

КВАЛІФІКАЦІЙНА РОБОТА

КР.ПЗ – 21.00.00.000 ПЗ

Група ПЗс–2017

Іванюк Т.Я.

2021

**ЗАКЛАД ВИЩОЇ ОСВІТИ  
УНІВЕРСИТЕТ КОРОЛЯ ДАНИЛА**

**Факультет суспільних і прикладних наук  
Кафедра інформаційних технологій**

на правах рукопису

**Іванюк Тарас Ярославович**

УДК 004.415

**Розробка ізоморфної платформи для читання і розміщення коміксів**

Спеціальність 121 — «Інженерія програмного забезпечення»

Кваліфікаційна робота на здобуття освітнього ступеня бакалавра

Науковий керівник  
к.т.н., доцент,  
Пашкевич Олег Петрович

Івано–Франківськ — 2021

**ЗВО «Університет Короля Данила»  
Факультет суспільних і прикладних наук  
Кафедра інформаційних технологій**

Освітній ступінь: «бакалавр»

Спеціальність: 121 «Інженерія програмного забезпечення»

**ЗАТВЕРДЖУЮ  
Завідувач кафедри**

«\_\_\_\_\_» \_\_\_\_\_ 202\_\_ року

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

*Іванюку Тарасу Ярославовичу*

(прізвище, ім'я, по- батькові)

1. Тема роботи

*Розробка ізоморфної платформи для читання і розміщення коміксів*

керівник роботи

*Пашкевич Олег Петрович, кандидат технічних наук, доцент*

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від «\_\_» \_\_\_\_\_ 202\_\_ року

№ \_\_\_\_\_

2. Строк подання студентом роботи

01.06.2021 р.

3. Зміст бакалаврської роботи (перелік питань, які потрібно розробити)

*1. Опис аналогів та використаних технологій*

*2. Проектування та розробка вебдодатку*

*3. Реалізація та огляд платформи*

*4. Охорона праці*

4. Дата видачі завдання

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Огляд існуючих аналогів	25.03.2021	
2.	Проектування вебдодатку	09.04.2021	
3.	Проектування розробка вебдодатку	15.04.2021	
4.	Формування висновків	22.05.2021	
5.	Охорона праці	25.05.2021	
6.	Оформлення пояснювальної записки	28.05.2021	
7.	Оформлення графічного матеріалу та підготовка до захисту роботи	1.06.2021	

Студент \_\_\_\_\_  
(підпис) (прізвище та ініціали)

Керівник роботи \_\_\_\_\_  
(підпис) (прізвище та ініціали)

### Вихідні дані:

*Мова програмування TypeScript. Фреймвоки Next, Nest. РСУБД PostgreSQL.*

---

---

---

---

---

**Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)**

Сторінка	Опис граф. матеріалу	Сторінка	Опис граф. матеріалу
9	Комікс від marvel	27	Навігаційна панель
			Обкладинка манги
11	Статистика популярності західних коміксів	28	Сторінка манги
		29	Налаштування розміру
14	Головна сторінка вебсайту		
15	Навігаційна панель	31	Клієнт-сервер-сервер
	Тіло головної сторінки	32	Приклад моделі (сутності)
16	Вкладка Featured		
		Сторінка манги	33
17	Сторінка Manga List	34	Service BLL взаємодіє з іншими шарами
	Сторінка About Us	35	Взаємодія програми між шарами
18	Manga Catalog	36	Приклад роботи додатка на рівні технологій
19	Карточки манг	37	Один і той же код JavaScript в обох середовищах.
	Слайдер сайту	38	Ізоморфний потік програми
			Діаграма візуалізації сторінки
20	Популярні манги	39	Візуалізація браузера та завантажувальна стрічка
21	Коментарі до комікса	40	Scrum backlog
	Сповіщення оновлення коміксу		
22	Проблема з темною темою	41	Kanban дошка



## ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	8
ВСТУП.....	9
РОЗДІЛ 1. ОПИС ТА ПОРІВНЯННЯ ІСНУЮЧИХ АНАЛОГІВ.....	13
1.1 Огляд предметної області .....	13
1.2 Огляд та аналіз вебдодатків для читання манги .....	14
Висновки до розділу 1.....	30
РОЗДІЛ 2. ПРОЕКТУВАННЯ АРХІТЕКТУРИ ТА ПЛАН РОЗРОБКИ ДОДАТКУ .....	31
2.1 Архітектура.....	31
2.2 Data Access Layer.....	32
2.3 Business Logic Layer .....	33
2.4 Presentation Layer.....	34
2.5 Технологічний стек технологій та ізоморфність .....	36
2.6 Огляд ізоморфної архітектури .....	38
2.7 Управління розробкою дипломного проекту.....	40
2.8 Mind Map.....	42
2.9 Journey map .....	43
2.10 User stories.....	44
Висновки до розділу 2.....	46
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ІЗОМОРФНОГО ДОДАТКА.....	47
3.1 Розробка серверної частини додатку з вибору комікса.....	47
3.2 Розробка інтерактивних речей.....	55
Висновок до розділу 3.....	59

					<b>КР.ПЗ – 04.00.000 ПЗ</b>			
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	<i>Розробка ізоморфної платформи для читання і розміщення коміксів</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Розроб.</i>		<i>Іванюк Т.Я.</i>						
<i>Перевір.</i>		<i>Пашкевич О.П</i>					6	75
<i>Реценз.</i>		<i>Кузнецов С.В</i>				ЗВО «УКД» ПЗс – 2017		
<i>Н. Контр.</i>		<i>Зорін В. О.</i>						
<i>Затверд.</i>		<i>Пашкевич О.П</i>			<i>Пояснювальна записка</i>			

РОЗДІЛ 4. ОХОРОНА ПРАЦІ.....	60
4.1 Умови роботи на робочому місці з комп'ютером .....	60
4.2 Забезпечення правильного мікроклімату у приміщенні .....	62
4.3 Вимоги до безпечної експлуатації комп'ютерної техніки .....	70
Висновки до розділу 4.....	71
ВИСНОВКИ .....	72
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	73

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
						7
Змн.	Арк.	№ докум.	Підп.	Дата		



## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

- XHR – XMLHttpRequest
- HTML – hyper text markup language
- DOM – document object model
- ORM – об'єктно-реляційне відображення
- CSS – cascading style sheets
- SPA – single– page application
- СУБД – система управління базами даних
- API – application program interface
- SSR – server side rendering
- SSG – server side generation
- HTTP – протокол передачі гіпертексту
- UI – user interface

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
						8
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		

## ВСТУП

**Актуальність теми.** За останні 15 років людство все частіше почало використовувати засоби електронної інформації, для вирішення поставлених задач. Вони проводять різноманітні онлайн обчислення, слухають музику, переглядають фільми, мають доступ до безлічі навчальних матеріалів та багато іншого. Ще з великих переваг можна виділити те, що в інтернеті можна читати книги, адже значно зручніше читати або переглядати електронний варіант контенту, ніж шукати потрібну літературу у магазинах. Це заощаджує кошти, час та дозволяє не наражати себе лишній раз на небезпеку у час пандемії.

На фоні простої літератури хочеться виділити так звані комікси. Головною відмінністю від книги є наявність ілюстраційних малюнків. Комікс – це впорядковані, об'єднані певною тематикою рисунки. На даних малюнках зазвичай присутні короткі текстові описи події, які зображені (рис. 1).



Рисунок 1 – Зразок комікса від Marvel

					КР.ПЗ – 04.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		9

Існує Японський варіант коміксів, так звана «манга» (рис. 2). Манга, комікси вперше створені в Японії.

Її визнали формою образотворчого мистецтва, літературним мистецтвом. Випускається вона в різних жанрах, з різною тематикою та для різних споживачів. Людина яка створює мангу називається мангака.

Країни як Китай та Корея мають свої власні назви “манхуа” та “манхва” які можуть відрізнятися не тільки назвою, а й стилем малювання і не бути обов’язково чорно-білою, а випускатися в електронному форматі (в Японії мангу малюють в більшості на художніх аркушах).



Рисунок 2 – Зразок манги

На основі манги знімають аніме та повнометражні мультфільми. Його сюжет в більшості повністю базується на манзі. Що робить мангу ще більш відомою та популярною на сьогодні.

					КР.ПЗ – 04.00.000 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підп.	Дата		

Одним з великих поштовхів була пандемія. На рисунку 3 зображено статистику за останні 5 років і можемо побачити що у 2021 тобто тепер, комікси стали популярнішими на чверть.

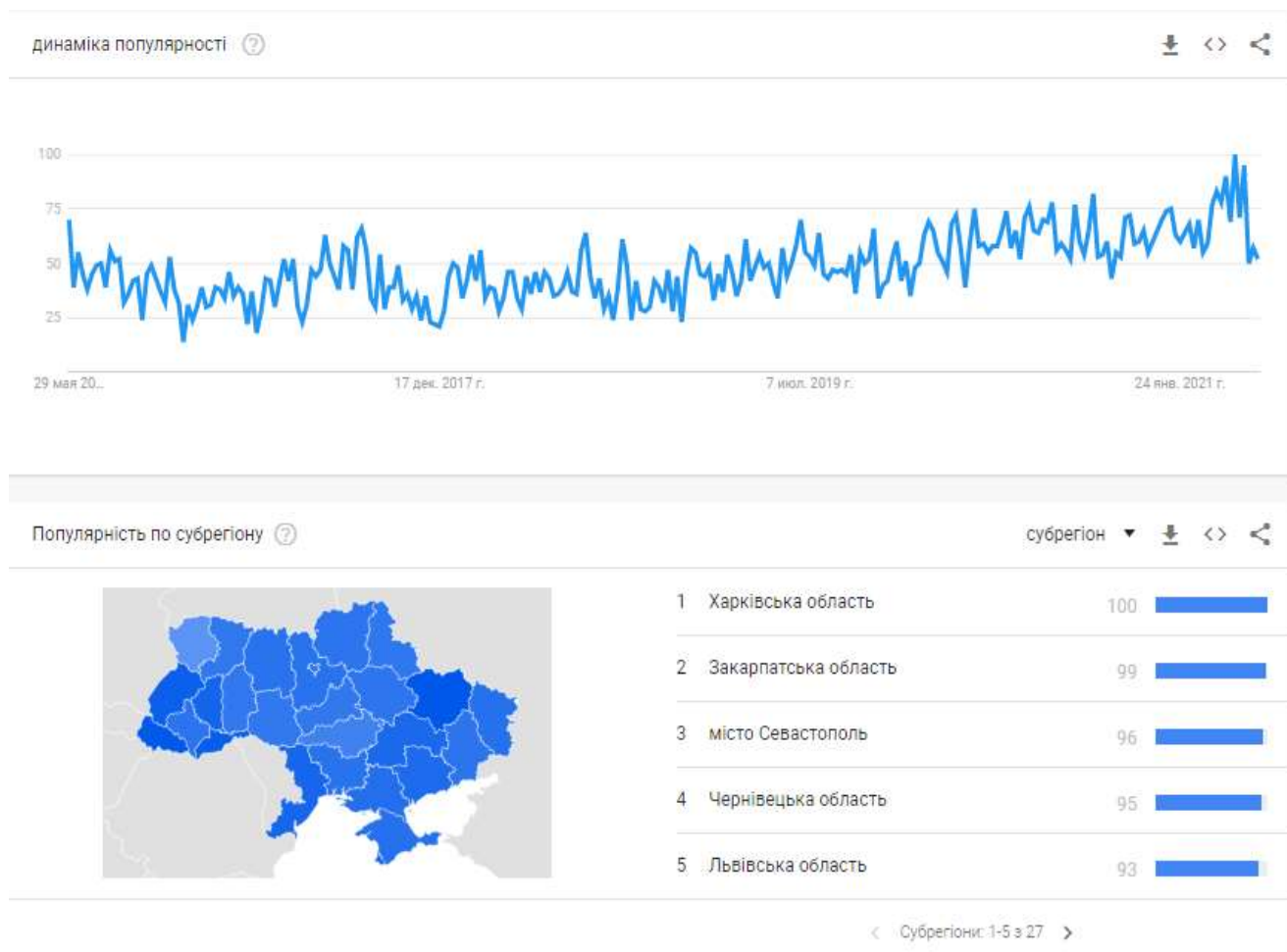


Рисунок 3 – Статистика популярності західних коміксів

Оскільки людям протягом довгого часу було заборонено відвідувати людні місця і взагалі виходити з дому, а якось весело проводити час в умовах пандемії потрібно було, великою перевагою була наявність сервісу який допоміг би користувачам ознайомлюватися з мангою не виходячи із дому.

