

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ПРАТ «ПВНЗ «ЗАПОРІЗЬКИЙ ІНСТИТУТ ЕКОНОМІКИ  
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ»

Кафедра інформаційних технологій

ДО ЗАХИСТУ ДОПУЩЕНА

Заст. зав. кафедри \_\_\_\_\_

д.е.н., проф. Н. Р. Полуектова

МАГІСТЕРСЬКА ДИПЛОМНА РОБОТА  
РОЗРОБКА СИСТЕМИ УПРАВЛІННЯ КОРИСТУВАЧАМИ ТА  
КОНТЕНТОМ СОЦІАЛЬНОЇ МЕРЕЖІ

Виконав

ст. гр. ІПЗ-211м

\_\_\_\_\_

Шляга О. В.

Науковий керівник

доцент

\_\_\_\_\_

Резніченко Ю. С.

Запоріжжя

2023 р.

ПРАТ «ПВНЗ «ЗАПОРІЗЬКИЙ ІНСТИТУТ ЕКОНОМІКИ  
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ»  
Кафедра інформаційних технологій

ЗАТВЕРДЖУЮ

Заст. зав. кафедри \_\_\_\_\_

д.е.н., проф. Полуктова Н. Р.

03.10.2022 р.

З А В Д А Н Н Я

НА МАГІСТЕРСЬКУ ДИПЛОМНУ РОБОТУ

студенту гр. ІІЗ-211м, спеціальності 121 «Інженерія програмного  
забезпечення» ОП «Інженерія програмного забезпечення»

Шлязі Ользі Володимирівні

(прізвище, ім'я, по батькові)

1. Тема : Розробка системи управління користувачами та контентом  
соціальної мережі

затверджена наказом по інституту № 02-16 від 03.10.2022 р.

2. Термін здачі студентом закінченої роботи: 12.01.2023 р.

3. Перелік питань, що підлягають розробці

1. Провести огляд сучасних CRM-систем та переваг від їх впровадження

2. Описати принцип роботи систем управління контентом та  
користувачами

3. Ознайомитися з документацією фреймворка Laravel та платформи  
SendGrid для розробки панелі

4. Розробити адмін-панель для керування користувачами, контентом та їх  
бізнесом

5. Розробити панель користувача для управління контентом, профілями та  
бізнесом

6. Розробити модуль сповіщення через email

7. Створити аналітику та звіти в реальному часі у вигляді таблиць з детальними даними

8. Розробити модуль експорту статистики у CSV (Excel)

9. Проаналізувати отримані результати. Оформити звіт за результатами роботи

4. Календарний графік підготовки магістерської дипломної роботи:

№ етапу	Зміст	Термін виконання	Готовність по графіку (%), підпис керівника	Підпис керівника про повну готовність етапу, дата
1.	Формулювання теми магістерської дипломної роботи (збір практичного матеріалу за темою магістерської дипломної роботи)	20.10.22		
2.	I атестація I розділ магістерської дипломної роботи	27.10.22		
3.	II атестація II розділ магістерської дипломної роботи	17.11.22		
4.	III атестація III та IV розділ магістерської дипломної роботи, висновки та рекомендації, додатки, реферат, перевірка програмою «Антиплагіат»	29.12.22		
5.	Доопрацювання магістерської дипломної роботи, підготовка презентації, отримання відгуку керівника і рецензії	10.01.22		
6.	Попередній захист магістерської дипломної роботи	12.01.22		
7.	Подача магістерської дипломної роботи на кафедру	за 3 дні до захисту		
8.	Захист магістерської дипломної роботи	18.01.22		

Дата видачі завдання: 03.10.2022 р.

Керівник магістерської роботи \_\_\_\_\_ Ю. С. Резніченко  
(підпис) (прізвище та ініціали)

Завдання отримав до виконання \_\_\_\_\_ О. В. Шляга  
(підпис студента) (прізвище та ініціали)

## РЕФЕРАТ

Магістерська дипломна робота містить: 85 сторінок, 43 рисунки, 40 бібліографічних посилань.

Мета роботи – розробити панель для соціальної мережі, яка дозволить повністю керувати контентом, користувачами, реалізацією ними середнього та малого бізнесу, збором статистики та аналітики.

Об’єкт дослідження – автоматизація бізнес-процесів соціальної мережі завдяки панелі управління користувачами та контентом. Предметом дослідження є методи і підходи автоматизації системи управління бізнес-процесами соціальної мережі.

В першому розділі здійснено детальний огляд предметної області та обґрунтовано вибір об’єкту дослідження. Виявлено переваги і недоліки CRM-систем. Оглянуто економічний ефект від впровадження CRM-систем та наведено його приклади.

У другому розділі проведено огляд використаних інструментів для розробки. Описано середовище розробки, менеджери залежностей, пакунків NPM, системи контролю версій та відладчики, фреймворк та сервіси для розсилок..

В третьому розділі проведено огляд інтерфейсу та основних функцій для розробки. Описано основний принцип роботи створеної CRM-системи, Розроблено адмін-панель для керування користувачами, контентом та їх бізнесом, панель користувача для управління контентом, профілями та бізнесом, функціональні модулі.

PHP7, LARAVEL, APACHE, PHPSTORM, SENDGRID, JS, JQEURY, PAYPAL

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	7
ВСТУП.....	8
РОЗДІЛ 1 ОБҐРУНТУВАННЯ ВИБОРУ ОБ’ЄКТУ ДОСЛІДЖЕННЯ.....	11
1.1 Навіщо потрібні CRM-системи.....	11
1.2 Переваги впровадження CRM-систем.....	13
1.3 Недоліки CRM.....	17
1.4 Економічний ефект від впровадження CRM.....	19
1.5 Висновки за першим розділом.....	23
РОЗДІЛ 2 ВИКОРИСТАНІ ІНСТРУМЕНТИ ДЛЯ РОЗРОБКИ.....	24
2.1 Середовище розробки.....	24
2.2 Локальний сервер.....	26
2.3 Менеджер залежностей.....	29
2.4 Відладчик.....	30
2.5 Система контролю версій.....	33
2.6 Інструмент Webpack.....	36
2.7 Менеджер пакунків NPM.....	39
2.8 PHP фреймворк.....	41
2.9 Сервіс для email розсилок.....	42
2.10 Висновки за другим розділом.....	44
РОЗДІЛ 3 ОГЛЯД ІНТЕРФЕЙСУ ТА ОСНОВНІ ФУНКЦІЇ ДЛЯ РОЗРОБКИ.....	45
3.1 Основний принцип роботи розробленої CRM.....	45

3.2 Головна сторінка CRM панелі.....	47
3.3 Сторінка публікацій.....	51
3.4 Сторінка управління і редагування публікації.....	57
3.5 Сторінка користувачів.....	60
3.6 Сторінка управління користувачем.....	63
3.7 Сторінка статистики пожертвувань.....	65
3.8 Сторінка аналітики.....	67
3.9 Розділ розгляду магазинів.....	68
3.10 Управління продуктами.....	70
3.11 Управління магазинами.....	73
3.12 Панель користувача.....	74
3.13 Реалізація бази даних.....	79
3.14 Висновки за третім розділом.....	80
ВИСНОВКИ.....	81
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	83

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

Слово/словосполучення	Скорочення	Умова використання
Customer Relationship Management	CRM	В тексті
Comma-Separated Values	CSV	В тексті
International Business Machines	IBM	В тексті
Cascading Style Sheets	CSS	В тексті
Syntactically Awesome Stylesheets	SASS	В тексті
Leaner Style Sheets	LESS	В тексті
HyperText Markup Language	HTML	В тексті
Representational State Transfer	REST	В тексті
Database	DB	В тексті
Structured Query Language	SQL	В тексті
Node Package Manager	NPM	В тексті
Java Script	JS	В тексті
Application Programming Interface	API	В тексті
Simple Mail Transfer Protocol	SMTP	В тексті
Integrated Development Environment	IDE	В тексті
HyperText Transfer Protocol	HTTP	В тексті
Internet Service Provider	ISP	В тексті
DomainKeys Identified Mail	DKIM	В тексті
What You See Is What You Get	WYSIWYG	В тексті

## ВСТУП

В останні роки все більше і більше користувачів витрачають величезну кількість часу в соціальних мережах. І якщо потрібно побудувати бізнес в інтернеті, то потрібно бути там, де знаходяться клієнти, тобто в соціальних мережах.

Актуальність теми обумовлена тим, що величезна кількість людей користуються соціальними мережами, а тому на сьогодні соціальні мережі є хорошим інструментом для просування і реалізації середнього і малого бізнесу. Система управління користувачами і контентом допоможе автоматизувати процеси бізнесу і ефективніше будувати діалог з покупцем та не допускати помилок в роботі, і в підсумку продавати йому більше.

Актуальність теми визначається кількома ключовими факторами, які відображають сучасні тенденції розвитку цифрових технологій і суспільства, зокрема:

- зростанням ролі соціальних мереж. Соціальні мережі сьогодні є одним із найпотужніших інструментів комунікації та самовираження для мільярдів користувачів по всьому світу. Вони впливають на інформаційні потоки, громадську думку, політичні процеси, а також на бізнес та маркетинг. Соціальні медіа стали основними платформами для обміну різноманітним контентом, від текстових повідомлень до відео, зображень і мультимедійних презентацій;

- необхідністю управління користувачами. Управління великою кількістю користувачів вимагає високого рівня безпеки та захисту персональних даних. Невпорядковане або неефективне управління може призвести до серйозних порушень конфіденційності або витоку даних, що стає критично важливим в умовах зростаючих кіберзагроз. Адміністрування доступу, модерація поведінки користувачів, відстеження активностей і



взаємодій є необхідними для підтримки гармонійного середовища в соціальній мережі;

- важливістю управління контентом. З кожним днем обсяги контенту в соціальних мережах зростають експоненціально, що вимагає ефективної системи управління для забезпечення релевантності, достовірності та якості інформації. Управління контентом також включає модерацію для дотримання етичних стандартів, законів і правил платформи, що є викликом в умовах поширення дезінформації, фейкових новин і небажаного контенту;

- технічними викликами та інноваціями. Стрімке зростання числа користувачів вимагає ефективних рішень для масштабованості, що включає автоматизацію процесів управління та моніторингу користувачів і контенту.

- наявністю комерційних можливостей. Ефективна система управління контентом дозволяє краще працювати з таргетованою рекламою, надаючи компаніям більше можливостей для монетизації соціальних мереж. Збір і аналіз даних про активність користувачів дозволяє точніше прогнозувати їхні потреби та пропонувати персоналізовані послуги, що підвищує комерційну цінність платформи.

Таким чином, розробка ефективної системи управління користувачами та контентом у соціальних мережах є актуальною не тільки через зростання популярності таких платформ, але й через виклики, пов'язані з безпекою, масштабованістю, законодавчими обмеженнями та потребою в персоналізації. Інноваційні підходи в цьому напрямку можуть суттєво покращити досвід користувачів і забезпечити більш стабільне та безпечне функціонування соціальних мереж.

Мета роботи – розробити панель для соціальної мережі, яка дозволить повністю керувати контентом, користувачами, реалізацією ними середнього та малого бізнесу, збором статистики та аналітики.

Виходячи із мети, завданнями роботи є:

- провести огляд сучасних CRM-систем та переваг від їх впровадження;

- зрозуміти принцип роботи систем управління контентом та користувачами;
- ознайомитися з документацією фреймворка Laravel та платформи SendGrid для розробки панелі;
- розробити адмін-панель для керування користувачами, контентом та їх бізнесом;
- розробити панель користувача для управління контентом, профілями та бізнесом;
- розробити модуль сповіщення через email;
- створити аналітику та звіти в реальному часі в вигляді таблиць з детальними даними;
- розробити модуль експорту статистики у CSV (Excel).

Об'єкт дослідження – автоматизація бізнес-процесів соціальної мережі завдяки панелі управління користувачами та контентом. Предметом дослідження є методи і підходи автоматизації системи управління бізнес-процесами соціальної мережі.

## РОЗДІЛ 1

### ОБҐРУНТУВАННЯ ВИБОРУ ОБ'ЄКТУ ДОСЛІДЖЕННЯ

#### 1.1 Навіщо потрібні CRM-системи

CRM розшифровується як – Customer Relationship Management, тобто «управління відносинами з клієнтами». CRM за допомогою автоматизації процесів допомагає ефективніше вибудувувати діалог з покупцем, не допускати помилок в роботі і в підсумку продавати йому більше.

CRM-системи (Customer Relationship Management – система управління взаємовідносинами з клієнтами) є важливим інструментом для будь-якого бізнесу, оскільки допомагають ефективно керувати взаємодією з клієнтами та покращувати їх обслуговування. Розглянемо основні причини, чому потрібні CRM-системи.

CRM-система збирає та зберігає всі дані про клієнтів в одному місці, включаючи контактну інформацію, історію взаємодій, покупки, запити та відгуки. Це дає компанії повне уявлення про кожного клієнта і дозволяє персоналізувати обслуговування.

CRM дозволяє відстежувати всі контакти з клієнтами через різні канали, такі як телефон, електронна пошта, соціальні мережі або чати. Це допомагає уникнути втрати важливої інформації та забезпечити більш ефективне і швидке реагування на запити клієнтів.

CRM-системи автоматизують багато рутинних процесів, таких як відправка листів, планування зустрічей, нагадування про наступні кроки тощо. Це знижує навантаження на співробітників, покращує ефективність їх роботи і дозволяє зосередитися на важливіших завданнях.

CRM-системи збирають та аналізують дані про клієнтів, що дозволяє оцінювати ефективність маркетингових та продажних кампаній. Також вони

допомагають прогнозувати майбутні продажі, визначати найприбутковіших клієнтів і виявляти можливості для збільшення доходу.

Персоналізовані підходи та швидке реагування на потреби клієнтів підвищують їхню лояльність і задоволеність. CRM-системи дозволяють відслідковувати задоволення клієнтів і вчасно вирішувати проблеми.

CRM об'єднує різні департаменти (маркетинг, продажі, підтримка клієнтів), надаючи доступ до єдиної бази даних. Це покращує внутрішню координацію і дозволяє командам працювати більш злагоджено.

CRM дозволяє сегментувати базу клієнтів за різними ознаками (вік, місцезнаходження, історія покупок тощо), що допомагає точніше націлювати маркетингові кампанії та збільшувати їхню ефективність.

CRM надає інструменти для відстеження продажних процесів на різних етапах, що дозволяє виявляти «вузькі місця» і оптимізувати роботу менеджерів з продажів. Завдяки цьому можна скоротити цикл продажу та підвищити конверсію.

За рахунок кращого розуміння клієнтів, персоналізації пропозицій і покращення сервісу CRM сприяє збільшенню повторних покупок, підвищенню середнього чека і залученню нових клієнтів через рекомендації.

CRM-системи надають детальні звіти про роботу співробітників, стан угод і результати кампаній. Це дозволяє керівництву контролювати ефективність бізнес-процесів і вносити своєчасні корективи.

Як бачимо, CRM-системи є важливим інструментом для організації та покращення взаємодії з клієнтами, підвищення ефективності роботи, а також для збільшення продажів і прибутку компанії. Вони допомагають створювати довготривалі, продуктивні стосунки з клієнтами та забезпечують конкурентну перевагу на ринку.

При цьому CRM не замінить живого менеджера, не продасть продукт одним натисканням кнопки, але допоможе вашим співробітникам провести продаж швидше і ефективніше.

## 1.2 Переваги впровадження CRM-систем

Ми звикли, що наші бажання вгадують, а постпродажний супровід є в будь-який час доби. Все це практично неможливо реалізувати без CRM-системи, адже в ній на кожного клієнта заведена окрема картка з повною інформацією про покупця:

- контакти;
- дані про клієнта (марка авто, день народження, наявність дітей, переваги і т.д.);
- записи дзвінків і листи;
- файли і документи;
- історія покупок і платежів;
- примітки менеджерів і історія завдань.

Ці дані представлені в хронології, і можна за пару хвилин переглянути всю історію роботи з клієнтом. Ще один плюс впровадження CRM в тому, що ваша база даних, нарешті, стане єдиною. Можна забути про стікери, блокноти та Excel-таблицях: якщо вам потрібен e-mail або телефон клієнта, ви просто вводите його ім'я в пошуку CRM і відкриваєте картку.

Загалом, впровадження CRM-систем має безліч переваг для компаній, оскільки вони допомагають оптимізувати процеси взаємодії з клієнтами та покращувати бізнес-результати.

CRM-системи допомагають зібрати всю інформацію про клієнтів в єдиному місці, що забезпечує комплексне бачення кожного клієнта і дозволяє краще розуміти їхні потреби та очікування.

Використання CRM дозволяє автоматизувати багато рутинних завдань, таких як відправка листів, нагадувань, формування рахунків та звітів, що знижує навантаження на співробітників і покращує їх продуктивність.

CRM-системи надають інструменти для відстеження етапів продажів, планування наступних дій, управління потенційними клієнтами і оптимізації продажних процесів. Це дозволяє скоротити цикл продажу, збільшити конверсію та зростити доходи.

Завдяки централізованому доступу до даних і автоматизації завдань, співробітники можуть зосередитися на стратегічних задачах, таких як пошук нових клієнтів чи підвищення якості обслуговування.

Завдяки CRM компанії можуть швидше реагувати на запити клієнтів, надавати більш персоналізовані послуги та покращувати рівень задоволеності. Це сприяє підвищенню лояльності клієнтів і їх утриманню.

CRM дозволяє сегментувати базу клієнтів за різними критеріями, такими як вік, інтереси, попередні покупки, і проводити персоналізовані маркетингові кампанії, що підвищує ефективність реклами і конверсію.

CRM-системи надають детальні звіти та аналітику щодо ефективності продажів, маркетингових кампаній, роботи співробітників та взаємодії з клієнтами. Це допомагає керівництву приймати обґрунтовані рішення для покращення стратегії.

CRM об'єднує різні департаменти компанії (продажі, маркетинг, підтримку клієнтів) і надає їм доступ до єдиної бази даних. Це забезпечує кращу координацію та співпрацю між відділами, знижуючи кількість помилок і підвищуючи загальну ефективність.

CRM-системи допомагають краще розуміти клієнтські потреби, швидко вирішувати проблеми та персоналізувати взаємодію, що сприяє підвищенню рівня лояльності та утриманню клієнтів.

CRM дозволяє компанії оптимізувати свої внутрішні процеси, що може знизити витрати на обслуговування клієнтів і підвищити ефективність використання ресурсів.

Отже, впровадження CRM-системи надає бізнесу інструменти для підвищення ефективності роботи, покращення взаємодії з клієнтами,

збільшення продажів і зниження витрат. Це допомагає компаніям не тільки утримувати існуючих клієнтів, але й залучати нових, завдяки покращеному обслуговуванню та персоналізації пропозицій.

Ми вважаємо, що серед безлічі переваг впровадження CRM-систем особливий акцент слід зробити на автоматизації бізнес-процесів.

Кожному знайома ситуація, коли ти записав, щоб не забути, а потім забув, куди записав. З користувачами CRM-систем такого не трапляється. Програма не тільки нагадує зателефонувати, написати лист, обробити лід, але і об'єднує всі канали комунікації: ви зможете прямо з CRM дзвонити, писати листи, листуватися з клієнтами в соцмережах, відправляти їм автоматичні sms, отримувати їх заявки з форми на сайті.

Але це ще не все. Серед ключових переваг CRM – автоматизація майже будь-яких рутинних завдань, що дозволяє зекономити час і ресурси, автоматизуючи майже будь-які повторювані процеси. Прикладами таких автоматизованих завдань у CRM можуть бути:

- ведення бази клієнтів – CRM автоматично зберігає й оновлює інформацію про клієнтів, контакти, історію покупок і взаємодій;

- розсилки й комунікація – CRM може автоматично надсилати персоналізовані електронні листи, смс-повідомлення або нагадування клієнтам про важливі події (наприклад, нагадування про завершення підписки або акції);

- створення нагадувань для менеджерів – CRM автоматично створює нагадування для співробітників про наступні дії щодо клієнтів, дзвінки чи зустрічі, що знижує ризик пропустити важливі кроки;

- обробка заявок та запитів – CRM може автоматично реєструвати заявки, запити клієнтів через різні канали, включно з вебсайтом або соціальними мережами, та розподіляти їх між відповідними менеджерами;

генерація звітів – автоматичне формування аналітичних звітів і графіків для керівництва на основі поточних даних про продажі, маркетингові кампанії чи роботу відділу;

управління завданнями – CRM може автоматично призначати завдання, відстежувати їх виконання та генерувати нагадування про дедлайни;

інтеграція з іншими системами – автоматизована синхронізація з обліковими програмами, складськими системами чи системами підтримки клієнтів.

Завдяки цьому бізнес може працювати ефективніше, скорочуючи витрати часу на рутинні операції й концентруючись на стратегічних завданнях.

Також слід детальніше зупинитися на актуальній та наочній аналітиці.

Завдяки CRM аналітика стане зрозумілою, наочною і швидкою. Клік-клік – перед вами підсумки роботи за день, тиждень або вісім місяців і чотири дні. Ще один клік – і ви розумієте, що Петров працює краще Іванова, Сидоров порушує алгоритм роботи, а заявок з сайту надходить все більше – значить, маркетолог заслужив премію. Приклад показаний на рисунку 1.1.

