

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Факультет комп'ютерних наук, фізики та математики
Кафедра комп'ютерних наук та програмної інженерії

Розробка та реалізація веб-додатку
“Сайт фандрайзингу Херсонського державного університету”

Кваліфікаційна робота (проект)
на здобуття ступеня вищої освіти «бакалавр»

Виконав: здобувач 4 курсу 12-441 групи
Спеціальність: 121 Інженерія програмного
забезпечення

Освітньо-професійна програма: Інженерія
програмного забезпечення

Дригваль Андрій Олексійович

Керівник: к. пед. н., доцент Вінник М.О

Рецензент: фріланс програміст Кльонон Д.М

Івано-Франківськ — 2025

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1. ПОНЯТТЯ ФАНДРАЙЗИНГУ	5
1.1. Огляд готових платформ для фандрайзингу	6
1.2. Методи інтеграції платіжних систем	8
1.3. Висновки щодо вибору оптимального підходу розробки	13
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА МОДЕЛЮВАННЯ МОДУЛІВ	
САЙТУ	15
2.1. Прототипування сторінок сайту	15
2.2. Архітектура веб-платформи	18
2.3. Функціональні модулі	19
РОЗДІЛ 3. ПРОГРАМУВАННЯ ТА ІНТЕГРАЦІЯ.....	20
3.1. Розробка фронтенду	20
3.2. Розробка бекенду	25
ВИСНОВКИ	35
ВИКОРИСТАНІ ДЖЕРЕЛА.....	36
ДОДАТКИ	39

Умовні позначення

1. **API** (Application Programming Interface): інтерфейс прикладного програмування, набір інструментів та протоколів для створення програмного забезпечення.
2. **PCI DSS**: стандарт безпеки даних індустрії платіжних карток.
3. **3D Secure**: технологія додаткової аутентифікації для забезпечення безпеки онлайн-платежів.
4. **ACS** (Access Control Server): сервер, що використовується в 3D Secure для аутентифікації власника картки.
5. **DS** (Directory Server): сервер, що використовується в 3D Secure для маршрутизації запитів між емітентом картки та продавцем.
6. **3DSS** (3D Secure Server): сервер, що використовується продавцем для ініціювання 3D Secure аутентифікації.
7. **Фронтенд (Frontend)** Відповідає за відображення інтерфейсу, взаємодію користувача з платформою та обробку введених даних перед передачею на сервер.
8. **Бекенд (Backend)** Обробляє запити від фронтенду, забезпечує авторизацію, логіку бізнес-процесів та обробку платежів.
9. **Сервер (Server)** Виконує роль середовища для розгортання бекенду, обробляє запити від клієнтів та керує потоками даних.
10. **Контейнеризація (Docker)** Використовується для ізоляції та управління середовищем виконання веб-додатку, спрощує розгортання.
11. **Блок-схема (FLOWCHART)** — Представлення алгоритму розв'язування або аналізу задачі за допомогою геометричних елементів, які позначають операції, потік, дані тощо.

ВСТУП

Данна робота присвячена розробці веб-додатку фандрайзингу університету, дослідженню реалізації подібного додатку, зокрема архітектури системи, інтеграції платіжних сервісів та створенню зручного механізму управління проєктами та транзакціями.

Актуальність теми: це створення такого рішення що дозволить підвищити рівень залученості не тільки студентської спільноти, але й партнерських компаній який спростить механізм підтримки університетських ініціатив та зробити процес донатів безпечним і зручним.

Мета цієї роботи – розробка веб-додатку для фандрайзингу університету, який буде включати у себе - зручний та безпечний механізм онлайн-платежів, представлення університету та його потреб для широкої аудиторії та адміністративну панель для керування новинами та прозорою звітністю навчального закладу.

Розробка сайту фандрайзингу для університету є необхідною через зростаючу потребу у додаткових джерелах фінансування освітніх та інфраструктурних ініціатив. На сьогодні більшість таких зборів відбувається без централізованої цифрової платформи, що ускладнює процес як для адміністрації, так і для потенційних донорів.

Запропонований сайт із вбудованим адміністративним модулем та інтеграцією платіжної системи надає можливість автоматизувати процес збору коштів, а також полегшити взаємодію з донорами, студентами й партнерами, завдяки швидкому піднесенню інформації та потреб. Такий інструмент не лише сприятиме розвитку університету, а й зробить його більш конкурентоспроможним у сфері залучення ресурсів і підтримки освітніх проєктів.

Досягнення мети роботи можливе через реалізацію такого плану:

1. **Аналіз існуючих рішень** у сфері університетського фандрайзингу та онлайн-платежів.

2. **Розробка архітектури сайту**, включаючи функціонал адміністративного та платіжного модулів.
3. **Реалізація зручного та безпечного механізму платежів** через інтеграцію сучасних платіжних систем.
4. **Створення інтуїтивного інтерфейсу** для користувачів (донатерів, студентів, адміністраторів).
5. **Тестування та оптимізація системи** для забезпечення безперебійної роботи та високої продуктивності.

РОЗДІЛ 1. ПОНЯТТЯ ФАНДРАЙЗИНГУ

Сучасні університети все частіше стикаються з проблемою фінансування важливих студентських ініціатив, наукових проєктів та соціальних програм. Звичайні методи збору коштів такі як благодійні заходи, кампанії зі збору на ту чи іншу ціль не дають великого впливу на аудиторію, від чого спроби зібрати суму на конкретний проєкт стають все важчими. [6] Все дедалі більше стає популярним інструмент фандрайзинг збору коштів залучаючи великі маси людей до себе.

Фандрайзинг — ефективний інструмент залучення коштів через благодійні внески, гранти та партнерські програми. [7] Впровадження цифрової платформи для збору коштів спрощує процес донатів, підвищує прозорість фінансових операцій та залучає ширше коло благодійників.

Основними моделями фандрайзингу в університетах є:

- 1) Благодійні внески випускників та спонсорів – найпоширеніша форма фінансування.
- 2) Грантові програми – отримання коштів від міжнародних організацій.
- 3) Краудфандинг – залучення широкої аудиторії через онлайн-платформи.
- 4) Корпоративне партнерство – підтримка з боку бізнесу та компаній.[7-6]

У сфері залучення коштів до університетських програм більший відсоток займають саме партнерські програми. Через велику користь у

пожертві та співпраці з навчальними закладами, компанія-благодійник може отримати не тільки позитивний відгук, але й велику рекламу для себе і свого бізнесу.[17]

З розвитком цифрових технологій університети починають переходити на онлайн-платформи для збору коштів, що дозволяє значно розширити аудиторію донаторів, автоматизувати процеси та забезпечити зручність для всіх учасників.[8]

Саме тому розробка спеціалізованого веб-додатку, який дозволить адмініструвати проєкти, інтегрувати платіжну систему та забезпечити прозорість фінансування звітністю, є перспективним напрямком дослідження.

1.1. Огляд готових платформ для фандрайзингу

Університетські платформи фандрайзингу — це спеціалізовані веб-сайти, створені з метою залучення благодійних внесків від випускників, студентів, співробітників, меценатів та широкого загалу. Такі платформи дозволяють підтримувати наукові проєкти, інфраструктурні ініціативи, стипендіальні програми та інші освітні цілі.[18] У різних країнах і регіонах ці сервіси мають свої особливості реалізації, які залежать від культури благодійності, технічних рішень, законодавства та традицій взаємодії між університетом і громадськістю.

З метою виявлення найкращих практик та визначення функціональних і концептуальних особливостей, було здійснено огляд університетських фандрайзингових сайтів, розподілених за географічним принципом — по континентах. Такий підхід дозволяє порівняти платформи між собою, зважити регіональні підходи до реалізації механізмів збору коштів, а також визначити загальні тенденції в архітектурі, дизайні та користувацькому досвіді.

Під час аналізу розглядалися такі критерії:

- 1) Наявність прозорого механізму пожертв.
- 2) Зручність користування для відвідувачів сайту.

- 3) Інтуїтивно зрозумілий дизайн.
- 4) Функціональність адміністративної частини.
- 5) Інтеграція платіжних систем.
- 6) Доступність звітів та зворотного зв'язку.
- 7) Наявність підтримки багатомовності або орієнтація на локальну/глобальну аудиторію.

У Додатку А представлено аналіз фандрайзингових сайтів університетів з Америки, Європи, Азії та України. Кожна платформа охарактеризована з точки зору її переваг, недоліків, дизайну, загальної структури та можливості подальшого масштабування або адаптації до локального освітнього середовища.

Візьмемо для прикладу порівняння декількох університетів, та їх готових рішень фандрайзингу з Додатку А, та наведемо ці порівняння у Таблиці-1.1.

Таблиця 1.1.