А так, як спільнота яка цікавиться мангою постійно росте то такий сервіс стає просто необхідним. Він дозволить не тільки читати улюблену мангу прямо з дому, але і значно заощадить кошти своїм користувачам.

Сервіс дозволить користувачам отримувати швидкий доступ до манги як тільки вона отримає переклад на потрібну мову.

**Мета роботи.** Розробити для читання та публікації вебдодаток коміксів. Технології якими буде вирішуватися дана проблема на мовою програмування JavaScript, серверна частина використовує фреймворк Nest.js який своєю чергою використовує TypeORM для роботи з PostgreSQL, клієнтська частина буде розроблятися з допомогою фреймворку Next.js і дозволить нам використовувати SSR.

**Об'єкт роботи.** Процес розробки сучасного вебдодатку призначеного для читання та публікації коміксів.

**Предмет роботи.** Покращення роботи функціонування аналогів.

**Завдання роботи.** Відповідно до вибраної теми завдання розробити додаток який реалізує певні можливості, а саме:

- читання комікса;
- пошук;
- можливість переходу між розділами манги;
- сортування за різними критеріями;
- додавання в улюблене.

**Результати роботи.** Вебдодаток для читання та публікації коміксів.

**Структура роботи.** Розділи 4. Загальний обсяг головної частини 74. Список використаних джерел 16 позицій.

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		12

# РОЗДІЛ 1. ОПИС ТА ПОРІВНЯННЯ ІСНУЮЧИХ АНАЛОГІВ

## 1.1 Огляд предметної області

На цю мить в інтернеті є велика кількість різноманітних вебдодатків, які надають можливість переглядати комікси та мангу.

Проте, якщо переглянути найпопулярніші з них, то можна помітити, що функціонал, інтерфейс та дизайн цих вебсайтів уже досить застарілий для сучасних вимог. Тому було поставлено завдання створити сучасний вебдодаток для перегляду популярних манг та коміксів з урахуванням усіх вимог.

Поставлено вимоги, а саме, що повинен реалізовувати подібний проект. Перш за все, що впадає в очі кожному користувачу, який зайшов на вебсайт це UI. Під цим розуміється правильно підібраний дизайн відповідно до аудиторії вебсайту та зручність користування функціоналом. Дизайн повинен бути виконаний у простій манері, при цьому вебдодаток зобов'язаний містити повний спектр можливостей для перегляду контенту.

Для користувача повинна бути надано меню в якому показується відповідний контент та можливість для сортування, пошуку. Сортування реалізовується за жанром та критеріями, а пошук ми можемо виконати у спеціальній формі, яка буде знаходити потрібні нам манги по назві або автору. На вебсайті мають бути присутні відповідні сторінки, в яких показуватися найпопулярнішої манги на цей час, а також останні оновлення.

Наступним що, хотілось підмітити це система авторизації користувачів. При авторизації користувач отримує доступ до попередньо закритого для нього функціоналу.

Для початку, йому відкривається список улюблених манг, в якому він зберігає ті комікси, які власне йому сподобались і він би хотів прочитати їх у майбутньому. Зважаючи, які манги переглядав користувач, система буде підбирати рекомендовані манги.

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
						13
Змн.	Арк.	№ докум.	Підп.	Дата		

Повинна бути реалізована система сповіщень, користувач має можливість отримувати бажані йому повідомлення по оновлення улюбленої манги. При авторизації з'являється можливість давати свій особистий відгук про певну мангу та оцінювати по певній шкалі, яка буде враховуватись в загальний рейтинг манги. Це надасть можливість для інших користувачів, правильно підбирати матеріал для ознайомлення. Також під кожною мангою можна залишити коментарі для обговорення з іншими відвідувачами. Кожен авторизований користувач отримує доступ до особистого профілю та можливість для його редагування.

Говорячи про інтерфейс, не можливо не відмітити, що вебсайт повинен бути кросбраузерний, тобто має підтримуватись на більшості браузерів і пристроях та підлаштовуватись під їх систему та розмірність екрану. Читання манги є найважливішим елементом типового вебдодатку, тому їх подача повинна бути реалізована у максимально зручному форматі.

## 1.2 Огляд та аналіз вебдодатків для читання манги

Першим аналог виступає платформа MANGA Plus від SHUEISHA (рис. 1.1). Офіційно створений японською компанією журналу.

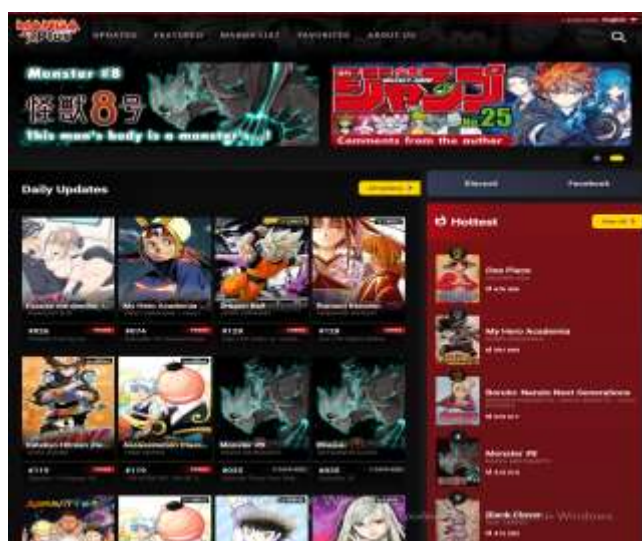


Рисунок 1.1 – Головна сторінка вебсайту

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		14



Манга Weekly Shonen Jump, розпочав свою роботу у 2019 році. Weekly Shonen Jump один з найпопулярніших журналів по манзі у всьому світі.

Вебдодаток містить як англійську, так і іспанську мови перекладу, що не тільки міняє інтерфейс додатку, але й самі розділи коміксів. У Manga Plus є свій додаток в App Store та Play market. Крім цього вебсайт адаптований під різні пристрої, його зручно переглядати, як на телефонах, так і на стаціонарних комп'ютерах. Переглянемо детальніше елементи вебсайту.

В шапці всього вебсайту розташована навігаційна панель (рис. 1.2). яка у свою чергу відкриває доступ до: оновлень, список рекомендованої манги, список усіх новинок, улюбленої манги та інформація про вебсайт.

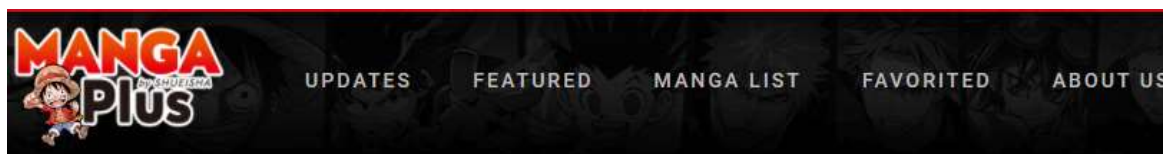


Рисунок – 1.2 Навігаційна панель

Також збоку розташована пошукова форма, в якій ви можете знайти потрібний вам том манги відповідно до назви або автора. В тілі сторінки оновлень в першому блоці знаходяться щоденні оновлення найпопулярнішої манги, карточка та вся потрібна інформація про неї. Лівіше блок з найпопулярнішою мангою та посилання на соціальні мережі проекту такі як discord та facebook (рис. 1.3).

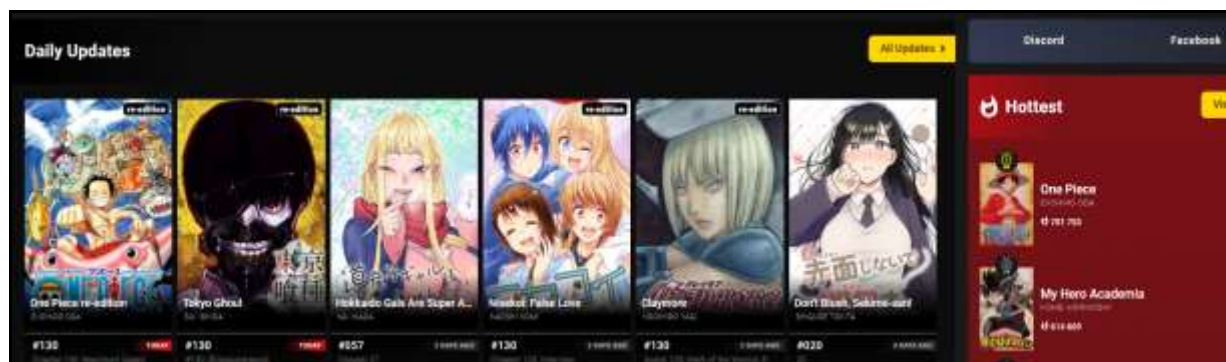


Рисунок 1.3 – Тіло головної сторінки

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		15



Перейшовши у вкладку Featured, користувач має можливість вибрати списки манги, які зображаються у вигляді слайдера та посортовані по відповідних списках. (рис. 1.4).



Рисунок 1.4 – Вкладка Featured

Перейшовши на вибраний комікс, відкривається нова сторінка (рис. 1.5) де є можливість читати опис, кількість загальних переглядів у вибраної манги, автора, список розділів, перехід за яким користувач може розпочати читати мангу, сортування цих розділів від меншого до більшого та навпаки.

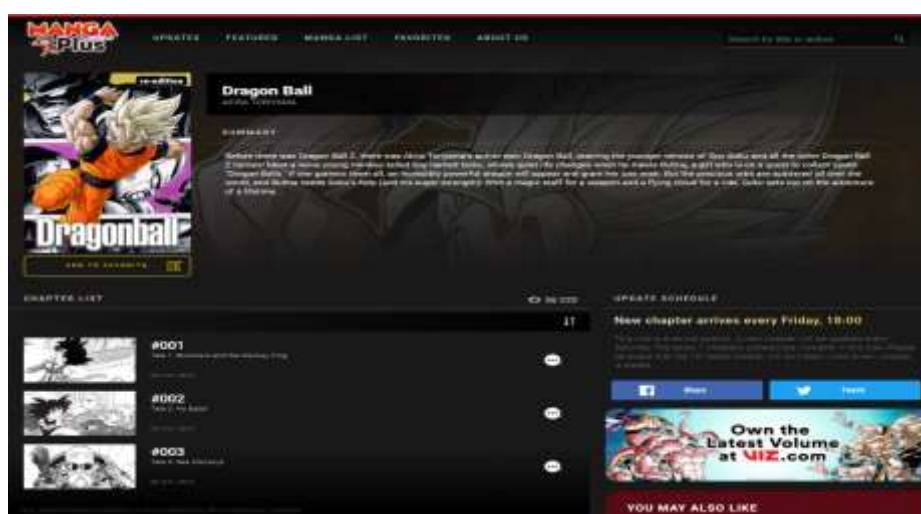


Рисунок 1.5 – Сторінка манги»

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		16

Кожний розділ зображений відповідною фотографією платформа дозволяє ділитися коміксом у соціальних мережах, дає можливість додати конкретну мангу в улюблене, що дозволить нам зберігати мангу, яку ми постійно читаємо, а для користувача це є величезним плюсом, що дозволяє вебдодатку Manga plus бути такою популярною у світі (рис. 1.6).

У вкладці Manga List зображена вся манга, яка присутня на вебсайті. Користувач має можливість їх посортувати або знайти необхідну мангу і розпочати її читання.

