

DOI: <https://doi.org/10.15276/ict.01.2024.06>

УДК 004.896

## Розробка інтерактивної платформи з використанням програмного інтерфейсу для сприяння соціальним зв'язкам

**Корніловська Наталя Володимирівна<sup>1)</sup>**

Канд. техніч. наук, доцент каф. Інформатики і комп'ютерних наук

ORCID: <https://orcid.org/0000-0002-8331-8027>; pypsiknata2015@gmail.com. Scopus Author ID: 57195531548

**Вишемирська Світлана Вікторівна<sup>1)</sup>**

Канд. техніч. наук, доцент каф. Інформатики і комп'ютерних наук

ORCID: <https://orcid.org/0000-0002-6343-7512> printvvs@gmail.com. Scopus Author ID: 57210284358

**Карамушка Марина Володимирівна<sup>1)</sup>**

Канд. техніч. наук, доцент каф. Інформатики і комп'ютерних наук

ORCID: <https://orcid.org/0000-0001-5982-4598>; karamushkamarina@gmail.com

**Січкарюк Руслан Костянтинович<sup>1)</sup>**

Магістр каф. Інформатики і комп'ютерних наук

ORCID: <https://orcid.org/0000-0001-8353-7679>; sichkaryk39@gmail.com. Scopus Author ID: 57195531548

<sup>1)</sup> Херсонський національний технічний університет, вул. Інститутська, 11. Хмельницький, 29016 Україна

### АНОТАЦІЯ

В епоху стрімкого розвитку технологій, коли людське спілкування сягнуло нових вершин, інформаційні платформи відіграють значну роль, трансформуючи багато аспектів життя. Вони започаткували нові способи організації стосунків між людьми, їх взаємодії та співпраці. За останні роки велика кількість факторів стала причиною низької соціалізації суспільства, що призвело до глобальної проблеми, коли соціальні навички змішуються з рекламними інтеграціями великих та малих компаній. Тому розробка комунікаційних платформ, які сприяють соціальній взаємодії, має великий потенціал для вирішення викликів сучасного життя та розширення нашого розуміння людської взаємодії в епоху цифрових технологій.

Виникає гостра потреба в більш етичному та відповідальному використанні інформаційних систем, щоб уникнути експлуатації людської взаємодії. Це потребує впровадження таких рішень, як комунікаційні платформи з відкритими інтерфейсами прикладного програмування (API), які сприяють прозорості та кращому контролю за соціальними взаємодіями.

Одним із прикладів, що показують глибокий вплив технологій на людські взаємини, є поширення медійних соціальних мереж. За винятком всі цифрові мережі все ще мають великі проблеми. Особливо широко поширені соціальні медіа-платформи надають перевагу профілям користувачів над реальними людськими відносинами, фактично призводячи до створення продукту соціальних взаємодій. Вони фіксують багато поверхневих взаємодій, розрізних від справжньої близькості та співпереживання. Так, цей розрив очевидний і є місцем для пошуку рішення подолання нагальних проблем та для забезпечення значущих змін у суспільстві.

Наша соціальна мережа надасть можливість подолання обмежень звичайних соціальних мереж, орієнтуючись на автентичність, емпатію та справжній контакт з людьми. У своїй суті така платформа прагне бути стимулом для створення важливих відносин, створювати затишок і здоровий спосіб життя та дбати про загальне добро. Філософія дизайну передбачає використання вже існуючих чи власних підходів в поєднанні з інтерактивними моментами та інтуїтивно зрозумілими інтерфейсами користувача для створення оптимального користувацького досвіду.

**Ключові слова:** інтерактивна платформа; інструменти шаблонизатора Jinja2; фреймворки; бази даних; інтерфейс прикладного програмування (API)

**Метою дослідження** є проектування та розробка інтерактивної платформи з використанням програмних інтерфейсів для покращення соціальної взаємодії в цифровому середовищі. Для досягнення цієї мети дослідження будуть зосереджені на визначенні потреб користувачів, створенні гнучкої взаємодії, використанні інтерактивних функцій і соціальній ефективності платформи в розвитку значущих взаємодій.

Об'єктом дослідження є інтерактивна платформа з використанням програмного інтерфейсу для сприяння соціальним зв'язкам. Об'єкт дослідження включає в себе всі аспекти цієї платформи, включаючи інтерфейс користувача, функціональні можливості, архітектуру програмного забезпечення, використані технології та інтерфейс прикладного програмування (API).