Аналіз декількох університетів

Регіон	Назва університету	Місце розташування	Структура сайту	Переваги	Недоліки
Азія	Сеульський національний університет	Сеул, Південна Корея	Лаконічна, розділена на кілька підрозділів (кампанії, проекти, контактна інформація) Навігація проста з чіткими розділами.	Простота <u>навігації</u> , <u>зрозуміла</u> структура <u>кампаній</u> .	<u>Відсутність інтерактивності</u> та <u>обмежена візуальна привабливість</u> .
Америка	<u>Harvard Alumni</u>	124 Mt Auburn St, Cambridge, MA 02138, США	<u>Звичайний дизайн</u> Головна сторінка <u>містить</u> напрямки <u>пожертв</u> . <u>Сторінка пожертв</u> виконана у вигляді форми до заповнення	<u>Поділ на окремі філії</u> для <u>пожертвування</u> При <u>авторизації</u> форма заповнюється автоматично Є <u>служба підтримки</u> у вигляді <u>пошти</u> та номера телефону	• <u>Немає конкретних напрямків для пожертв</u> • <u>Неможливе відслідкування на що саме підуть кошти і скільки</u>
Україна	Маріупольський державний університет	<u>Вулиця Преображенська, 6/4, Київ, 02000, Україна</u>	Сайт <u>має просту</u> структуру. В <u>головній сторінці</u> є <u>заклик до дій</u> , <u>пожертвування</u> , <u>контакти</u> , <u>прєкти</u> та <u>відео заклик</u> .	1. <u>Простий але привабливий</u> дизайн 2. <u>Відео заклик</u>	Невеличка структура сайту. Одна сторінка
Європа	<u>Оксфордський університет</u>	"Wellington Square, Oxford OX1 2JD, Велика Британія"	"Сайт складається з Головної сторінки, та розділів Про (About), Історії впливу (Impact Stories), Новини (News), Пожертви (Donate), Спадщина (Legacies), Вакансії (Jobs), Контакти (Contact)."	1. Сайт <u>повністю присвячений</u> фандрайзингу; 2. <u>Містить зрозумілий інтерфейс</u>; 3. <u>Добре структурований</u>; 4. Сайт <u>не перенавантажений</u> інформацією; 5. <u>Звітування про залучені кошти</u>;	Дизайн сайту верхньої частини головної сторінки, виконана у <u>поєнанні кольорів, яке змушує користувачів напружувати зір</u> .

За таблицею зазначеною вище, аналіз платформ фандрайзингу провідних університетів світу демонструє суттєві відмінності в підходах

до організації процесу збору коштів, які переважно залежать від рівня розвитку цифрової інфраструктури та фінансових можливостей.

Зокрема, Гарвардський та Оксфордський університети мають високофункціональні платформи з продуманим дизайном, широкими можливостями вибору фондів, напрямків пожертви та зручними платіжними інтерфейсами. [Додаток А, «Аналіз регіону Америка», «Аналіз регіону Англія»]

У свою чергу, Сеульський національний університет також реалізує фандрайзингову платформу на високому технічному рівні, з підтримкою кількох мов, мобільною адаптацією та зручним UX/UI. Основний акцент зроблено на розвиток міжнародної участі та корпоративної підтримки, що відображає глобальну стратегію розвитку університету. [Додаток А «Аналіз регіону Азія»]

Загалом заведеною таблицею порівнянь декількох світових університетів, підкреслюється актуальність розробки власного цифрового рішення, адаптованого до потреб закладу вищої освіти, з урахуванням найкращих світових практик.

1.2. Методи інтеграції платіжних систем

Онлайн-платіжні системи відіграють ключову роль у забезпеченні зручності та безпеки фінансових операцій, дозволяючи користувачам здійснювати платежі через інтернет, оплачувати товари та послуги.[12] У зв'язку з цим, компанії та підприємці активно інтегрують платіжні системи у свої бізнес-процеси для автоматизації фінансових операцій та підвищення рівня обслуговування клієнтів.

В Україні діє низка платіжних сервісів, які пропонують API (інтерфейс прикладного програмування) для взаємодії з іншими системами. Серед них варто виокремити LiqPay, Portmone, EasyPay та ПРОСТІР, які є найбільш популярними серед підприємців, інтернет-магазинів, фінансових установ та державних сервісів. Тому слід

розглянути та порівняти кожен з них, щоб обрати найбільш підходящий до задачі.

Однією з ключових вимог до розробленого веб-додатку є забезпечення високого рівня безпеки платіжних операцій. З огляду на те, що обробка фінансової інформації та персональних даних користувачів є чутливою сферою, прийнято рішення про інтеграцію із зовнішнім сертифікованим платіжним сервером.

Такий підхід дозволяє:

- Виключити зберігання платіжної інформації (включно з реквізитами банківських карток) на стороні серверу університету.
- Дотримуватись вимог сучасних стандартів безпеки, зокрема PCI DSS (Payment Card Industry Data Security Standard).
- Передавати дані за захищеним з'єднанням з використанням HTTPS та TLS.
- Забезпечити відповідальність за обробку платежів уповноваженій фінансовій установі або платіжному сервісу.

Таким чином, реалізовано принцип делегованої безпеки: сервер університету не обробляє та не зберігає фінансові дані, що зменшує ризики компрометації і покращує загальну безпеку платформи.

Інформація про системи

1. **LiqPay:** — це платіжна система, розроблена ПриватБанком, яка забезпечує можливість швидкої та безпечної інтеграції платежів на сайти та в мобільні додатки. Підтримує платежі з карток Visa, Mastercard, Google Pay та Apple Pay. [15]
2. **Portmone:** — одна з найстаріших платіжних систем України, яка підтримує платежі з карток, мобільні платежі та інтернет-банкінг. [14]
3. **EasyPay** — українська платіжна система, яка підтримує оплату через термінали, мобільні додатки та сайти. Підходить для малого та середнього бізнесу. [16]

4. **ПРОСТІР**: Національна платіжна система, яка створена для безпечних розрахунків в Україні, незалежно від міжнародних компаній. [13]

Аналіз платіжних систем

LiqPay: [15]

Переваги:

- 1) Простий та зручний API.
- 2) Високий рівень безпеки завдяки двохфакторній аутентифікації.
- 3) Широка підтримка платіжних методів.
- 4) Підтримка платежів у різних валютах.

Недоліки:

- 1) Залежність від ПриватБанку (проблеми для користувачів інших банків).
- 2) Можливі обмеження для іноземних клієнтів.

Безпека:

- 1) Відповідає стандарту **PCI DSS**.
- 2) Підтримує технологію **3D Secure**.

Portmone: [14]

Переваги:

- 1) Надійність і довготривала присутність на ринку.
- 2) Підтримка регулярних платежів.
- 3) Доступні тарифи для бізнесу.
- 4) Легка інтеграція в e-commerce системи.

Недоліки:

- 1) Комісія може бути вищою, ніж у конкурентів.
- 2) Менша популярність серед малого бізнесу порівняно з LiqPay.
- 3) Обмежена підтримка міжнародних платежів.

Безпека:

- 1) Відповідає стандарту **PCI DSS**.
- 2) Підтримує **3D Secure**.

EasyPay: [16]**Переваги:**

- 1) Велика кількість платіжних методів, включаючи термінали.
- 2) Низькі комісії для малого бізнесу.
- 3) Підтримка великої кількості сервісів для платежів (комунальні послуги, податки, мобільний зв'язок тощо).

Недоліки:

- 1) Менш гнучкий API порівняно з LiqPay та Portmone.
- 2) Інколи можливі затримки з проведенням платежів.
- 3) Обмежений міжнародний функціонал.

Безпека:

- 1) Відповідає стандарту **PCI DSS**.
- 2) Підтримує **3D Secure**.

ПРОСТІР: [13]**Переваги:**

- 1) Високий рівень безпеки та захист даних.
- 2) Незалежність від міжнародних систем Visa/Mastercard.
- 3) Державна підтримка та регулювання.

Недоліки:

- 1) Обмежена інтеграція з міжнародними платіжними системами.
- 2) Менша популярність серед малого бізнесу.
- 3) Вужча аудиторія користувачів.

Безпека:

- 1) Відповідає стандарту **PCI DSS**.
- 2) Використовує національні механізми безпеки.

Для додаткового порівняння долі використання платіжних систем на ринку, на рисунку діаграми представлено кількість транзакцій у кожній з платіжних систем.



Рис. 1.1. Діаграма кількості транзакцій.

Представлена інформація на (Рис.1.1.) щодо кількості транзакцій та обсягів коштів, що пройшли через зазначені платіжні системи в Україні:

- 1) EasyPay- У 2024 році користувачі здійснили понад 140 мільйонів платежів через EasyPay [11]
- 2) LiqPay - Кількість транзакцій: У 2023 році через LiqPay було здійснено понад 108 мільйонів транзакцій.[9]
- 3) Portmone - За останній рік через Portmone було опрацьовано близько 155 мільйонів транзакцій.[10]

Аналіз переваг і недоліків обраних систем

- 1) **Зручність інтеграції:** LiqPay має найбільш простий і зрозумілий API для інтеграції, тоді як EasyPay дещо поступається в гнучкості. Portmone також має хороший API, але може вимагати більше налаштувань.
- 2) **Підтримка платіжних методів:** LiqPay та Portmone підтримують більшість популярних методів.

- 3) **Комісії:** EasyPay пропонує найбільш вигідні умови для малого бізнесу, в той час як Portmone має відносно високі комісії.
- 4) **Безпека:** Усі платіжні системи відповідають стандарту PCI DSS, але ПРОСТІР виділяється завдяки державній підтримці та незалежності від міжнародних систем.
- 5) **Міжнародні платежі:** LiqPay та Portmone краще підходять для роботи з іноземними клієнтами, тоді як ПРОСТІР та EasyPay більше орієнтовані на внутрішній ринок.

1.3. Висновки щодо вибору оптимального підходу розробки

На основі проведеного раніше аналізу платіжних систем, вибір пав саме на LiqPay через легшу інтеграцію API на сайт для проведення транзакцій та розробки власного платіжного модулю на базі цієї системи. Також через доволі низьку комісію та широку підтримку платіжних методів – наприклад для переказів з-за кордону, найкращім варіантом буде використати LiqPay.