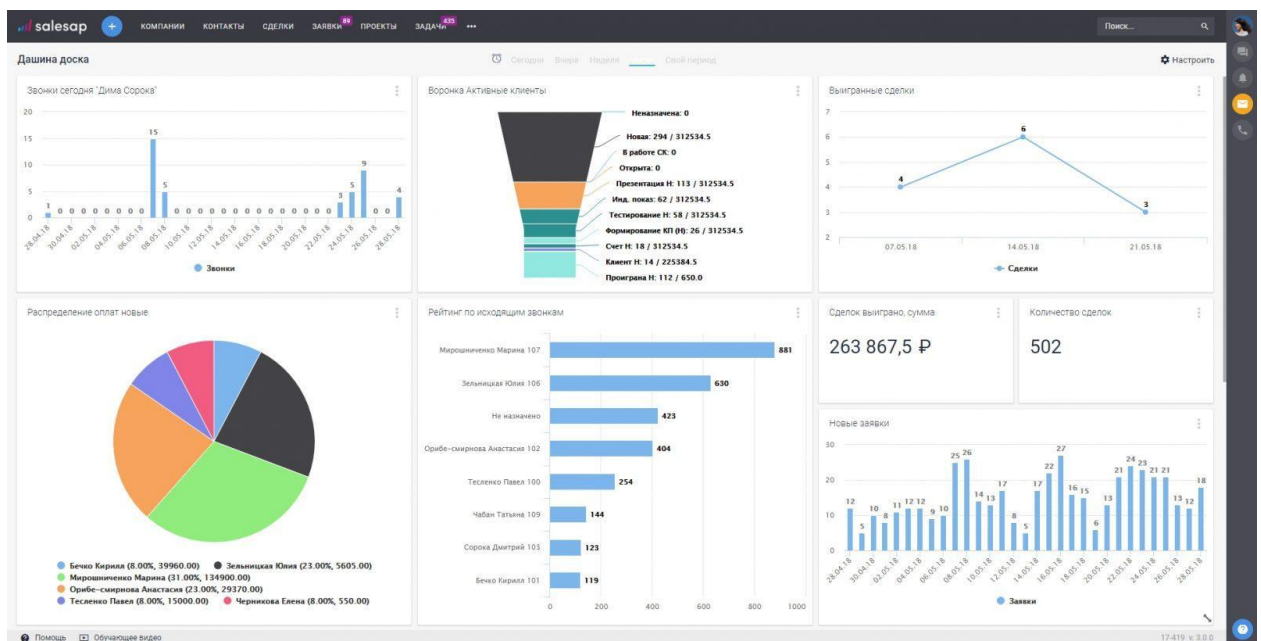


Рис. 1.1 – Наглядна статистика



Так, CRM-системи значно полегшують процес аналітики, роблячи її зрозумілою, наочною і швидкою. Вони дозволяють автоматизувати збір та аналіз даних про клієнтів, угоди, продажі та маркетингові кампанії. Завдяки інтеграції з різними каналами комунікації, CRM забезпечує централізоване зберігання інформації про взаємодію з клієнтами, що дозволяє створювати звіти, діаграми та прогнози в реальному часі.

Основні переваги CRM для аналітики:

користувач отримує централізовані дані – усі дані про клієнтів зберігаються в одному місці;

користувач швидко аналізує дані завдяки автоматизованим звітам – створення динамічних звітів і графіків, які допомагають легко відстежувати продуктивність і результати;

користувачу доступне «легке» прогнозування – CRM дозволяє використовувати дані для прогнозування продажів і аналізу поведінки клієнтів;

користувач використовує візуалізацію – аналітичні дані подаються у вигляді графіків, діаграм і панелей, що робить їх легкими для сприйняття.

Таке рішення допомагає приймати ефективні бізнес-рішення на основі даних.

### 1.3 Недоліки CRM

Отже, CRM веде облік клієнтів і продажів, автоматизує процеси і сама зводить звіти, в результаті чого зростає прибуток і т.д. і т.п. Звучить дуже заманливо, але ... Якби все було так просто, то всі компанії вже давно впровадили б CRM: стартапи злітали б як один, а CRM-система була б визнана панацеєю від усіх проблем. Але чомусь це не так.

Справа в тому, що впровадження такого ПЗ передбачає багато складнощів. І про них краще дізнатися заздалегідь.

По-перше, це висока ціна.

Хороший софт коштує грошей. Вартість доступу в хмарні CRM-програми починається від 180 гривень за користувача в місяць. І чим більше ваш штат, тим дорожче вийде. Плюс потрібно додати вартість самого впровадження програми ... Звичайно, практично у всіх великих CRM-систем є безкоштовні версії. Але потрібно розуміти, що вони надають усічений функціонал, а значить, ефект від впровадження CRM буде менше, ніж обіцяє статистика.

По-друге, опір співробітників.

Перша проблема – не всім легко дається вивчення нового софта, і у бухгалтера тітки Каті можуть виникнути об'єктивні складності. Але якщо вона колись замість калькулятора навчилася користуватися Excel, то і з CRM у неї все вийде. Друга і більш серйозна проблема - саботаж з боку менеджерів. Вони можуть просто не захотіти працювати в CRM - будуть забувати вносити дані або свідомо спотворювати картину. Адже якщо раніше вони могли півгодини сидіти на холодному обдзвоні, а півтори години байдикувати, то тепер керівник в онлайн-режимі відстежує звіти: скільки дзвінків зробив кожен співробітник, скільки завдань закрив, скільки угод обробив і т.д. Справитися з цим може тільки мудре керівництво: з кимось із співробітників, можливо, доведеться розлучитися, а іншим пояснити, що CRM працює на їх благо.

По-третє, низький ефект від впровадження CRM-системи. Саме по собі наявність у компанії CRM-системи ще не гарантує високих продажів і потоку лояльних клієнтів, адже:

- CRM не перетворить поганого працівника в хорошого.
- CRM буде непереливки вірні звіти, якщо ви внесете в програму не всі дані.

– CRM не нагадала про зустріч з клієнтом, якщо ви не поставите собі таке завдання і т.д.

Одним словом, якщо цим інструментом користуватися неправильно, то розраховувати на зростання прибутку не варто. Потрібно спочатку чітко розуміти, яка мета впровадження CRM, і постійно вдосконалювати свої бізнес-процеси.

Наприклад, сама по собі програма не виправить погану воронку продажів, але допоможе побачити, на яких етапах йде найбільше клієнтів, хто з менеджерів не вміє обробляти заперечення і скільком лідам забули передзвонити. Аналіз цієї інформації допоможе усунути слабкі місця, оптимізувати воронку продажів, скорегувати скрипти менеджерів і в кінцевому підсумку підвищити прибуток.

#### 1.4 Економічний ефект від впровадження CRM

Інвестиції в CRM-системи можуть бути надзвичайно вигідними для бізнесу. За даними IBM, кожен долар, витрачений на CRM, повертає п'ять доларів прибутку, що підкреслює значний вплив на рентабельність. Це означає, що використання CRM не тільки окупує вкладені кошти, але й приносить значний дохід за рахунок поліпшення ефективності бізнесу.

Збільшення конверсії до 300% завдяки CRM свідчить про те, що цей інструмент значно покращує управління продажами і процесами залучення клієнтів. CRM допомагає компаніям краще розуміти своїх клієнтів, пропонувати персоналізовані послуги й активніше залучати потенційних клієнтів, що призводить до більшої кількості успішних угод.

Цифри гарні, але незрозуміло, звідки вони беруться? Давайте розберемося, наскільки великий ефект від впровадження CRM-системи і за рахунок чого він досягається.

Масштаби економічного ефекту від впровадження CRM-системи найкраще оцінювати через реальні кейси компаній і галузеві дослідження. Тому я звернувся до профільного порталу «Практика CRM», який проаналізував досвід сотень компаній і вивів цікаву статистику.

За його даними, витрати на впровадження CRM відбиваються в термін від 3 місяців (мінімум). А якщо взяти для прикладу компанію з оборотом 1 млн на місяць, то при введенні CRM очікуваний ефект складатиме від 80 до 180 тисяч гривень на місяць – як додаткового прибутку.

Однак, CRM-система сама по собі не генерує прибуток, але вона створює можливості для підвищення ефективності, автоматизації процесів та поліпшення взаємодії з клієнтами, що в кінцевому підсумку впливає на зростання доходів компанії. Розглянемо, як саме CRM допомагає компаніям оптимізувати роботу:

#### 1. Покращує управління продажами

CRM дозволяє відстежувати кожен етап продажів, від першого контакту до завершення угоди. Це допомагає менеджерам не пропустити важливі можливості та працювати з максимальною ефективністю. Система автоматично фіксує всі взаємодії з клієнтами, полегшуючи прийняття рішень щодо наступних кроків.

#### 2. Збільшує рівень утримання клієнтів

CRM допомагає компаніям краще розуміти потреби та поведінку клієнтів, що дає змогу пропонувати персоналізовані рішення та підтримувати більш тісні відносини. Автоматизовані нагадування про важливі події (закінчення контракту, дні народження) допомагають зберігати лояльність клієнтів.

#### 3. Автоматизує рутинні процеси

Завдяки автоматизації рутинних завдань, таких як управління контактами, обробка заявок або ведення звітності, працівники мають більше часу для стратегічної роботи, що призводить до підвищення продуктивності.

Зниження людських помилок через автоматизовані процеси підвищує точність і ефективність.

#### 4. Оптимізує маркетингові кампанії

CRM дозволяє сегментувати базу клієнтів за різними критеріями (поведінка, інтереси, попередні покупки), що дає можливість розробляти більш таргетовані та персоналізовані маркетингові кампанії. Система автоматично аналізує результати кампаній, що допомагає оперативно коригувати стратегії та покращувати їх ефективність.

#### 5. Підвищує якість обслуговування клієнтів

CRM забезпечує повну історію взаємодій із кожним клієнтом, що дозволяє швидше та якісніше вирішувати проблеми або запити. Відділи підтримки клієнтів можуть більш оперативно реагувати на запити, що покращує загальний досвід клієнтів.

#### 6. Прогнозує продажі і полегшує планування

Завдяки аналітиці, CRM допомагає прогнозувати майбутні продажі на основі історичних даних, що дозволяє компаніям планувати свою діяльність і ресурси більш точно. Візуалізація даних через звіти й діаграми допомагає керівникам приймати обґрунтовані рішення. Як це виглядає в CRM-системі SalesapCRM показано на рисунку 1.2.

#### 7. Координує команду

CRM дозволяє різним відділам компанії (продажі, маркетинг, підтримка) працювати в єдиній системі, забезпечуючи кращу координацію та доступ до актуальної інформації. Це підвищує ефективність комунікації та зменшує кількість внутрішніх затримок.

#### 8. Інтегрується з іншими бізнес-системами

Сучасні CRM можуть інтегруватися з бухгалтерськими програмами, інструментами управління проектами, аналітичними платформами та іншими бізнес-додатками. Це дозволяє автоматизувати більшість процесів компанії, покращуючи їхню синхронізацію та управління.

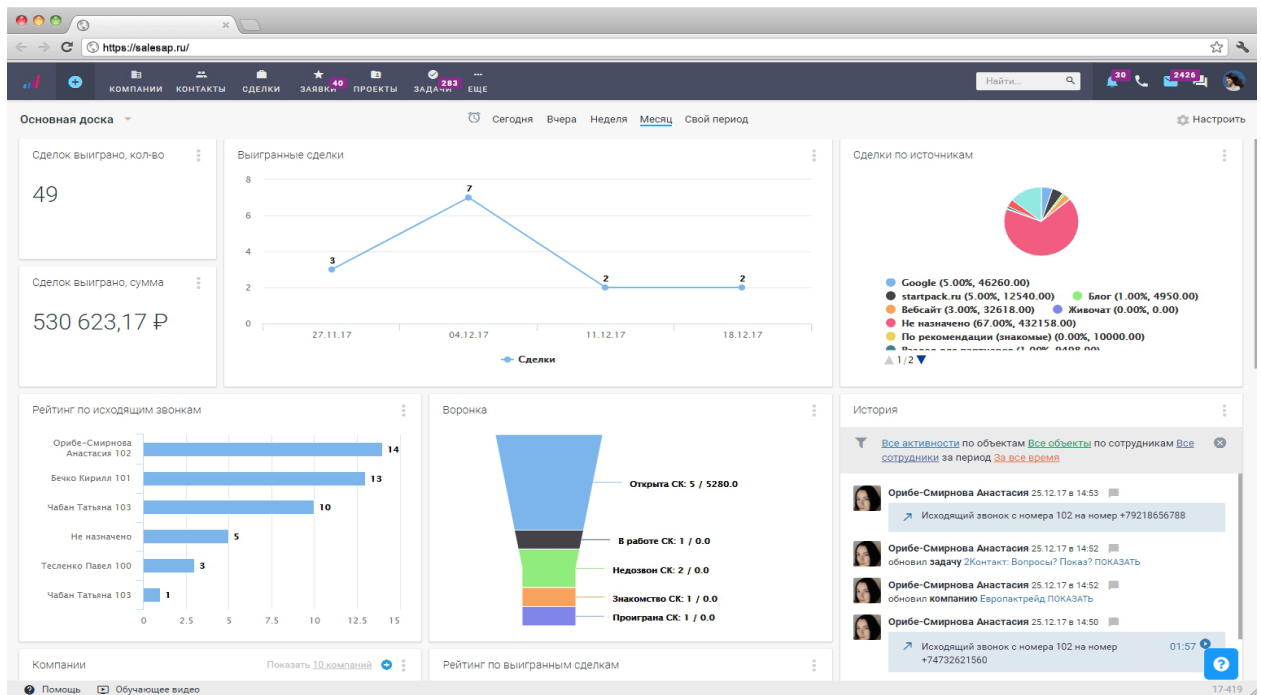


Рис. 1.2 – Аналітика в CRM-системі SalesapCRM

CRM-система виступає інструментом, який допомагає компанії ефективніше організувати свою роботу, автоматизувати рутинні завдання та оптимізувати управління взаєминами з клієнтами. Це не лише дозволяє зменшити витрати, а й сприяє збільшенню доходів за рахунок покращення якості обслуговування, підвищення продажів та покращення маркетингових стратегій.

Розглянемо деякі приклади економічного ефекту від впровадження CRM.

Краще один раз побачити, ніж сотні разів почути, тому ми пропонуємо оцінити економічний ефект від впровадження CRM на прикладі реальних компаній.

– Логістичний оператор PONY EXPRESS

Компанія вирішила налагодити комунікацію менеджерів з клієнтами. Після впровадження CRM цикл продажів по телефону скоротився з 60 до 25

днів, а термін укладення договорів скоротився з 24 до 3 годин. При цьому рівень задоволеності клієнтів, як і рівень утримання клієнтів виріс на 5%.

– Компанія з продажу та обслуговування авто «БЦР МОТОРС»

Компанія, що займається продажем автомобілів вирішила підвищити ефективність роботи відділу продажів, забезпечити ефективну роботу call-центру і поліпшити взаємодію між підрозділами. Після впровадження CRM продажі зросли на 40%, число втрачених клієнтів знизилося в 2 рази, час на паперові процедури знизилося в 3 рази, швидкість обслуговування клієнтів підвищилася в 2 рази.

– Дослідницько-консультантська компанія FORRESTER

Компанія вирішила систематизувати інформацію про клієнтів, зробити зручний доступ до цих даних і скоротити зайві операції працівників. Після впровадження CRM-системи окупність інвестицій в неї склала 302%, у працівників вивільнилися по 53 зайвих хвилини кожен день, а прибуток виріс на 1,8%, що склало 17 мільйонів доларів.

### 1.5 Висновки за першим розділом

В ході роботи було обґрунтовано об'єкт дослідження, детально розібрані переваги та недоліки. Також були приведені реальні приклади економічного ефекту від впровадження CRM різних компаній, їх статистичні данні до, та після впровадження. Було доказано фактами та підтверджено аналітикою, що автоматизована система управління користувачами та контентом є потужним інструментом, який може значно спростити і покращити роботу компанії з контентом та клієнтами. Інакше кажучи, автоматизована система управління користувачами та контентом дозволяє компаніям більш ефективно працювати з інформацією, підвищувати залученість користувачів і швидко реагувати на їхні потреби і загалом спрощувати процес створення та управління контентом.

## РОЗДІЛ 2

### ВИКОРИСТАНІ ІНСТРУМЕНТИ ДЛЯ РОЗРОБКИ

#### 2.1 Середовище розробки

Для розробки було обрано IDE – PhpStorm 2020, оскільки для розробки використовувалася мова програмування Php 7.3. PhpStorm – це потужне середовище розробки (IDE) від JetBrains, призначене для роботи з PHP, а також іншими мовами програмування та вебтехнологіями.

Основні переваги PhpStorm, які роблять його популярним серед розробників:

1. Широка підтримка PHP. PhpStorm підтримує всі сучасні функції PHP, включаючи найновіші версії мови, що дозволяє використовувати найсучасніші інструменти і фреймворки. Він забезпечує інтелектуальну підсвітку коду, автодоповнення, рефакторинг та перевірку коду на наявність помилок, що значно полегшує розробку.

2. Підтримка інших мов і технологій. Крім PHP, PhpStorm підтримує HTML, CSS, JavaScript, TypeScript, SQL та багато інших мов, що робить його універсальним інструментом для фронтенд і бекенд розробки. Інтеграція з популярними бібліотеками та фреймворками (React, Angular, Vue.js), а також із CSS-препроцесорами (SASS, LESS).

3. Розширені можливості для роботи з базами даних. PhpStorm має вбудовану підтримку для роботи з базами даних, дозволяючи виконувати SQL-запити безпосередньо з IDE, переглядати структуру баз даних та працювати з ними у зручному графічному інтерфейсі. Це зручно для розробників, які працюють з базами даних, такими як MySQL, PostgreSQL, Oracle та іншими. Повний функціонал PhpStorm включений в PhpStorm, а також додана повноцінна підтримка PHP і підтримка баз даних / SQL. Основні підтримувані компоненти показані на рисунку 2.1.





Рис. 2.1 – Основні підтримувані компоненти

4. Інтеграція з системами контролю версій. PhpStorm має повну інтеграцію з Git, SVN, Mercurial та іншими системами контролю версій, що дозволяє зручно керувати репозиторіями прямо з середовища розробки. Вбудовані інструменти для злиття гілок, перегляду історії комітів та вирішення конфліктів.

5. Підтримка фреймворків. PhpStorm підтримує найпопулярніші PHP-фреймворки, такі як Laravel, Symfony, Yii, Zend Framework, а також системи керування контентом (CMS), як-от WordPress, Drupal, Magento. IDE надає спеціальні інструменти для спрощення роботи з цими фреймворками, автоматизуючи багато завдань, таких як налаштування маршрутизації або автозавантаження класів.

6. Підтримка інструментів тестування. PhpStorm має інтеграцію з PHPUnit, Behat, Codeception та іншими інструментами для тестування, що дозволяє легко писати, запускати й відслідковувати тести прямо в IDE. Це допомагає підвищити якість коду та виявляти помилки на ранніх етапах розробки.

7. Швидкість і продуктивність. PhpStorm відомий своєю швидкістю та високою продуктивністю навіть на великих проєктах із великою кількістю

файлів. Він оптимізований для швидкого завантаження проєктів і роботи з великими обсягами коду.

8. Інтеграція з інструментами розгортання. IDE підтримує інтеграцію з різними системами розгортання, включаючи FTP, SFTP, SSH, а також хмарні платформи. Це дозволяє легко розгортати додатки або виконувати віддалену налагодження безпосередньо з середовища розробки.

9. Інтелектуальні інструменти рефакторингу. PhpStorm пропонує інструменти для безпечного рефакторингу коду, включаючи перейменування класів, функцій, змінних та автоматичне оновлення посилань на них по всьому проєкту. Це значно зменшує ризик виникнення помилок під час змін у великому коді.

10. Підтримка інтеграцій з Docker та іншими контейнеризаційними платформами. PhpStorm дозволяє інтегруватися з Docker, що дає змогу створювати, керувати та тестувати додатки в контейнерах безпосередньо з IDE, спрощуючи налаштування робочого середовища та зменшуючи час на розгортання.

PhpStorm – це комплексне середовище розробки, яке надає широкі можливості для PHP-програмістів і веброзробників загалом. Висока продуктивність, потужні інструменти для роботи з кодом і багатофункціональність роблять його ідеальним вибором для професійних команд і індивідуальних розробників.

## 2.2 Локальний сервер

Apache HTTP Server (або просто Apache) – це одне з найпопулярніших і найбільш використовуваних веб-серверних програмних забезпечень у світі. Він відкритий, кросплатформний і безкоштовний, розроблений та підтримуваний Apache Software Foundation. Розглянемо його особливості, переваги та недоліки.

Основні особливості Apache:

1. Модульна архітектура. Apache використовує модульну архітектуру, що дозволяє додавати або вимикати функціональні модулі залежно від потреб проєкту. Наприклад, можна використовувати модулі для обробки різних мов програмування (PHP, Python) або SSL для забезпечення безпеки.

2. Підтримка різних операційних систем. Apache може працювати на більшості популярних операційних систем, таких як Linux, Windows, macOS та Unix. Це робить його універсальним для різних серверних середовищ.

3. Підтримка різних протоколів. Крім стандартного протоколу HTTP, Apache підтримує HTTPS для безпечного з'єднання, а також інші протоколи, такі як FTP і HTTP/2.

4. Висока гнучкість налаштувань. Користувачі можуть детально налаштовувати конфігураційні файли для оптимізації продуктивності, безпеки, керування ресурсами, логування та інших аспектів вебсервера.