Останньою вкладкою головної навігації є сторінка “про засновників проекту”. В ньому показується коротко інформація про даний вебдодаток, а також відгуки деяких авторів (рис. 1.7).

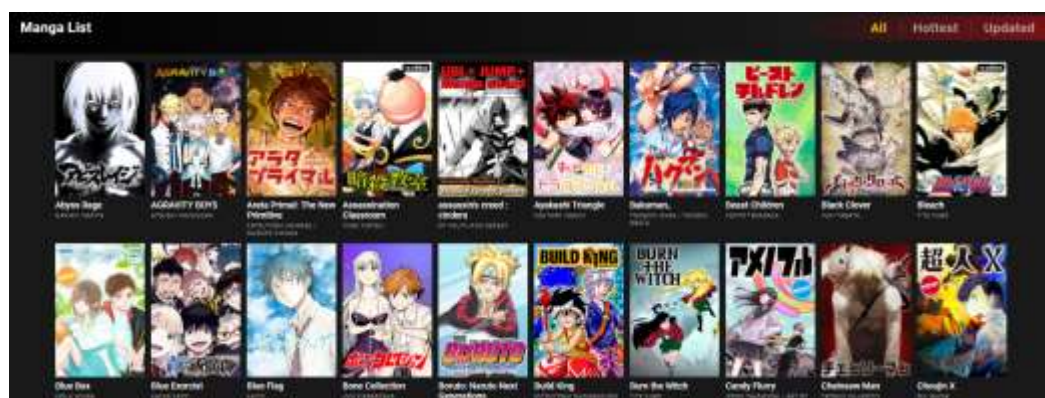


Рисунок 1.6 – Сторінка Manga List.



Рисунок 1.7 – Розділ About Us

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		17

Мінуси платформи MANGA Plus можна виділити:

- зберігання персональної інформації в cookies що є досить великою вадою, адже при зміні пристроїв дані, які були на іншому пристрої ніяк не передадуться;
- також, мінусом є те, що на додатку є тільки одна чорна тема. Тому якщо користувач буде використовувати мобільний пристрій або планшет у світлу погоду текст на чорному фоні буде значно важче читати;
- відсутність авторизованих користувачів зменшує інтерес до ресурсу через відсутність коментарів і неможливість орієнтуватися чи манга хороша чи ні;
- немає можливості побачити скільки розділів конкретної манги уже було прочитано, що є суттєвим мінусом, адже при виході новинки раз у місяць, досить важко слідкувати за прочитаними сторінками. Цю проблему можна вирішити давши позначку про оновлення, яка буде розташована на кожній картці манги;
- немає сповіщень про оновлення манги. Це те що мене як користувача найбільше розчаровує;

Аналог Manga Catalog (рис. 1.8).

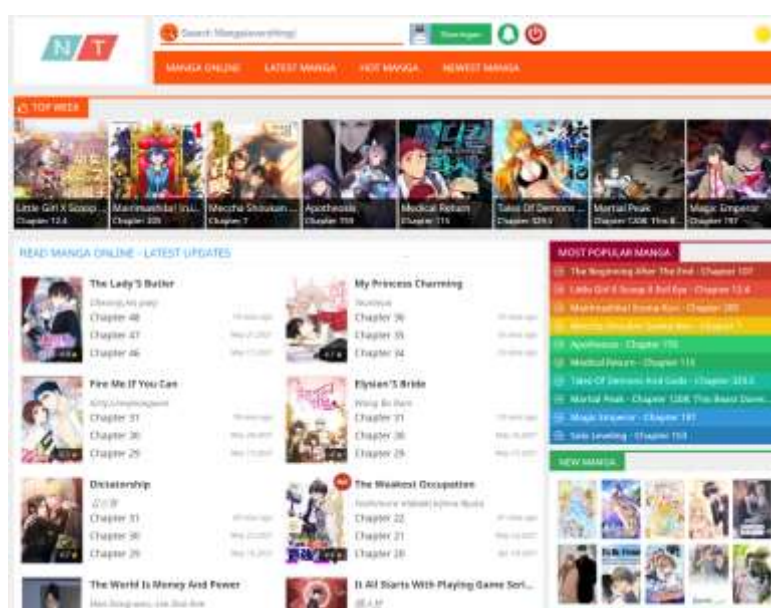


Рисунок 1.8 – Manga Catalog

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		18



Вебсайт працює уже довгий час і зарекомендував себе як один із найшвидших ресурсів перекладу та публікації манги. Дизайн вебсайту не можна схарактеризувати як щось дуже інноваційне, але функціонал який він надає є набагато більшим, ніж Manga Plus.

Порівнюючи з попереднім варіантом тут також реалізований пошук та зручна навігація по вебсайту. На картці коміксу зображено останні три розділи та дату останнього оновлення (рис. 1.9).



Рисунок 1.9 – Карточка манги

Нижче шапки вебсайту виступає слайдер, де користувач може переглянути найпопулярнішу мангу (рис. 1.10).



Рисунок 1.10 – Слайдер сайту

Ще нижче можна побачити просунуту фільтрацію коміксів, де можна вибрати певну мангу та жанри. Величезним плюсом є те, що на сайті реалізований перемикач тем, чого не було в Manga Plus. З правого боку

вебсайту розташований sidebar, який показує найбільш популярні комікси, вони позначені різними кольорами (рис. 1.11).

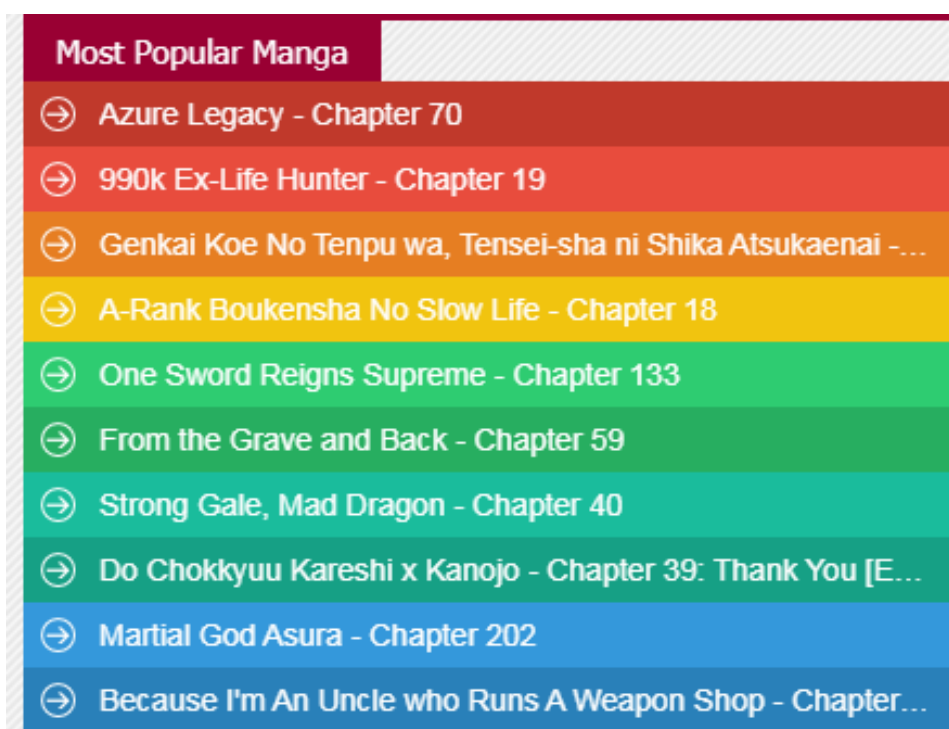


Рисунок 1.11 – Популярні манги

Зайшовши на конкретний комікс відкривається велика кількість корисної інформації, а саме: статус, рейтинг манги, кількість переглядів, автор, опис і так далі. Головною функціональністю таких ресурсів є можливість додавати цікаву вам мангу в улюблене.

Після зміни пристроїв ви можете авторизуватися, не боячись, що збережені вами дані не пропадуть, на відмінно від Manga Plus. Реалізована можливість ділитися мангою у соціальних мережах, що є корисною фішкою для тих, кому це потрібно.

Ще однією корисною функцією є стрілка, яка є позиціонована в правій частині внизу, що дозволяє перейти з будь-якої точки вебсайту на початковий екран. На відмінну від Manga Plus.

Список розділів доповнюють інформаційні блоки як «кількість переглядів в розділі» та «дата публікації».

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
						20
Змн.	Арк.	№ докум.	Підп.	Дата		

Ще одною корисною функцією, є те що на вебсайті є можливість залишати коментарі, це дозволить людям економити свій час і дізнатися чи варто читати дану мангу чи немає ніякого сенсу (рис. 1.12).

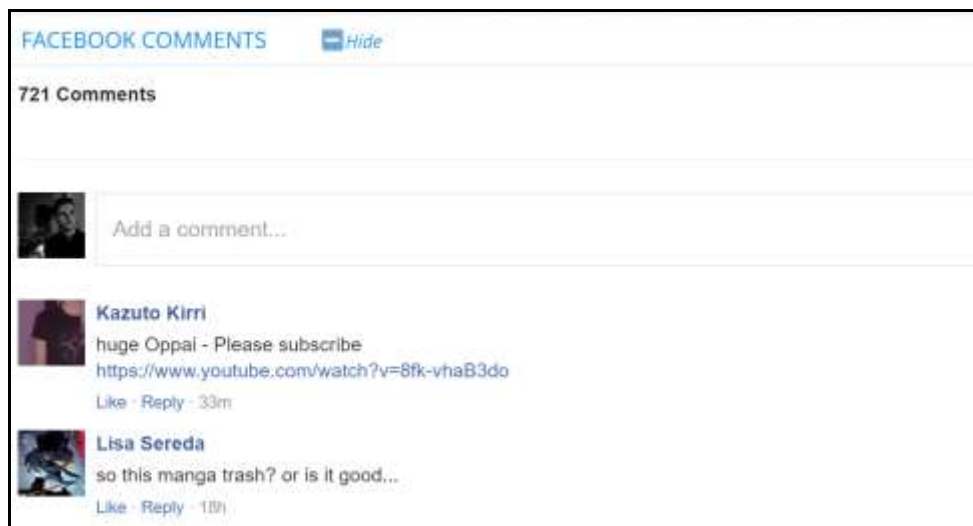


Рисунок 1.12 – Коментарі до комікса

В теперішній час більшість ресурсів повинні містити в собі режим сповіщень, через те, що людину як користувача, приваблює користуватися ресурсом, тобто ті комікси які ви додали собі в закладки при оновленні будуть повідомляти в іконці з дзвіночком цифрою (рис. 1.13).

У вебсайту також є якір посилання, який дозволить користувачу, у будь-якій частині вебсайту повернутися на початок сторіки.

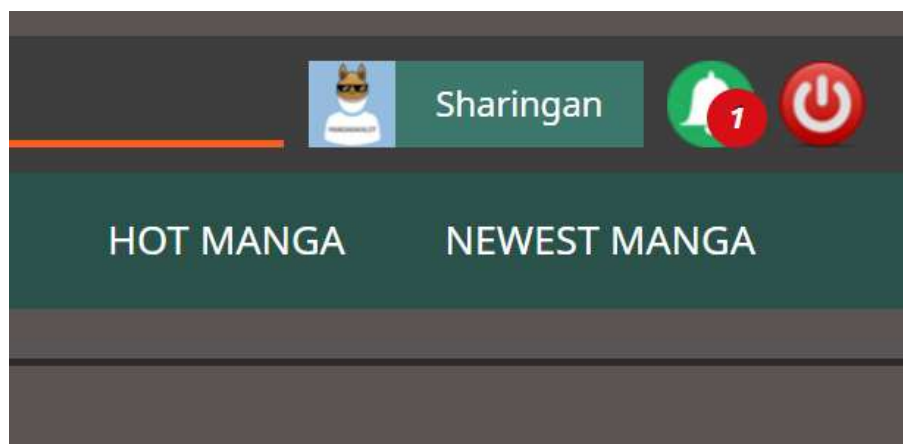


Рисунок 1.13 – Сповіщення оновлення коміксу

					КР.ПЗ – 04.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		21

З основних недоліків вебсайту Manga Catalog можна виділити:

Якщо увімкнути темну тему, то при переході на інші вкладки спершу зображається оригінальна тема, а через одну-дві секунди темна, що є досить неприємно (рис. 1.14).