Розробка фандрайзингового веб-додатку потребує системного підходу, що враховує специфіку даної сфери. На основі аналізу існуючих рішень та доступних платіжних систем можна визначити оптимальний підхід до створення ефективного та надійного рішення.

Ключовим етапом було дослідження та аналіз конкурентних платформ. Аналіз показав, що важливими факторами є:

- 1) **Інтуїтивно зрозумілий інтерфейс** – простота у користуванні збільшує залученість користувачів і сприяє зростанню обсягів пожертв.
- 2) **Прозорість фінансових операцій** – надання користувачам детальної інформації про збори та витрати підвищує довіру до платформи.
- 3) **Гнучкість та масштабованість** – можливість адаптації системи до змінних потреб (наприклад, підтримка додаткових валют, інтеграція нових платіжних методів).

Другим важливим аспектом є вибір платіжної системи, яка буде інтегрована у веб-додаток. Оптимальний підхід передбачає врахування таких параметрів:

- 1) **Сумісність із технологічним стеком** – платіжний сервіс повинен мати добре документоване API, що забезпечить швидку та безпроблемну інтеграцію.
- 2) **Мінімізація комісійних витрат** – обрання системи з оптимальним рівнем комісій дозволить максимізувати суму, що надходить до отримувачів пожертв.
- 3) **Підтримка широкого спектра платіжних методів** – можливість здійснювати оплату банківськими картками, електронними гаманцями, розширює потенційну аудиторію.
- 4) **Безпека та відповідність стандартам** – система повинна відповідати стандартам PCI DSS, мати захист від шахрайства та підтримувати двофакторну автентифікацію.

На основі аналізу даних чинників, оптимальним підходом до розробки є використання модульної архітектури, що дозволить забезпечити гнучкість та масштабованість проєкту. Важливим є поділ системи на незалежні компоненти:

1. **Ядро веб-додатку** – забезпечує основний функціонал, додавання та керування проєктами, новинами, збір та обробку даних.
2. **Платіжний модуль** – окремий компонент, який реалізує взаємодію з обраною платіжною системою, що дозволяє за необхідності змінити постачальника платіжних послуг без суттєвих змін у загальній структурі системи.
3. **Фронтенд частина** – візуальна частина сайту. Відображає основну інформацію щодо університету, поточних проєктів, новин щодо реалізацій та звітності коштів для прозорості.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА МОДЕЛЮВАННЯ МОДУЛІВ САЙТУ

2.1. Прототипування сторінок сайту

Для забезпечення зручності користування та відповідності сучасним вимогам веб-дизайну було здійснено прототипування та моделювання інтерфейсу веб-додатку за допомогою онлайн-інструменту Figma. Даний інструмент дозволив ефективно створити адаптивні макети сторінок, визначити взаємодію між елементами інтерфейсу та організувати зручну навігацію між основними розділами сайту.

У межах проєктування було створено прототипи таких сторінок:

Головна сторінка — містить вступну інформацію про ініціативу фандрайзингу, заклик до дії, а також блоки з останніми новинами, проєктами та звітами. Особлива увага приділена візуальній ієрархії та першому враженню користувача. (Рис.2.1.)

Сторінка "Новини" — реалізована у вигляді стрічки новин, де кожна публікація представлена у форматі картки з коротким описом, зображенням. Передбачено можливість переходу до повного тексту новини.

Сторінка "Проєкти" — містить список актуальних ініціатив, на які ведеться збір коштів. Кожен проєкт має опис, цільову суму, зібрану суму, кнопку "Детальніше". Інтерфейс реалізовано з урахуванням прозорості та відкритості даних.

Сторінка "Звіти" — доступ до фінансових та організаційних звітів за проєктами. Прототип передбачає завантаження документів та короткий опис кожного звіту.

Сторінка "Донат" — реалізовано як зручну форму для здійснення пожертви. У макеті передбачено вибір суми, контактну інформацію, способи оплати, а також підтвердження згоди з політикою конфіденційності. Особливу увагу приділено безпеці та довірі користувача до процесу платежу.

Приклади перших прототипів дизайну сайту :

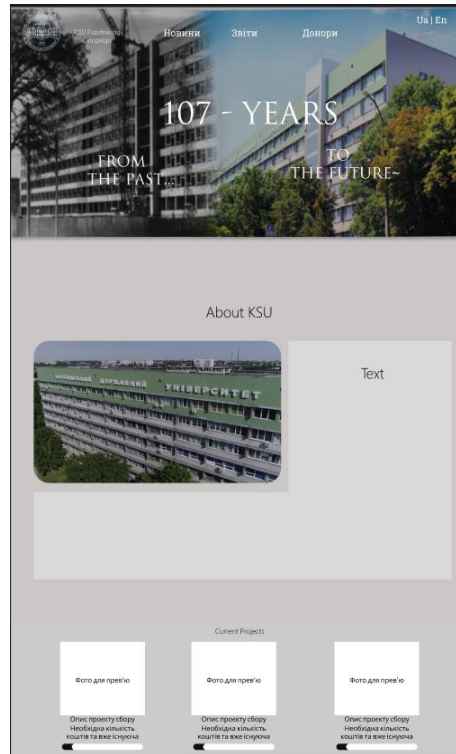


Рис. 2.1. Дизайн головної сторінки

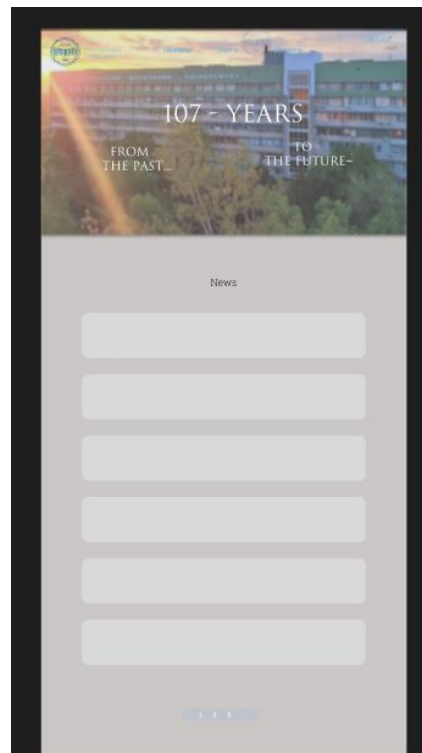


Рис. 2.1 Приклад вигляду сторінки новин

Розроблений дизайн сторінки (Рис.2.1.) для новин включає у себе логіку показу новин, за всі можливі дні коли їх було опубліковано. При необхідності можна перемкнути сторінку на іншу і завантажити новини датами раніше.

Сторінка пожертви (Рис.2.3.) включає у себе можливість обрати бажаний проєкт для підтримки та суму пожертвування. Якщо людина хоче зробити анонімну одноразову пожертву, є варіант вибору «анонімного пожертвування» без внесення особистих даних (ім'я, прізвище, номер телефону)

The image shows a mobile application interface for a donation page. At the top, there is a banner with a cityscape and the text "107 - YEARS FROM THE PAST... TO THE FUTURE-". Below the banner, the page is divided into two main sections: "Donation" and "Анонімне пожертвування".

The "Donation" section contains a form with the following fields:

- "Вибір напряду" (Dropdown menu)
- "Сума" (Dropdown menu)
- "Виктаєть дані" (Section header)
- "Ім'я" (Text input)
- "Прізвище" (Text input)
- "Е-mail" (Text input)
- "Телефон" (Text input)
- "Donate" button

The "Анонімне пожертвування" section contains a form with the following fields:

- "Вибір напряду" (Dropdown menu)
- "Сума" (Dropdown menu)
- "Повідомлення з побажаннями" (Text input)
- "Donate" button

At the bottom of the page, there is a footer with social media icons and text.

Рис. 2.3. Сторінка для пожертв

2.2. Архітектура веб-платформи

Основні компоненти веб-платформи

Веб-платформа складається з декількох основних компонентів приклади наведено у таблиці 2.1:

Таблиця 2.1.

Компоненти веб-платформи

Компонент	Опис
Фронтенд (Frontend)	Відповідає за відображення інтерфейсу, взаємодію користувача з платформою та обробку введених даних перед передачею на сервер.
Бекенд (Backend)	Обробляє запити від фронтенду, забезпечує авторизацію, логіку бізнес-процесів та обробку платежів.
Сервер (Server)	Виконує роль середовища для розгортання бекенду, обробляє запити від клієнтів та керує потоками даних.
Контейнеризація (Docker)	Використовується для ізоляції та управління середовищем виконання веб-додатку, спрощує розгортання.

Загальна схема побудови клієнт-серверної взаємодії

Архітектура веб-платформи базується на класичній клієнт-серверній моделі:

1. Користувач взаємодіє з веб-інтерфейсом через браузер.
2. Фронтенд - відображає дані та надсилає запити до сервера.
3. Бекенд (Python + Flask) обробляє запити, виконує бізнес-логіку та працює з адміністративним та платіжним модулями.
4. Сервер приймає запити, керує розподілом навантаження та забезпечує доступність додатку.
5. Docker використовується для контейнеризації та розгортки веб-додатку.

Вибір стеку технологій

Для реалізації веб-платформи обрано наступні технології:

Фронтенд:

- 1) Bootstrap 5 – адаптивний дизайн та стилізація сторінок.
- 2) HTML – структура веб-сторінок.
- 3) CSS – стилізація та анімації.
- 4) JavaScript – логіка та інтерактивність веб-додатку.