This is an open access article under the CC BY license (<https://creativecommons.org/licenses/by/4.0/deed.uk>)

Дослідження об'єкту передбачає аналіз різноманітних аспектів платформи, таких як її роль у сприянні соціальним зв'язкам, ефективність взаємодії з користувачами, можливості розширення функціоналу через API та інші ключові аспекти. Вивчення об'єкта дослідження передбачає аналіз і оцінку різних параметрів платформи з метою розробки та вдосконалення інтерактивного середовища, яке сприятиме покращенню соціальних зв'язків користувачів.

Для досягнення поставленої мети ми будемо використовувати такі сучасні інструменти, як інструменти візуалізації та проектування схеми бази даних, інструменти шаблонізатора Jinja2 для шаблонів, методи інтеграції інтерактивної платформи з базою даних та зовнішніми сервісами, методи інтеграції API та зовнішніх джерел.

Під час розробки проекту, ми умовно поділили його на 3 частини, які працюють як єдине ціле. Кожна з цих частин займає свою ступінь у формуванні проекту та має своє значення, яке не можна переоцінити. Першою з важливих частин є моделі котрі описують правила побудови, та вид кожного об'єкта у базі даних. Другою частиною розробка програмного забезпечення є створення вью, котрі надають можливість системі обробляти всі вхідні та вихідні запити користувачів. Дана частина є найважливішою, з усіх, та на своїх плечах тримає всю функціональність. Та остання але не менш важлива – це створення та обробка шаблонів, котрі виводять інформацію сформовану у базі даних на екран.

Почнемо з того, що жоден проект не може бути успішним без міцної архітектурної основи. Розумний спосіб створення програми закладає основу для її успіху, забезпечуючи масштабованість, зручність обслуговування та ефективність. Архітектурна схема нашого фреймворку формує основу комунікаційної платформи, описуючи її системні елементи, потоки даних та елементи інтерфейсу користувача.

У своїй основі наша архітектура включає добре відому структуру Model-View-Controller (MVC), яка є перевіреним методом для розділення проблем керування даними, логіки презентації та взаємодії з користувачем. На практиці це означає, що наша структура логічно розділена на три взаємопов'язані частини: управління даними та модель бізнес-логіки; представлення, яке контролює, як інформація подається користувачам; і контролер, який діє як посередник між моделлю та сценою, координуючи введення користувача та відповідь системи [1].

Наша діаграма архітектури служить візуальним представленням цих багаторівневих зв'язків, надаючи розуміння того, як ці компоненти взаємодіють і взаємодіють у системі. Використовуючи стрілки та з'єднувальні вузли, діаграма показує, як дані перетікають між моделлю, представленням і рівнями контролера. Крім того, він описує процеси, що використовуються для обробки вмісту, і направляє користувачів під час взаємодії з платформою.

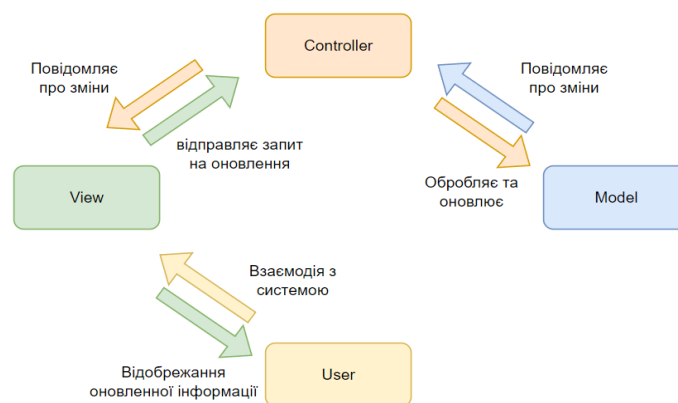


Рис. 1. Діаграма архітектури Model-View-Controller

Дотримуючись структури дизайну MVC, наша структура дизайну є модульною, розширюваною та зручною для обслуговування, що дозволяє нам легко проектувати та створювати платформу [2]. Ця архітектура не тільки покращує дизайн і читабельність коду, але також полегшує співпрацю між розробниками, оскільки кожен компонент є унікальною частиною функціональності програми. Зрештою, архітектура нашої системи закладає основу для надійної та масштабованої платформи, готової задовольнити потреби наших користувачів, що постійно розвиваються.