5. Підтримка віртуальних хостів. Apache дозволяє хостити кілька вебсайтів на одному сервері, використовуючи віртуальні хости (Virtual Hosts). Це зручно для хостинг-провайдерів або компаній, які управляють кількома проєктами на одному сервері.

6. Масштабованість. Apache добре масштабується на середніх і великих проєктах. Його можна налаштувати для роботи на кластерах серверів і використовувати з різними балансувальниками навантаження.

7. Підтримка SSL/TLS. За допомогою модулів (наприклад, mod\_ssl), Apache дозволяє забезпечити захищені з'єднання через SSL/TLS, що необхідно для безпеки даних і дотримання сучасних стандартів захисту.

8. Велика спільнота та підтримка. Оскільки Apache існує понад 20 років і є відкритим проєктом, він має величезну спільноту користувачів і розробників, що забезпечує регулярні оновлення, виправлення помилок і безліч навчальних матеріалів.

Переваги Apache цілком очевидні:

1. Безкоштовний та відкритий код. Apache – це безкоштовне програмне забезпечення з відкритим кодом, що дозволяє будь-кому використовувати, змінювати та розповсюджувати його.

2. Підтримка великої кількості модулів. Користувачі можуть додавати різноманітні модулі, такі як `mod_rewrite` для перенаправлення URL-адрес, `mod_security` для захисту або `mod_cache` для кешування контенту, що розширює можливості вебсервера.

3. Гнучкість у налаштуванні. Apache забезпечує безліч варіантів конфігурації та налаштувань, що дозволяє створювати індивідуальні рішення для різних потреб. Можна керувати доступом до ресурсів, контролювати кешування та маршрутизацію запитів.

4. Сумісність із багатьма технологіями. Apache добре працює з популярними мовами програмування (PHP, Python, Ruby), системами баз даних (MySQL, PostgreSQL) та іншими серверними програмами.

5. Надійність і стабільність. Apache є дуже стабільним і надійним рішенням для вебсайтів будь-якого розміру. Він використовується великими організаціями та має довгу історію безперебійної роботи.

6. Підтримка багатопоточності та балансування навантаження. Apache дозволяє обробляти велику кількість одночасних з'єднань через різні робочі моделі, такі як `worker`, `event` або `prefork`, що дозволяє краще керувати ресурсами сервера.

7. Безпека. Apache активно підтримується і оновлюється для виправлення вразливостей. Модулі безпеки, такі як `mod_ssl`, `mod_security` та інші, дозволяють налаштувати захист вебсайтів на високому рівні.

Незважаючи на безліч переваг, Apache має певні недоліки, зокрема:

1. Продуктивність на високих навантаженнях. Apache може працювати повільніше в порівнянні з новішими вебсерверами (наприклад, Nginx) під час обробки великої кількості одночасних запитів. Його багатопроцесорна модель не завжди оптимально використовує ресурси під високим навантаженням.

2. Складність конфігурації. Хоча Apache пропонує велику гнучкість, його налаштування можуть бути досить складними для новачків. Для оптимальної роботи потрібне глибоке розуміння конфігураційних файлів і роботи модулів.

3. Споживання ресурсів. Apache може використовувати більше пам'яті та процесорних ресурсів порівняно з іншими вебсерверами, особливо в режимі `prefork`, де кожен запит обробляється окремим процесом.

4. Повільніше оновлення. Оскільки Apache є зрілим проєктом, швидкість його оновлення може бути нижчою порівняно з новішими вебсерверами, такими як Nginx або LiteSpeed, які швидше адаптуються до нових вимог ринку.

5. Не ідеальний для статичних сайтів. Якщо вебсайт складається переважно зі статичних файлів (наприклад, HTML, зображення), Apache може бути не найкращим вибором через його архітектуру, яка краще підходить для динамічних сайтів.

Таким чином, Apache – це надійний і перевірений часом вебсервер, який підходить для різних типів проєктів, від невеликих вебсайтів до великих корпоративних додатків. Він пропонує багату функціональність і гнучкість завдяки модульній архітектурі та підтримці різних платформ. Однак у випадках, коли важливі продуктивність на високих навантаженнях або ефективна обробка статичних файлів, можуть знадобитися додаткові налаштування або використання альтернативних рішень, таких як Nginx.

### 2.3 Менеджер залежностей

Composer – це інструмент, яким користуються усі. Менеджер пакетів і залежностей, який, відносно не так давно зробив революцію у світі розробки, дав величезні можливості розробникам і спростив їх життя.

Це інструмент є менеджером залежностей, який дає можливість розробникам створювати або використовувати окремі розширення,

компоненти гнучкіше та зручніше. Він допомагає розв'язати проблему бракуючих залежностей, у випадку якщо для роботи проекту потрібно сторонні бібліотеки, які треба завантажити та встановити.

Раніше це викликало проблеми і доставляло незручності, через те, що усі дії з установки доводилося робити вручну, додаючи усі залежності. І це без урахування різниці у версіях, коли необхідний розробникові віджет вимагав установки актуальної версії бібліотеки, і доводилося все оновлювати вручну.

Composer призначений для автоматичної установки бібліотек. Установка відбувається таким чином, що бібліотека встановлюється всередину кожного проекту окремо, а не глобально.

Припустимо є певний проект, що залежить від бібліотек, які в свою чергу залежать від інших бібліотек. Для установки всіх залежностей, досить описати їх у файлі «composer.json» і запустити виконання скрипта. Далі Composer сам завантажить і встановить останні версії описаних раніше бібліотек. Такий метод хороший тим, що якщо розробник вирішить опублікувати цей проект на якийсь ресурс, наприклад github, або зробити комміт в системі контролю версій, то йому достатньо завантажити в репозиторій саму папку зі скриптами і «composer.json», в якому будуть описані бібліотеки, необхідні для роботи цього проекту. Тобто, немає необхідності завантажувати в репозиторій всі файли бібліотек, вони завантажаться самі при запуску скрипта. Такий проект буде займати менше місця і їм буде простіше ділитися.

## 2.4 Відладчик

Відладка програми – це процес виявлення і усунення багів. Наприклад, якщо треба знати значення змінної на певному етапі виконання програми, подивитися чи виконалася умова або ж запит, і навіть перевірити усі етапи рекурсії або циклу. У цьому допомагає відладчик.

Xdebug – це розширення для відладки PHP-коду, яке має гнучкі і просунуті інструменти.

Функціонал, який надає Xdebug:

- Трасування стека – поетапне виведення шляху, який привів додаток до якої-небудь помилки, включаючи параметри, передані у функції. Для зручності пошуку всі дії відбуваються по порядку.
- Приємніше виведення `var_dump`, яке створює підсвічування синтаксису коду і структурований вид разом з дампом суперглобальних змінних.
- Профайлер для пошуку вузьких місць коду з можливістю візуалізованого представлення графіків продуктивності зовнішніми інструментами.
- Віддалений відладчик, який може бути використаний при з'єднанні з Xdebug для запуску та виконання коду в IDE або браузері порядкове через брейк-пойнти. Найчастіше це плагін для браузера, який з'єднується з IDE.
- Наочність коду показує яка частина була виконана в процесі запиту. Це функція потрібна здебільшого для тестів і отримання інформації про те наскільки добре ваш код покритий тестами.

У моєму випадку Xdebug працює в зв'язці з PhpShtorm і Apache2. Налаштування відбувалося таким чином:

1. Включення розширення Xdebug у файлі конфігурації PHP.
2. Далі потрібно розкоментувати рядки, показані на рисунку 2.2.

```
zend_extension="%sprogdir%/modules/php/%phpdriver%/ext/php_xdebug.dll"  
xdebug.remote_enable = 1  
xdebug.remote_port = 9000  
xdebug.remote_handler = "dbgp"  
xdebug.remote_connect_back = 1  
xdebug.idekey = "PHPSTORM"
```

Рис. 2.2 – Конфігурація Xdebug

3. Налаштування відладки в IDE PhpStorm. Треба зайти в меню налаштувань «Settings» і перейти в розділ «Languages & Frameworks -> PHP -> Debug» і вказати наступні налаштування:

- Debug port: 9000.
- Force break at first line when no path mapping specified: прибрати галочку.
- Force break at first line when a script is outside the project: прибрати галочку.

Також треба додати конфігурацію для Xdebug зі значеннями з OpenServer. У моєму випадку були значення, що приведені на рисунку 2.3.

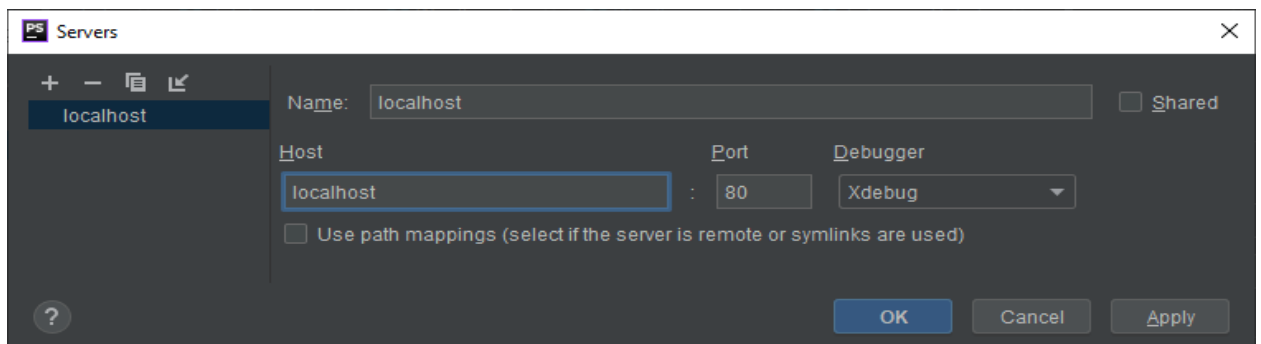


Рис. 2.3 – Конфігурація сервера для Xdebug

Щоб зв'язати Xdebug з розширенням у браузері, треба в пункті «Ide key (session id)» прописати ключ: PHPSHTORM.

4. Налаштування відладки в розширенні для браузера Google Chrome.
5. Треба перейти на сторінку офіційних розширень для браузера і викачати плагін – Xdebug helper. Далі в налаштуваннях встановити ключ який показаний на рисунку 2.4.

### IDE key

Select your IDE to use the default sessionkey or choose other to use a custom key.

PhpStorm ▼ PHPSTORM Save

Рис. 2.4 – Установка ключа в розширенні для браузера



Після цього етап налаштування і зв'язки відладчика Xdebug завершені.

## 2.5 Система контролю версій

Система контролю версій – це система, яка запам'ятовує зміни у файлах проекту для відновлення їх початкового стану у разі потреби. Наприклад, щоб не витратити велику кількість часу на пошуки дрібної помилки, яка приводить увесь код або ділянку в непрацездатний стан, то легше відкотити останні зміни і наново продовжити розробку з минулого місця. Git – це розподілена система контролю версій (DVCS), яка дозволяє розробникам відстежувати зміни в коді, працювати над різними версіями проекту одночасно і зберігати історію всіх модифікацій. Вона широко використовується в програмній розробці завдяки своїй ефективності та гнучкості.

Основні особливості Git наступні:

1. Розподілена система контролю версій. Кожен користувач Git має повну копію всього репозиторію, включаючи всю історію змін, що дозволяє працювати в автономному режимі без необхідності постійного підключення до центрального сервера.

2. Гілкування та злиття (branching and merging). Git дозволяє легко створювати гілки для паралельної роботи над різними функціями або виправленнями. Гілки можуть бути створені швидко та ефективно, а злиття різних гілок зазвичай проходить без проблем.

3. Швидкість і ефективність. Git спроектований для роботи з великими проектами та дозволяє виконувати більшість операцій дуже швидко, оскільки працює локально на машині розробника.

4. Збереження історії змін. Git зберігає історію всіх змін, дозволяючи переглядати попередні версії файлів і відстежувати, хто, коли і які зміни вносив.

5. Підтримка віддалених репозиторіїв. Git дозволяє працювати з віддаленими репозиторіями (наприклад, на GitHub, GitLab, Bitbucket), що спрощує співпрацю в команді. Розробники можуть синхронізувати свої зміни, обмінюватися гілками та об'єднувати їх.

6. Хешування за допомогою SHA-1. Кожен коміт у Git ідентифікується унікальним хешем (SHA-1), що забезпечує надійний механізм перевірки цілісності даних та відстеження змін.

7. Система індексації. Git має систему індексації (staging area), що дозволяє користувачам підготувати (stage) певні файли або частини файлів для коміту, даючи змогу більш точно керувати змінами.

8. Підтримка сценаріїв і гачків (hooks). Git підтримує сценарії (scripts) для автоматизації різних процесів, таких як тестування коду або відправка повідомлень після коміту, що підвищує ефективність і продуктивність розробки.

#### Переваги Git:

1. Розподілена модель роботи. Кожен розробник має повну копію репозиторію, тому можна працювати без підключення до мережі або сервера. Це зручно для роботи в автономному режимі та зберігає всю історію змін.

2. Швидке виконання операцій. Більшість операцій у Git, таких як створення гілок, злиття та коміти, виконуються швидко, оскільки всі дані обробляються локально.

3. Ефективна система гілкування. Легке створення гілок для роботи над різними функціями або виправленнями, а також потужні інструменти для злиття дозволяють команді розробників працювати паралельно без конфліктів.

4. Велика кількість інструментів та інтеграцій. Існує багато інструментів для роботи з Git (GitHub, GitLab, Bitbucket), що забезпечують зручний інтерфейс, можливості для автоматизації і DevOps, а також інтеграції з системами управління проектами та CI/CD.

5. Безпека та цілісність даних. Кожна зміна в Git фіксується за допомогою хешування, що забезпечує високу безпеку і дозволяє відстежувати всі зміни без можливості їх підробки або втрати даних.

6. Співпраця в команді. Git дозволяє ефективно працювати в команді через віддалені репозиторії, надаючи можливість синхронізувати зміни і працювати з різними гілками одночасно.

7. Відкрита і безкоштовна система. Git – це безкоштовне програмне забезпечення з відкритим вихідним кодом, що дозволяє будь-кому використовувати його та вносити зміни відповідно до потреб.

Недоліки Git:

1. Крива навчання для новачків. Git може бути складним для початківців, особливо через наявність багатьох команд, концепцій і складніших операцій, таких як вирішення конфліктів або робота з віддаленими репозиторіями.

2. Потенційно складне злиття гілок. Хоча Git потужний у злитті, в разі серйозних конфліктів злиття (merge conflicts) може вимагати значного часу і ручної роботи для вирішення цих конфліктів.

3. Великі репозиторії можуть уповільнити роботу. У великих проєктах з дуже великою кількістю файлів та історії змін продуктивність Git може знижуватися. Це призводить до повільнішого клонування або виконання деяких операцій.

4. Вимагає додаткових інструментів для простого використання. Git, як правило, використовується через командний рядок, що може бути складним для нових користувачів. Хоча існують графічні інтерфейси для роботи з Git, вони можуть не покривати всіх можливостей командної роботи.

5. Відсутність централізації. Через те, що Git є розподіленою системою, важко встановити централізований контроль над репозиторієм або примусити певні робочі процеси на рівні всіх учасників команди, як це можна зробити в централізованих системах контролю версій.

Відтак, Git – це потужна і гнучка система контролю версій, яка підходить для проєктів будь-яких розмірів. Вона дозволяє ефективно співпрацювати над кодом і зберігати повну історію змін. Проте вона може вимагати значного навчання для новачків і бути складною в деяких аспектах, таких як вирішення конфліктів або робота з великими репозиторіями. Завдяки своїм перевагам Git є стандартом в індустрії програмної розробки.

## 2.6 Інструмент Webpack

Webpack – це потужний та гнучкий інструмент для збірки модулів JavaScript, CSS, зображень та інших ресурсів на стороні клієнта (frontend). Він широко використовується для розробки вебдодатків, дозволяючи легко керувати залежностями та оптимізувати продуктивність проєктів. Webpack використовує концепцію модулів, де кожен файл вважається модулем, і об'єднує всі ці модулі в один або кілька вихідних файлів (bundle).

Основні особливості Webpack:

1. Модульна система. Webpack дозволяє працювати з модулями JavaScript (ES6, CommonJS, AMD) та іншими типами файлів (CSS, зображення) як модулями. Він об'єднує всі модулі в єдиний або кілька бандлів, що значно спрощує роботу з великими проєктами.

2. Лоадери (loaders). Лоадери дозволяють Webpack обробляти різні типи файлів (крім JavaScript). Наприклад, за допомогою CSS-лоадера можна імпортувати стилі безпосередньо в JavaScript-код, або за допомогою Babel-лоадера трансформувати сучасний код ES6+ у код, сумісний зі старими браузерами.

3. Плагіни (plugins). Плагіни розширюють функціональність Webpack. Наприклад, вони можуть мінімізувати файли JavaScript, оптимізувати зображення, витягувати CSS у окремі файли або генерувати HTML-

документи. Вони є потужним інструментом для налаштування складних процесів збірки.

4. Кешування та гаряче завантаження модулів (Hot Module Replacement). Webpack підтримує гаряче завантаження модулів, що дозволяє оновлювати модулі без перезавантаження сторінки під час розробки. Це прискорює процес розробки та покращує досвід роботи з кодом у реальному часі.

5. Код-сплітинг (Code Splitting). Webpack дозволяє розділяти код на менші частини, так звані "chunks", які можна завантажувати асинхронно за потреби. Це допомагає зменшити початковий час завантаження сторінки, адже браузер завантажує лише необхідні частини коду.

6. Tree shaking. Ця функція дозволяє видаляти не використовуваний код із бандла, що допомагає мінімізувати розмір файлів і покращити продуктивність. Webpack аналізує код і прибирає непотрібні залежності.

7. Підтримка кількох вихідних точок (Entry Points). Webpack дозволяє налаштовувати кілька вихідних точок для проєкту, що корисно для багатосторінкових застосунків, коли кожна сторінка може мати свої скрипти.

9. Конфігураційні файли. Webpack використовує конфігураційні файли (webpack.config.js), що дає змогу налаштовувати процес збірки проєкту з великою гнучкістю. Тут можна налаштувати плагіни, ладери, вказати вихідні точки, цілі (production чи development) тощо.

Переваги Webpack:

1. Універсальність. Webpack працює з різними типами ресурсів – JavaScript, CSS, зображення, шрифти тощо. Він об'єднує всі залежності в один пакет або кілька пакетів, що робить його універсальним рішенням для збірки сучасних фронтенд-додатків.

2. Гнучкість налаштувань. За допомогою плагінів і ладерів Webpack можна налаштувати під конкретні потреби проєкту, що робить його дуже гнучким для будь-якого типу додатків.

3. Оптимізація продуктивності. Webpack автоматично мінімізує файли, виконує код-сплітинг і видаляє зайві модулі (Tree shaking), що допомагає зменшити розмір бандлів і підвищити швидкість завантаження сторінок.

4. Активне ком'юніті. Webpack має велике і активне ком'юніті, яке постійно випускає нові плагіни, оновлення і підтримку нових технологій. Це робить Webpack надійним і постійно вдосконаленим інструментом.

5. Інтеграція з різними фреймворками. Webpack легко інтегрується з популярними фреймворками, такими як React, Angular, Vue, і допомагає автоматизувати збірку проєктів на їх основі.

6. Підтримка сучасних стандартів. Webpack підтримує нові стандарти JavaScript (ES6+, модулі), що дозволяє розробникам використовувати найновіші функції без турбот про сумісність з різними браузерами.

#### Недоліки Webpack:

1. Складність налаштувань. Webpack може бути важким для розуміння та налаштування, особливо для новачків. Його конфігураційний файл може стати досить складним, якщо проєкт великий або містить багато специфічних налаштувань.

2. Крива навчання. Новачкам у світі збірників може бути важко зрозуміти всі аспекти Webpack і його функції, такі як ладери, плагіни, entry points, output тощо.

3. Повільний процес збірки для великих проєктів. У дуже великих проєктах Webpack може працювати повільніше, особливо під час гарячого завантаження або збірки проєкту з великою кількістю модулів. Для цього можна використовувати плагіни для прискорення процесу, але це вимагає додаткової конфігурації.

4. Потреба в постійному налаштуванні для оптимізації. Для досягнення максимальної продуктивності Webpack потребує тонких налаштувань і оптимізації, що може вимагати додаткового часу і знань.

Принцип роботи Webpack зображений на рисунку 2.5.

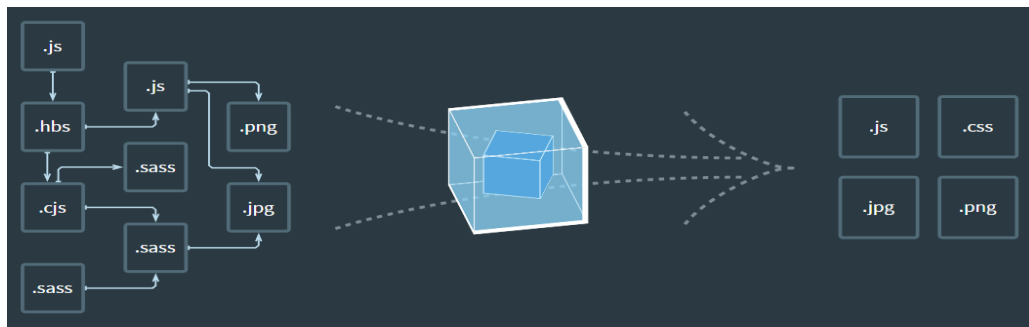


Рис 2.5 – Принцип роботи Webpack

Вибір Webpack обумовлений саме тим, що це один із найпотужніших інструментів для збірки frontend-проектів, що надає безліч можливостей для роботи з модулями, оптимізації продуктивності та управління залежностями. Він відмінно підходить для великих і складних проектів, однак може мати деякі труднощі в освоєнні через складність налаштувань.