Бекенд:

- 1) Python – основна мова програмування.
- 2) Flask – веб-фреймворк для реалізації логіки додатку.

Інфраструктура:

- 1) Docker – контейнеризація та розгортання.
- 2) Nginx – модуль для взаємодії з файлами сайту на їх розподілом у «контейнері».

2.3. Функціональні модулі

Користувацька частина: Надає можливість перейти до веб-додатку та ознайомитись з актуальними на даний час проєктами для університету, прочитати його історію детальніше оглянути певний проєкт та перейти до його підтримки через інтегрований модуль платежів.

Адміністративна панель: включає у себе управління проєктами, фінансова звітність, управління новинами і публікаціями.

Вимоги до адміністративного модулю:

- 1) Управління проєктами – створення, редагування фандрайзингових кампаній.
- 2) Управління новинами – створення, редагування.
- 3) Модерація контенту – можливість додавати та редагувати текст, фото, відео для сторінок кампаній та новин.
- 4) Звіти та аналітика – візуалізація даних про збори, статистика донатів, ефективність кампаній.

Нефункціональні вимоги:

1. Зручний інтерфейс – просте управління без потреби в технічних знаннях.
2. Адаптивність – підтримка мобільної версії адміністративної-панелі.
3. Безпека – захист від XSS-атак, шифрування даних.

Платіжний модуль: інтеграція з вибраною платіжною системою, забезпечення безпеки транзакцій.

Вимоги до платіжного модулю:

- 1) Інтеграція з платіжними системами – підтримка API для роботи модулю.
- 2) Різні способи оплати – картки (Visa/Mastercard, Apple Pay, Google Pay.)
- 3) Генерація платіжних запитів – створення посилань або QR-кодів для оплати.

РОЗДІЛ 3. ПРОГРАМУВАННЯ ТА ІНТЕГРАЦІЯ**3.1. Розробка фронтенду**

Фронтенд-частина сайту розроблена з використанням HTML5, CSS3, JavaScript та фреймворку Bootstrap. Дизайн побудований з урахуванням принципів зручності користувача (UI/UX) — логічне розміщення елементів, читабельність тексту та інтуїтивна навігація забезпечують комфортну взаємодію відвідувача з сайтом.

Було реалізовано такі ключові сторінки:

- 1) Головна сторінка з загальною інформацією.
- 2) Сторінка з новинами.
- 3) Окрема сторінка для перегляду повного тексту новини.
- 4) Сторінка з переліком проєктів.
- 5) Детальна сторінка проєкту з можливістю підтримки.
- 6) Сторінка пожертви з формою для введення даних.

Окрім цього, реалізовано підтримку мультимовності для зручності користувачів. Всі елементи сайту адаптовані для коректного відображення на мобільних пристроях і екранах різного розміру.

Далі наведено опис основних графічних елементів які були створені для користувача сайту.

Головна сторінка. На цій сторінці знаходиться уся інформація про університет. Наявна візуалізована картинка та слоган університету. Оскільки кількість інформації невелика, то вона чудово розміщується на ній без шкоди для дизайну.

Хедер - у хедері сайту відображається логотип відділу закладу та сайту, перемикач мов а також кнопки навігації по сторінках. Рис.3.2.

Футер - у футері також знаходиться логотип але самого університету, наявна контактна інформація для контакту з відділом та університетом, юридична адреса університету, посилання на головну сторінку навчального закладу, та копірайт лейбінг. Рис.3.4.

Також для привабливості та цікавості сайту, на головну сторінку було додано анімацію логотипу, який плавно переходить у верхній лівий кут, на місце логотипу на сайті. Зроблено для невеликого запасу часу, поки необхідні елементи сайту завантажуються, без білого фону на який буде дивитись користувач.

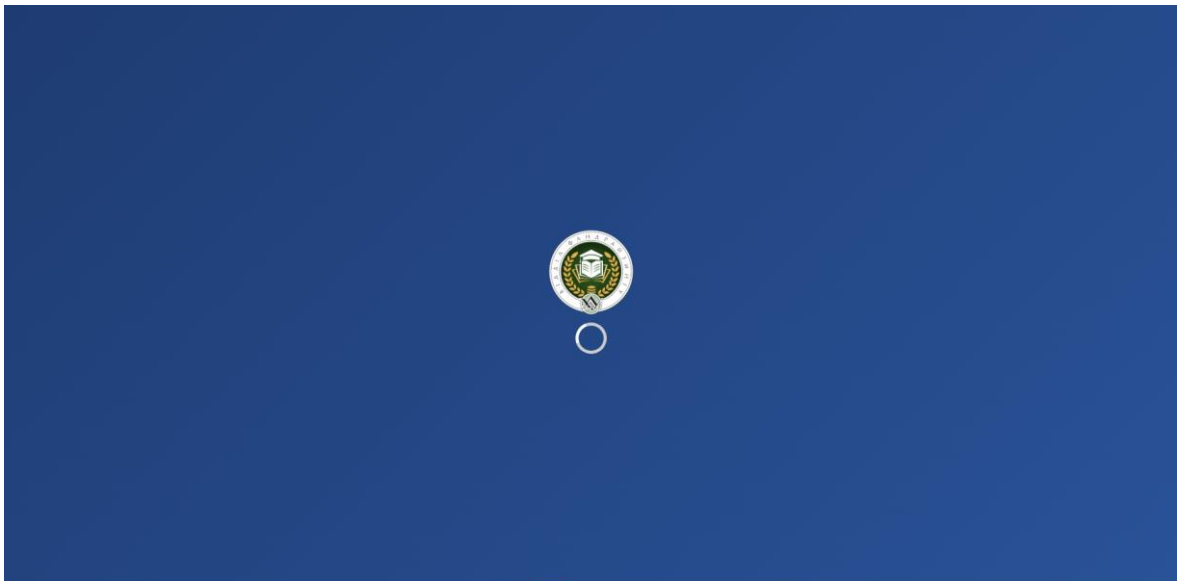


Рис. 3.1. Візуалізація головної сторінки та анімованого логотипу

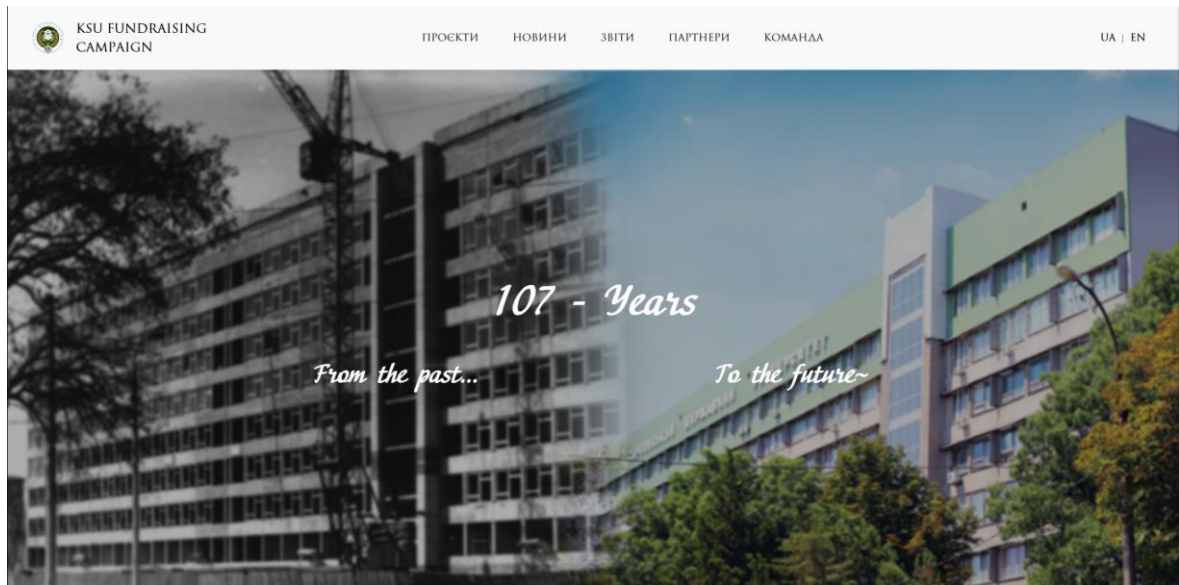


Рис.3.2. Візуалізація головної сторінки та слогану

Додатково на головній сторінці розташовується блок з пріоритетними проєктами на які з акцентовано увагу, та блок недавніх новин.

