаптації при переїзді в іншу країну Connect Abroad було проведено мануальне тестування, спрямоване на виявлення можливих помилок і перевірку коректності функціонування всіх реалізованих модулів. Метою тестування було забезпечення стабільної роботи програмної системи, перевірка її відповідності заявленим вимогам і забезпечення зручного користувацького досвіду.

На етапі первинного тестування особливу увагу було приділено механізму аутентифікації користувачів. Перевірялася робота форми логіну з різними наборами вхідних даних: правильними, неправильними та порожніми. У разі некоректних даних система повертала зрозуміле повідомлення про помилку без перезавантаження сторінки, що свідчило про правильну обробку виключних ситуацій. Після введення правильних даних користувач успішно отримував доступ до особистого кабінету.

Наступним етапом тестування стало тестування модуля профілю користувача. Після авторизації перевірялася коректність відображення персональної інформації. Була перевірена можливість редагування даних профілю – настрою, хобі, цілей та мови спілкування. Система коректно надсилала оновлені запити до API, і зміни відображалися в інтерфейсі без затримок та перезавантаження сторінки.

Особливу увагу було приділено тестуванню модуля друзів. Для цього було створено кілька тестових облікових записів. Перевірялися сценарії надсилання запитів у друзі, прийняття запитів та їх відхилення. Інтерфейс коректно відображав статус запитів і змінювався після взаємодії.

Модуль груп і підписок також пройшов всебічну перевірку. Була перевірена можливість перегляду доступних груп, підписки та відписки. У разі помилок, наприклад при спробі приєднатися до закритої групи, система коректно повідомляла користувача через toast-повідомлення.

У групових чатах тестувалися надсилання та отримання повідомлень у реальному часі та плавна прокрутка вікна повідомлень.

Модуль подій показав стабільну роботу. Користувачі могли переглядати заходи, реєструватися на них, бачити список учасників та відмовлятися від участі. Окрема увага приділялася перевірці коректної роботи при участі в кількох заходах одночасно.

Тестування модуля статей включало створення тестових публікацій із повними даними – назвою, описом та текстом. Статті коректно зберігалися та відображалися у списку. Також перевірялося редагування й видалення публікацій без помилок у роботі бази даних.

На завершення тестування було змодельовано ситуації спроби несанкціонованого доступу до захищених сторінок без авторизації, а також некоректні запити. В усіх випадках система демонструвала стабільну обробку помилок, перенаправлення на логін або виведення інформаційних повідомлень.

Таким чином, за результатами тестування встановлено, що програмна система Connect Abroad відповідає заявленим функціональним вимогам.

5.2 Виявлені недоліки в процесі тестування

Під час тестування застосунку було виявлено кілька дрібних недоліків, які не впливають критично на основну функціональність системи, однак врахування цих аспектів допоможе покращити користувацький досвід у майбутньому.

Одним із таких моментів стало обмежене інформування користувача у випадку втрати з'єднання з сервером. Доцільно в майбутньому впровадити глобальне сповіщення про втрату підключення.

Також виявлено відсутність обмеження на кількість символів у деяких формах редагування профілю. Це може вплинути на зовнішній вигляд інтерфейсу в разі надмірно довгих текстів. Рекомендується встановити максимальну кількість символів для таких полів, як «Ім'я», «Статус» та «Опис цілей».

Ще одним аспектом є необхідність автоматичного оновлення інтерфейсу після відправки запиту в друзі або підтвердження дружби. У деяких випадках

користувачеві доводилося вручну перезавантажувати сторінку для відображення змін.

Останнім важливим недоліком є відсутність підтвердження деяких критичних дій. Наприклад, при виході з групи або скасуванні участі в події дія виконувалася миттєво без запиту на підтвердження користувача. Для зручності та безпеки взаємодії доцільно додати діалогове вікно з підтвердженням перед виконанням незворотних змін.

Усі виявлені недоліки мають несуттєвий характер, не впливають на основні функції системи та можуть бути виправлені в межах наступних ітерацій розробки. В цілому результати тестування показали високу якість реалізації програмної системи та готовність продукту до подальшого розвитку й масштабування.

6 ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Після завершення розробки й тестування застосунку Connect Abroad на наступному етапі передбачено впровадження програмного забезпечення в реальні умови експлуатації. Процес впровадження заплановано здійснити поступово, з чітким поділом на окремі послідовні етапи, що гарантуватиме якісну інтеграцію розробленого програмного продукту у повсякденну практику користувачів.