Схема бази даних на основі MySQL є наріжним каменем нашої системи, тобто зберігання даних, у яких є структура та порядок [3]. Він детально описує структуру таблиці бази даних та її зв'язки, які утворюють структуру для ефективного пошуку та обробки інформації. Базуючись на функціональних можливостях нашої платформи, ми розробляємо систему, яка підтримує наші цілі, узгоджену з ефективними методами збору, аналізу та обробки придатних для використання даних.

Створення наших планів бази даних відбувається за системним підходом, починаючи з детального аналізу функціональних можливостей нашої платформи. Ми розглядаємо взаємодію та зв'язки між ними та визначаємо основні функції, які складають нашу модель даних. Використовуючи MySQL Workbench або подібний інструмент, ми можемо перевести це в логічну схему, тобто таблиці, поля та межі, необхідні для точного представлення нашої моделі даних

Після завершення початкового проектування системи ми виконуємо подальшу оптимізацію та оптимізацію на основі покращеної продуктивності та масштабованості. Ми підтримуємо використання вікі-служб і служб MySQL для індексування, нормалізації та інших методів оптимізації для забезпечення доступу до даних і зберігання [4]. Ми гарантуємо точність і надійність наших систем шляхом ретельного тестування та перевірки, незалежно від будь-яких аномалій або потенційних проблем із збоями.

Нижче наведено графічну схему, яка показує структуру бази даних нашої нової платформи соціальної мережі.

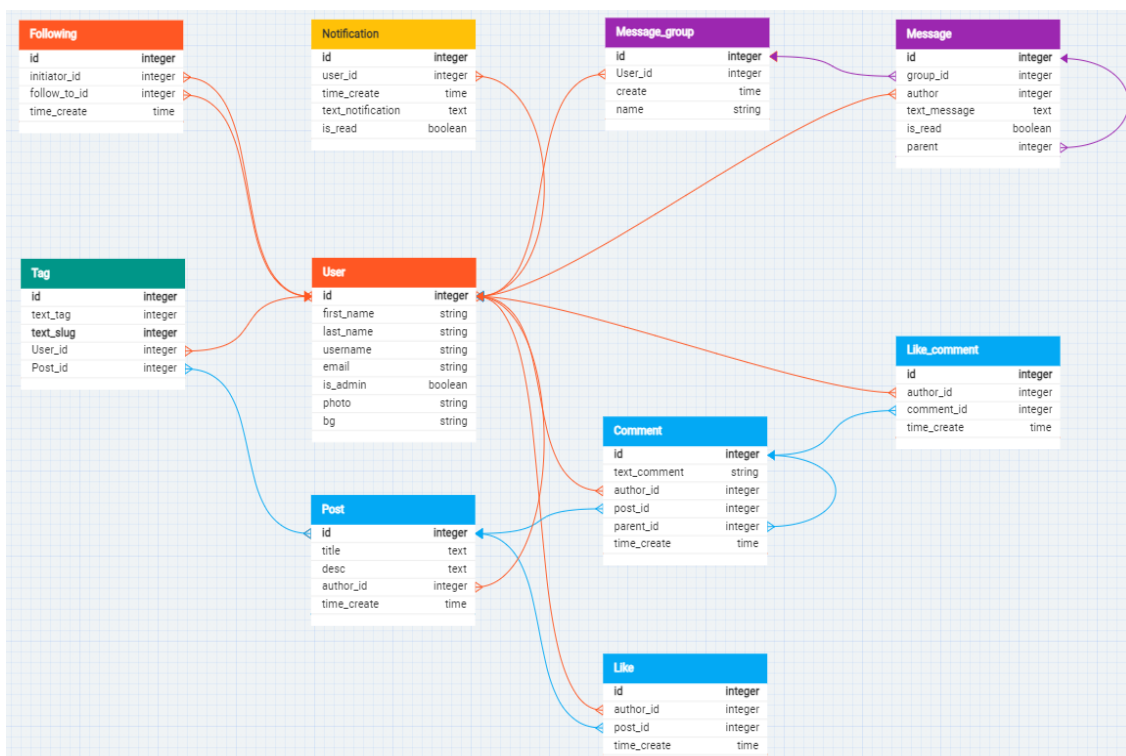


Рис. 2. Схема бази даних

Ця детальна схематична діаграма є чітким візуальним представленням моделі зв'язування сутностей, чітко показуючи взаємозв'язок таблиць, полів і зв'язків між різними сутностями в даних. За допомогою цього наглядного посібника, зацікавлені сторони отримують безцінну інформацію, що до базової структури бази даних, сприяючи глибокому розумінню складних зв'язків і взаємодії між об'єктами даних у системі.