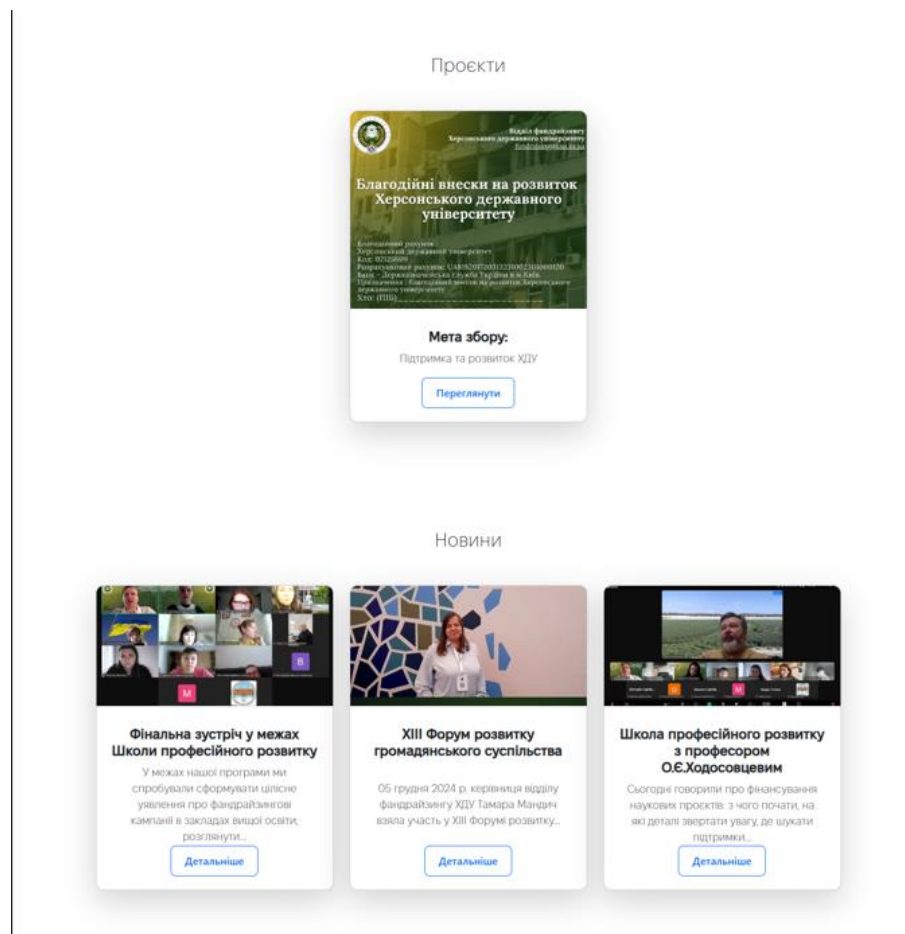


Рис.3.3. Блок проєктів та новин

Сторінка з переліком проєктів включає у себе візуальну картку з короткою інформацією про проєкт, та прогрес баром де позначено грошову ціль збору, і актуальну інформацію про вже зібрану кількість коштів.

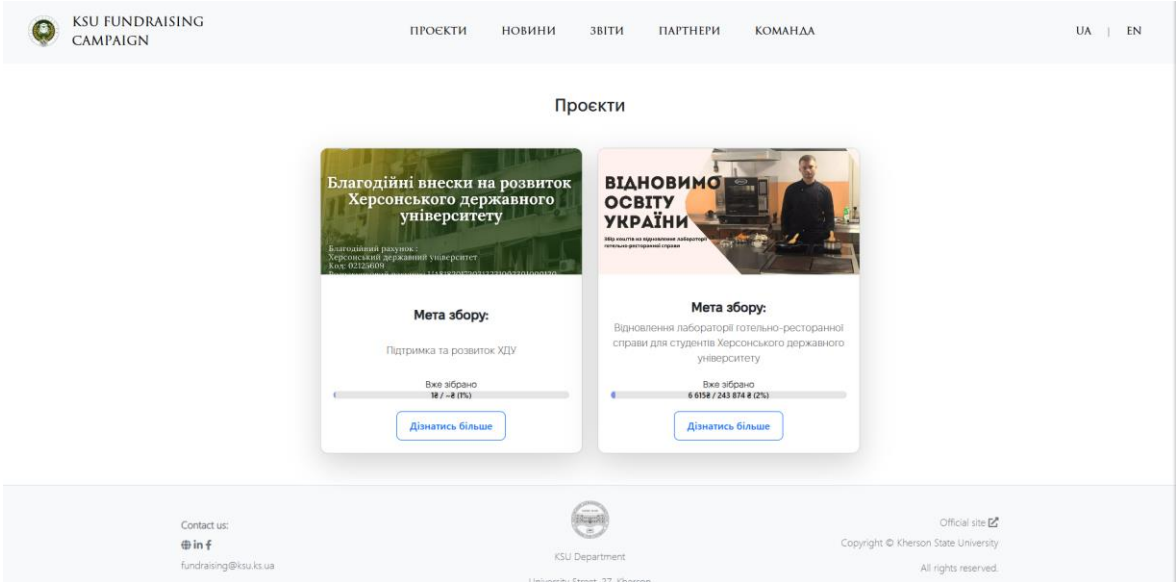


Рис.3.4. Картки проєктів

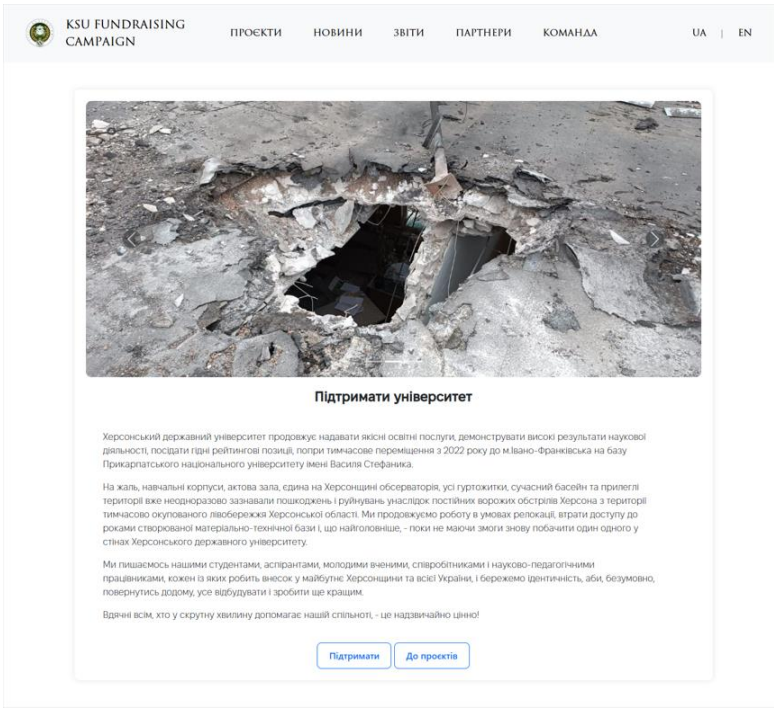


Рис.3.5. Внутрішня частина проєкту

Детальна сторінка проєкту з можливістю підтримки демонструє нам конкретний опис певного проєкту, з карусель-баром де знаходяться фотографії які відносяться до проєкту. Наявні кнопки для повернення до усіх проєктів а також кнопка переходу до сторінки для пожертви

Перейдемо до сторінки з новинами де також знаходяться картки з новинами натиснувши на які можна перейти до детального ознайомлення з НОВИНОЮ

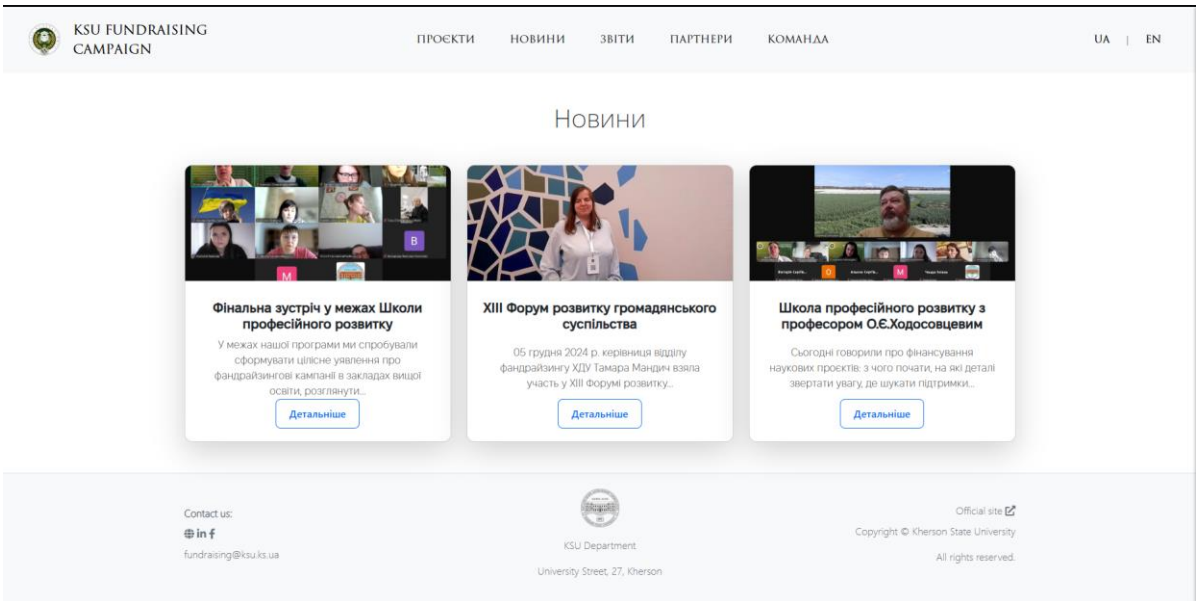


Рис.3.6. Внутрішня частина проєкту

Окрема сторінка для перегляду повного тексту новини включає у себе текст новини та картинки для її більш уявного представлення



Рис.3.7. Візуалізація сторінки з новиною

Сторінка для пожертв була у довгій розробці, що призвело до двох функціонально та візуально різних варіантів. Наявна сторінка для пожертв за певними реквізитами, де для зручності користувача було додано кнопки для швидкого копіювання необхідного тексту, а також у подальшому додано QR-код для оплати з телефону, що натовхнуло до реалізації цієї ідеї на іншій сторінці оплати.