Незважаючи на це, його переваги в універсальності, продуктивності та підтримці сучасних стандартів роблять його невід’ємною частиною більшості сучасних вебдодатків.

## 2.7 Менеджер пакунків NPM

Менеджер пакунків NPM (Node Package Manager) є ключовим інструментом у світі веб-розробки, особливо для тих, хто працює з JavaScript і платформою Node.js. Ось основні причини, навіщо потрібен NPM і його роль у процесі розробки:

NPM дозволяє легко керувати сторонніми бібліотеками та модулями, які використовуються в проекті. Він автоматично завантажує, оновлює та встановлює всі необхідні пакети, зазначені в файлі `package.json`, щоб проект працював коректно.

Замість того, щоб писати весь код вручну, розробники можуть використовувати тисячі готових бібліотек і модулів з репозиторію NPM. Це

значно прискорює розробку, оскільки не потрібно винаходити велосипед, а можна використовувати вже готові рішення. NPM надає простий і зрозумілий інтерфейс для встановлення пакетів командою `npm install`, після чого всі залежності автоматично завантажуються і готові до використання в проєкті. Це усуває необхідність ручного завантаження і інтеграції стороннього коду. NPM дозволяє контролювати версії встановлених пакетів, що допомагає уникнути проблем з сумісністю. Розробники можуть визначати конкретні версії пакетів, які слід використовувати, або дозволяти автоматичне оновлення до останніх стабільних версій.

У великих командах або при роботі над відкритими проєктами NPM полегшує координацію, оскільки всі учасники можуть встановити ті ж самі залежності і працювати з одними й тими ж версіями пакетів.

NPM дозволяє не лише використовувати сторонні пакети, але й публікувати свої власні рішення. Це робить NPM важливим інструментом для спільнот розробників, які можуть обмінюватися кодом, бібліотеками та інструментами через публічні або приватні репозиторії. NPM робить проєкти масштабованими завдяки своїй структурі модулів. Ви можете легко додавати нові функції у вигляді пакетів або модулів, не впливаючи на решту проєкту.

Крім керування пакетами, NPM дозволяє запускати різні скрипти для автоматизації рутинних завдань (наприклад, запуск серверу розробки, тестування коду, деплой). Це спрощує управління проєктом і полегшує робочі процеси.

NPM – це найбільший репозиторій JavaScript-пакетів у світі. Завдяки цьому розробники мають доступ до безлічі інструментів і бібліотек для різноманітних завдань, від обробки даних до побудови користувацьких інтерфейсів. NPM працює через командний рядок, що дозволяє автоматизувати багато процесів і інтегрувати його в різні системи без необхідності використання графічних інтерфейсів. Це робить його гнучким і універсальним інструментом.



Отже, менеджер пакунків NPM є незамінним інструментом для розробників, які працюють з Node.js і JavaScript. Він спрощує управління залежностями, полегшує співпрацю в командах, забезпечує швидкий доступ до тисяч готових рішень і значно прискорює процес розробки завдяки автоматизації і стандартизації управління пакетами.

Для завантаження, встановлення та використання фронтенд компонентів було обрано – npm як ключовий інструмент для будь-якого JavaScript-розробника, який спрощує управління залежностями, забезпечує доступ до величезної кількості бібліотек і модулів та дозволяє автоматизувати важливі аспекти розробки. Незважаючи на деякі недоліки, npm залишається одним із найпопулярніших менеджерів пакетів у світі програмної розробки.

## 2.8 PHP фреймворк

Laravel спочатку проектувався під створення багатофункціональних веб-додатків. Він може похвалитися відмінним движком шаблонів, який виконує безліч звичайних завдань. Що стосується функціоналу, потрібно відзначити Restful Routing, аутентифікацію і кешування. Завдяки цих опціям Laravel робить розробку більш швидкою.

Але є й інші причини, за якими ми радимо застосовувати цей фреймворк:

- Високий рівень безпеки. Ми вже акцентували увагу на тому, що для PHP-фреймворку безпека є дуже важливою характеристикою. В даному випадку ви будете працювати в безризиковою середовищі, тому як Laravel блокує всі шкідливі активності і володіє чудовим захистом від випадкових і прихованих впроваджень SQL-коду.

- Вбудовані поштові сервіси. За рахунок інтегрованого поштового сервісу розроблене вами веб-додаток зможе відправляти користувачам SMS і Slack повідомлення. Фреймворк має простий у використанні API, а також драйвера для Mailgun, Amazon SES, SMTP і Mandrill.

- Великий функціонал. Розробники можуть впроваджувати різні види готового функціонала завдяки пакетам, плагинам, компонентам і модулів, що дозволяє істотно скоротити час розробки.
- Покращена аутентифікація. Фреймворк дозволяє користувачам без особливих зусиль створювати логіку аутентифікації. А це, як відомо, далеко не найпростіше завдання.

## 2.9 Сервіс для email розсилок

SendGrid – це платформа, яка покращує транзакційні розсилки і дозволяє масштабувати їх для вирішення задач email-маркетингу. Вона пропонує гнучкі веб та SMTP API-інтерфейси, може легко інтегруватися з будь-якою хмарною інфраструктурою. SendGrid підтримує різні фреймворки, мови і додатки.

Основні характеристики SendGrid: SMTP-сервіс; відстеження відкриття та кліків; відмова від відстеження; кастомні інтеграції API; Шаблони повідомлень; управління списками; моніторинг репутації; виділені IP-адреси; SMTP API; моніторинг ISP; аналіз Webhook; балансування навантаження; тестування фільтра; зворотній зв'язок; DKIM налаштування; SMTP Relay; підтримка 24/7.

SendGrid – це один з провідних сервісів для email-розсилок, який широко використовується компаніями для відправки як маркетингових, так і транзакційних листів. Ось основні переваги SendGrid, які роблять його популярним вибором:

1. Висока масштабованість. SendGrid підходить як для малого бізнесу, так і для великих корпорацій. Він дозволяє надсилати невелику кількість листів або ж керувати масштабними кампаніями з мільйонами електронних листів. Його інфраструктура здатна обробляти великі обсяги трафіку без проблем із затримками або втратою даних.

2. Висока швидкість і надійність доставки. Завдяки потужній інфраструктурі та ефективному управлінню IP-адресами, SendGrid забезпечує високий рівень доставки email. Це знижує ймовірність того, що ваші листи потраплять у спам. Також сервіс автоматично аналізує, як отримувачі взаємодіють з вашими листами, і відповідно адаптує доставку, покращуючи репутацію відправника.

3. Аналітика та звітність. SendGrid пропонує розширену аналітику щодо результатів email-кампаній. Ви можете відслідковувати відкриття листів, кліки за посиланнями, відписки, доставку і навіть скарги на спам. Це дозволяє швидко аналізувати ефективність розсилки і коригувати кампанії для кращих результатів.

4. Транзакційні листи. SendGrid відмінно підходить для відправки транзакційних листів, таких як підтвердження замовлення, зміна паролю або повідомлення про доставку. Він забезпечує швидку і надійну доставку важливих повідомлень у реальному часі.

5. Зручний API. SendGrid пропонує потужний та гнучкий API, який дозволяє легко інтегрувати email-розсилки з вашими додатками чи веб-сайтами. Ви можете автоматизувати процес відправки листів і використовувати різні функції сервісу, не заходячи в його інтерфейс.

6. Інтуїтивно зрозумілий інтерфейс. SendGrid має простий і зрозумілий інтерфейс, який дозволяє легко налаштовувати та керувати кампаніями без необхідності глибоких технічних знань. Також доступний конструктор шаблонів, що дозволяє швидко створювати професійні листи без навичок програмування.

7. Підтримка різних типів листів. Сервіс підтримує як масові маркетингові кампанії, так і персоналізовані транзакційні листи. Це дозволяє комбінувати різні стратегії розсилок в рамках одного сервісу, не переходячи до інших платформ.

8. Управління контактами. SendGrid дозволяє зручно керувати базою контактів, сегментувати її та відправляти персоналізовані листи окремим групам отримувачів. Це значно підвищує ефективність кампаній і знижує ризик потрапляння в спам.

9. Висока безпека. SendGrid приділяє велику увагу безпеці. Він забезпечує захист даних клієнтів і електронних листів через шифрування, двофакторну автентифікацію і підтримку стандартів GDPR. Це важливо для компаній, що працюють з конфіденційною інформацією.

10. Підтримка та документація. SendGrid має обширну документацію та якісну підтримку, що спрощує інтеграцію та використання сервісу. Їхня команда технічної підтримки доступна для вирішення будь-яких проблем, що виникають у користувачів.

SendGrid – це потужний і надійний інструмент для email-розсилок, який забезпечує високу швидкість доставки, детальну аналітику, простоту використання та масштабованість. Завдяки своїм можливостям він підходить для різних бізнесів, від стартапів до великих корпорацій, які прагнуть покращити свої маркетингові кампанії та відправляти надійні транзакційні повідомлення.

## 2.10 Висновки за другим розділом

Для створення системи управління користувачами та контентом була вибрана мова програмування PHP 7.3 та IDE – PhpStorm 2020. У якості локального серверу було взято – Apache. Також використовувався менеджер залежностей Composer, для установки бібліотек. Для завантаження, встановлення та використання фронтенд компонентів було використано – npm. В якості СУБД обрана MySQL 5.6 тому що вона використовувалася для розробки соціальної мережі. За тестування і налагодження відповідає XDebug.

## РОЗДІЛ 3

### ОГЛЯД ІНТЕРФЕЙСУ ТА ОСНОВНІ ФУНКЦІЇ ДЛЯ РОЗРОБКИ

#### 3.1 Основний принцип роботи розробленої CRM

Принцип роботи системи управління користувачами і контентом побудований таким чином:

##### 1. Реєстрація користувача.

Спочатку користувач повинен зареєструватися і пройти різні етапи реєстрації. На панелі управління користувачами новий профіль стане доступним навіть при незакінченій реєстрації. Тобто людині досить пройти перший етап реєстрації, і після чого він може повернутися назад до використання сайту, але вже як зареєстрований користувач і пізніше заповнити профіль до кінця. Спочатку користувач проходить реєстрацію звичайного профілю, далі він має можливість створити бізнес профіль який буде прив'язаний до основного. Наступним необов'язковим кроком буде створення профілю магазину і додавання перших товарів. Залежно від кількості пройдених кроків реєстрації, користувач буде потрапляти в різні розділи панелі управління користувачами. Наприклад, якщо користувач не заповнив свій профіль до кінця або зареєструвався за допомогою гугл аккаунта, то він потрапить в розділ користувачів з незакінченою реєстрацією. Або якщо користувач не підтвердив свій аккаунт по електронній пошті, то він потрапляє в розділ неверифікованих користувачів. Такі користувачі теж повинні відображатися, так як вони все одно є потенційним клієнтом, тому що в майбутньому вони можуть повернутися на сайт.

На панелі управління користувачами доступні всілякі маніпуляції з акаунтами – редагування, видалення, зв'язок з користувачем, допомога в заповненні профілю, повторна відправка листа підтвердження аккаунта і

багато іншого. Також доступні статистичні дані і аналітика по користувачам, з можливістю експорту.

## 2. Користування сайтом.

Після реєстрації або авторизації у користувача відкривається можливість створювати публікації, ставити лайки, залишати коментарі, спілкуватися з користувачами через особисті повідомлення, створювати бізнеси і магазини з товарами та багато іншого. Все це проходить в першу чергу через панель управління контентом і користувачами. Наприклад після того як користувач створив публікацію, то вона не буде відразу опублікована. Спочатку вона потрапляє в панель адміністратора для подальшого розгляду редакторами або модерацією. Вони перевіряють контент, чи відповідає він правилами сайту та інші критерії. Далі вони вирішують публікувати пост чи ні. Також є можливість редагування публікацій, залишення побажань від адміністрації або редакторів (наприклад прохання про завантаження превью кращої якості). Теж саме відбувається і з додаванням магазинів і продуктів, спочатку вони потрапляють на панель адміністратора на розгляд.

## 3. Палень користувача.

У користувачів також є панель управління своїм контентом, профілями, продуктами та іншим. Так як користувач може реалізувати свій бізнес через соціальну мережу, він теж має свою панель керування своїм магазином, замовленнями, оплатами і т.д.

## 4. Просування.

Адміністрація має можливість не тільки управляти контентом і користувачами, а й просувати їх. Наприклад виходячи зі статистики переглядів і лайків конкретної публікації, її можна додати в розділ «Рекомендовані публікації» для користувачів. А автора публікації додати в рекомендації людей за якими потрібно стежити. Також це працює і з просуванням бізнес профілів, магазинів та продуктів. Кращі магазини і продукти потрапляють на банери і в рекомендації.

### 3.2 Головна сторінка CRM панелі

Доступ до CRM панелі та до всього функціоналу мають тільки адміністратори сайту. Головна сторінка панелі показана на рисунку 3.1.

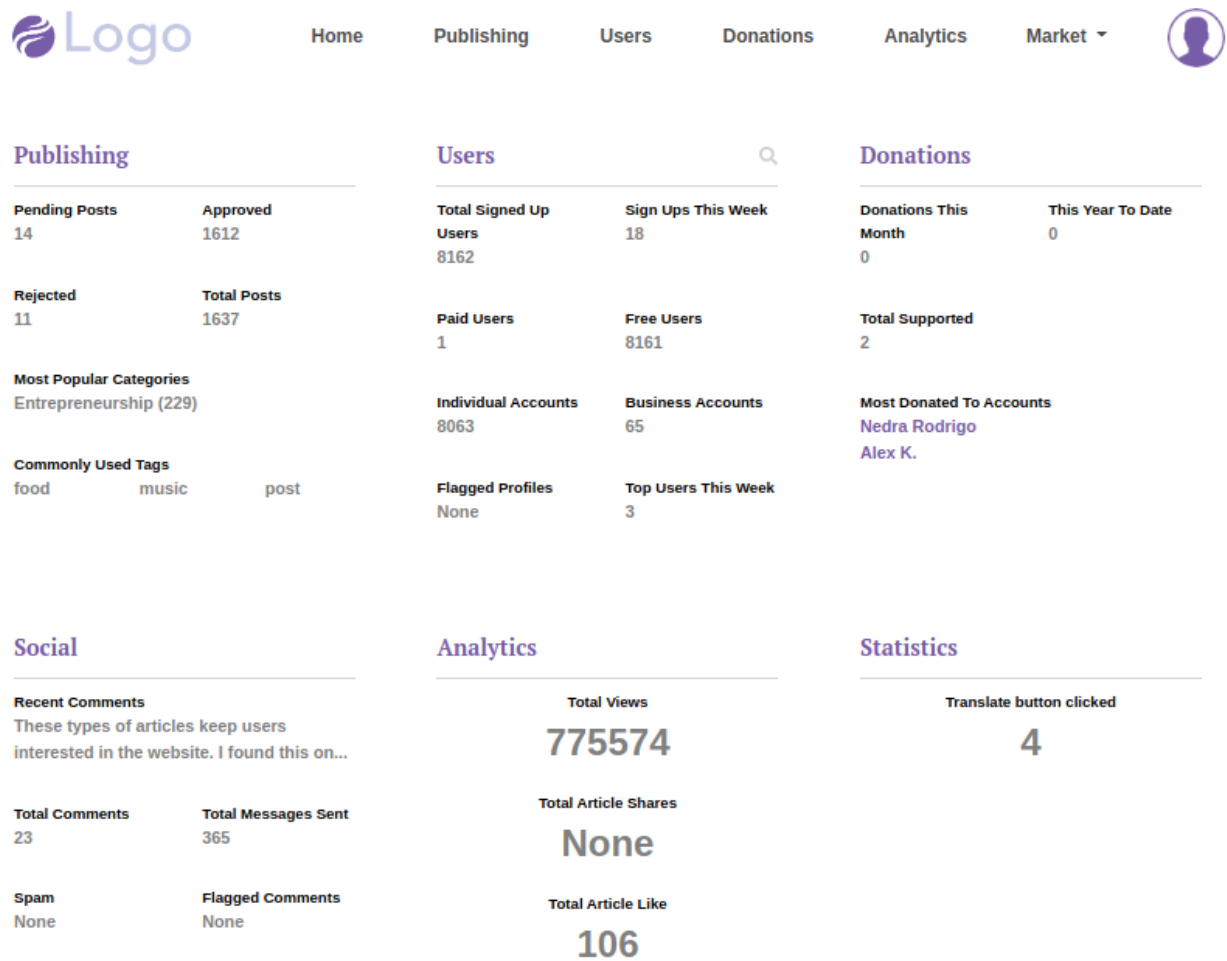


Рис. 3.1 – Голова сторінка CRM панелі

На цій сторінці знаходиться найголовніша статистика по всім розділами соціальної мережі. Як видно з малюнка вище, панель розділена на секції з навігацією та блоки з основною статистикою за різними категоріями.

Блок публікацій містить наступну статистику:

- Кількість публікацій користувачів, які чекають підтвердження адміністрацією або модераторами;

- Кількість підтверджених публікацій;
- Публікації які відкинула адміністрація або модератори;
- Кількість всіх публікацій;
- Найпопулярнішу категорію публікації;
- Найпопулярніші теги які були згадані в тексті публікацій.

Наступний блок містить інформацію про користувачів, а саме:

- Кількість зареєстрованих користувачів, включаючи тих хто ще не верифікував свій аккаунт;

- Кількість зареєстрованих користувачів за останній тиждень;
- Кількість користувачів які придбали підписку;
- Кількість звичайних користувачів;
- Кількість користувачів зі звичайним профілем;
- Кількість користувачів з бізнес профілем;
- Топ користувачів за останній тиждень. Визначається по авторам публікацій, які мають найбільше число переглядів за останній тиждень, за лайкам та іншими критеріями.

Також цьому блоці міститься пошук по користувачах.

Далі йде блок з грошовими пожертвами:

- Пожертвування за місяць;
- Пожертвування за рік;
- Кількість унікальних користувачів які вчинили пожертвування;
- Імена та посилання на профілі користувачів які вчинили пожертвування;

Після знаходиться соціальний блок який відображає наступне:

- Недавні коментарі;
- Кількість коментарів;
- Кількість відправлених повідомлень;
- Наявність спаму;



- Зазначені коментарі;

Наступний блок включає в себе аналітику про загальну кількість переглядів публікацій, кількості публікацій якими поділилися та про максимальну кількість лайків яку було зафіксовано на унікальній публікації.

Останній блок включає в себе статистику кількості натискань на кнопку перекладу сайту.

До CRM панелі сайту мають доступ тільки ті користувачі які мають роль адміністратора. Для цього було об'єднано всі роути панелі адміністраторів в групу admin і використовував middleware які входять до складу фреймворка Laravel. Група роутів продемонстрована на рисунку 3.2.

```

/** Admin only **/
$route->group(['prefix' => 'admin', 'as' => 'admin.', 'middleware' => ['role:admin']], function () use ($router) {

    //Admin panel
    $router->get( uri: '/', action: "AdminController@index")->name( name: 'admin');

    //Posts
    $router->get( uri: '/posts', action: "AdminController@posts")->name( name: 'posts');
    $router->get( uri: '/posts_pending', action: "AdminController@posts_pending")->name( name: 'posts_pending');
    $router->get( uri: '/posts_approved', action: "AdminController@posts_approved")->name( name: 'posts_approved');
    $router->get( uri: '/posts_rejected', action: "AdminController@posts_rejected")->name( name: 'posts_rejected');
    $router->get( uri: '/publish_calendar', action: "AdminController@publish_calendar")->name( name: 'publish_calendar');
    $router->get( uri: '/publish_calendar/details/{date}', action: "AdminController@publish_calendar_details")->name( name: 'publish_calendar_details');
    $router->get( uri: '/posts_search', action: "AdminController@posts_search")->name( name: 'posts_search');
    $router->get( uri: '/posts_search_form', action: "AdminController@posts_search_form")->name( name: 'posts_search_form');
    $router->get( uri: '/posts/{id}', action: "AdminController@post_edit")->name( name: 'post_edit');
    $router->post( uri: '/edit_post', action: "AdminController@post_edit_submit")->name( name: 'post_edit_submit');
    $router->post( uri: '/reject_post', action: "AdminController@reject_post")->name( name: 'reject_post');
    $router->post( uri: '/delete_post/{id}', action: "AdminController@delete_post")->name( name: 'delete_post');
    $router->post( uri: '/edit_comment', action: "AdminController@edit_comment")->name( name: 'edit_comment');
    $router->post( uri: '/post_comment', action: "AdminController@post_comment")->name( name: 'post_comment');
    $router->post( uri: '/delete_comment/{id}', action: "AdminController@delete_comment")->name( name: 'delete_comment');
}

```

Рис. 3.2 – Реалізація групи роутів

HTTP Middleware (посередники) – це фільтри обробки HTTP-запиту. Наприклад до складу Laravel входять middlewares для перевірки аутентифікації користувача. Якщо користувач не залогінений, middleware перенаправляє його на сторінку логіна. Якщо ж навпаки то middleware не втручається в проходження запиту, пропускаючи його далі по іншим middleware-посередникам до додатка. У моєму випадку було використано перевірку користувача на [«role: admin»].