Першим етапом стане вибір та налаштування хмарного середовища для розгортання застосунку. Передбачається використати платформу DigitalOcean або AWS, які надають широкі можливості масштабування, управління ресурсами та моніторингу системи. На цьому етапі буде налаштовано віртуальний сервер, на якому розгорнеться Docker-середовище, що дозволить легко керувати розгортанням клієнтської та серверної частини, а також бази даних MySQL. Використання контейнеризації забезпечить стабільність роботи застосунку та полегшить подальші оновлення й технічне обслуговування.

Наступним кроком стане налаштування доменного імені та захищеного SSL-з'єднання. Для цього заплановано реєстрацію домену, наприклад, connect-abroad.com, і налаштування DNS-записів для забезпечення швидкого доступу до сайту. Для безпеки передачі даних планується інтегрувати SSL-сертифікат, що буде отриманий через сервіс Let's Encrypt. Це забезпечить надійне шифрування всіх даних користувачів та захист їх персональної інформації від можливих зловмисних атак.

Після розгортання системи на сервері буде проведено етап внутрішнього (альфа) тестування. Для цього планується створити тестову групу користувачів, яка включатиме розробників та невелику кількість волонтерів (до 10 осіб). Цей етап дозволить провести ретельну перевірку основних функцій у реальному середовищі, виявити критичні помилки, що можуть виникнути лише при фактичному використанні, та оперативно їх виправити. Альфа-тестування буде зосереджено на перевірці реєстрації, авторизації, взаємодії у чатах, підписки на події та роботи з профілями.

Наступний запланований етап – це проведення відкритого бета-тестування, до якого планується залучити більшу кількість потенційних користувачів (50-100 осіб). Бета-тестування дозволить отримати широкий зворотний зв'язок про зручність інтерфейсу, актуальність запропонованих функцій і реальну користь застосунку для вирішення завдань соціальної адаптації. Для збору відгуків та пропозицій щодо покращення буде створено спеціальну форму зворотного зв'язку всередині застосунку, що дозволить врахувати всі побажання й зауваження користувачів перед фінальним релізом.

Після завершення бета-тестування буде здійснено аналіз отриманих даних та внесено необхідні корективи в інтерфейс та функціонал застосунку. Ці зміни будуть спрямовані на покращення користувацького досвіду: оптимізацію швидкодії, вдосконалення дизайну сторінок, уточнення текстових повідомлень про помилки й поліпшення механізмів обробки запитів до API.

Для чіткішого планування всіх перелічених вище етапів впровадження програмної системи було складено відповідний план-графік. Нижче, у таблиці 6.1, наведено детальний опис етапів із зазначенням точних дат їх початку та завершення.

Таблиця 6.1 – План впровадження програмної системи для підтримки соціальної адаптації при переїзді в іншу країну

Назва етапу	Дата початку	Дата завершення
Вибір і налаштування хмарного сервісу	15.05.2025	20.05.2025
Реєстрація домену та отримання SSL	21.05.2025	23.05.2025
Розгортання Docker-контейнерів на сервері	24.05.2025	28.05.2025
Внутрішнє альфа-тестування	29.05.2025	05.06.2025
Аналіз результатів альфа-тестування	06.06.2025	08.06.2025
Виправлення виявлених недоліків	08.06.2025	15.06.2025
Відкрите бета-тестування	16.06.2025	23.06.2025

Аналіз результатів бета-тестування	24.06.2025	27.07.2025
Фінальні покращення та оптимізація	28.06.2025	02.07.2025
Налаштування системи моніторингу	03.07.2025	05.07.2025
Офіційний запуск у «Production»	06.07.2025	06.07.2025
Створення служби підтримки користувачів	07.07.2025	10.07.2025

Після офіційного запуску системи передбачається постійне технічне супроводження й регулярні оновлення, спрямовані на підтримку високої якості роботи, актуалізацію даних та додавання нового функціоналу відповідно до потреб користувачів. Також планується створення системи підтримки, яка забезпечуватиме оперативну відповідь на запитання та вирішення проблем, з якими можуть зіштовхнутися користувачі.

Таким чином, запланований процес впровадження застосунку у майбутньому передбачає комплекс заходів, спрямованих на якісну інтеграцію системи у реальне користувацьке середовище та забезпечення її стабільної й ефективної роботи у довгостроковій перспективі.

ВИСНОВКИ

У процесі роботи було успішно розроблено програмну систему Connect Abroad, яка покликана вирішити актуальну проблему соціальної адаптації людей, які переїжджають до нової країни. Головною метою роботи стало створення ефективної та зручної веб-платформи, що надає користувачам можливості для швидкої інтеграції через пошук однодумців, спілкування, створення тематичних заходів і спільнот, а також отримання релевантної інформації.