На нашій платформі соціальних медіа є кілька базових прикладів, такі як користувачі, публікації, оцінки «лайки», коментарі, повідомлення, теги, тощо. Ці моделі представляють собою фундаментальні об'єкти нашої системи, інкапсулюючи основні функції, такі як профілі користувачів, опрацювання повідомлень та створення контенту.

Кожна модель, представлена на схемі, несе важливу роль у загальній роботі системи, та є невід'ємною її частиною.

Основними моделями є:

– Модель користувачі (User): представляє собою модель, в якій зберігаються користувачі системи, та їх данні. Це є основна модель, так як у наступних, менш значних моделях, екземпляр користувача використовується як зв'язуюче поле з іншими записами. Таким чином, це є найголовніша модель, навколо якої будується вся наступна інфраструктура нашого проекту.

– Модель пости (Post): представляє собою не менш важливою моделлю для частини системи пов'язаною зі створенням контенту у мережі. Вона є основною для низки інших моделей, та пов'язана з моделлю “User” полем “author\_id”.

– Модель коментарів (Comments): представляє собою данні, які застосовуються для опису коментарів під конкретним постом, є залежною моделлю від моделей Posts та User.

– Модель лайків коментарів (Like\_comment): призначена для зберігання та обробки лайків для коментарів, на основі цієї моделі формується кількість лайків, та список користувачів, які взаємодіяли з потенційним коментарем.

– Модель лайків (Likes): призначена для зберігання та обробки лайків для постів, на основі цієї моделі формується кількість лайків, та список користувачів, які взаємодіяли з потенційним компостом.

– Модель тегів (Tags): є дуже важливою моделлю для сортування та релевантного пошуку постів у системі, на її основі базується реалізація рекомендацій. Результатом даної моделі є легкий пошук, сортування, оцінка інтересів користувача та долучення користувачів до глибокого пізнання зацікавленого контенту.

– Модель підписників (Following): створена для зберігання та обробки взаємодії між користувачами у системі. Їх слідкування один за одним будуть відображатись у даній моделі, та у подальшому формувати список «друзів», у яким проводитиметься вибірка постів, з тегами, які цікаві підписникам. Тому дана таблиця є основоположником релевантного пошуку, та рекомендацій у системі, так як грає немаловажну роль у будіванні соціального аспекту нашої системи.

– Модель повідомлення (Message): Створена для обробки повідомлень у групах. Вона є основою переписки у системі, але залежною від груп. Додавши цю модель до нашого плану бази даних, ми створюємо основу нашої платформи соціальної мережі, яка дозволяє користувачам легко спілкуватися, спілкуватися та взаємодіяти один з одним. Ми прокладаємо шлях до переконливого та захоплюючого досвіду користувача, що сприяє успіху та зростанню нашої платформи.

Також важливим елементом нашої розробки є – інтерфейс прикладного програмування (API), який служить важливим каналом, який плавно долає інформаційний розрив між різним програмним забезпеченням і програмами. API служать архітектурними рамками, що

визначають методи та протоколи, в яких різні програмні компоненти гармонійно взаємодіють.

По суті, API слугують каналами співпраці, дозволяючи розробникам використовувати зовнішні пули ресурсів і можливостей для покращення функціональності та можливостей своїх додатків, інтегруючи платіжні шлюзи для безперервних транзакцій, підключаючи API соціальних мереж для додаткового залучення користувачів або в режимі реального часу - Використання оновлених API погоди дозволяє розробникам API створювати динамічні та працездатні програми, які резонують із користувачами та відповідають їхнім мінливим потребам [5].

Розглянемо можливість інтеграції API соціальних мереж, наприклад Facebook Graph API або Twitter API, у нашу програму. Увімкнувши вхід із соціальних мереж через ці API, користувачі можуть легко отримати доступ до нашої платформи за допомогою своїх наявних облікових записів у соціальних мережах, усуваючи потребу в окремому створенні облікового запису та облікових даних для входу. Це не тільки покращує взаємодію з користувачем, але й сприяє соціальній комунікації та взаємодії, сприяючи почуттю спільності та зв'язку на нашій платформі.

Модульний характер інтеграції API дає нам неперевершену гнучкість і гнучкість у розширенні та адаптації наших програм відповідно до мінливих потреб і вподобань наших користувачів, так що, коли нові платформи соціальних медіа випускаються або існуючі додають нові функції, ми легко інтегруємо додаткові API для підтримки цих платформ. Ми можемо покращити соціальні можливості програм.