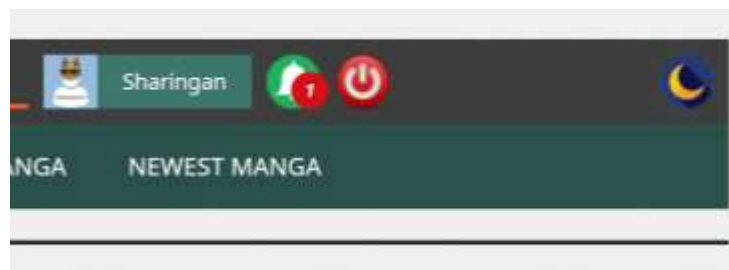


Рисунок 1.14 – Проблема з темною темою

Також дизайн вебсайту виглядає досить застарілим, він не йде порівнянню з Manga Plus. Аналог ReadManga (рис. 1.15).

Є один із найпопулярніших вебсайтів для читання манги. ReadManga зарекомендував себе за те, що він перекладає та публікує мангу одним із перших на просторах інтернету.

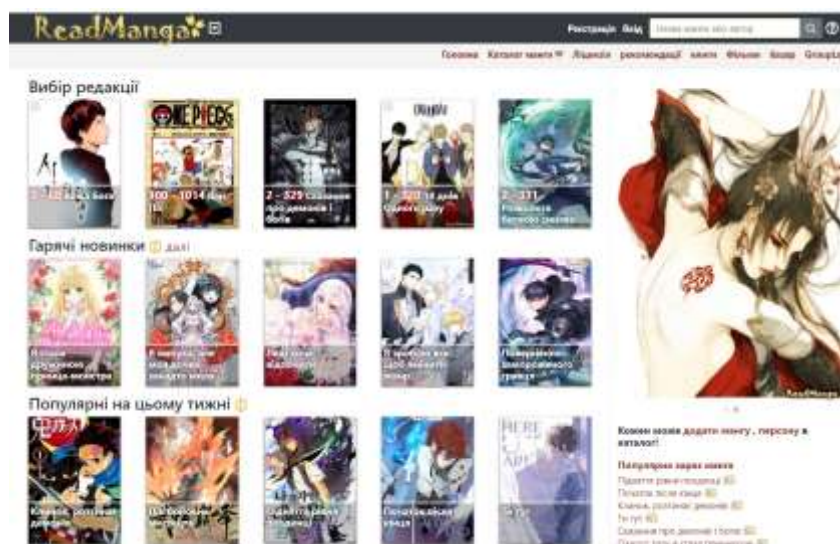


Рисунок 1.15 – Головна сторінка

Секції на вебсайті розділені дуже зручним чином, на головній сторінці є три секції:

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
						22
Змн.	Арк.	№ докум.	Підп.	Дата		

- вибір редакції;
- гарячі новинки;
- популярні на цьому тижні;

Кожна секція містить в собі п'ять коміксів. Карта з мангою містить в собі:

- назву манги;
- кількість манги;
- кількість прочитаної манги користувача;

Останні дві секції містять в собі кнопку «далі» перейшовши на яку відкривається сторінка з каталогом де уже є певні вибрані фільтри.

Сторінка каталогу містить в собі просунутий механізм фільтрування коміксів (рис. 1.16), а саме:

- жанр;
- персонажі;
- категорії;
- сортування.

Тобто це найкраща фільтрація серед усіх аналогів. Також можна скористатися розширеним пошуком, і будь-який користувач з великою ймовірністю знайде те що він шукає.



Рисунок 1.16 – Фільтрація вебсайту ReadManga

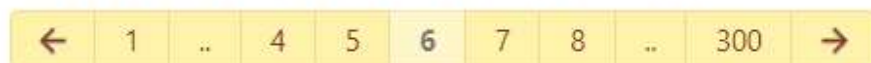
У вебсайту є цифрова та стрілкова пагінація, для переходу на сторіки яка розміщена як і на початку розташування карток вебсайту, так і в кінці (рис. 1.17).

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		23



Карточка манги на сторінці каталогу відрізняється від карточки головної сторінки.

## Список манги, новинки - с



Димо -Увертюра-  
DEEMO -Prelude-

повсякденність , фентезі , драма

Рисунок 1.17 – Пагінація та катка коміксу

Є можливість зареєструватися та зберігати комікси в закладках, також зареєстровані користувачі можуть писати коментарі до коміксів і оцінювати їх. Вебсайт дозволяє встановлювати задній фон вебсайту на різні кольори.

Але на жаль це функція є можливою тільки для сторінок з читанням розділу, на інших сторінках такої функції не було знайдено (рис. 1.18).



Рисунок 1.18 – Перемикач фону розділу

У шапки додатка вебсайту є ще розділи: книги (в цьому посиланні вебсайт надає доступ до ранобе, це книги по яких пишуться манги, манхви, махухи), фільми (в цьому посиланні вебсайт надає доступ до аніме) (рис. 1.19).

					КР.ПЗ – 04.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		24

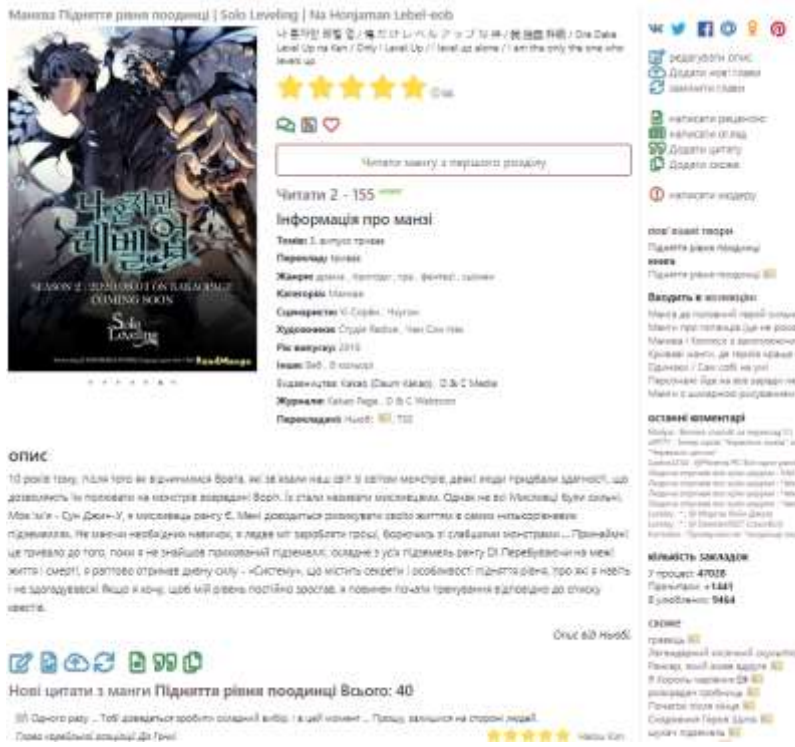


Рисунок 1.19 – Сторінка вибраної манги

Секція з вибраною мангою не може порадувати користувача крутим дизайном, але інформація яка є на сторінці та її розміщення є найбільш зручним для читання. Також тут навіть є цитати з коміксу, що робить вебсайт досить унікальним. Є можливість писати коментарі не тільки під коміксом, а і під кожним розділом. Можливість подивитися на популярність манги, тобто скільки людей поставили оцінку.

Також у додатку є секція з останніми оновлення коміксів біля назви є цифра з номером останнього оновлення коміксу. Основна особливість є те що є можливість сортування по днях оновлення, цієї можливості не було знайдено серед усіх аналогів та кандидатів (рис. 1.20).

Оновлення за червня 2021				
← травня 2021				
понеділок	вівторок	середа	четвер	п'ятниця
31	01	02	03	04
112	188	74	167	212

Рисунок 1.20 – Пошук оновлення за датою

З мінусів вебсайту, уже не актуальний дизайн, та мінус секції «останні оновлення» в якій немає можливості побачити фото манги (рис. 1.21).

Останні оновлення каталогу Календар		Дата
Мінімалізм 1 - 52 Ловець		12.06
Мінімалізм 1 - 51 На допомогу!		12.06
Нестерпний аромат апельсинів 1 - 3		12.06
Нестерпний аромат апельсинів 1 - 2		12.06
Початок після кінця 4 - 23 Перше зіткнення		12.06
Міф: Дистиляція 1 - 1 Хто вона?		12.06

Рисунок 1.21 – Останні оновлення

Останнім прикладом для розгляду є ресурс Manga.in.ua (рис. 1.22).

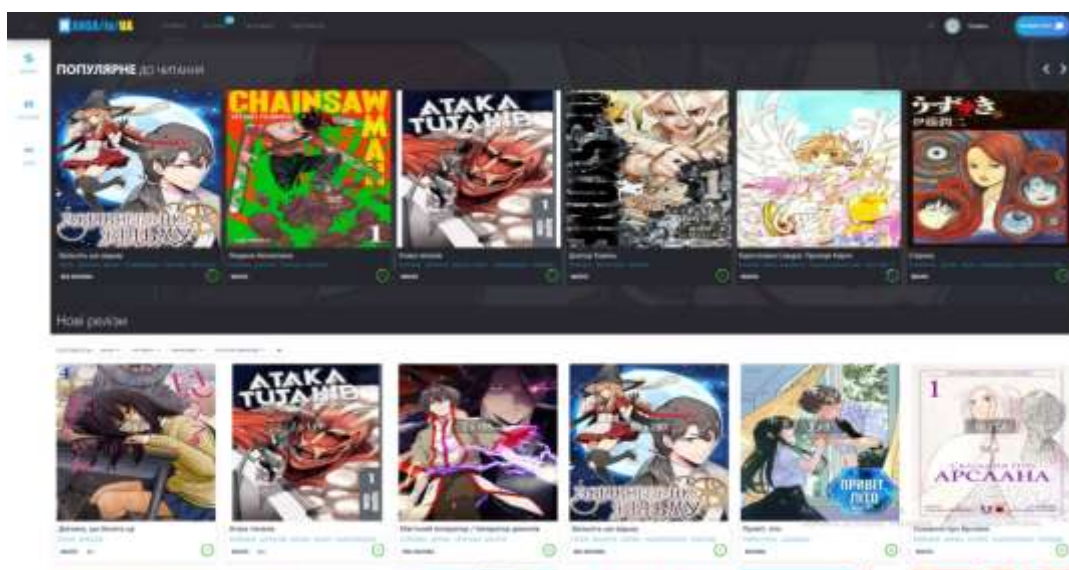


Рисунок 1.22 – Головна сторінка

Досить популярний український вебсайт коміксів, який перекладається волонтерами. Manga in ua має досить сучасний дизайн та цікавий функціонал, який виділяє його серед інших додатків.

Кількість розділів, які переведені на українську та ще одну інтерактивність дає лівий sidebar, де після кліку в модальному вікні можна вибрати жанри. В шапці вебсайту ми можемо перейти на каталог, читальню, підтримку, пошук, профіль та коментарі (рис. 1.23).

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		26



Рисунок 1.23 – Навігаційна панель

Знизу від шапки знаходиться слайдер з популярними коміксами перемикання здійснюється за допомогою відповідних кнопок які зображені стрілкою.

Досить унікальна сама карта, яка дає інформацію якої країни комікс (чи це манга, махва чи манхуа). Частина жанрів коміксу, оцінка та назва (рис. 1.24).