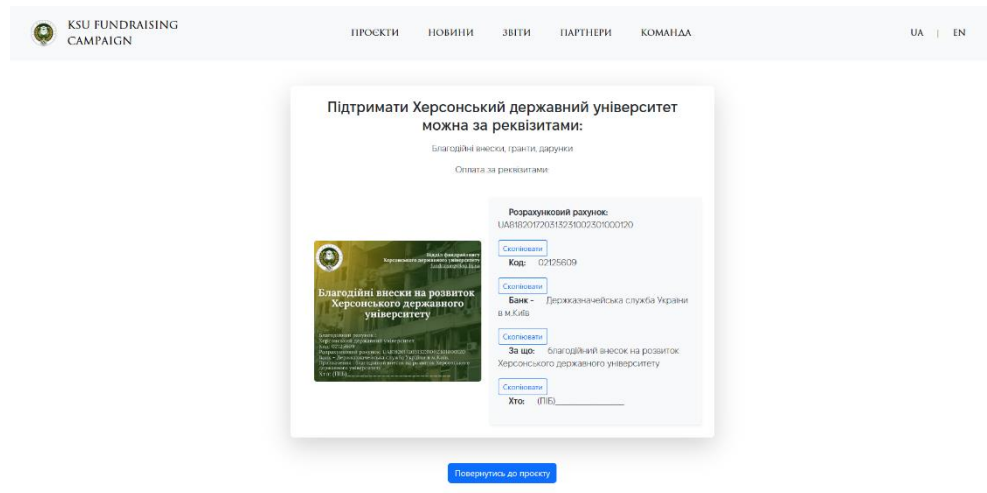


Рис.3.8. Сторінка для пожертв

3.2. Розробка бекенду

Для забезпечення функціональності сайту фандрайзингу було реалізовано серверну частину (бекенд) на основі мови програмування Python з використанням мікрофреймворку Flask, що дозволяє швидко створювати веб-додатки та гнучко організовувати маршрутизацію запитів.

Ключовим елементом бекенду стало ядро адміністративної панелі, яке забезпечує:

1. Додавання, редагування та видалення новин;
2. Управління інформацією про проєкти фандрайзингу;
3. Завантаження та публікація звітів;
4. Автоматичне формування та оновлення HTML-структури для виводу контенту на фронтенді;
5. Обробку зображень та графіки за допомогою бібліотеки Pillow.

Модель роботи адміністративного модулю продемонстровано у (Таб.2)

Для парсингу HTML-шаблонів та динамічного оновлення вмісту використовується бібліотека BeautifulSoup4, що дозволяє коректно обробляти розмітку та вбудовувати змінний контент безпосередньо у сторінки сайту.

Всі дані (текст, фото) передаються у форматі JSON, а з'єднання з ядром забезпечується з урахуванням CORS-політик за допомогою модуля flask-cors, що дозволяє здійснювати кросдоменні запити з клієнтської частини.

Особливу увагу приділено реалізації платіжного модуля (Таб. 1), який інтегрується з системою LiqPay через їхній API. Користувач має можливість підтримати обраний проєкт безпосередньо на сайті. Всі дані транзакцій передаються через платіжну систему не залишаючись на сервері навчального закладу.

Опис логіки роботи адміністративного ядра

Адміністративне ядро веб-додатку відповідає за обробку запитів, що надходять з інтерфейсу адміністратора. Його основне призначення — забезпечити зручне керування контентом сайту без потреби прямого редагування файлів вручну. Адміністративне ядро функціонує на сервері й взаємодіє з користувачем через веб-інтерфейс адміністративної панелі.

1. Авторизація адміністратора

Першим етапом є авторизація адміністратора на адміністративній сторінці. (Рис.3.10.) Користувач вводить логін та пароль, після чого ці дані надсилаються на сервер, де здійснюється перевірка. У разі успішної аутентифікації адміністратор отримує доступ до основного функціоналу панелі керування.

2. Вибір розділу для редагування

Після авторизації адміністратору відкривається навігаційне меню, яке розміщене ліворуч на сторінці. У цьому меню надається вибір розділів, зокрема (Новини, Проекти, Звіти.)

Після вибору одного з розділів здійснюється перехід на відповідну підсторінку, яка завантажує функціонал для роботи з вибраним типом контенту.

3. Робота з контентом розділів

На прикладі розділу **"Новини"** можна простежити загальну логіку взаємодії адміністратора з контентом за допомогою ядра:

3.1. Отримання списку новин

На сторінці розділу **"Новини"** адміністратору доступна кнопка **"Вивести всі новини"**. Після її натискання відправляється запит до ядра. У свою чергу, ядро сканує відповідну директорію на сервері, де зберігаються HTML-файли новин, витягує з них назви або інші ідентифікаційні дані та повертає у вигляді списку. (Рис.3.11.) Цей список відображається на сторінці адміністратора у вигляді заголовків новин з відповідними кнопками для подальшої взаємодії.

3.2. Редагування новин

Якщо адміністратор вмикає режим редагування (наприклад, шляхом встановлення чекбокса), під кожною новиною стає доступною кнопка **"Редагувати"**. При натисканні на неї формується окремий запит до ядра, в якому вказується конкретний файл новини. (таблиця 3.2.)

Ядро відкриває відповідний HTML-файл і здійснює парсинг розмітки для знаходження потрібних блоків, таких як заголовки (<h1>), основний текст (<div class="content"> тощо). Вміст цих блоків відправляється у відповідь на клієнтську частину, де він відображається у текстових полях для редагування.

3.3. Збереження змін

Після редагування адміністратор може натиснути кнопку **"Зберегти"**, унаслідок чого змінений вміст (заголовок, текст тощо) у вигляді JSON-структури надсилається назад до ядра. Ядро отримує ці дані, знову відкриває відповідний HTML-файл, знаходить необхідні елементи розмітки, оновлює їх новим вмістом і зберігає файл.

У такий спосіб забезпечується повноцінне редагування вмісту сайту без необхідності прямого доступу до файлової системи або HTML-коду з боку адміністратора.

4. Аналогічна логіка для інших розділів

Розділ **"Проекти"** реалізований за аналогічним принципом.

- 1) Отримання списку доступних елементів (сторінок).
- 2) Можливість перегляду та редагування вмісту.
- 3) Збереження змін з оновленням відповідного HTML-файлу.
- 4) Завдяки уніфікованій структурі запитів і єдиному підходу до обробки, ядро може масштабуватися або доповнюватися іншими типами контенту за потреби.