Ретельне визначення архітектури системи, підкріплене розумним вибором мов програмування та фреймворків, служить основою, на якій зведено нашу інтерактивну платформу. Через призму інтеграції API та об'єднання зовнішніх джерел ми виходимо за межі нашої програми, використовуючи колективну потужність зовнішніх ресурсів для покращення взаємодії з користувачами та розширення горизонтів функціональності [6]. Симбіотичний зв'язок між архітектурним дизайном і технологічними рішеннями ще більше підкреслюється бездоганною інтеграцією інструментів розробки, таких як Visual Studio Code, що дає розробникам можливість створювати елегантні рішення з точністю та тонкістю.

Інтерактивна платформа має інтуїтивний та зрозумілий кожному інтерфейс, із-за можливості роботи на окремому сервері, має значну підтримку девайсів, так як дизайн є адаптивним під різні розміри пристроїв.

workshop\$Order Entity CRUD operations		
GET	/entities/workshop\$Order	Gets a list of entities: workshop\$Order
POST	/entities/workshop\$Order	Creates new entity: workshop\$Order
GET	/entities/workshop\$Order/{entityId}	Gets a single entity by identifier: workshop\$Order
PUT	/entities/workshop\$Order/{entityId}	Updates the entity: workshop\$Order
DELETE	/entities/workshop\$Order/{entityId}	Deletes the entity: workshop\$Order
GET	/entities/workshop\$Order/search	Find entities by filter conditions: workshop\$Order
POST	/entities/workshop\$Order/search	Find entities by filter conditions: workshop\$Order

Рис. 3. Приклад використання автодокументації інтерфейс прикладного програмування