Рисунок 1.24 – Обкладинка манги

Середня частина вебдодатку займає секція з новими релізами, де є досить просунута функція сортування коміксів і є можливість відсортувати за перекладом чи статусом перекладу (продовжується чи завершений), чого не було на інших сайтах, які були досі розглянуті. Особливою функцією є те, що натиснувши на коментарі в шапці, ви зможете переглянути всі останні коментарі за день на різну мангу, що також що робить сайт більш унікальним ніж інших.

Вебпрограма надає статистику перекладених розділів манги, а також кількість сторінок, яку ви прочитали якщо користувач зареєстрований, що дуже виділяє даний вебсайт серед інших вебдодатків. Також присутня особливість

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		27

«В списках користувачів», де користувач може побачити хто в який тип збережених додав мангу.

На основі цієї інформації користувач може зрозуміти, як окремий комікс можна посортувати в збережених:

- читаю;
- прочитано;
- заплановано;
- перечитую;
- відкладено;
- покинуто;
- улюблене;

Сайт розроблений у стилі full-screen, тобто сайт розтягнутий на всю ширину екрану, що для читача дає можливість сховати у браузері верхні вкладки та дозволить не масштабувати сторінку, щоб побачив всю картину. Що виділяється дуже сильно, і це робить оцінку манги стильною (рис. 1.25).

Сайт [manga.in.ua](http://manga.in.ua) дає величезну кількість інформації про мангу, в порівнянні з іншими застосунками. Причина цьому, команди перекладачів, тому читачі знатимуть наперед, який переклад і статус манги.

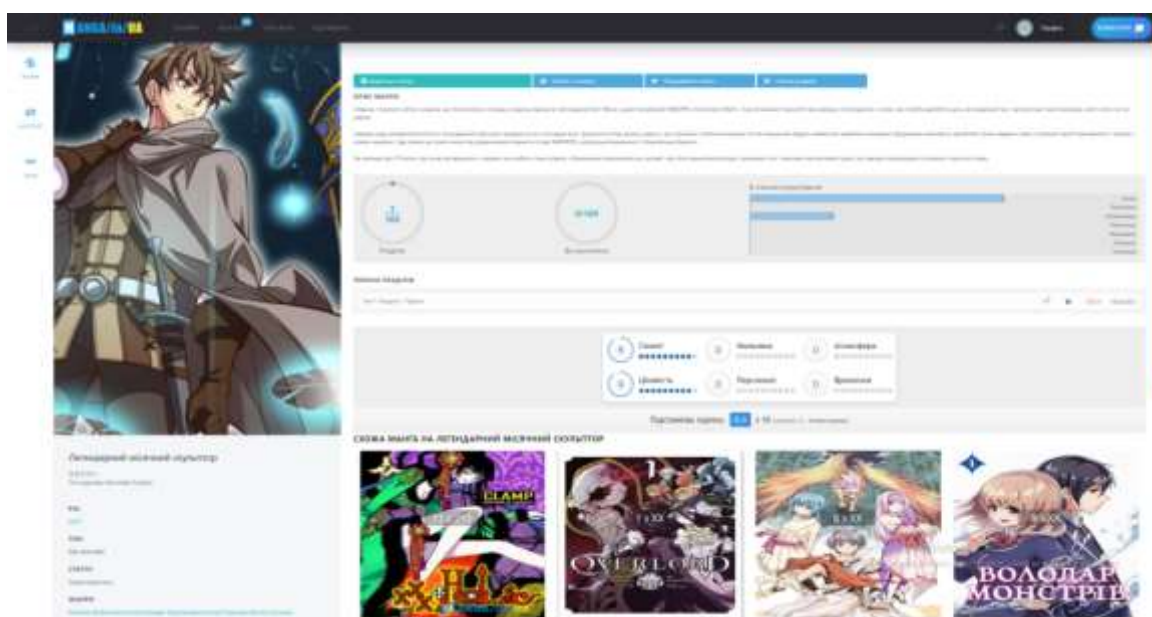


Рисунок 1.25 – Сторінка манги

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		28



Починаючи читати розділи, з'являється зручний перемикач розміру картинки (рис. 1.26).

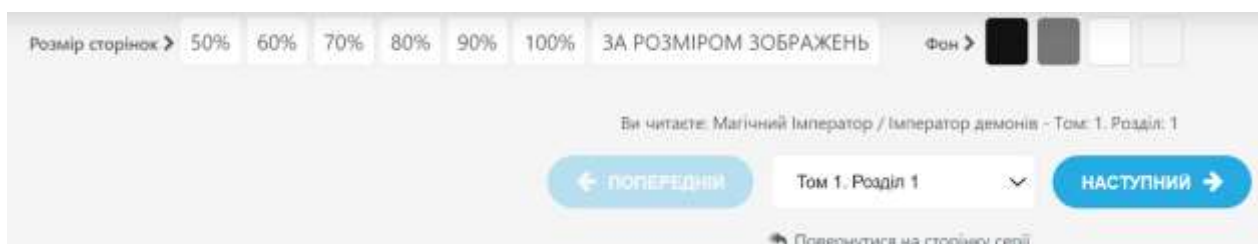


Рисунок 1.26 – Налаштування розміру

Серед усіх розглянутих сайтів про комікси, це перший сайт, де реалізована така функція. Також перемикач хороший тим, що дозволяє вибрати 4 кольори, що є досить зручно.

Перейдемо до основних мінусів сайту manga.in.ua. Перш за все сайт не адаптується під великі розширення екрану (рис. 1.27), для прикладу користувач з комп'ютера з монітором в якого розширення 4k (4096 на 2160) буде бачити розтягнуті картинки сайту. Також проблемою є великий об'єм сайту, що спричиняє повільне завантаження сторінок на слабких комп'ютерах.



Рисунок 1.27 – Проблема адаптації сайту для екранів з великим розширенням

Також може виникати певна помилка, яка просить вас почекати деякий проміжок часу і повторити знову. Це може відібрати бажання у користувача

читати на цьому ресурсі. На даному ресурсі може бути проблема з карточкою манги, при якій велика кількість інформації починає випадати за межі блоку, що явно не є плюсом для сайту.

Проблема виникає, коли користувач натискає хрестик і при цьому він нічого не вибрано у сортуванні манги, що перезавантажує сайт.

### **Висновки до розділу 1**

В розділі розглянуто предметну область роботи, проведено дослідження про комікси та мангу. Було проаналізовано найпопулярніші аналоги. Описано загальну характеристику та можливості ресурсів надання читання коміксів. Базуючись на проведеному аналізі було сформовано вимоги до розробки програмного забезпечення.

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
						30
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		

## РОЗДІЛ 2. ПРОЕКТУВАННЯ АРХІТЕКТУРИ ТА ПЛАН РОЗРОБКИ ДОДАТКУ

### 2.1 Архітектура

Опираючись на те якою повинна бути хороша для проекту, на початковому етапі в систему має сенс закладати лише необхідний функціонал (принцип YAGNI – you is not gonna need it, «Вам це не знадобиться»), що робить процес розробки й супроводу програми простішим та ефективним. Програму з хорошою архітектурою буде легше розширювати та змінювати.

Було поставлено список універсальних критеріїв:

- Гнучкість системи. Зміна одного фрагмента системи не повинно впливати на інші фрагменти. Тому код повинен бути слабо зв'язаним, щоб внесення найбільш ймовірних змін вимагало найменших зусиль.
- Масштабованість процесу розробки. Як щоб до цього проекту прийшов ще один програміст то архітектура повинна дозволяти розподілити процес розробки.
- Можливість використання коду повторно. Систему бажано проектувати так, щоб її фрагменти можна було повторно використовувати в інших системах.

Приклад архітектури на рівні клієнт сервера (рис. 2.1).

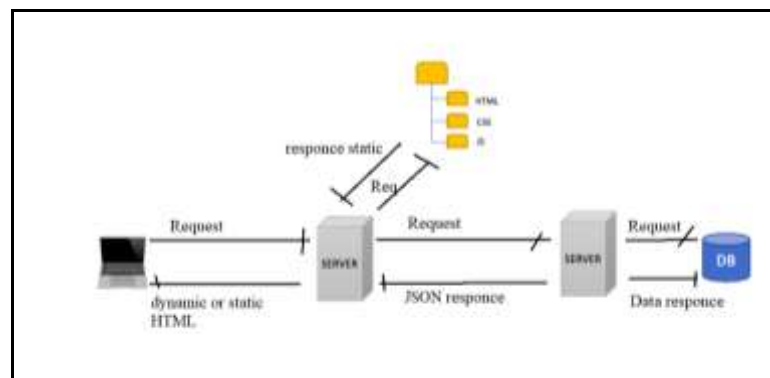


Рисунок 2.1 – Клієнт-сервер-сервер

					КР.ІІЗ – 04.00.000 ІІЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підп.	Дата		



## 2.2 Data Access Layer

Розробка шару доступу до даних (Data Access Layer) далі DAL. При проектуванні життєвого циклу додатку велику роль грає питання як організувати доступ до даних, проблему поділу бізнес-логіки та шару доступу до даних вирішує архітектурний шаблон DAL. В теперішній час напряду звертатися до реляційної СУБД (РСУБД) не дозволить швидко розробити прототип та бути одним засобом для вирішення завдання по зберіганню даних при розробці додатків, тому тут було прийнято рішення виділити в окремий шар роботу з даним.

Рівень DAL рівень містить в собі доменні моделі із проекрованої таблиці (Entity), тобто це як у випадку із більшістю випадків виступає клас у якого поля описані як поля в базі даних. Модель завжди повинна властивості ідентифікатора (поле id). Сутність може також складатися з інших властивостей відносин, які моделюють зовнішні ключі між таблицею до іншої таблиці в базі даних (рис. 2.2).

```
@Entity(name = "chapter")
export class Chapter {
    @PrimaryGeneratedColumn()
    id: number;

    @Column({ unique: {default: 1, nullable: true}})
    mangaId: number;

    @Column({ unique: {type: "real", scale: 1}})
    chapter: string;

    @Column()
    title?: string;

    @Column({ unique: {type: "int", default: 1, nullable: true}})
    volume: number;

    @CreateDateColumn({ name: "created_at" })
    createdAt: Date;

    @UpdateDateColumn({ name: "updated_at" })
    updatedAt: Date;

    @ManyToOne({ type: "many_to_many" }) => Manga, { onDelete: "cascade" }
    manga: Manga;

    @OneToMany({ type: "one_to_many" }) => ChapterTags, { onDelete: "cascade" }
    chapterTags: ChapterTags[];
}
```

Рисунок 2.2 – Приклад моделі (сутності)

					КР.ІІЗ – 04.00.000 ІЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		32

Метою DAL є відділення бізнес – логіки від шару з даним. Кожного разу, коли бізнес-об’єкту потрібно отримати доступ до рівня даних, ви використовуєте виклики методів у DAL. У якості DAL виступає ORM яка дозволить писати SQL або noSQL запити використовуючи об’єктноорієнтовану парадигму тобто незалежно від мови програмування просто повинна бути написана ORM бібліотека і використовувати можна будь-де, тобто використовуючи код мови програмування. Хоча використання ORM має в собі багато недоліки, такі як лишні запити, та те що вона в якомусь сенсі порушує об’єктно орієнтоване програмування, але відмова від ORM призведе до ще одного шару абстракцій.

Реалізація запитів до бази даних вебдодатку здійснюється через те що в ORM реалізований шаблон Repository а сам запит складається з шаблону builder де кожний метод робить певну sql дію , кожний в конструкторі та що основне повертає сам інстанс класу.

В організованому багатшаровому додатку відповідальність за отримання даних з бази даних повинна лежати на рівні інфраструктури (рис. 2.3).