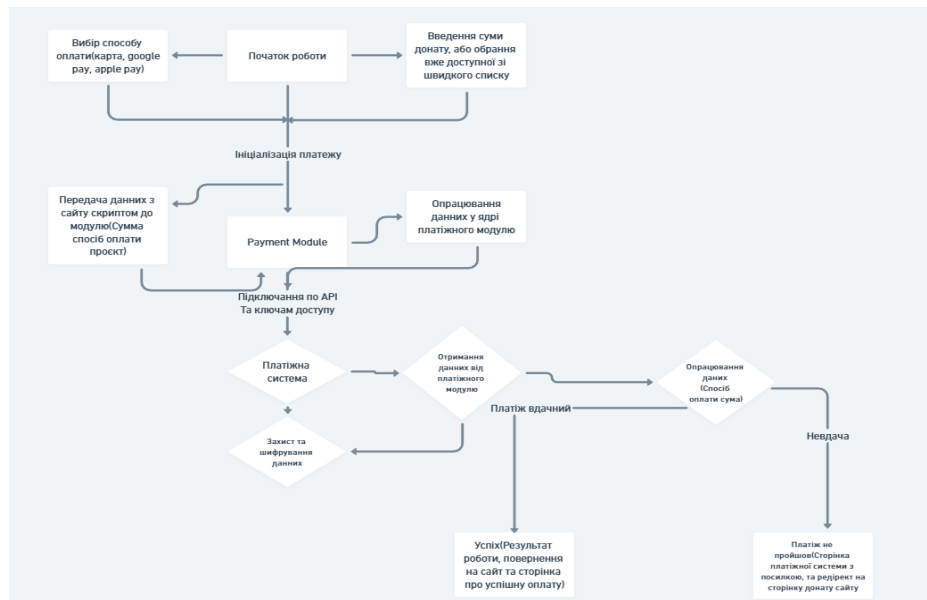


Рис.3.9. Візуалізація роботи платіжної системи методом FLOWCHART

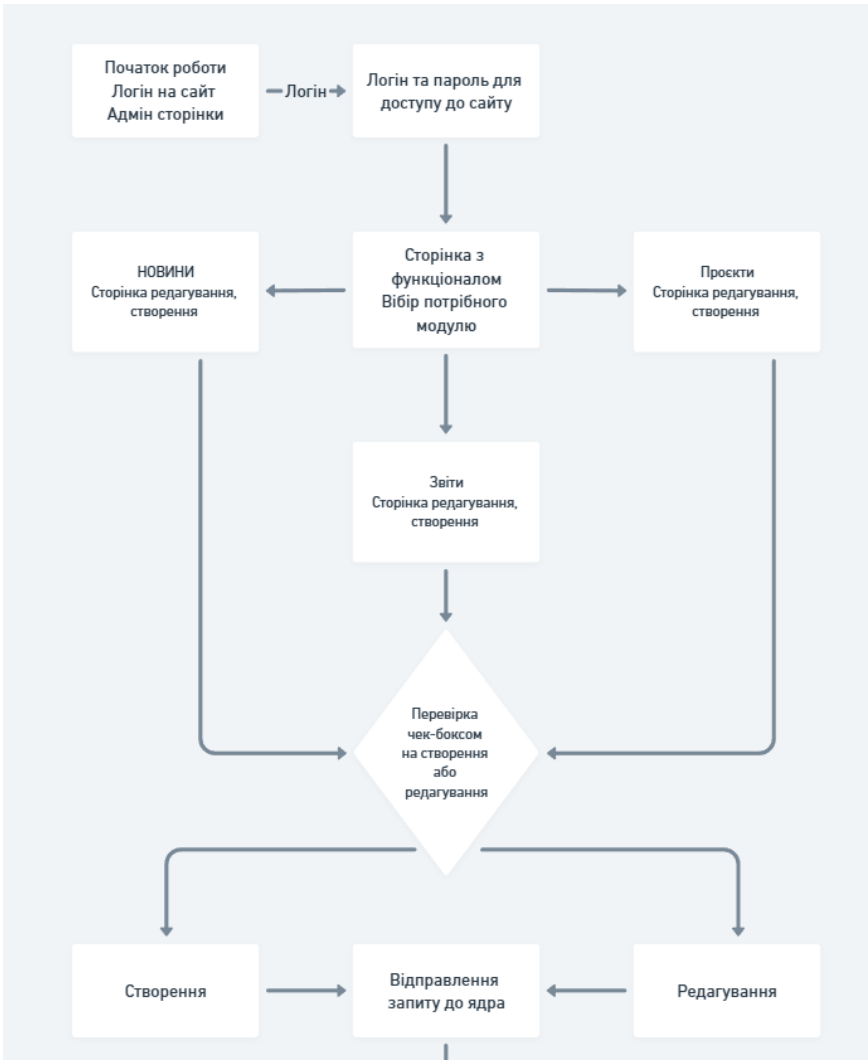


Рис.3.10. Візуалізація адміністративного модулю

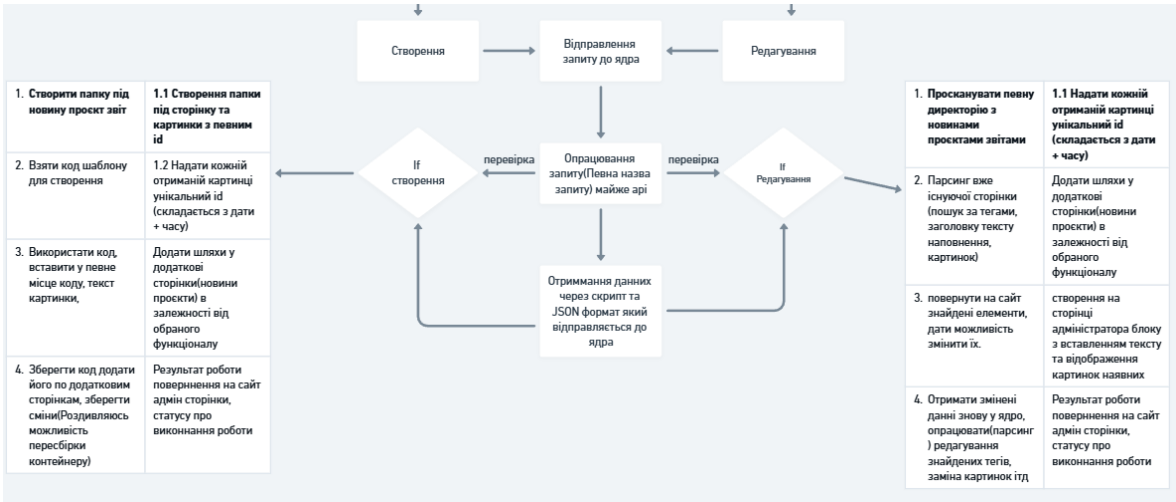


Рис.3.11. Візуалізація адміністративного модулю

Опис логіки роботи платіжного модуля

Для реалізації функціоналу прийому онлайн-платежів у фандрайзинговому веб-додатку розроблено окреме серверне ядро

платіжного модуля, яке взаємодіє з платіжною системою LiqPay через API-з'єднання. Цей модуль забезпечує безпечну передачу та обробку транзакцій, відповідно до стандартів захисту даних користувачів.

1. Підключення до платіжної системи

Ядро платіжного модуля взаємодіє з LiqPay (Рис.3.9) за допомогою API, для чого використовуються два ключі доступу:

- 1) Публічний ключ — використовується на клієнтській стороні для ініціалізації базових параметрів запиту;
- 2) Приватний ключ — зберігається на сервері та використовується для формування цифрового підпису запиту. Він має бути захищеним і не розголошуватися, оскільки забезпечує безпеку транзакцій.

Ці ключі дозволяють формувати захищений запит до платіжної системи, що підтверджується криптографічно підписаними даними.

2. Користувацький інтерфейс форми донату

На головному сайті, на сторінці фандрайзингу, реалізована форма, яка дозволяє користувачу здійснити пожертву на вибраний проєкт. Форма складається з наступних елементів:

- Кнопки швидкого вибору суми пожертви (наприклад, 10, 20, 50, 100, 500 грн);
- Список проєктів — для вибору, на який саме проєкт буде спрямовано пожертву;
- Форма введення особистих даних, таких як ім'я (як на платіжній картці) та номер телефону.

3. Обробка даних на сервері

Після заповнення форми та підтвердження наміру пожертвувати, усі введені користувачем дані збираються в структурований формат JSON і надсилаються до серверного ядра платіжного модуля. (таблиця 3.1.) Там дані проходять обробку та підготовку до надсилання в LiqPay API.

Зокрема:

- 1) Генерується підпис.

- 2) Формується запит з параметрами платежу (сума, валюта, опис, callback URL тощо).
- 3) Ініціалізується платіжний процес через захищене з'єднання.

4. Переадресація на платіжну сторінку

Користувача автоматично перенаправляє на сторінку платіжної системи LiqPay. Ця сторінка реалізована зі всіма необхідними протоколами безпеки:

- 1) Шифруванням з'єднання (HTTPS).
- 2) Захистом персональних даних.
- 3) Відсутністю збереження чутливої інформації на стороні університету.

Таким чином, університетський сервер не зберігає жодних платіжних або банківських даних користувача, що підвищує рівень безпеки як для користувача, так і для організації.

5. Обробка результату платежу

Після здійснення транзакції, платіжна система повертає користувача на одну з двох сторінок:

- 1) Успішна оплата — при позитивному результаті, користувачу відображається сторінка з повідомленням про успішну оплату, після чого він автоматично повертається на сайт;
- 2) Помилка платежу — при виникненні проблем (недостатньо коштів, скасування, помилка системи тощо), користувачу відображається відповідне повідомлення про невдачу.

Крім того, серверна частина може приймати callback-запити від LiqPay з підтвердженням результату транзакції, що дозволяє зафіксувати факт платежу в базі даних або надіслати адміністратору повідомлення.

Візуальні приклади коду з деякими функціями адміністративного та платіжного модулів зображені на рисунках (3.12., 3.13., 3.14.).

```

from flask import Flask, jsonify, send_from_directory, request
from flask_cors import CORS
from bs4 import BeautifulSoup
import os
import json

app = Flask(__name__)
CORS(app)

# Правильний шлях до папки News, яка знаходиться в KREP
NEWS_DIR = os.path.abspath(os.path.join(os.path.dirname(__file__), '..', '..', 'try', 'KSUREP', 'News'))
IMAGES_DIR = os.path.join(NEWS_DIR, 'images') # Шлях до папки з зображеннями

print(f"Шлях до папки з новинами: {NEWS_DIR}")

# Перевірка на існування папки
if not os.path.exists(NEWS_DIR):
    print("Папка з новинами не знайдена!")
else:
    print("Папка з новинами знайдена!")

# Функція для пошуку HTML файлів та зображень
def find_files_and_images(directory):
    news_files = []
    images_files = []

    # Проходимо всі підпапки і файли в директорії
    for root, dirs, files in os.walk(directory):
        for file in files:
            file_path = os.path.join(root, file)

            # Якщо файл HTML
            if file.endswith('.html'):
                news_files.append(os.path.basename(file_path)) # Зберігаємо тільки назву файлу
            # Якщо файл зображення (наприклад .jpg, .png)
            elif file.lower().endswith(('jpg', 'jpeg', 'png', 'gif', 'bmp', 'webp')):
                relative_image_path = os.path.relpath(file_path, IMAGES_DIR) # Отримуємо відносний шлях до картинки
                images_files.append(relative_image_path)

    return news_files, images_files

```

Рис.3.12. Приклад коду адміністративного ядра

```

# Обробник для запиту на пошук новин
@app.route('/api', methods=['POST'])
def api():
    data = request.get_json()

    if data and 'command' in data and data['command'] == 'parse-folder':
        print("Запит отримано!")
        print("Шукаємо новини...")

        # Шукаємо новини та зображення
        news_files, images_files = find_files_and_images(NEWS_DIR)
        print("News files:", news_files) # Логування, щоб перевірити, що на сервері правильні дані
        result = {
            'news_files': news_files,
            'images_files': images_files
        }

        return jsonify(result)

    return jsonify({"error": "Невідомий запит"}), 400

```

Рис.3.13. Приклад коду адміністративного ядра

```

liqpay = LiqPay(public_key, private_key)
html = liqpay.cnb_form({
    'action': 'pay',
    'amount': '1',
    'currency': 'USD',
    'description': 'description text',
    'order_id': 'order_id_1',
    'version': '3'
})

```

Рис.3.14. Приклад коду для взаємодії платіжного модулю

З міркувань безпеки, зокрема через ризик зламів або атак, у роботі наведено лише обмежену кількість прикладів коду з двох серверних ядер. Такий підхід є обґрунтованим з огляду на дотримання вимог конфіденційності.