Реалізація отримання інформації для головної сторінки адмінської панелі показана на рисунку 3.3.

```
class AdminController extends Controller
{
    public function index(Post $post)
    {
        $count = $post->resentPostCountToDashboard($post);
        $allPostsCount = Post::whereNull( column: 'in_draft')->count();
        $pendingCount = Post::OfPendingPosts()->count();
        $approvedCount = Post::OfApprovedPosts()->count();
        $lastComment = Comment::orderBy( column: 'id', direction: 'desc')->first()->body;
        $commentsCount = Comment::count();
        $regularAccountsCount = User::where( column: 'profile_type', operator: User::REGULAR_PROFILE)->count();
        $businessAccountsCount = User::where( column: 'profile_type', operator: User::BUSINESS_PROFILE)->count();
        $viewsCount = Ip_users_post::count();
        $likesCount = UsersLike::count();
        $messagesCount = Message::count();
        $rejectedPostsCount = Post::whereNull( column: 'in_draft')->whereNotNull( column: 'rejected_reason')->count();

        $allUsersCount = User::count();

        $weekNewUsers = User::whereDate( column: 'created_at', operator: '>', Carbon::now()->subWeek()->count();

        $paidUsersCount = PremiumUsersEndDate::count();
    }
}
```

Рис. 3.3 – Отримання інформації для головної сторінки

Відповідно до принципів ООП, запити до бази даних були винесені в окремі методи. Приклад запитів на отримання затверджених, популярних і рекомендованих публікацій показаний на рисунку 3.4.

```
public function scopeOfPendingPosts($query)
{
    return $query->where('is_approved', '=', 0)->whereNull('in_draft')->whereNull('rejected_reason')->orderBy('created_at', 'desc');
}

public function scopeOfPopularPosts($query)
{
    return $query->where('is_approved', '=', 1)->where('popular', '=', 1)->orderBy('created_at', 'desc');
}

public function scopeOfFeaturedPosts($query)
{
    return $query->where('is_approved', '=', 1)->where('featured', '=', 1)->where('show_date', '<=', Carbon::now()->orderBy('created_at', 'desc');
}
```

Рис. 3.4 – Запит на отримання публікацій з бази

### 3.3 Сторінка публікацій

Вигляд сторінки публікацій показано на рисунку 3.5.

Title	Author	Category	Tags	Location	Submission Date	Date Published	Assigned Editor
13. sdsds	Roman1 R.	Entrepreneurship	N/A	Algeria	14-08-2020	N/A	Unassigned
12. adsdasdas	Roman1 R.	Love & Relationships	N/A	Afghanistan	14-08-2020	N/A	Unassigned
11. New post	Roman1 R.	Videos	s	Albania	13-08-2020	N/A	Unassigned
10. TEst title	Roman1 R.	Society & Culture	N/A	Ukraine	24-04-2020	N/A	Unassigned
9. #HearMeToo - 16 Days of Activism Against Gender Based Violence	Guest C.	Society & Culture	unwomen, hearmetoo, orangethe world, metoo, genderbasedviolence, womensempowerment, tamilwomensdevelopmentgroup, twdg, twdgshiningalight	Australia	02-11-2018	N/A	Niluja Albert
8.							

Рис. 3.5 – Сторінка публікацій

На даній сторінці знаходяться панель управління публікаціями. Після того як користувач створив публікацію, вона потрапляє на цю сторінку для подальших перевірок контенту модераторами. І після перевірки модератори вирішують який статус давати публікації. Сторінка розділена на різні категорії залежно від статусів.

Публікація може мати такі статуси:

- В очікуванні. Сюди приходять публікації користувачів та чекають перевірки модераторами.
- Затверджені. Після того як модератори перевірили публікацію, щоб її контент не суперечив правилам сайту, щоб були якісні зображення і т.д. вона потрапляє в категорію «затверджених».

- Всі публікації. Тут знаходяться всі публікації з усіма можливими статусами.
- Відхилені. Сюди потрапляють публікації які суперечать правилам сайту або не підходять для модерації.
- Календар. У публікацій є можливість планування дати виходу. В цьому розділі знаходяться схвалені публікації, які очікують поки не настане запланований час публікації, зазначений користувачем.

Також для наочності для кожної публікації виведені наступні поля:

- Назва;
- Автор;
- Категорія;
- Теги які були згадані в контенті публікації;
- Локація на яку орієнтована публікація;
- Дата створення користувачем публікації;
- Дата коли опублікували;
- Призначений редактор. Тут можна вибрати редактора, кому прийде публікація на розгляд.

Також на сторінці є кнопка повідомлення редакторів. При натисканні здійснюється email розсилка призначена для сповіщення редакторів про нові публікації призначених ним.

Для зручності є секція пошуку, яка відкривається після натискання на кнопку пошуку. Меню пошуку показано на рисунку 3.6.

Пошук включає в себе:

- Пошук по назві публікації. При натисканні на назву відкриється сторінка редагування і огляду публікації;
- Пошук публікації за її категоріями;
- За ролями автора публікації;
- За ім'ям користувача;
- За датою створення публікації.

Back to Main Notify Editors

Pending Approved All Posts Rejected Calendar

Title Of Article

Category  User Role  Name  Date Submitted

Title	Author	Category	Tags	Location	Submission Date	Date Published	Assigned Editor
15. sdsds	Roman1 R.	Entrepreneurship	N/A	Algeria	14-08-2020	N/A	Unassigned <input type="button" value="v"/>
12. adsdasdas	Roman1 R.	Love & Relationships	N/A	Afghanistan	14-08-2020	N/A	Unassigned <input type="button" value="v"/>

Рис. 3.6 – Меню пошуку

Реалізація пошуку за ім'ям, назві публікації та категоріями показана на рисунку 3.7.

```
public function posts_search_form(Request $request)
{
    $query = $request->except( keys: ['_token', '_method']);
    $title = 'Search';

    $queryPosts = Post::query();

    if ((!empty($query['role'])) && ($query['role'] != 0)) {
        if (empty($query['author_name'])) {
            $usersList = Role::with( relations: 'users')->where( column: 'name', $query['role']->get()[0]->users->pluck( value: 'id'));
            $queryPosts->whereIn( column: 'user_id', $usersList);
        }
    }

    if (!empty($query['post_title'])) {
        $queryPosts->where( column: 'title', operator: 'LIKE', value: '%' . $query['post_title'] . '%');
    }

    if ($query['post_category'] != 0) {
        $queryPosts->where( column: 'category', $query['post_category']);
    }
}
```

Рис. 3.7 – Реалізація пошуку

Реалізація виведення списку всіх публікацій показана на рисунку 3.8.

```

public function posts()
{
    if (Auth::user()->hasRole('name: 'admin')) {
        $posts = Post::orderBy('column: 'id', 'direction: 'desc')->whereNull('column: 'in_draft')->paginate('perPage: 20');
    } else {
        $posts = Post::where('column: 'assigned_to', Auth::user()->id)->orderBy('column: 'id', 'direction: 'desc')->whereNull('column: 'in_draft')
            ->paginate('perPage: 20');
    }

    $editorsList = Role::with('relations: 'users')->where('column: 'name', 'operator: 'editors')->get()[0]->users;

    return view('view: 'pages.admin.posts.all_posts')->with([
        'posts' => $posts,
        'editorsList' => $editorsList,
    ]);
}

```

Рис. 3.8 – Виведення списку всіх публікацій

Для прискорення завантаження сторінки і роботи бази даних, вивід був обмежений пагінацією.

Пагінація – посторінковий вивід даних. Такий вивід інформації може стати великою проблемою, але не з фреймворком Laravel. Він робить цей процес безболісним, тому що здатний сам генерувати діапазон посилань щодо поточної сторінки. Згенерована розмітка сумісна з фреймворком Twitter Bootstrap.

Є кілька способів поділу даних на сторінки.

Найпростіший – використовуючи метод `paginate` об'єкта-будівника запитів або в зв'язці з моделями Eloquent. В даному випадку було обмежено кількість публікацій до 20 на 1 сторінку за допомогою `paginate(20)`.

Отримати інформацію про поточний стан можна за допомогою таких методів:

- `currentPage`;
- `lastPage`;
- `perPage`;
- `total`;
- `count`.

Для реалізації оповіщення редакторів через email розсилку використовувався сервіс SendGrid.

Laravel надає простий інтерфейс до популярної бібліотеки SwiftMailer. Головний файл налаштувань - `app / config / mail.php` – містить всі можливі параметри, що дозволяють змінювати SMTP-сервер, порт, логін, пароль, а також встановлювати глобальний адресу `from` для вихідних повідомлень.

Можна використовувати будь-який SMTP-сервер, або стандартну функцію PHP `mail` – для цього потрібно встановити параметр `driver` в значення `mail`. Крім того, доступний драйвер `sendmail`.

Реалізація відправки оповіщення показана на рисунках 3.9, 3.10 та 3.11.

```
public function notify_editors()
{
    $notifyIds = NotifyEditor::orderBy( column: 'editor_id')->groupBy( ...groups: 'editor_id')->get()->pluck( value: 'editor_id');
    if (empty($notifyIds[0])) {
        return back();
    }

    foreach ($notifyIds as $editorId) {
        $assignedPosts = NotifyEditor::where( column: 'editor_id', $editorId)->get()->pluck( value: 'post_id');
        $editor = User::find($editorId);
        $posts = Post::whereIn( column: 'id', $assignedPosts)->get();

        foreach ($posts as $post) {
            $editorEmailService =
                EditorEmailService::getInstance( emailType: EditorEmailService::NEW_ASSIGNED_POST_NOTIFICATION, [
                    'receiver' => $editor,
                    'post' => $post,
                ]);
            $editorEmailService->sendEmail();
        }
    }

    DB::table( table: 'notify_editors')->delete();
    return back();
}
```

Рис. 3.9 – Реалізація відправки оповіщення

```

abstract class EditorEmailService extends EmailService
{
    const NEW_ASSIGNED_POST_NOTIFICATION = 'NewAssignedPostNotificationEmail';

    public function __construct(string $emailType, array $data)
    {
        parent::__construct($emailType, $data);
    }

    /** @param string $emailType ...*/
    public static function getInstance(string $emailType, array $data)
    {
        $class = __NAMESPACE__ . "\\\" . $emailType;
        return new $class($emailType, $data);
    }

    /** @param Personalization $personalization ...*/
    protected function setReceiverEmail(Personalization $personalization)
    {
        $receiverEmail = new To($this->getReceiverName(), $this->getReceiverEmail());
        $personalization->addTo($receiverEmail);

        $this->getMail()->addPersonalization($personalization);
    }

    /** @return string ...*/
    protected function getReceiverName()
    {
        return $this->receiver->getUserName();
    }
}

```

Рис. 3.10 – Сервіс оповіщення редакторів

```

/** @return string ...*/
protected function getReceiverEmail()
{
    if (getenv( varname: 'SENDGRID_MODE') === 'test') {
        return explode( delimiter: ';', env( key: 'DEV_EMAILS'))[0];
    }
    return $this->receiver->getUserEmail();
}

/** @param Personalization $personalization ...*/
protected function setUnsubscribeLinks(Personalization $personalization)
{
    $encryptedMessage = encrypt( value: 'unable to disable this emails');

    $unsubscribeLink = route( name: 'disableEmailNotifications', ['encryptedMessage' => $encryptedMessage]);
    $unsubscribePreferencesLink = route( name: 'account_settings');

    $personalization->addSubstitution( substitution: "[Unsubscribe]", $unsubscribeLink);
    $personalization->addSubstitution( substitution: "[Unsubscribe_Preferences]", $unsubscribePreferencesLink);

    return $personalization;
}

/** @return bool ...*/
protected function emailCanBeSent()
{
    return true;
}

```

Рис. 3.11 – Сервіс оповіщення редакторів



### 3.4 Сторінка управління і редагування публікації

На цій сторінці редактори оглядають публікацію і вирішують що з нею робити далі.

Перша секція сторінки зображена на рисунках 3.12 та 3.13.

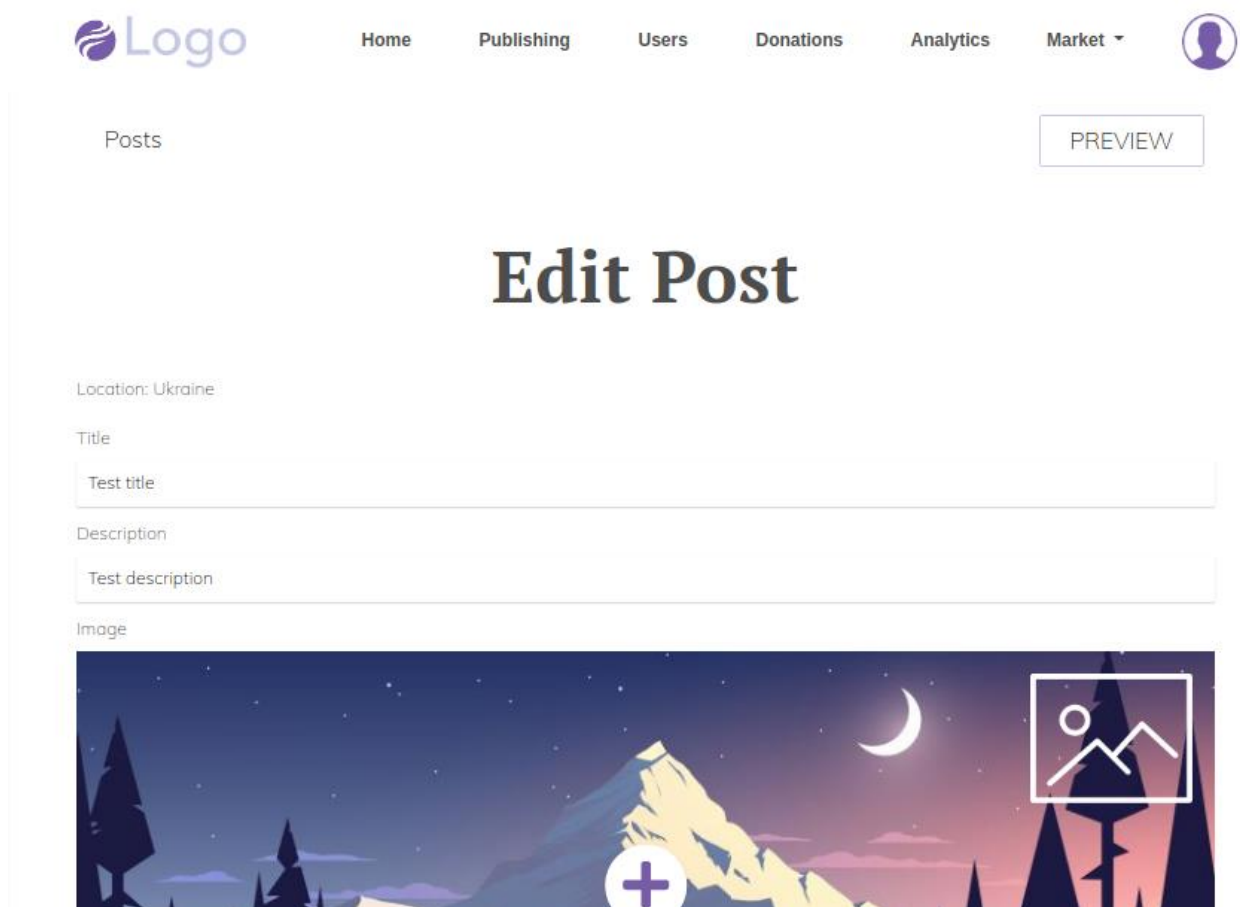


Рис. 3.12 – Перша секція сторінки управління публікацією

Logo Home Publishing Users Donations Analytics Market

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

P » SPAN POWERED BY TINYMCE

Video

Podcast

Category  
Society & Culture

Tags  
Enter tags ...

Is Approved

Featured

Select date x

UPDATE

DELETE POST

Рис. 3.13 – Перша секція сторінки управління публікацією

Ця секція містить в собі:

- Локацію на яку орієнтована публікація;
- Назва публікації з можливістю редагування;
- Опис публікації з можливістю редагування;
- Картинка для превью;
- WYSIWYG редактор тексту TinyMCE з дуже великим функціоналом і гнучкими настройками;
- Поле для додавання посилання на відео (Працює з усіма популярними відео сервісами);
- Поле для додавання посилання на подкаст. (Працює з усіма популярними медіа ресурсами);

- Поле вибору категорії публікації;
  - Поле додавання і редагування тегів публікації;
  - Меню вибору статусу публікації (наприклад схвалення статті або додавання її в рекомендовані);
  - Меню вибору дати публікації;
  - Кнопка поновлення змін;
  - Кнопка видалення публікації;
  - Кнопка відхилення публікації;
  - Також є кнопка попереднього погляду публікації, яка відкриває нову сторінку, яка показує як буде виглядати публікація на сайті після її схвалення.
- Приклад передперегляду показаний на рисунку 3.14.

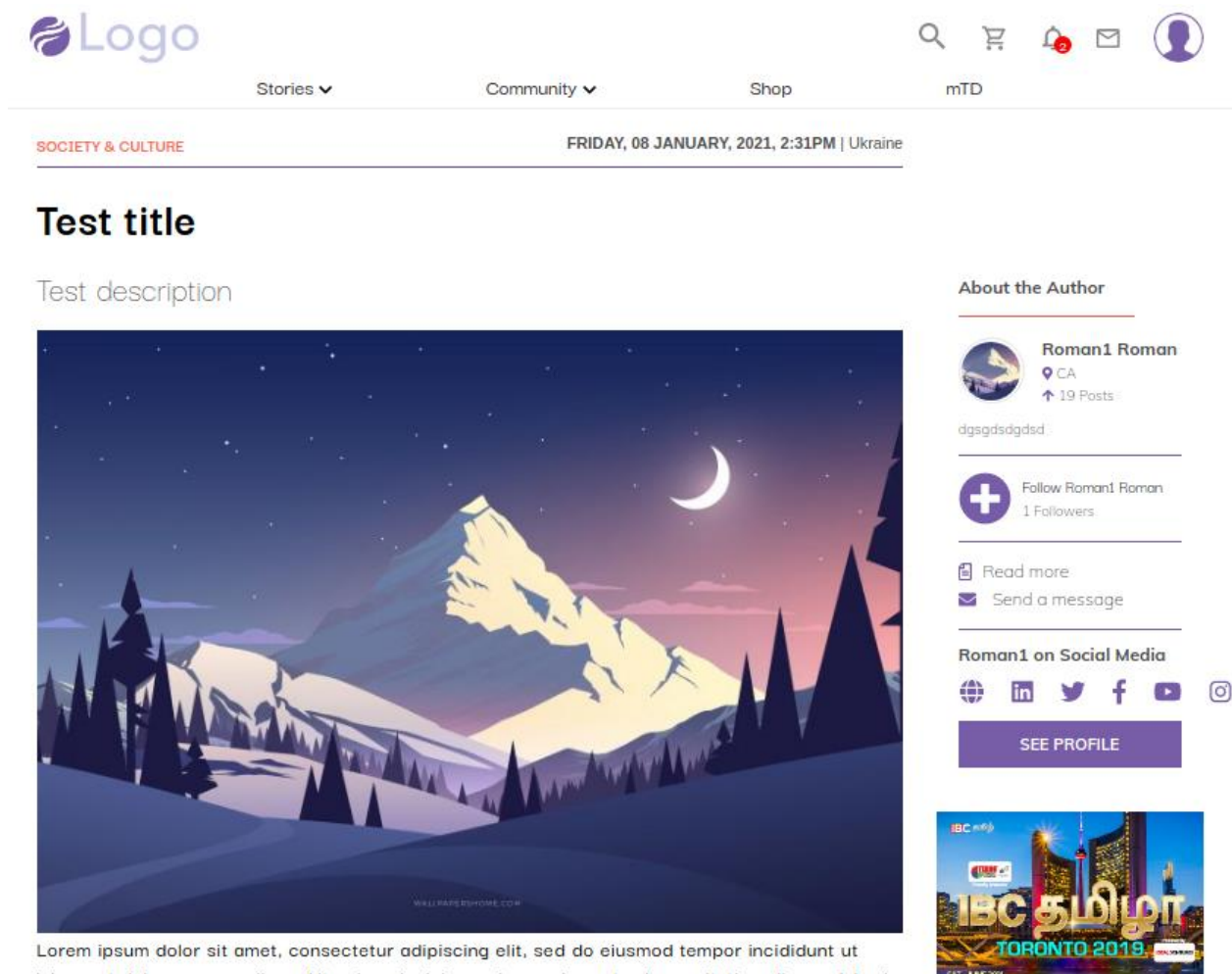


Рис. 3.14 – Огляд публікації

У наступних секціях показаний автор публікації з посиланням на його профіль і меню коментарів. Адміністратор або ж редактор може залишити коментарі про статтю, наприклад попросити користувача додати превью в кращій якості. Ці секції продемонстровані на рисунку 3.15.



Рис. 3.15 – Інші секції сторінки управління публікацією

### 3.5 Сторінка користувачів

На даній сторінці знаходяться панель управління користувачами. Це показано на рисунку 3.16. Сторінка розділена на різні категорії сортування користувачів:

- Схвалені;
- Неверифіковані акаунти, які не підтвердили реєстрацію через посилання в email листі;

- Що чекають схвалення;
- Відключені акаунти;
- Медійні особистості;
- Редактори;
- Адміністратори.