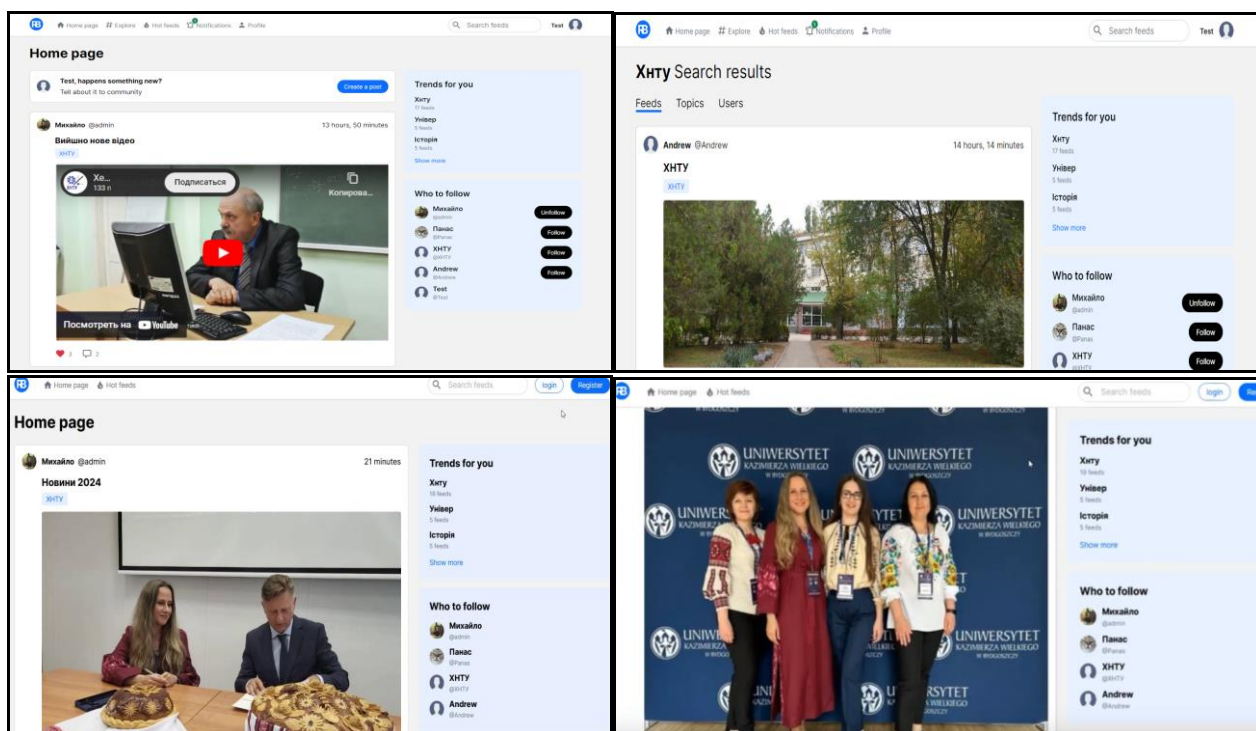


Рис.4. Інтерфейс розробленої інтерактивної платформи

Результати цього дослідження не лише сприятимуть розвитку інтерактивних платформ для спілкування та співпраці, але й стануть важливим внеском у подальше дослідження та розвиток соціальних технологій. Впровадження передових інструментів та розробка оптимальних практик у цій області сприятимуть покращенню якості віртуального спілкування та зближенню людей у цифровому просторі.

#### СПИСОК ЛІТЕРАТУРИ

1. Greenfeld D. R., Greenfeld A. R. “Two Scoops of Django: Best practices for Django 1.11”. 2017. p. 250–430.
2. Obe R. O., Hsu L. S. “PostgreSQL: Up and Running: A practical guide to the advanced open source database”. 2019. p. 105–306.
3. Chodorow K., Dirolf M. “MongoDB: The definitive guide”. 2018. p. 60–320.
4. Biehl M. “RESTful API Design: Best Practices in API Design with REST”. 2018. p. 52–295.
5. Martin R. C., Martin M. “Agile Software Development, Principles, Patterns, and Practices”. 2019. p. 125–360.
6. Martin R. C., Martin M. “Clean Code: A handbook of agile software craftsmanship”. 2008. p. 124–349.

DOI: <https://doi.org/10.15276/ict.01.2024.06>

UDC 004.896

## Development of an interactive platform using an Application Programming Interface to facilitate social connections

**Natalia V. Kornilovska<sup>1)</sup>**

PhD, Associate Professor, Department of Informatics and Computer Science

ORCID: <https://orcid.org/0000-0002-8331-8027>; pypsiknata2015@gmail.com. Scopus Author ID: 57195531548

**Svitlana V. Vyshemyrska<sup>1)</sup>**

PhD, Associate Professor, Department of Informatics and Computer Science

ORCID: <https://orcid.org/0000-0002-6343-7512>; printvvs@gmail.com. Scopus Author ID: 57210284358

**Marina V. Karamushka<sup>1)</sup>**

PhD, Associate Professor, Department of Informatics and Computer Science

ORCID: <https://orcid.org/0000-0001-5982-4598>; karamushkamarina@gmail.com

**Ruslan K. Sichkariuk<sup>1)</sup>**

Master, Department of Informatics and Computer Science

ORCID: <https://orcid.org/0000-0001-8353-7679>; sichkaryk39@gmail.com. Scopus Author ID: 57195531548

<sup>1)</sup> Kherson National Technical University, 11 Instytutska Street. Khmelnytskyi, 29016, Ukraine

### ABSTRACT

In an era of rapid technological advancement, where human communication has reached new heights, informational platforms play a significant role in transforming many aspects of life. They have introduced new ways of organizing relationships between people, their interactions, and collaboration. In recent years, numerous factors have led to a decline in socialization, contributing to a global issue where social skills are increasingly intertwined with advertising integrations of both large and small companies. Therefore, developing communication platforms that promote social interaction holds great potential for addressing the challenges of modern life and expanding our understanding of human interaction in the digital age.

There is an urgent need for a more ethical and responsible use of information systems to prevent the exploitation of human interaction. This requires the implementation of solutions such as communication platforms with open Application Programming Interfaces (APIs), which promote transparency and better control over social interactions.

One example demonstrating the deep impact of technology on human relationships is the widespread adoption of social media platforms. Despite their prevalence, all digital networks still face significant challenges. Particularly, popular social media platforms prioritize user profiles over real human relationships, effectively turning social interactions into a product. These platforms often capture superficial exchanges, disconnected from genuine closeness and empathy. This gap is evident and presents an opportunity to find solutions that address pressing issues and bring about meaningful changes in society.

Our social network will overcome the limitations of traditional social media by focusing on authenticity, empathy, and genuine human contact. At its core, this platform aims to be a catalyst for fostering meaningful relationships, creating a sense of comfort, promoting a healthy lifestyle, and caring for the common good. The design philosophy involves leveraging existing or proprietary approaches combined with interactive elements and intuitive user interfaces to create an optimal user experience.

**Keywords:** Interactive platform; templating tools Jinja2; frameworks; databases; Application Programming Interface (API)