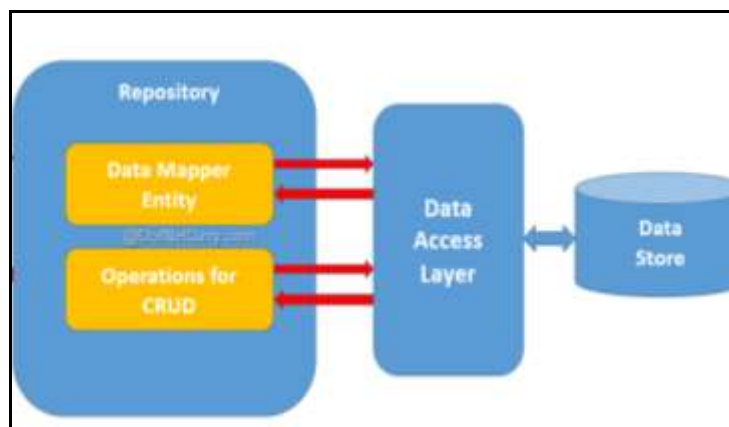


Рисунок 2.3 – Приклад взаємодії Repository і DAL

## 2.3 Business Logic Layer

Було спроектовано шар бізнес-логіки в цьому шарі виконуються основні use case операції, специфічні для роботи, яку виконує програма. Бізнес-рівень

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		33

який повинен інкапсулювати всю бізнес-логіку, отримує об'єкти з рівня доступу до даних і передає їх на рівень API, або, навпаки, отримує дані з рівня API та передає їх на рівень даних.

Наприклад, на проєкті з коміксами Business Logic, який обробляє запит клієнта на збереження коміксів у закладки: перевіряє чи в cookies користувача є token доступу, при успішній перевірці зберігається в таблицю id користувача та id манги.

Основна робота буде відбуватися саме в цьому шарі. Сутності для роботи з BLL заведено називати сервісам (Services) Шар бізнес-логіки ніякому разі не повинен напряму звертатися до бази даних, чи працювати на пряму з протоколом передачі даних (рис. 2.4).

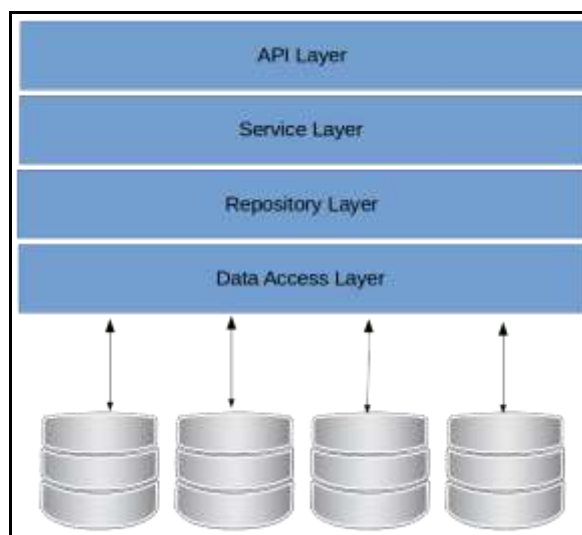


Рисунок 2.4 – Service BLL взаємодіє з іншими шарами

## 2.4 Presentation Layer

Presentation Layer у проєкті відповідає за представлення та взаємодію з користувачем. Це шар поділити на два підшари. Підшар API використовується, щоб захистити зовнішній світ або клієнтів від знання структури. Іншими словами, шар API являє собою міст між клієнтом та додатком, міст, який передає дані від програми до клієнта. За допомогою цього механізму буде

забезпечена стабільна безпека разом із передачею даних. API забезпечують уніфікований (у єдиній формі) спосіб відкрити інформаційні ресурси та легко ними ділитися. Передача даних в шарі здійснюється через протокол HTTP на рівні кодової абстракції в проекті працює через сутність Controller. Ця сутність може виступати як класом або функцією яка відпрацьовує за певним маршрутом (посиланням), і може відати HTML чи JSON.

Приклад взаємодії клієнтської частини API та іншими шарами програми (рис. 2.5).

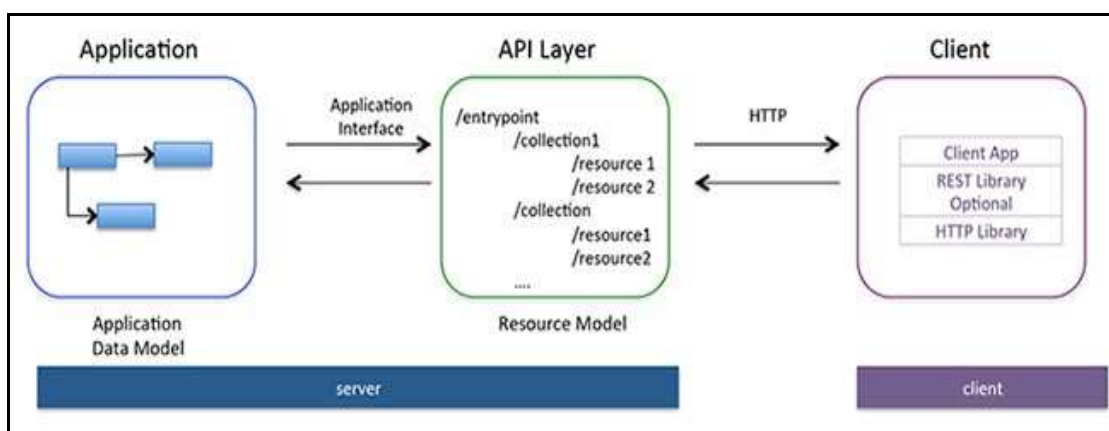


Рисунок 2.5 – Взаємодія програми між шарами

Server side rendering / generation як частина PresentationLayer, яку було впроваджено в проект і що дозволило створювати візуалізацію на стороні сервера та статичні вебдодатки, це буде досягнуто за допомогою за допомогою ізоморфних методів фреймворку які були імпортовані з Next.js. Він має безліч чудових функцій та переваг, що може зробити розробку на ньому набагато кращою альтернативою, а ніж будь-які інші програми для створення шаблонів. Це допомогло зручніше масштабувати вебдодаток. Через те, що клієнтська частина буде орієнтуватися на ту саму архітектуру DAL, BLL, PL.

Плюси використання такого підходу полягає в тому що у нас появляється спільний код, яким ми можемо ділитись між нашими шарами, для прикладу інтерфейси в додатку є спільними починаючи від доменних entity до клієнтського сховища.

На рисунку (рис. 2.6) показано докладніше як працює SSR.

- повинен бути сервер, який виконує наш додаток точно так, як робив би це користувач в браузері. Роблячи запити на необхідні ресурси, відображаючи весь необхідний HTML, наповнюючи стан;
- сервер віддає клієнту наповнений HTML, наповнений стан, а також віддає всі необхідні JS, CSS та інші ресурси;
- клієнт, отримуючи HTML і ресурси, синхронізує стан і працює з додатком як зі звичайним Single Page Application. При цьому важливим моментом є те, що стан має синхронізуватися;

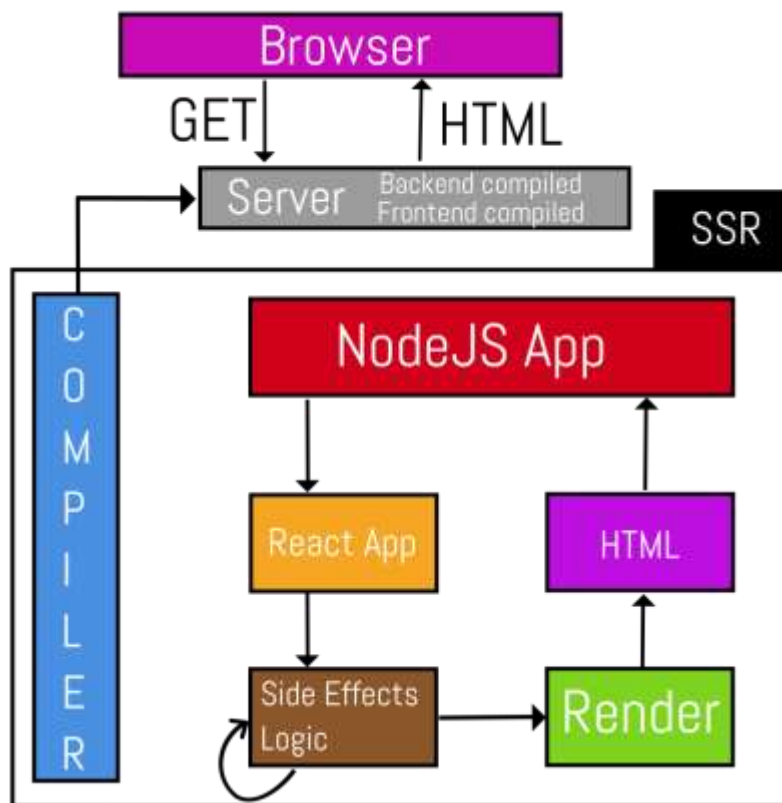


Рисунок 2.6 – Приклад роботи додатка на рівні технологій

## 2.5 Технологічний стек технологій та ізоморфність

Проект з коміксами є розроблявся як ізоморфним вебдодатком. Під час першого запити користувача на отримання ресурсу візуалізація відбувається на стороні сервер передається браузеру. Візуалізація робиться за допомогою

					<b>КР.ІІЗ – 04.00.000 ІЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		36

серверних методів з NodeJs. Під час навігації по сторінках, кожна сторінка показує візуалізацію HTML в браузері. При використанні інтерактивних речей для цього використовується SPA принципи, це й принцип має назву «ізоморфною передачею». На сервері ви зберігаєте стан програми, а потім надаєте цей стан браузеру. Браузер використовує цей стан для завантаження SPA-версії програми.

Без цієї ізоморфної передачі обслуговування користувач повинен дочекатися завантаження серверної сторінки, а потім довше чекати повного повторного відтворення вмісту у браузері. Цей процес називається гідрацією (hidryte).

Вебдодаток коміксів покладається на повторне використання якомога більшої кількості коду між сервером та браузером. Він використовує можливість JavaScript працювати в кількох середовищах.

Сумуючи все вище: вебдодаток, який потрібно розробити серверну вебпрограму з SPA програмою. З одного боку, потрібно скористатися швидкою, продуктивною візуалізацією із сервера.

З іншого боку, потрібно обробляти складні дії користувача у браузері (наприклад, відкриття модального вікна). На рисунку 2.7 потік у дії.

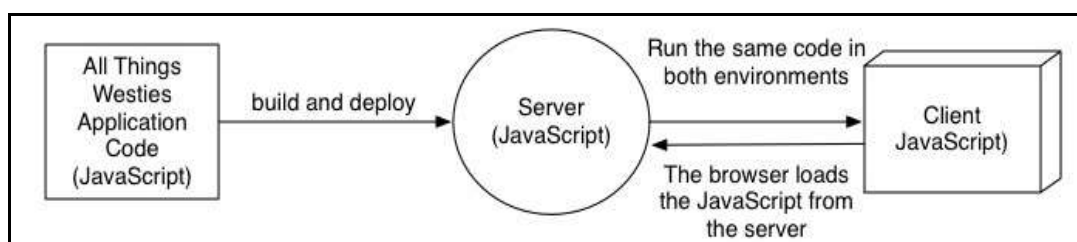


Рисунок 2.7 – Один і той же код JavaScript в обох середовищах.

Він показує, як код для додатка з коміксами розгортається на сервері та клієнті. Оскільки вебдодаток користуємось перевагами JavaScript, що працює в обох середовищах, той самий код, який працює в браузері та розмовляє з нашим API також одночасно працює на сервері для спілкування з базою даних.

## 2.6 Огляд ізоморфної архітектури

На рисунку 2.8 показані всі етапи отримання ізоморфної програми, яка зображається та реагує на введення та дії зроблені користувачем, як SPA програма, починаючи з того часу, коли користувач переходить на localhost веб адресу.