Таблиця 3.1.

Модель роботи платіжної системи

Етап	Опис дії	Учасники процесу
1. Ініціація платежу	Користувач вводить платіжні дані (сума, метод оплати) на веб-додатку	Користувач, веб-додаток
2. Відправка запиту	Дані передаються на сервер бекенду для обробки	Веб-додаток → Сервер
3. Передача до платіжного шлюзу	Сервер надсилає запит до API платіжної системи	Бекенд → Платіжний шлюз
4. Взаємодія з банком	Платіжний шлюз пересилає запит до банку для перевірки та обробки платежу	Платіжний шлюз → Банк
5. Обробка платежу	Банк перевіряє наявність коштів, проводить списання або відхиляє платіж	Банк
6. Відповідь від платіжного шлюзу	Банк надсилає відповідь до платіжної системи, яка пересилає її серверу додатку	Банк → Платіжний шлюз → Бекенд
7. Оновлення статусу	Сервер оновлює статус платежу в базі даних і сповіщає користувача	Бекенд → Веб-додаток → Користувач

**Модель роботи адміністративної панелі (збереження та генерація
HTML-сторінки**

Етап	Опис дії	Учасники процесу
1. Введення даних	Адміністратор вводить інформацію (текст, зображення, відео) через веб-форму	Адміністратор, веб-додаток
2. Збереження у JSON	Дані формуються у JSON-структуру та передаються на сервер	Веб-додаток → Сервер
3. Обробка сервером	Сервер на Python приймає JSON-файл, розпаковує його та парсить дані	Бекенд (Python)
4. Формування HTML-шаблону	Оброблені дані вставляються у відповідні блоки HTML-шаблону	Бекенд (Python)
5. Збереження медіафайлів	Зображення та відео зберігаються у папку з унікальним ID сторінки	Файлова система сервера
6. Генерація сторінки	Сформований HTML-файл зберігається та стає доступним через веб-додаток	Сервер
7. Відображення сторінки	Адміністратор або користувач переглядає згенеровану сторінку	Веб-додаток → Користувач

ВИСНОВКИ

У ході даної кваліфікаційної роботи була поставлена мета розробити веб-додаток для фандрайзингу університету. В результаті реалізовано функціонал, який дозволяє ефективно управляти проєктами, приймати пожертви та забезпечувати зручну взаємодію користувачів із платформою

Під час реалізації роботи було виконано такі пункти

Проаналізовані готові рішення сайтів та платіжних систем їх переваги та недоліки, визначено ключові вимоги до функціоналу.

- 1) Розроблений веб додаток Фандрайзингу університету.
- 2) Розроблений інтуїтивний користувацький інтерфейс.
- 3) Розроблено та впроваджено модуль ядра адміністрації сайту.
- 4) Розроблено та введено у дію платіжний модуль.
- 5) Протестовано функціонал модулів та платежі.

Розроблений веб-додаток як окремий проєкт надає можливість глобальної підтримки університету, представленням необхідностей з подальшим удосконаленням систем та нововведень.

Результати дослідження підтверджують доцільність використання веб-технологій для оптимізації процесів залучення коштів та підвищення ефективності управління фандрайзинговими ініціативами у закладах вищої освіти.

ВИКОРИСТАНІ ДЖЕРЕЛА

- 1) BeautifulSoup4 Documentation [Електронний ресурс]. – Інтернет-ресурс. – Режим доступу:
<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
- 2) Flask Documentation [Електронний ресурс]. – Режим доступу:
<https://flask.palletsprojects.com/>
- 3) Pillow Documentation [Електронний ресурс]. – Режим доступу:
<https://pillow.readthedocs.io/>
- 4) Bootstrap Documentation [Електронний ресурс]. – Режим доступу:
<https://getbootstrap.com/docs/5.3/getting-started/introduction/>
- 5) LiqPay API documentation [Електронний ресурс]. – Режим доступу:
https://www.liqpay.ua/doc/api/internet_acquiring/checkout?tab=1
- 6) Pérez-Esparrells C., Torre E. M. The challenge of fundraising in universities in Europe // International Journal of Higher Education. – 2012. – Vol. 1, № 2. – С. 1-23. – Стаття. – Режим доступу:
<https://eric.ed.gov/?id=EJ1057209>
- 7) Благодійність в Україні. Фандрейзинг: що це, основні види та особливості // Philanthropy in Ukraine. – [Електронний ресурс]. – Режим доступу: <https://philanthropyinukraine.org/knowledge-hub/publication/fandrejzing-sho-ce-osnovni-vydy-ta-osoblyvosti>
- 8) Lau S. C. Future Trends in Higher Education Fundraising // Sandra Chen Lau. – [Електронний ресурс]. – Режим доступу:
<https://sandrachenlauca.com/future-trends-higher-education-fundraising/>
- 9) LiqPay // Forbes. – [Електронний ресурс]. – Режим доступу:
<https://forbes.ua/profile/liqpay-1392>
- 10) Portmone: 23 роки інновацій // Portmone Blog. – 2023. – [Електронний ресурс]. – Режим доступу:
<https://blog.portmone.com.ua/uk/news-uk/portmone-birthday-23>
- 11) EasyPay: підсумки 2024 року // EasyPay Blog. – 2024. – [Електронний ресурс]. – Режим доступу: <https://blog.easypay.ua/11-6-mln->

[koristuvachiv-1-4-tis-novih-servisiv-i-novi-platizhni-rishennya-chim-zapamyatayetsya-2024-rik-v-easypay](#)

12) Переваги використання платіжних систем // Детектор медіа. – 2024. – [Електронний ресурс]. – Режим доступу:

<https://detector.media/withoutsection/article/222827/2024-02-12-perevagy-vykorystannya-platizhnykh-system>

13) Національна платіжна система "ПРОСТІР" // Prostir.gov.ua. – [Електронний ресурс]. – Режим доступу: <https://prostir.gov.ua/ua/about-us>

14) Документація для інтеграції Portmone // Portmone.com.ua. – [Електронний ресурс]. – Режим доступу:

<https://www.portmone.com.ua/ecommerce/faq>

15) Документація для інтеграції LiqPay: посібник // LiqPay.ua. – [Електронний ресурс]. – Режим доступу:

<https://www.liqpay.ua/information/handbook/activation?tab=0>

16) Документація для інтеграції EasyPay // Docs.easypay.ua. – [Електронний ресурс]. – Режим доступу: <https://docs.easypay.ua/ua>

17) Боголіб Т. М. Освітній та науковий фандрайзинг в Україні // Наукові праці Донецького національного технічного університету.

Серія: Економічна. – 2014. – № 5. – С. 7–17. – Наукова стаття (PDF). –

Режим доступу: irbis-nbuv.gov.ua

18) Top Trends in Higher Education Fundraising // DickersonBakker Blog. – 2024. – Стаття. – Режим доступу:

<https://blog.dickersonbakker.com/higher-education-fundraising-trends>

19) Булатов О. С. Огляд фреймворків та бібліотек для розробки веб-додатку онлайн-кінотеатру // Free and Open Source Software: матер. XVI Міжнар. наук.-практ. конф., Харків, 13–14 лют. 2025 р. – Харків: ХНЕУ ім. С. Кузнеця, 2025. – С. 116. – Матеріали конференції. – Режим

доступу: <https://repository.hneu.edu.ua/bitstream/123456789/35624/1/foss-2025-theses.pdf#page=117>

- 20) Mouat A. Using Docker: Developing and Deploying Software with Containers. – O'Reilly Media, Inc., 2015. – 332 с. – Книга. – Режим доступу: https://books.google.com.ua/books?hl=uk&lr=&id=wpYpCwAAQB_AJ&oi=fnd&pg=PP1&dq=docker+container+development&ots=QiI8xUm_bV&sig=
- 21) Гнатюк С. С. Що таке API? Вступ до інтерфейсів прикладного програмування // Збірник наукових праць ІТ та МС. – 2022. – С. 71. – Наукова стаття (PDF). – Рекомендовано до друку Вченою радою Поліського національного університету (протокол № 10 від 26.04.2023 р.). – Режим доступу: http://eprints.zu.edu.ua/37557/1/Збірник_ІТтаМС_2023_друк.pdf#page=71
- 22) Patkar U. та ін. Python for web development // International Journal of Computer Science and Mobile Computing. – 2022. – Vol. 11, № 4. – С. 36. – Стаття. – Режим доступу: <https://d1wqtxts1xzle7.cloudfront.net/87115279/V11I4202208-libre.pdf>
- 23) Janarthanan N. Web development and server hosting [Електронний ресурс]. – Звіт / книга. – 2023. – Режим доступу: https://jana9047.github.io/Janarthanan_Portfolio/Internship%20Reports/Future%20Report%20-%20Web%20Development%20and%20Server%20Hosting%20-%20Janarthanan%20N.pdf
- 24) Schommer I., Broschart S. SilverStripe: The Complete Guide to CMS Development. – Hoboken: John Wiley & Sons, 2010. – 408 с. – Книга. – Режим доступу <https://books.google.com.ua/books?hl=uk&lr=&id=4qIW7f6AkBAC&oi=fnd>

ДОДАТКИ*Додаток А*

«Аналіз фандрайзингу університетів» файл у форматі (Microsoft Excel) Analysis2025.xlsx