Під час виконання було проведено детальний аналіз предметної галузі, який дозволив виявити основні недоліки існуючих аналогів, такі як недостатній рівень персоналізації, відсутність комплексного підходу та недостатня увага до психологічних аспектів адаптації. В результаті цього аналізу було сформульовано концепцію комплексного застосунку, який інтегрує функції соціальної мережі, системи рекомендацій, тематичних груп і заходів.

Використання сучасних технологій (NestJS, Next.js, Prisma ORM, MySQL, TypeScript) забезпечило високу стабільність і гнучкість створеного рішення. Розроблена система підтримує базові функції соціальної мережі, такі як реєстрація, авторизація, пошук друзів, створення подій, груп, статей та спілкування в чатах, а також надає персоналізовані рекомендації, які ґрунтуються на інтересах і потребах користувачів.

У процесі тестування система продемонструвала стабільну роботу та відповідність заявленим вимогам. Були виявлені лише незначні недоліки, які стосувалися переважно інтерфейсних рішень і відсутності деяких додаткових підтверджень для критичних дій. Ці проблеми не впливають на загальну працездатність та можуть бути швидко усунені під час наступних етапів розробки.

Наступним важливим кроком стане впровадження платформи в реальне середовище, що включатиме хмарне розгортання, доменну реєстрацію та SSL-захист, а також поетапне тестування. Проведення відкритого бета-тестування дозволить отримати реальний зворотний зв'язок від потенційних користувачів і внести додаткові покращення перед офіційним запуском.

Отже, створена програмна система відповідає поставленим цілям та має значний потенціал для вирішення соціальних завдань, пов'язаних з адаптацією мігрантів, а також демонструє високу якість реалізації та готовність до практичного використання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. United Nations DESA. (2024). Megatrends: International Migration. URL: https://www.un.org/development/desa/pd/sites/www.un.org.development.desa.pd/files/undesapd_2024_megatrends3-international_migration.pdf (дата звернення: 20.04.2025).
2. Situation Ukraine Refugee Situation. Operational Data Portal. URL: <https://data.unhcr.org/en/situations/ukraine> (дата звернення: 20.04.2025).
3. Meetup | Find Local Groups, Events, and Activities Near You. Meetup. URL: <https://www.meetup.com> (дата звернення: 20.04.2025).
4. Community for expatriates & global minds | InterNations. Community for expatriates & global minds | InterNations. URL: <https://www.internations.org> (дата звернення: 20.04.2025).
5. Expatica: The largest online resource for expats. Expatica. URL: <https://www.expatica.com> (дата звернення: 20.04.2025).
6. InterPals Penpals. InterPals. URL: <https://www.interpals.net> (дата звернення: 20.04.2025).
7. Couchsurfing. Stay with locals and meet travelers. URL: <https://www.couchsurfing.com> (дата звернення: 20.04.2025).
8. Community of Ukrainians in Glasgow / AUGB Glasgow. Facebook. URL: https://www.facebook.com/GlasgowUkrainians/?locale=ua_UA (дата звернення: 20.04.2025).
9. Telegram Web. Telegram Web. URL: <https://web.telegram.org> (дата звернення: 20.04.2025).
10. NestJS - A progressive Node.js framework. NestJS - A progressive Node.js framework. URL: <https://nestjs.com> (дата звернення: 20.04.2025).
11. Prisma | Simplify working and interacting with databases. Prisma. URL: <https://www.prisma.io> (дата звернення: 20.04.2025).
12. Next.js by Vercel - The React Framework. Next.js by Vercel - The React Framework. URL: <https://nextjs.org> (дата звернення: 20.04.2025).

13. Designing For Accessibility And Inclusion - Smashing Magazine. Smashing Magazine.

URL: <https://www.smashingmagazine.com/2018/04/designing-accessibility-inclusion/> (дата звернення: 20.04.2025).

14. Neusser T. UX Basics: Study Guide. Nielsen Norman Group. URL: <https://www.nngroup.com/articles/ux-basics-study-guide/> (дата звернення: 20.04.2025).

15. Material Design - Version 2. Material Design. URL: <https://material.io/design/color/dark-theme.html> (дата звернення: 20.04.2025).

16. GitHub - NureBarsukovVladyslav/2025_B_PI_PZPI-21-1_Barsukov_V_O. GitHub. URL: https://github.com/NureBarsukovVladyslav/2025_B_PI_PZPI-21-1_Barsukov_V_O (дата звернення: 28.05.2025).