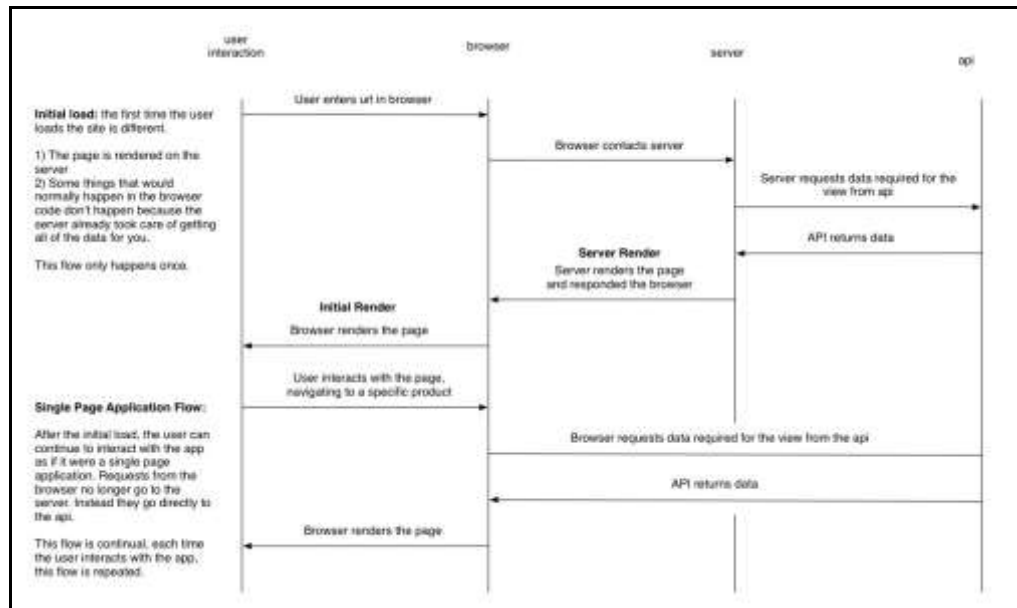


Рисунок 2.8 – Ізоморфний потік програми

На рисунку 2.9 показано потік у браузері від точки, в яку браузер отримує html, до точки завантаження програми.

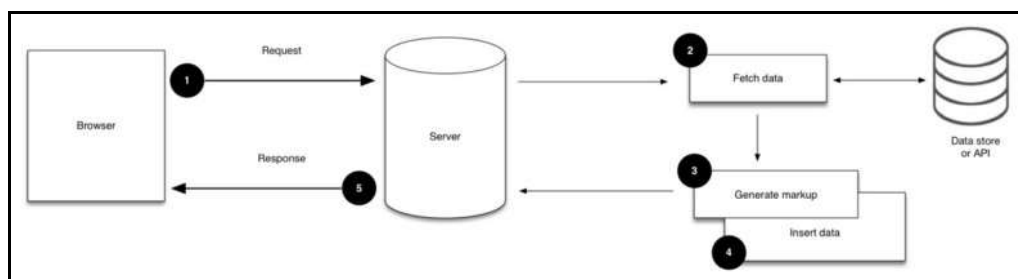


Рисунок 2.9 – Діаграма візуалізації сторінки

Алгоритм серверної візуалізації роботи наотримання запитів:

- Сервер отримує необхідні дані для цього запиту. Це може бути як з постійного сховища даних, як-от база даних MySQL або NoSQL, так і із зовнішнього API;
- Після отримання даних сервер може створити HTML. Він генерує розмітку за допомогою віртуального DOM React за допомогою методу `renderToString`;
- Сервер вводить дані з кроку 2 в HTML, дозволяючи браузеру отримати доступ до них пізніше;

Сервер відповідає на запит браузеру повністю побудованим HTML. Візуалізація в браузері:

- сервер отримує запит;
- на маршрутизатор додатків обробляє запит і збирає дані, необхідні для частини нашої програми, яку запитують.
- сервер генерує HTML для нашої вебсторінки, використовуючи дані, зібрані для сторінки;
- сервер відповідає на запит для `localhost:300/catalog` повністю побудованим HTML;

Сервер робить дві важливі речі. Спочатку він отримує дані, необхідні для подання.

Потім вони беруть ці дані та використовують їх для візуалізації DOM.

Рисунок 2.10, на якому показано потік на сервері.

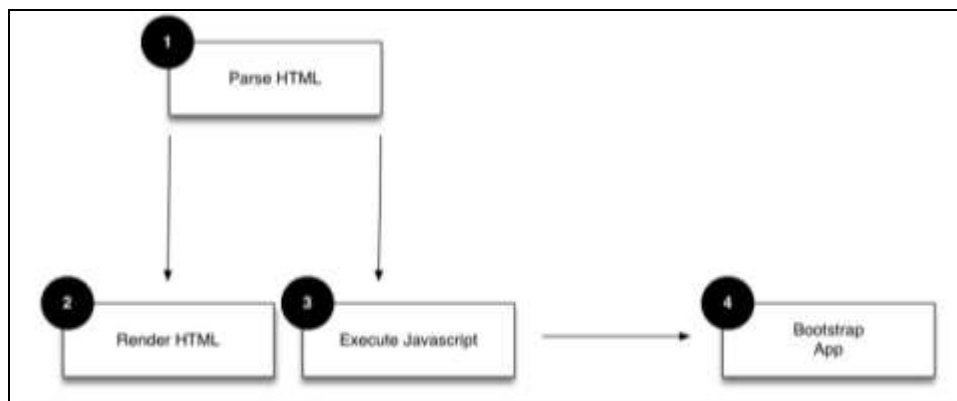


Рисунок 2.10 – Візуалізація браузера та завантажувальна стрічка



- браузер аналізує DOM, який він отримав від сервера;
- це призводить до відображення елемента HTML, або Запуск JavaScript;
- коли браузер досягає точки входу для програми, програма запускається сама;

На цьому етапі наш односторінковий потік програми знову запускається. Це найпростіша частина. Він обробляє події користувачів, здійснює виклики XHR та оновлює програму за потреби.

## 2.7 Управління розробкою дипломного проекту

Для управління розробки було використано програму програмне забезпечення Jira, яка дозволила використовувати багато різних функцій, включаючи розробку програмного забезпечення, гнучкий менеджмент проєктів, відстеження помилок, керування scrum методологією, управління вмістом, та багато іншого. Jira надала процес створення дошки Scrum, дозволяючи переносити кроки, створені на дошці kanban. Ці кроки допомогли визначити, де завдання знаходяться на стадії прогресу, що допомогло полегшити розробку програми зберегти час, структурувати задачі про пріоритети, зрозуміти з чого мочати і які модулі залежать від інших модулів (рис. 2.11).

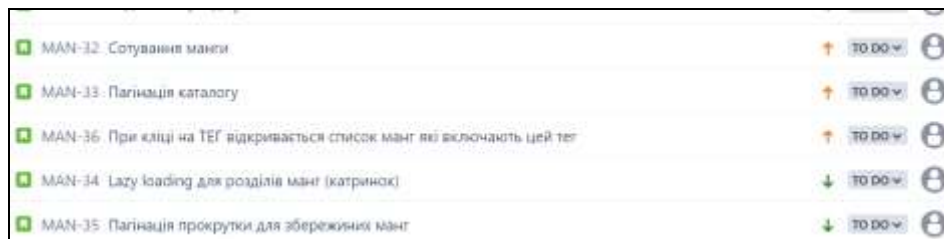


Рисунок 2.11 – Scrum backlog

Сучасні цифрові дошки Kanban допомогли автоматично збирати інформацію про час циклу завдань, час виконання та інші ключові показники ефективності.

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		40

Картки Канбана допомогла зобразити візуальне представлення завдань. Кожна картка містить інформацію про завдання та його статус (рис. 2.12) і опис. Кожен стовпець на дошці зображає різні етапи розробки робочого процесу. Картки проходять робочий процес до повного їх завершення.

Обмеження WIP дозволяє швидше закінчувати робочі завдання, допомагаючи зосередитись лише на поточних завданнях. Це запобігає накопиченню незавершених робіт, які в іншому випадку могли затопити процеси проекту з коміксами.

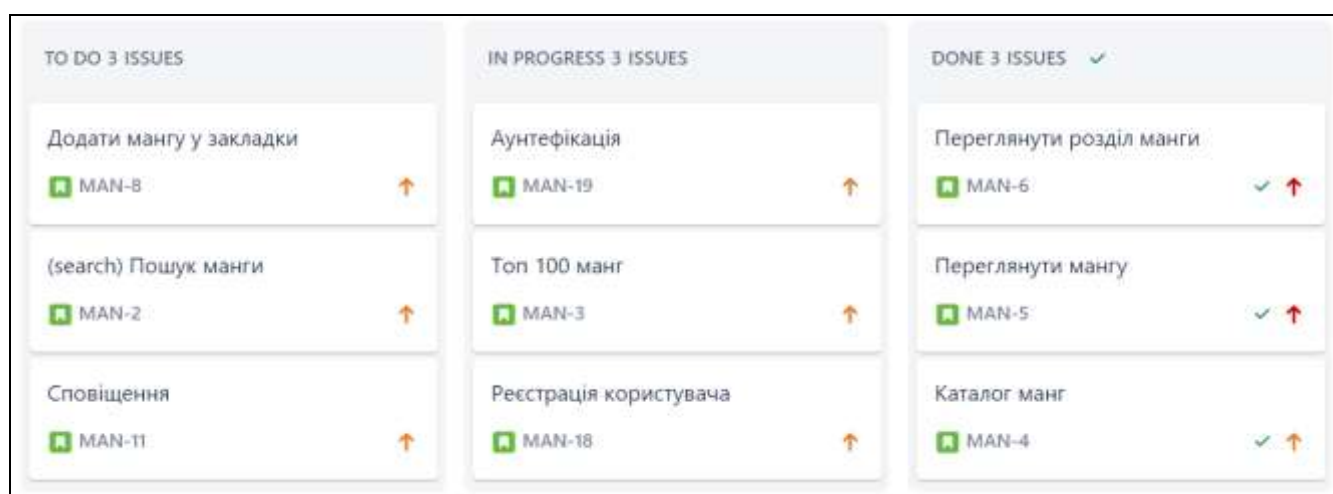


Рисунок 2.12 – Kanban дошка

В проекті потрібно було розроблено модель даних, на основі яких були створені таблиці та створені зв'язки між таблицями. В моделі даних були визначені та показані основні сутності, та ключові поля оскільки показує основну модель зв'язування таблиць та їх побудову.

Велика частина проекту спрямована на розробку і створення бази даних в рамках якої здійснюється обробка даних різної складності.

У якості СУБД використовується PostgreSQL, але через те, що проект працює через ORM, а саме TypeORM де може працювати в платформах NodeJS і в проекті були введені особливості такі, як:

- за допомогою кількох з'єднань з базою даних
- робота з декількома типами баз даних

- запити між базами даних та крос– схемами
- елегантний синтаксис, гнучкий та потужний QueryBuilder
- left та inner з'єднання
- правильна пагінація для запитів із використанням join
- кешування запитів

У проекті було реалізоване побудова таблиця використання, яка знаходиться у додатку та побудована у draw.io. На рисунку 2.13 зображена частина діаграми проекту.