Logo Home Publishing Users Donations Analytics Market

Back to Main EXPORT EMAILS TO CSV EXPORT TO CSV

Approved Unverified Pending Approval Rejected Influencers Editors Administrators Delete

Name or Email

Full registered Verified email Under Approve

Profile Type Location Category Membership Since

08-01-2021 Search

<input type="checkbox"/>	Name	Email	Account Type	Title	Location	Article Published	Category	Interests	Member Since
<input type="checkbox"/>	new user	5fbe51c199d02	Business	new user	Afghanistan	0	Activism	N/A	25-11-2020
<input type="checkbox"/>	just shop	5fbcf9512c412	Individual	N/A	Canada	0	Film & TV	N/A	24-11-2020
<input type="checkbox"/>	CacTus .	kaktysuwe@gmail.com	Individual	s	Afghanistan	0	Activism	N/A	24-11-2020

Рис. 3.16 – Панель управління користувачами

Також є пошук користувачів за наступними критеріями:

- За ім'ям або email;
- За акаунтами з повною реєстрацією. Це ті користувачі, які повністю заповнили свій профіль і пройшли всі етапи реєстрації.
- За верифікованими акаунтами;
- За акаунтами які затверджені;
- За типом профілю (звичайний або бізнес профіль);
- За місцем розташування користувача;
- За категорією профілю;

- За датою реєстрації.

Користувачі мають такі поля:

- Ім'я користувача з посиланням на огляд і редагування профілю;
- Email (профілі магазинів мають унікальний id замість email адреси);
- Тип профілю;
- Місцезнаходження;
- Кількість опублікованих постів;
- Категорія аккаунта;
- Теги інтересів;
- Дата реєстрації;

Існує можливість вибору профілів та їх видалення навіть без етапу перевірки користувача. Також є можливість експорту користувачів в CSV файл або експорту email адрес. Метод, який формує дані для експорту в csv файл, показаний на рисунку 3.17.

```
public function getInfoToCsv($data)
{
    $items      = $data['data'];
    $tableHeaders = $data['tableHeader'];
    $filename   = $data['filename'];
    $headers    = [
        'Cache-Control'      => 'must-revalidate, post-check=0, pre-check=0',
        'Content-type'       => 'text/csv',
        'Content-Disposition' => 'attachment; filename=' . $filename . '.csv',
        'Expires'            => '0',
        'Pragma'              => 'public'
    ];

    $callback = function () use ($items, $tableHeaders) {
        $FH = fopen('php://output', 'w');
        fputcsv($FH, $tableHeaders);
        foreach ($items as $row) {
            fputcsv($FH, $row);
        }
        fclose($FH);
    };

    return response()->stream($callback, status: 200, $headers);
}
```

Рис. 3.17 – Метод експорту у CSV файл

### 3.6 Сторінка управління користувачем

Ця сторінка служить для управління профілем користувача, наприклад зміна типу профілю або ж його блокування. Зовнішній вигляд продемонстрований на рисунку 3.18.

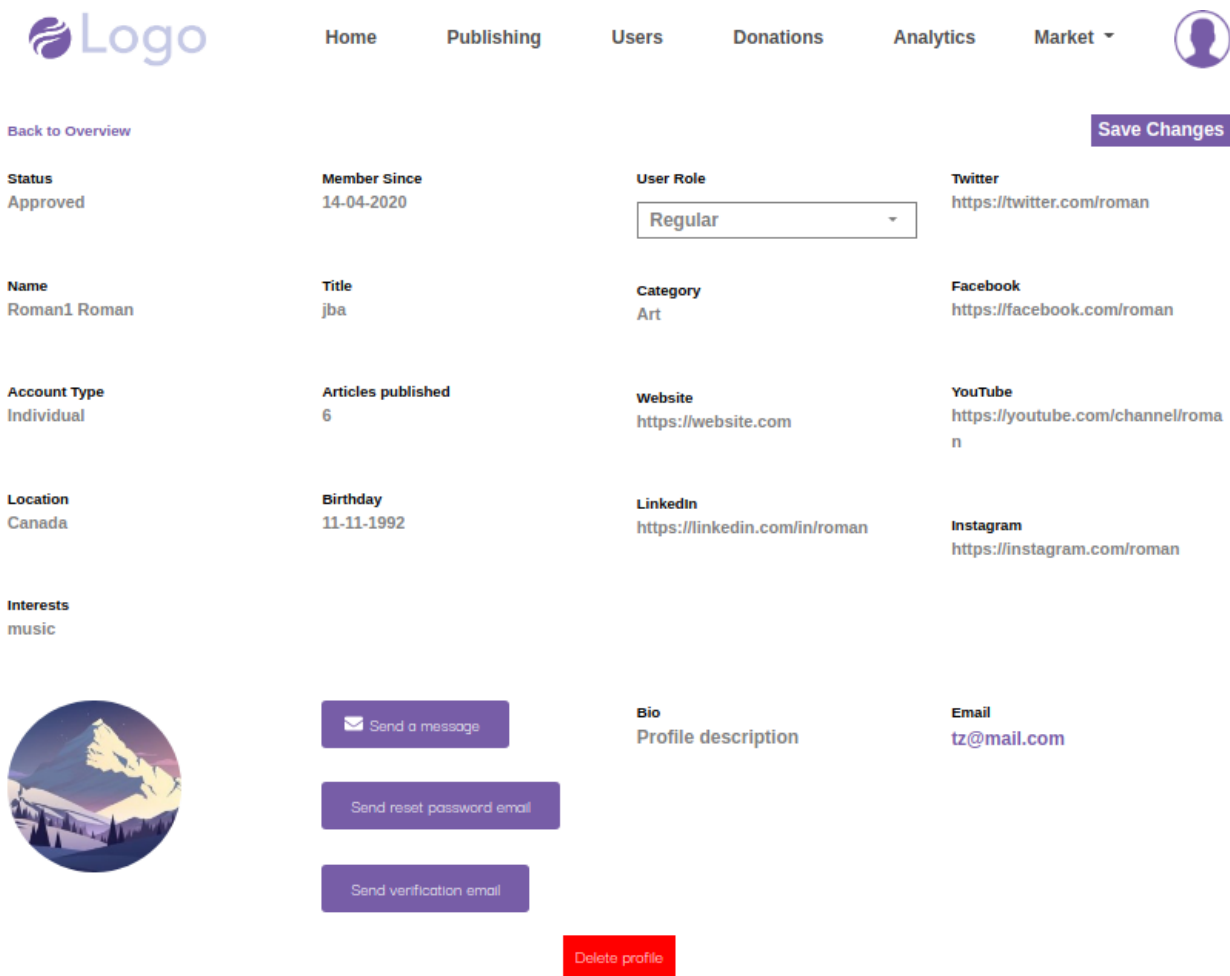


Рис. 3.18 – Сторінка управління профілем користувача

На сторінці відображена наступна інформація про профіль:

- Статус профілю;
- Ім'я;
- Тип профілю;

- Місцезнаходження;
- Інтереси;
- Дата реєстрації;
- Назва роботи;
- Кількість опублікованих постів;
- День народження;
- Меню вибору ролі аккаунта. Адміністратор може змінити роль профілю, наприклад призначити людину редактором або ж адміністратором і т.п.
- Категорія профілю;
- Посилання на вебсайт користувача і соцмережі (Linkedin, Twitter, Facebook, YouTube, Instagram);
- Опис профілю;
- Email профілю;
- Аватар;

Для комунікації з користувачем є кнопка відправки особистого повідомлення на сайті.

Також при натисканні на email відкриється програма за замовчуванням встановлена в операційній системі, для відправки email повідомлень.

Якщо користувач звернувся в техпідтримку сайту через те, що він забув пароль від аккаунта, або ж випадково видалив лист для верифікації профілю, є можливість відправити лист заново. Для цього є кнопка відправки листа зі збросом пароля або відправлення листа з верифікацією.

Також внизу сторінки є можливість видалення профілю або ж відхилення із зазначенням причини в поле вводу коментаря.

Реалізація відхилення профілю із зазначенням причини показано на рисунку 3.19.



```

public function reject(Request $request)
{
    $id          = $request->input('id');
    $rejectReason = RejectReason::whereShopId($id)->get();
    /** @var Shop $shop */
    $shop = Shop::findOrFail($id);
    $data = [
        'shop_id'    => $id,
        'reason'     => $request->input('reject_reason'),
        'is_rejected' => true,
    ];
    if ($rejectReason->isEmpty()) {
        RejectReason::create($data);
    } else {
        /** @var RejectReason $rejectReason */
        $rejectReason->first()->fill($data)->save();
    }

    event(new ShopRejected($shop));

    return redirect()->route('route: admin.market.shop.approval');
}

```

Рис. 3.19 – Відхилення профілю із зазначенням причини

### 3.7 Сторінка статистики пожертвувань

У соціальній мережі є можливість зробити грошову пожертву користувачеві, якщо він прив'язав до свого профілю PayPal аккаунт. Ця сторінка відображає список пожертвувань з можливістю сортування по недавнім або ж відображення всіх пожертвувань. Це можна побачити на рисунку 3.20.

Сторінка демонструє статистику:

- Кількість пожертвувань за поточний місяць;
- Кількість пожертвувань за рік;
- Загальна кількість пожертвувань;
- Середня сума пожертвувань.

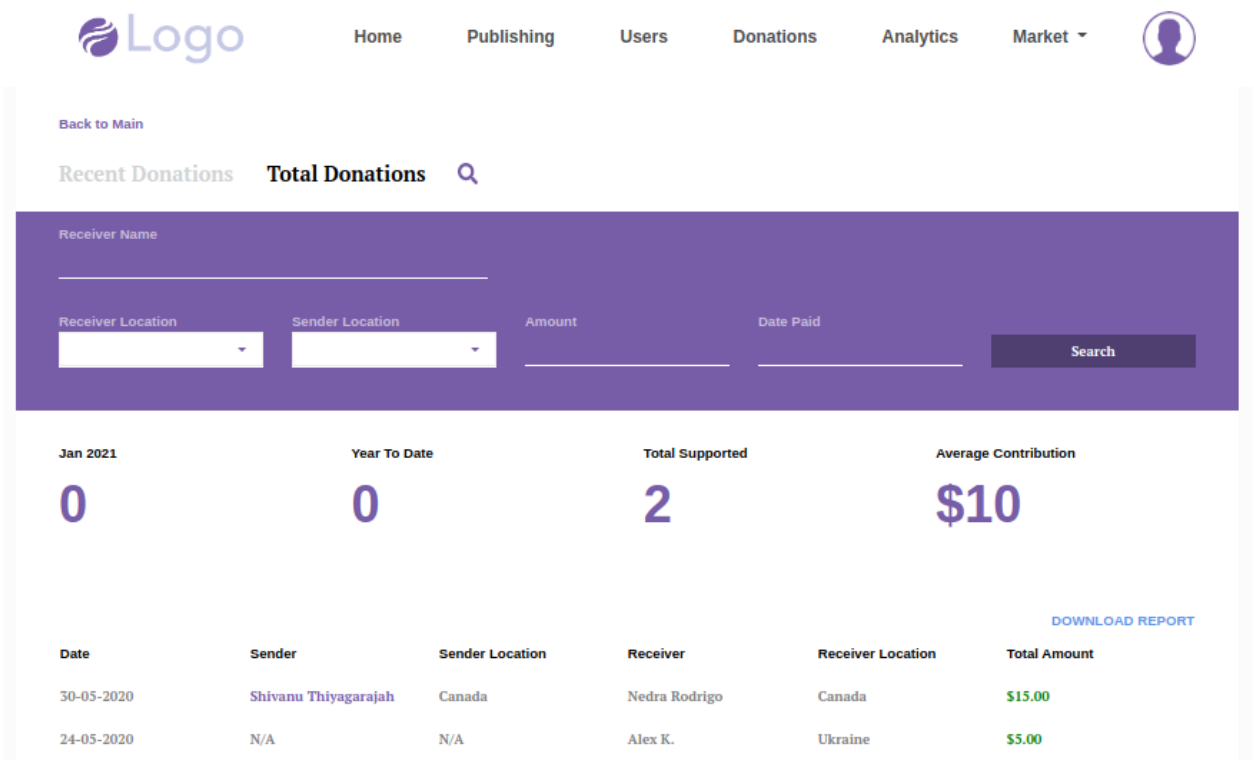


Рис. 3.20 – Сторінка грошових пожертвувань

Далі йде список пожертвувань який включає в себе:

- Дату;
- Аккаунт відправника. Якщо незареєстрований користувач зробив Донат то буде відображено «N / A»;

- Місцезнаходження відправника;
- Одержувач;
- Місцезнаходження одержувача;
- Сума.

Також є можливість пошуку за:

- Ім'ям одержувача;
- Розташуванням одержувача або відправника;
- Сумою платежу;
- Датою платежу.

Також при натисканні на кнопку «Завантажити звіт» буде сформований список пожертвувань в CSV файл.

Реалізація отримання інформації всіх пожертвувань показана на рисунку 3.21.

```
public function total_donations()
{
    $donations = PayoutForUser::with( relations: 'userSender', 'userReceiver')
        ->orderBy( column: 'created_at', direction: 'desc')
        ->paginate( perPage: 20);
    $middleCount = 0;
    try {
        $middleCount = round( (val: PayoutForUser::sum( column: 'payout_full') / PayoutForUser::count(), precision: 2, mode: PHP_ROUND_HALF_DOWN);
    } catch (\Exception $e) {
    }

    $monthDonations = PayoutForUser::whereDate( column: 'created_at', operator: '>', Carbon::now()->subMonth()->count();
    $yearDonations = PayoutForUser::whereDate( column: 'created_at', operator: '>', Carbon::now()->subYear()->count();

    $donationsCount = PayoutForUser::distinct( 'user_id')->count( columns: 'user_id');

    return view( view: 'pages.admin.donations.donations_total')->with([
        'donations' => $donations,
        'middleCount' => $middleCount,
        'monthDonations' => $monthDonations,
        'yearDonations' => $yearDonations,
        'donationsCount' => $donationsCount,
    ]);
}
```

Рис. 3.21 – Реалізація отримання інформації всіх пожертвувань

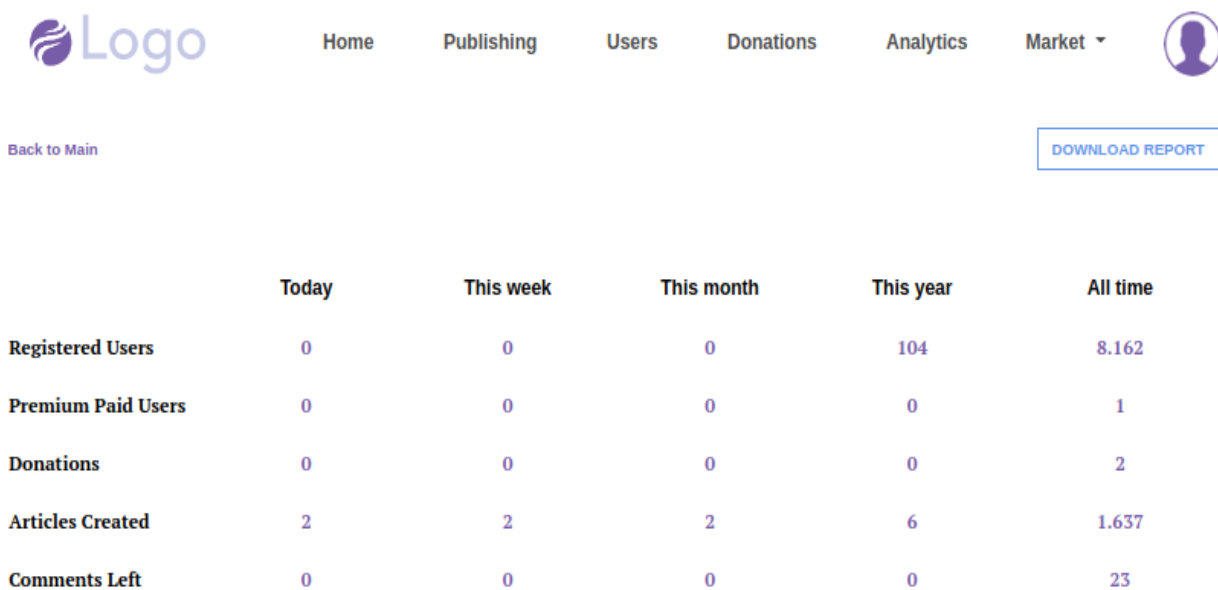
Донати на сайті реалізовані за допомогою платіжної системи PayPal.

### 3.8 Сторінка аналітики

Сторінка для наочного відображення різної аналітики за періоди часу показана на рисунку 3.22.

Тут відображена інформація про кількість:

- Зареєстрованих користувачів;
- Преміум користувачів;
- Пожертвувань;
- Створених публікацій;
- Залишених коментарів.



	Today	This week	This month	This year	All time
Registered Users	0	0	0	104	8.162
Premium Paid Users	0	0	0	0	1
Donations	0	0	0	0	2
Articles Created	2	2	2	6	1.637
Comments Left	0	0	0	0	23

Рис. 3.22 – Сторінка аналітики

Інформація розділена на різні періоди часу (за сьогодні, за поточний тиждень, місяць, рік і весь час). Вся інформація доступна для експорту в CSV файл.

### 3.9 Розділ розгляду магазинів

Дана соціальна мережа дає можливість не тільки створювати звичайні і бізнес профілі, але і створювати свої магазини для реалізації бізнесу. Вся статистика продажу, за товарам, магазинам користувачів також відображена в CRM панелі.

Після того як користувач створив магазин і додав перший товар, його магазин відправляється на розгляд адміністрацією. За це відповідає сторінка підтвердження магазинів яка відображена на рисунку 3.23.

На цій сторінці по черзі з'являються заявки на створення магазину. Зліва вказано число неперевіраних заявок. У заявці відображені назва магазину,

місце розташування, посилання на профіль власника, аватар і задній фон магазину. Після перевірки інформації є 2 рішення:

- Відхилення заявки із зазначенням причини відмови;
- Підтвердження заявки.

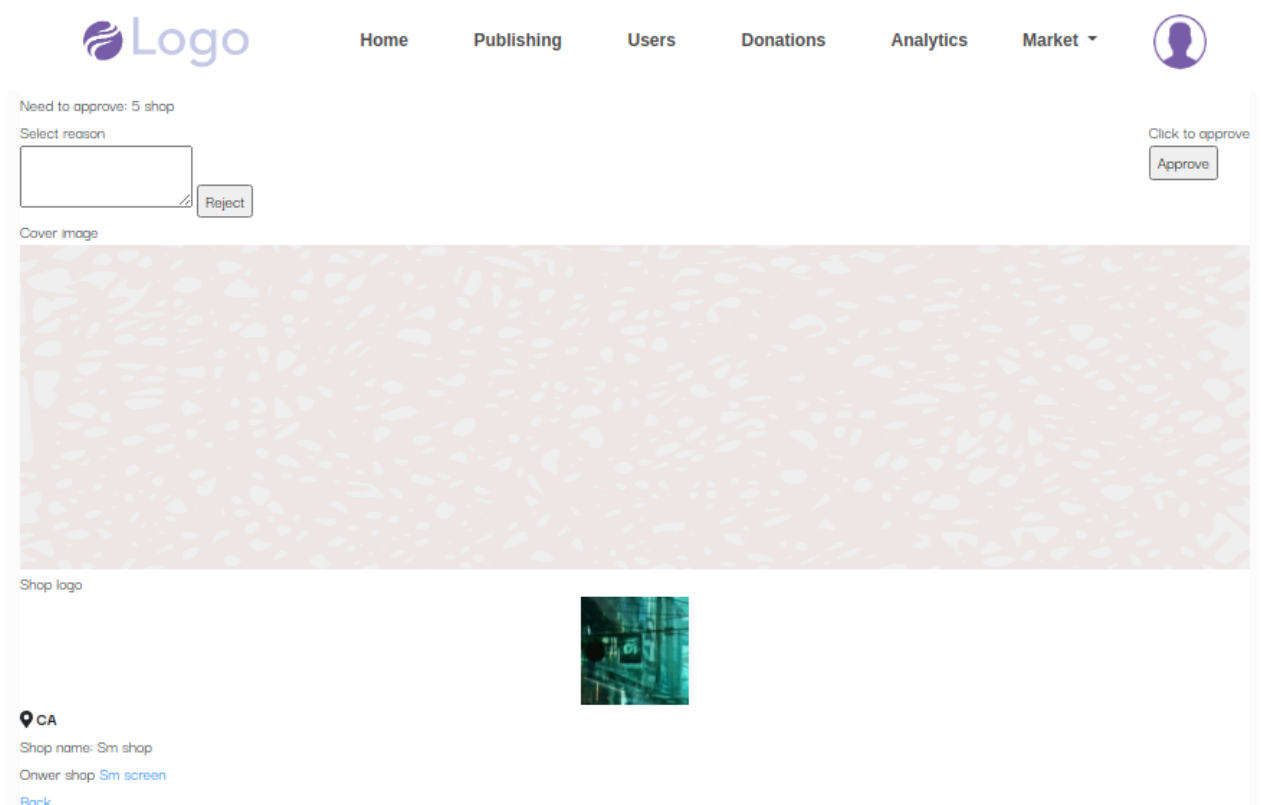


Рис. 3.23 – Сторінка підтвердження магазинів

Після підтвердження або відхилення, користувачеві приходить повідомлення електронної пошти з рішенням.

Формування запиту на отримання магазинів які чекають підтвердження показані на рисунку 3.24.

```

/**
 * @return Shop|Builder
 */
private function buildNoApprovedQuery()
{
    $queryBuilder = Shop::whereHas(
        relation: 'rejectReason', function ($query) {
            /** @var $query RejectReason */
            $query->NotRejected();
        }
    )
    ->orDoesntHave( relation: 'rejectReason')
    ->with(['listings.photos', 'ownerProfile', 'shopProfile', 'rejectReason'])
    ->ReadyForApproval();

    return $queryBuilder;
}

```

Рис. 3.24 – Запит на отримання магазинів які чекають підтвердження.

### 3.10 Управління продуктами

Другим пунктом меню управління магазинами є сторінка продуктів. Вона показана на рисунку 3.25.

Тут відображено картки з товарами які включають в себе:

- Головну картинку продукту. При натисканні на неї відкриється сторінка управління продукту;
- Назву продукту;
- Категорію;
- Чи знаходиться продукт в публічному доступі;
- Чи є продукт рекомендованим.

Для прискорення завантаження сторінки була додана пагінація. За раз на сторінку завантажуються максимум 10 карток з товарами.

Для перегляду докладної інформації про продукт потрібно натиснути на його картку і відкриється сторінка яка показана на рисунку 3.26

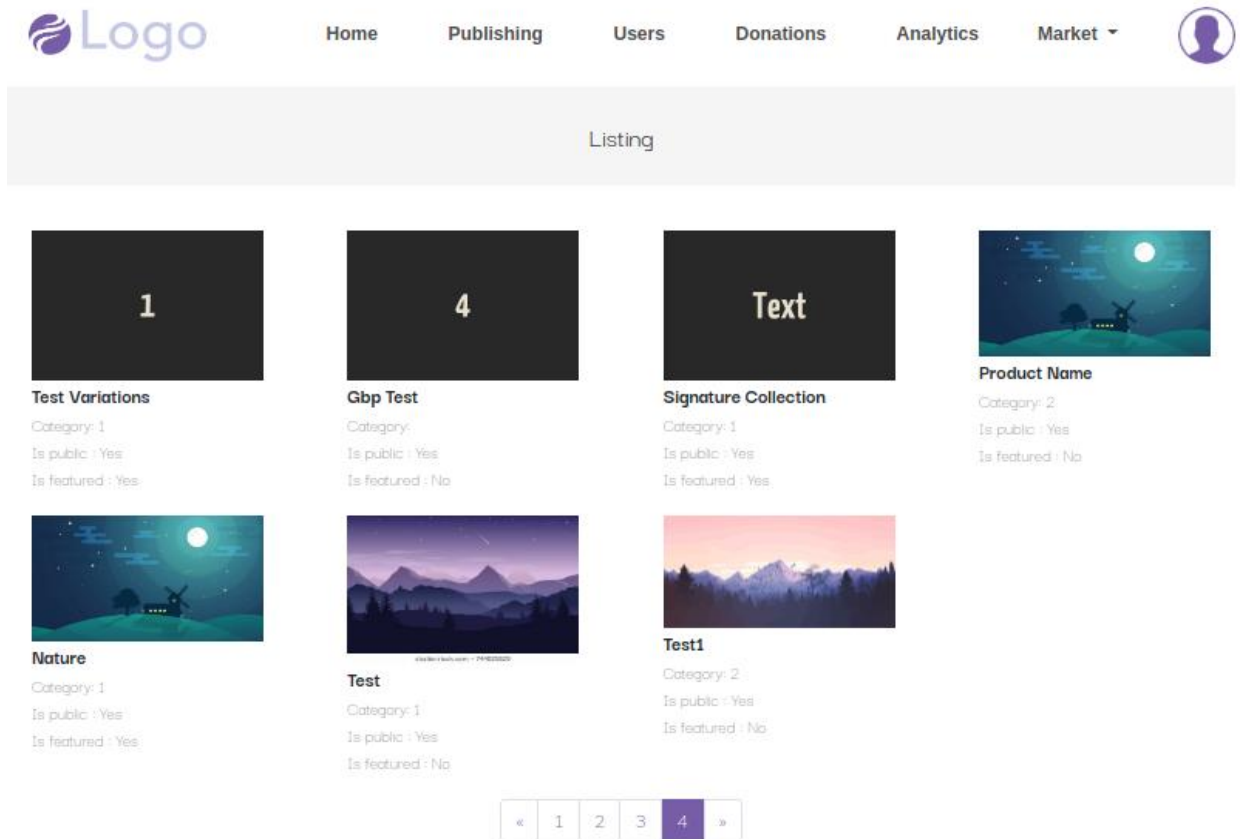


Рис. 3.25 – Сторінка продуктів

Тут можна подивитися всі фотографії товару які знаходяться в слайдері. Також відображена інформація про товар:

- Назва;
- Опис;
- Валюта, ціна;
- Чи можлива безкоштовна доставка;
- Категорія товару;
- Чи є товар рекомендованим користувачам;
- Інформація про габарити товару.

Адміністрація має право редагувати товар. Для цього потрібно натиснути на кнопку редагування. Після цього відкриється інша сторінка показана на рисунку 3.27.

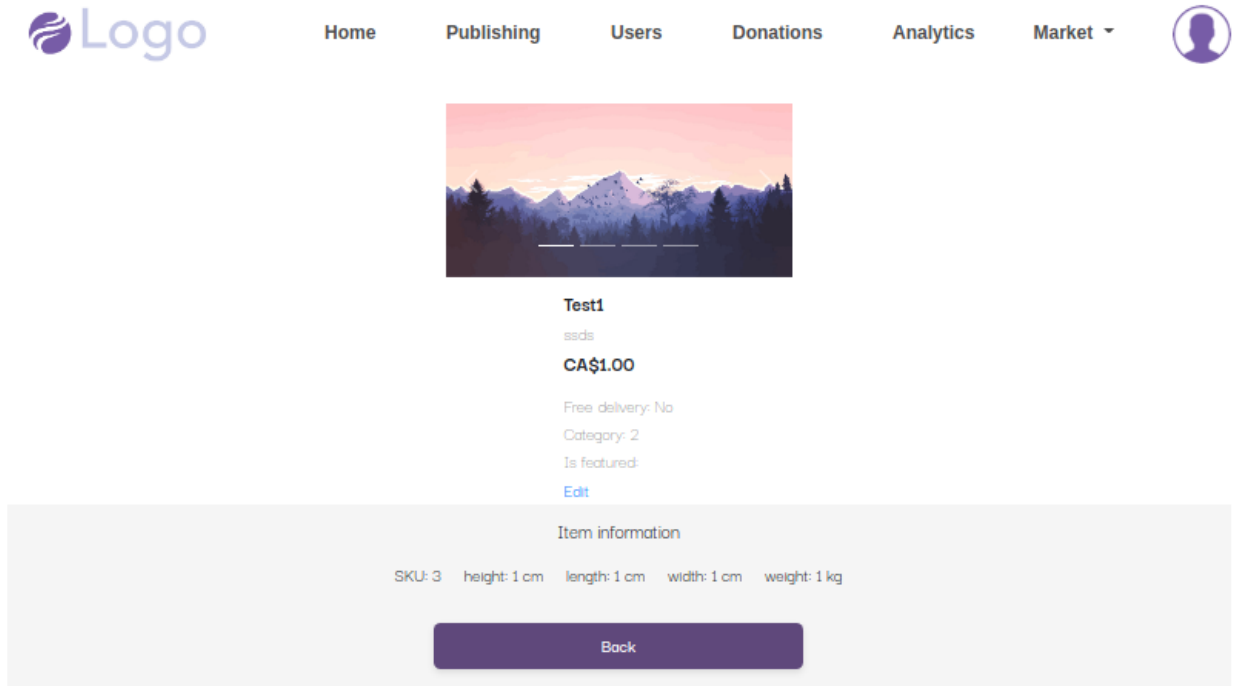


Рис. 3.26 – Сторінка докладної інформації про продукт

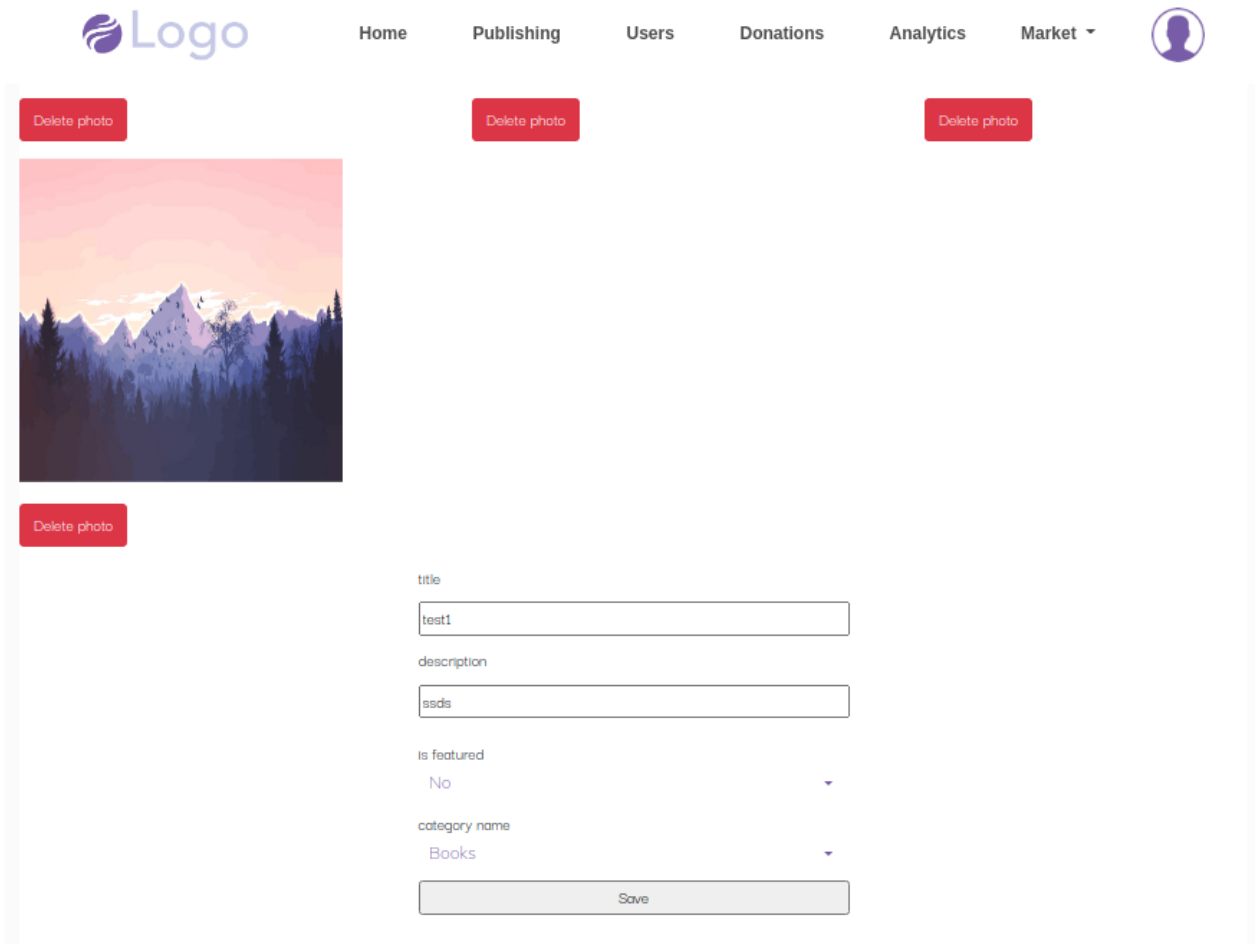


Рис. 3.27 – Сторінка редагування товару



На сторінці є можливість видалення фотографій, редагування назви і опису. Також можна вибрати чи буде відображатися продукт в рекомендаціях користувачів, та змінити категорію товару.

### 3.11 Управління магазинами

Третій пункт меню магазинів на панелі CRM веде на сторінку зі списком магазинів. Ця сторінка продемонстрована на рисунку 3.28.

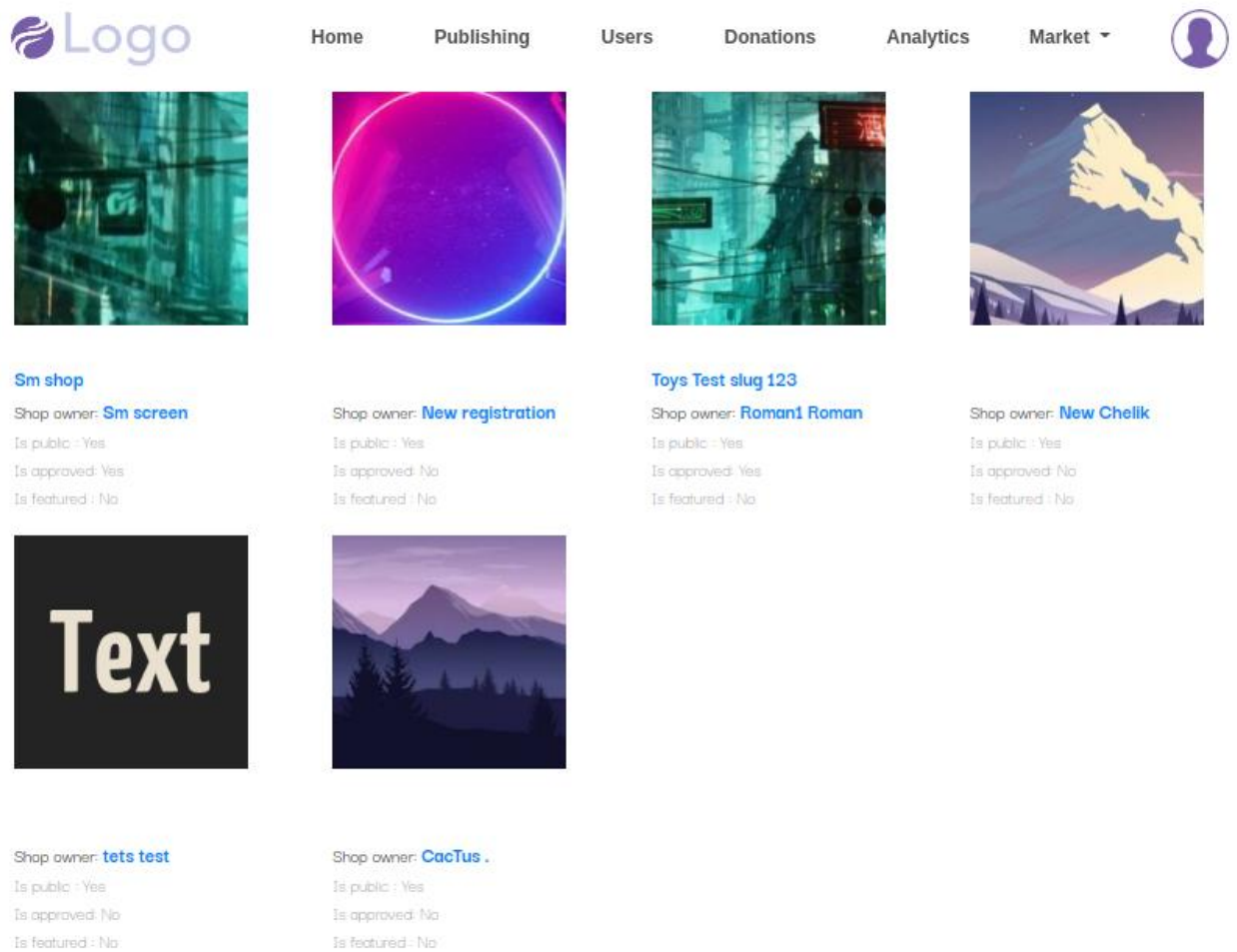


Рис. 3.28 – Сторінка списку магазинів

Тут відображено картки з магазинами які включають в себе:

- Назва магазину. Є посиланням на магазин;
- Власник магазину;
- Чи знаходиться магазин в публічному доступі;
- Схвалений чи магазин;
- Чи є магазин рекомендованим.

Для управління магазином потрібно натиснути на картинку магазину. Відкриється наступна сторінка зображена на рисунку 3.29.

Рис. 3.29 – Сторінка управління магазином

Тут можна відредагувати назву магазину. Також є можливість управління видимості магазину, зробити його в публічному доступі або поки що прихованим, наприклад якщо він не до кінця заповнений. Можна схвалити або ж навпаки поміняти рішення на відхилення магазину. Також можна додати магазин в рекомендації для користувачів.

### 3.12 Панель користувача

У користувачів також є панель управління своїм контентом, профілями, продуктами та іншими можливостями соціальної мережі. Так як у користувача

є можливість реалізувати свій бізнес через соціальну мережу, він теж має свою панель керування своїм магазином, замовленнями, оплатами і т.д.

Приклад панелі користувача для управління публікаціями показана на рисунку 3.30.

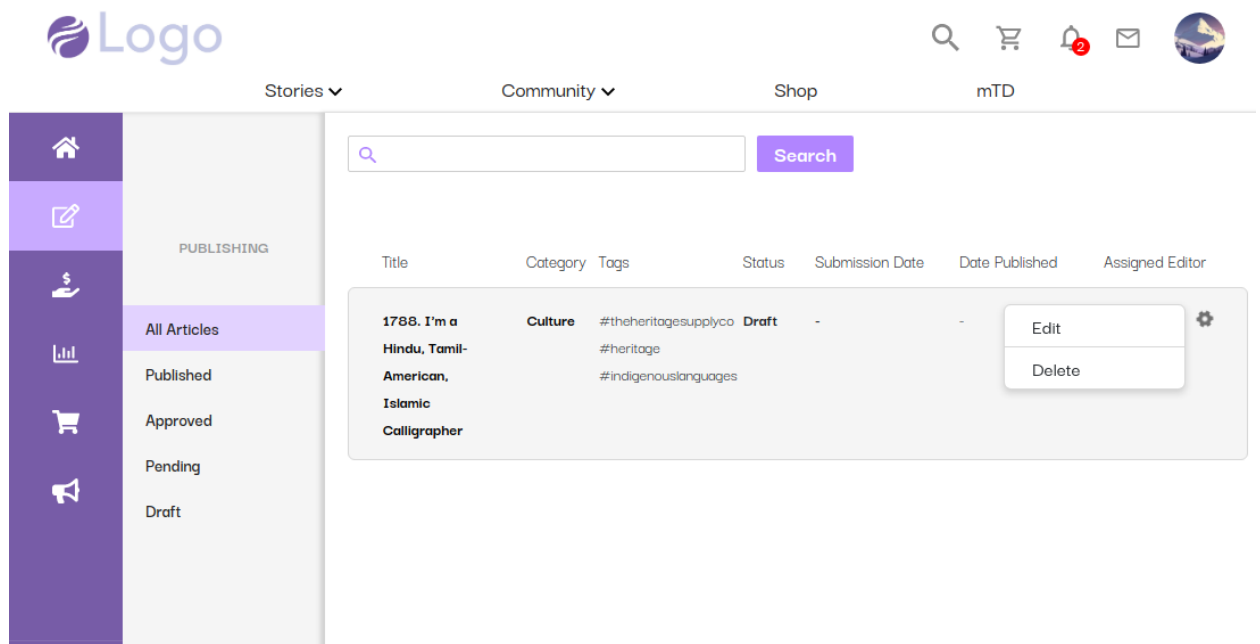


Рис. 3.30 – Панель користувача для управління публікаціями

Тут користувач бачить список своїх публікацій і їх статуси. Меню публікацій розділене за категоріями на наступні:

- Всі публікації;
- Опубліковані;
- Схвалені;
- Ті які чекають схвалення;
- Чернетки.

Користувач може редагувати або видалити публікацію. Також є пошук який шукає за назвою посади.

Приклад головної сторінки панелі користувача для управління магазином показана на рисунку 3.31.

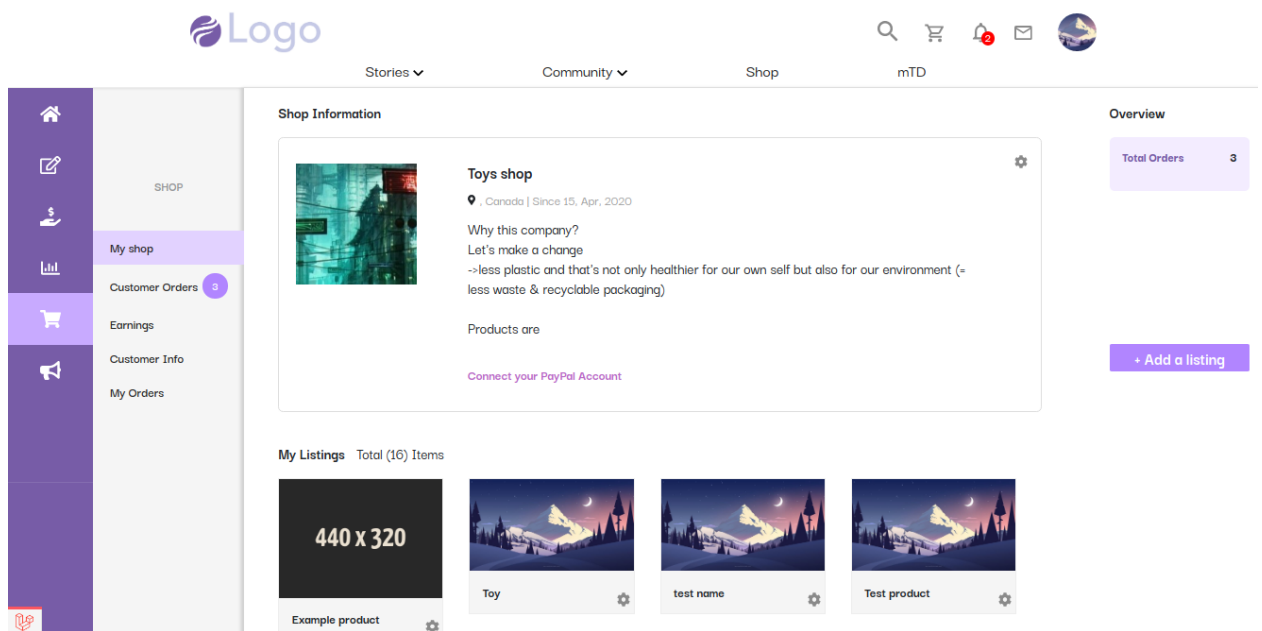


Рис. 3.31 – Головна сторінка управління магазином

Ця сторінка розділена на наступні секції:

- Інформація про магазин.

Тут знаходиться назва магазину, місце розташування, дата створення, опис і аватар. Знизу є кнопка прив'язки платіжної системи PayPal. Також у верхньому правому кутку знаходиться кнопка редагування інформації про магазин, яка відкриває форму з можливістю заміни аватара магазину, категорії магазину, адреси, опису і політики повернення товарів. Ця форма показана на рисунку 3.32.

- Інформація про товари.

Тут показується кількість товарів і їх картки з найменуванням товару, картинкою і кнопкою управління. Кнопка відкриває меню в якому є можливість редагування товару, видалення або приховування. Сторінка редагування товару показана на рисунку 3.33.

- Блок з інформацією про замовлення і кнопкою створення нового товару.

Сторінка управління замовленнями показаний на рисунку 3.34.

The screenshot shows a web application interface for managing a shop. At the top, there is a logo and navigation links for 'Stories', 'Community', 'Shop', and 'mTD'. A sidebar on the left contains icons for home, edit, earnings, orders, and announcements, with a 'SHOP' menu below. The main content area is titled 'Shop Information' and contains a circular profile picture, a text input field with 'Toys shop', a dropdown menu for 'Games & Toys', an 'Address' field, and separate input fields for 'City' (A1A 1A1) and a country dropdown (Canada). A text area contains the description: 'Why this company? <br>Let's make a change <br>->less plastic and that's not only healthier for our own self but also for our environment (= less waste & recyclable packaging) <br><br>Products are'. At the bottom, there is a 'Return and Policy' section.

Рис. 3.32 – Форма редагування інформації магазину

The screenshot shows a product editing form. On the left, there is a preview image of a snowy mountain landscape with a camera icon and the text 'Add another photo'. On the right, there are 'Delete' and 'Save' buttons. The form fields include: 'Product name' (Small toy), 'SKU' (empty), 'Original Price' (21.00) and 'Currency' (CAD), an unchecked checkbox for 'OFFER FREE SHIPMENT?', 'Item Details' section with 'Qty' (1 Total), and dimensions: Height (3 cm), Length (2 cm), Width (3 cm), and Weight (1 kg). The 'Product Description' field contains 'Small toy'. At the bottom, there is a 'Variants' section with an 'Add Variant +' button.

Рис. 3.33 – Сторінка редагування товару

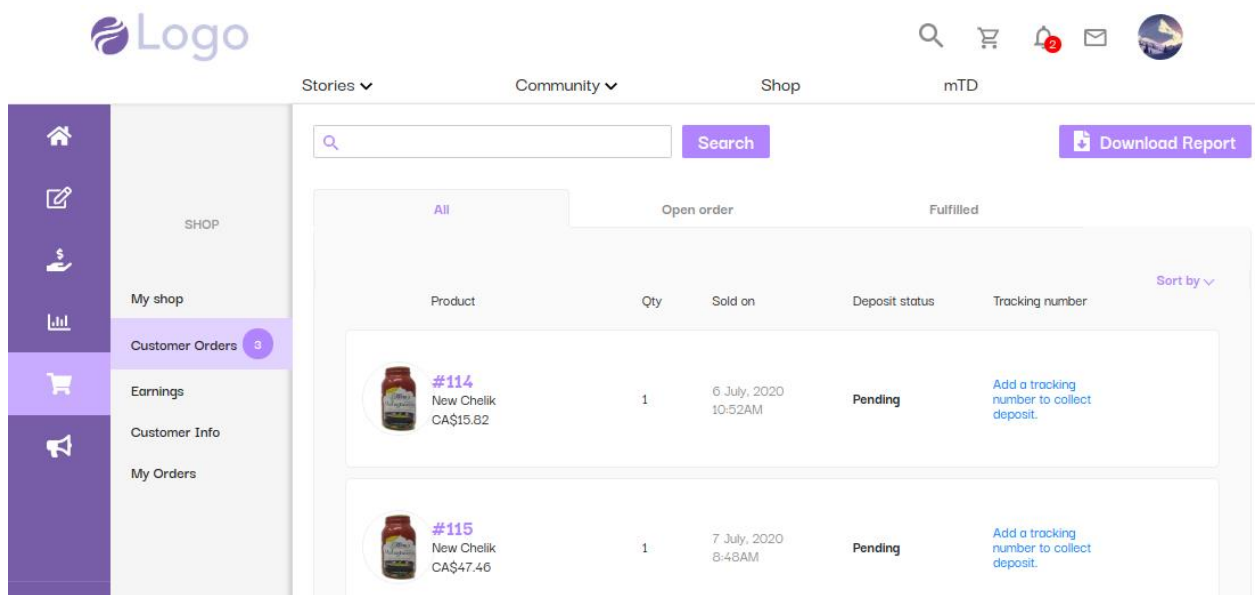


Рис. 3.34 – Сторінка управління замовленнями

Замовлення відфільтровані на різні вкладки в залежності від типу:

- Всі замовлення;
- Відкриті замовлення;
- Виконані.

Список продуктів містить наступні поля:

- Продукт. Тут знаходиться картинка товару, ціна, ім'я покупця, номер замовлення який є посиланням на сторінку деталей і управління замовленням;
- Кількість продуктів в замовленні;
- Дата продажу;
- Статус оплати;
- Посилання на додавання трек-номера для відстеження товару покупцем.

Також є можливість пошуку товару, сортування замовлень по новим або старим і експорт списку замовлень в CSV.

Сторінка деталей замовлення потрібна для повної інформації про замовлення і покупця. Вона показана на рисунку 3.35.

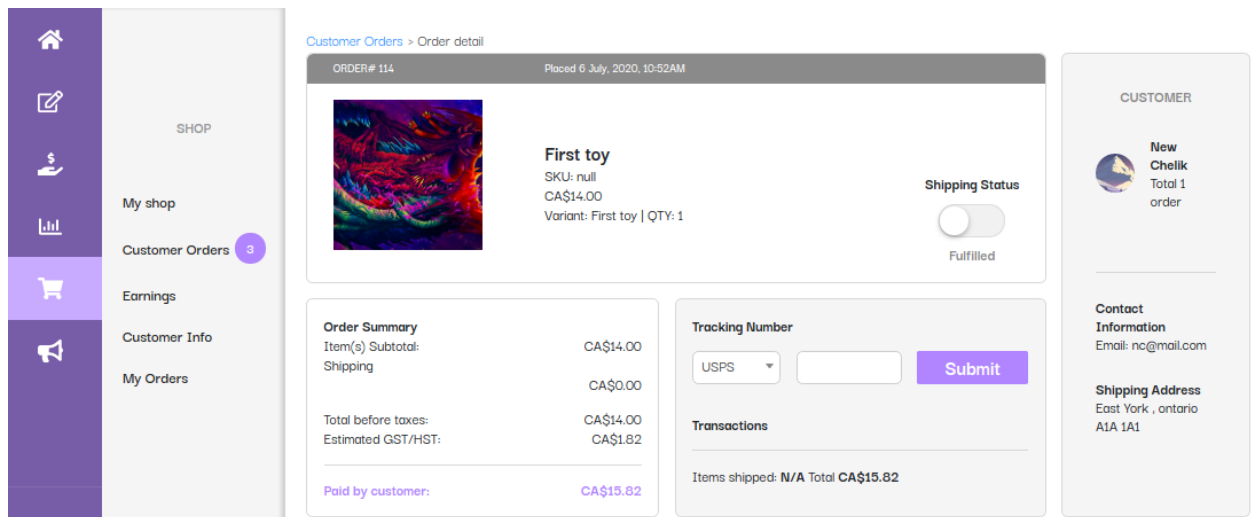


Рис. 3.35 – Сторінка деталей замовлення

Тут знаходиться повна інформація про замовлення, про його вартості, про вартість доставки, про покупця. Також кнопка управління статусом доставки і поле заповнення трек-номера замовлення для відстеження через поштові сервіси.

### 3.13 Реалізація бази даних

Система управління користувачами та контентом інтегрується в базу даних соціальної мережі.

В якості СУБД було використано MySQL 5.6 тому що вона використовувалася для розробки соціальної мережі. Основні таблиці та зв'язки соціальної мережі показані на рисунку 3.36.

Основними об'єктами для статистики та управління в CRM системі є користувачі, публікації, магазини, товари, і все що з ними пов'язано. Кожен об'єкт має зв'язки один до одного, один до багатьох, – це дуже важливо для правильного керування на панелі та збору достовірної статистики.

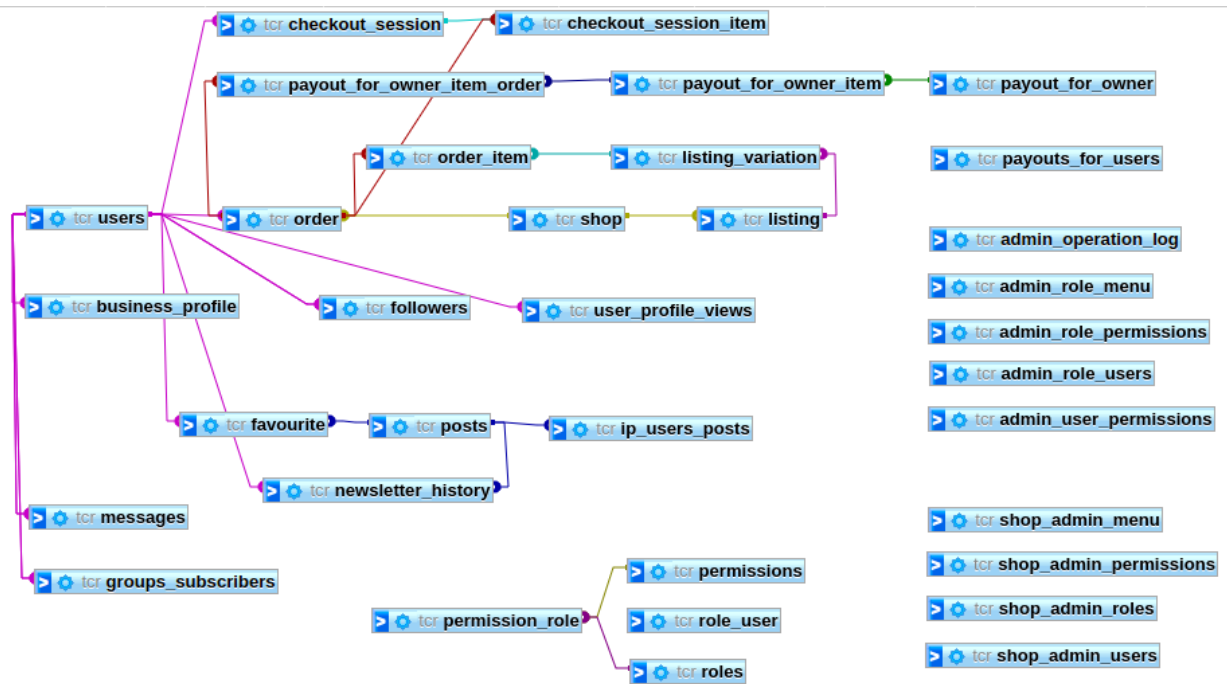


Рис. 3.36 – Структура та зв'язки основних таблиць

Для реалізації CRM потрібно було розширити стандартні ролі користувачів додавши ролі адміністраторів та редакторів. Завдяки тому що система управління користувачами та контентом має мінімальне втручання в структуру соціальної мережі – вона може буде інтегрована в інші різноманітні соціальні мережі.

### 3.14 Висновки за третім розділом

Вивчені принципи роботи системи управління користувачами та контентом соціальної мережі, розглянута документація фреймворку Laravel та платформи для Email розсилки – SendGrid. Було розроблена CRM панель яка має повний контроль над контентом, користувачами та їх бізнесами. Також розроблено збір статистика та аналітики. Був створений модуль експорту даних в CSV файли.



## ВИСНОВКИ

В ході роботи було обґрунтовано об'єкт дослідження, детально розібрані переваги та недоліки. Також були приведені реальні приклади економічного ефекту від впровадження CRM різних компаній, їх статистичні данні до, та після впровадження. Було доказано фактами та аналітикою, що система управління користувачами і контентом допоможе автоматизувати процеси бізнесу і ефективніше будувати діалог з покупцем та не допускати помилок в роботі.

Для створення системи управління користувачами та контентом була вибрана мова програмування PHP 7.3 так як вона є дуже гнучкою, не застарілою та зручною для виконання даної дипломної роботи. Для зручності обрано IDE – PhpStorm 2020, оскільки для розробки використовувалася мова програмування PHP. У якості локального серверу було взято – Apache, тому що він здатний обробляти великий обсяг трафіку і швидко масштабуватися, навіть без складного додаткового конфігурування. Також використовувався менеджер залежностей Composer, для установки бібліотек. Для завантаження, встановлення та використання фронтенд компонентів було використано – npm. В якості СУБД обрана MySQL 5.6 тому що вона використовувалася для розробки соціальної мережі. За тестування і налагодження відповідає XDebug.

Вивчені принципи роботи системи управління користувачами та контентом соціальної мережі, розглянута документація фреймворку Laravel та платформи для Email розсилки – SendGrid. Було розроблена CRM панель яка має повний контроль над контентом, користувачами та їх бізнесами. За допомогою панелі, адміністратори можуть керувати профілями користувачів а саме:

- Змінювати тип аккаунту;
- Видаляти користувача;
- Блокувати аккаунт та вказувати причину

- Заново відправляти email верифікації або відновлення пароля;
- Зв'язуватися через особисті повідомлення або пошту.

Також збір статистичних даних і аналітики по користувачам, з можливістю експорту.

Після того як користувач створив публікацію, вона потрапляє до панелі для подальших перевірок контенту модераторами. І після перевірки модератори вирішують який статус давати публікації. Публікацію можна редагувати, видаляти, блокувати, доповнювати, назначати редакторів для її розгляду. Доступний повний збір статистики по публікаціям, їх контенту, авторам та глядачам.

Дана соціальна мережа дає можливість не тільки створювати звичайні і бізнес профілі, але і створювати свої магазини для реалізації бізнесу. Вся статистика продажу, за товарам, магазинам користувачів також відображена в CRM панелі.

Після того як користувач створив магазин і додав товар, його магазин відправляється на розгляд адміністрацією.

Для користувачів також розроблена панель управління своїм контентом, профілями, продуктами та іншими можливостями соціальної мережі.

Розроблений модуль сповіщення за допомогою email розсилки, який сповіщає не тільки адміністрацію та редакторів о різних маніпуляціях у панелі, та змін статусів заявок, а і користувачів. Наприклад про схвалення їх публікації або бізнесу, адміністраторами.

Також розроблено збір статистики та аналітики у вигляді таблиць з детальними даними, та модуль експорту цих даних в CSV файли.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Статистичні дані CRM. URL : <https://community.ibm.com/>.
2. Як переглянути статистику з CRM. URL : <https://sendpulse.ua/knowledge-base/crm/statistics-and-analytics/overall>.
3. CRM. URL : <https://snov.io/glossary/ua/customer-relationship-management-crm-ua/>.
4. Аналітика та звіти в CRM. URL : <https://ua.keycrm.app/product/analytics>.
5. На що спроможні CRM-системи аналітики та звітності. URL : <https://wezom.com.ua/ua/blog/na-scho-spromozhni-crm-sistemi-analitiki-ta-zvitnosti>.
6. Бублик В. В. Об'єктно-орієнтоване програмування : Підручник. Київ : ІТ-книга, 2015. 624 с.
7. Щербаков О. В., Парфьонов Ю. Е., Федорченко В. М. Основи об'єктно-орієнтованого програмування : навчальний посібник. Харків : ХНЕУ ім. С. Кузнеця, 2019. 237 с.
8. Порєв В. М. Об'єктно-орієнтоване програмування: конспект лекцій : навч. посіб. для студ. освітньої програми «Інженерія програмного забезпечення комп'ютерних систем» спеціальності 121 «Інженерія програмного забезпечення». Київ : КПІ ім. Ігоря Сікорського, 2022. 271 с. URL : <https://ela.kpi.ua/server/api/core/bitstreams/854443cb-3375-4e76-961a-1f9a74a15b07/content>.
9. Порєв В. М. Об'єктно-орієнтоване програмування. Комп'ютерний практикум : Навч. посібник. Київ : КПІ ім. Ігоря Сікорського, 2022. 105 с.
10. Патерни проектування. REFACTORING GURU. URL: <https://refactoring.guru/uk/design-patterns>.
11. Alan Shalloway, James R. Trott. Design Patterns Explained: A New Perspective on Object-Oriented Design. Addison-Wesley, 2nd edition, 2004, 468 p.

12. Charles Petzold. Programming Windows. URL : [https://docs.google.com/file/d/0B73JwvIHVHaiSFdpekJCOUdoeE0/view?resourcekey=0-JFez95uS9jETJQdFGKY\\_1w](https://docs.google.com/file/d/0B73JwvIHVHaiSFdpekJCOUdoeE0/view?resourcekey=0-JFez95uS9jETJQdFGKY_1w).
13. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison Wesley, 1995, 368 p.
14. Gerbert Schildt. Java: A Beginner's Guide. McGrawHil (Eighth Edition), 2018, 814 p.
15. Martin Fowler, Kent Beck, John Brant, William Opdyke, Don Roberts. Refactoring: Improving the Design of Existing Code. Addison Wesley Professional, Second Edition, 2018, 448 p.
16. Martin Fowler. Inversion of Control Containers and the Dependency Injection pattern. URL : <https://www.martinfowler.com/articles/injection.html#FormsOfDependencyInjection>.
17. Martin Fowler. UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd Edition, 2003, 208 p.
18. Microsoft. Nested Class Declarations. URL : <https://learn.microsoft.com/en-us/cpp/cpp/nested-class-declarations?view=msvc170>.
19. OMG Standards Development Organization. Unified Modeling Language. URL : <https://www.omg.org/spec/UML/>.
20. Robert C. Martin. Agile Software Development, Principles, Patterns, and Practices. 1st ed., Pearson, 2002. URL : <https://dl.ebooksworld.ir/motoman/Pearson.Agile.Software.Development.Principles.Patterns.and.Practices.www.EBooksWorld.ir.pdf>.
21. Robert C. Martin. Design Principles and Design Patterns. 2000. URL : [http://staff.cs.utu.fi/~jounsmmed/doos\\_06/material/DesignPrinciplesAndPatterns.pdf](http://staff.cs.utu.fi/~jounsmmed/doos_06/material/DesignPrinciplesAndPatterns.pdf).
22. Wikipedia. Common User Access. URL : [https://en.wikipedia.org/wiki/IBM\\_Common\\_User\\_Access](https://en.wikipedia.org/wiki/IBM_Common_User_Access).

23. Wikipedia. Expression problem. URL : [https://en.wikipedia.org/wiki/Expression\\_problem](https://en.wikipedia.org/wiki/Expression_problem).
24. Deep dive into using patterns with Selenium: огляд 8 шаблонів, що стануть у пригоді. URL : <https://www.globallogic.com/ua/insights/blogs/dive-into-the-selenium-patterns/>.
25. The PHP Framework for Web Artisans. URL : <https://laravel.com/>.
26. Чому Laravel? URL : <https://laravel.pp.ua/>.
27. 12 головних переваг вибору Laravel Framework для вашого сайту. URL : <https://cases.media/en/article/12-golovnikh-perevag-viboru-laravel-framework-dlya-vashogo-saitu>.
28. Повний стек розробки на фреймворку Laravel. Онлайн курс. URL : <https://abiturients.info/uk/course/85005/85037>.
29. Документація фреймворку PhpStorm. URL : <https://www.jetbrains.com/phpstorm/documentation/>.
30. Документація фреймворку Laravel. URL : <https://laravel.com/docs/>.
31. Документація серверу Apache 2. URL : <https://httpd.apache.org/docs/2.4/>.
32. Документація менеджера пакунків Composer. URL : <https://getcomposer.ucoz.org/index/dokumentacija/0-7>.
33. Документація менеджера пакунків Composer. URL : <https://docs.npmjs.com/>.
34. Документація Git. URL : <https://git-scm.com/doc>.
35. Документація PhpMyAdmin. URL : <https://www.phpmyadmin.net/docs/>.
36. Документація MySQL. URL : <https://dev.mysql.com/doc/>.
37. Документація платформи SendGrid. URL : <https://sendgrid.com/docs/for-developers/sending-email/laravel/>.
38. Портал css-tricks. URL : <https://css-tricks.com/>.
39. Документація сервісу PayPal. URL : <https://developer.paypal.com/docs/>.
40. Налаштування PHP-коду за допомогою Xdebug в PhpStorm. URL : <https://webshake.ru/php-training-course/11>.