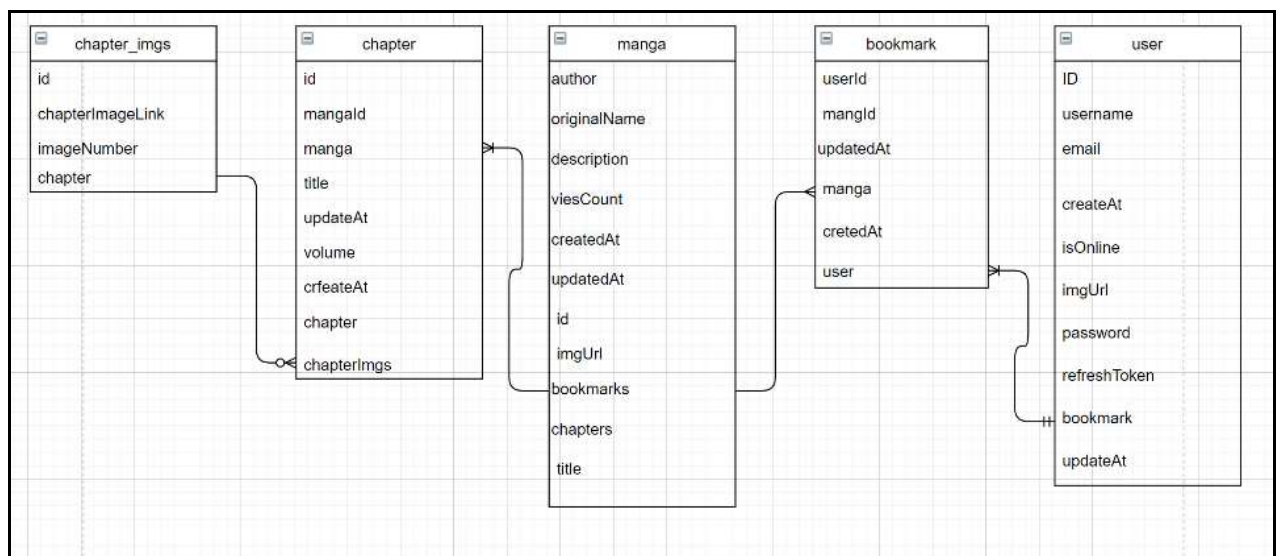


Рисунок 2.13 – Таблиці у проекті

## 2.8 Mind Map

Для представлення ідей та концепцій було використано карта вебсайту (Mind Map). Цей інструмент допоміг візуалізувати мислення, який структурувати інформацію, допомагаючи краще аналізувати, розуміти, синтезувати, згадувати та генерувати нові ідеї. На розумовій карті, на відміну від традиційних записок або лінійного тексту, інформація структурована таким чином, щоб набагато більше нагадувати те, як насправді має працювати вебдодаток коміксів. Оскільки це діяльність як аналітична, так і художня, вона

залучає ваш мозок значно, набагато багатше, допомагаючи у всіх його когнітивних функціях.

Як виглядає карта вебсайту (Mind Map) додатку можна подивитися на рисунку 2.8.

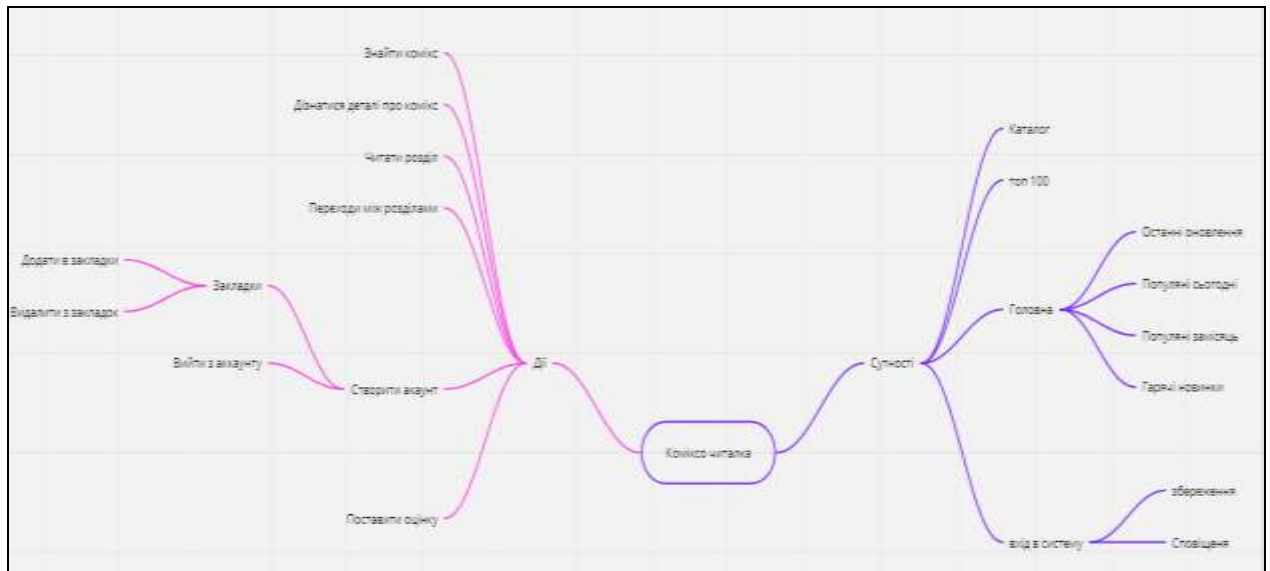


Рисунок 2.8 – Карта вебсайту (Mind Map)

Але для чого було використовувати карти розуму:

- конспектування
- ширше поглиблюватися в можливості
- вирішення проблеми
- вивчення та запам'ятовування
- планування
- дослідження та консолідація інформації з багатьох джерел
- представлення інформації
- поглиблення розуміння складних предметів

## 2.9 Journey map

Для розуміння розробки додатку, було розроблено CJM, це надало візуальне зображення досвіду роботи з додатками. Це дозволить визначити

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
						43
Змн.	Арк.	№ докум.	Підп.	Дата		

шлях, яким дотримується клієнт, коли він читає комікс, чи хоче знайти ресурс для читання.

Переваги використання Journey map:

- передбачення шляхів користувачів;
- розуміння проблем на яких спотикається користувач;
- розуміння що відчуває клієнт;
- поняття на яких проблемах може користувач спотикнутися;

Спочатку у клієнта виникає проблема, яку йому потрібно вирішити або бажання, яке він хоче задовольнити. Клієнт може зрозуміти сам, що у нього є проблема, і активно шукати її рішення. Кому і навіщо потрібна CJM.

Реалізація CJM базується на потребах аудиторії яка читає комікси та того, як вона споживає контент на ресурсах. Чим краще це вийде, тим вище будуть конверсії та попит на контент який може надати вебдодаток з коміксів. Без описаних CJM маркетинг буде хаотичним і некерованим, і чим складніше продукт, тим важче буде розуміння чого хоче аудиторія.

Побудовано карту шляху клієнта, коли вже був проведений весь аналіз, описані потреби, але через те, що проект є навчальним було проведено тільки поверхневий аналіз. Приклад CJM, створений для дипломного проекту з вебдодатку коміксів. На карті показані певні етапи взаємодії клієнта: пошук сайтів, оцінка всієї переглянутої манги, очікування на пошук манги, і подальшу взаємодію з ресурсом який був вибраним користувачем.

## 2.10 User stories

Так як user stories є частиною Agile підходу, який допоміг переключити фокус з написання вебдодатку. При розробці програмного забезпечення вебсайту манги та його управлінням, User stories дозволив написати неформальний природний опис декількох особливостей програмної системи. На табл. 2.1 зображено історії користувача.

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		44

## User stories для вебдодатку з коміксами

User stories	
<b>Value statement</b>	<b>Як адміністратор, я повинен мати доступ до додавання нових коміксів.</b>
<b>Acceptance criteria</b>	При додаванні коміксавони повинні відповідати таким умовам;; – У коміксаповинна бути оригінальна назва; – Комікс повинен бути з обкладенкою.
<b>Definition of Done</b>	Acceptance criteria задоволені Код– ревію здійснено Усі необхідні тести написано і пройдено (СІ білд пройдено) Крос– браузерне тестування здійснено Документацію оновлено
<b>Value statement</b>	<b>Як клієнт, я хочу мати доступ до пошуку манги, для того щоб знаходити цікаві мені речі.</b>
<b>Acceptance criteria</b>	Пошук має відбуватися у відповідному полі. Він буде здійснюватися за назвою
<b>Definition of Done</b>	Acceptance criteria задоволені Код– ревію здійснено Усі необхідні тести написано і пройдено (СІ білд пройдено) Крос– браузерне тестування здійснено Документацію оновлено
<b>Value statement</b>	<b>Як зареєстрований клієнт, я хочу мати доступ до функції збереження мангив закладках.</b>
<b>Acceptance criteria</b>	збереження буде відбуватися з бази даних після натиснення відповідної кнопки.
<b>Definition of Done</b>	Acceptance criteria задоволені Код– ревію здійснено Усі необхідні тести написано і пройдено (СІ білд пройдено) Крос– браузерне тестування здійснено Документацію оновлено
<b>Value statement</b>	<b>Як клієнт, я хочу мати можливість ставити оцінку коміксу.</b>
<b>Acceptance criteria</b>	Додавання оцінок буде здійснюватися після натиснення відповідної кнопки.
<b>Definition of Done</b>	Acceptance criteria задоволені Код– ревію здійснено Усі необхідні тести написано і пройдено (СІ білд пройдено) Крос– браузерне тестування здійснено Документацію оновлено
<b>Value statement</b>	<b>Як клієнт, я хочу мати доступ до навігаційної частини, для того щоб мені було зручно користуватися додатком.</b>
<b>Acceptance criteria</b>	Тут я зможу: – перейти до каталогу; – перейти на головну; – перейти на комікс; – перейти на розділ; – перескати між розділами;
<b>Definition of Done</b>	Acceptance criteria задоволені Код– ревію здійснено

Змн.	Арк.	№ докум.	Підп.	Дата

Усі Agile історії користувачів включають написане серію бесід про бажану функціональність. В user stories було описано тип користувача, що він хоче і чому. User stories допомогла створити спрощений опис вимоги. User stories були перенесені з backlog який був розроблений в Jira.

## **Висновки до розділу 2**

В розділі було розглянуто принцип роботи додатку з точки зору клієнт-серверної архітектури. Спроектовано серверну частину на основі трьох-шарової архітектури. Розглянуто взаємодію серверів вебдодатку та показано роботу на рівні ізоморфності. Показано підхід розробки та користувачську історію. Створено таблиці та структуру бази даних додатку.

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
						46
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		



## РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ІЗОМОРФНОГО ДОДАТКА

### 3.1 Розробка серверної частини додатку з вибору комікса

Розробка серверної частини є дуже важливою через те, що бізнес-логіка є первинна, а UI є вторинний, бо він в продукті є найбільш динамічним зі змін. В розробці серверної частини використовується мова Java Script яка запускається на платформі Node.js яка в собі містить chrome v8. Але в одному процесі може працювати тільки одна віртуальна машина, попри можливість використання декількох потоків.

У Chromium це обходиться мультипроцесовою архітектурою, підвищується також стабільність і безпека через реалізацію механізму «пісочниці».

В написанні серверної частини були використані два фреймворки Nest.js Next.js. Роль першого фреймворку в написанні бізнес-логіки роботи з базою даних та написання API, роль в іншого це зображення сторінок для користувача, та написання клієнтської частини. В проекті буде використовуватися мова програмування Type Script, щоб розширити можливості Java Script. Структура одного із компонентів серверної частини дивитися рисунок 3.1



Рисунок 3.1 – Структура серверного компонента»

					КР.ПЗ – 04.00.000 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підп.	Дата		

Оскільки проект опирається на принципах трьох шарової архітектури, тож кожна компонента приставляє з себе модуль в кому є під компонентні частини DAL, BLL, PL.

Розробка почалася з модулів ницого рівня, де потрібно реалізувати domain entity тобто сутності з якими буде взаємодіяти користувач. Кожна сутність виступає у вигляді класу. Кожне поле у нас застосовується шаблоном декоратором, який надає мета дані для полів. Кожний клас має бути експортований для використання. Кожне поле яке позначене окличним знаком не може бути пустим!

Частика коду бізнес сутностей:

```
@Entity('user')
export class User {
  @PrimaryGeneratedColumn()
  id: number;
  @Column({ unique: true })
  username: string;
  @Column()
  email: string;
  @Column()
  password: string;
  @Column({ default: 0 })
  gender: number;
}
```

Декоратори PrimaryGeneratedColumn, Column, Entity які імпортуються з бібліотеки TypeOrm, яка дозволить використовувати шаблон Repository, це надасть можливість використовувати його через dependency injection, для того, щоб робити запити до бази даних. В сервіс потрібно впровадити orm-entity для використання запитів до бази даних.

Впровадження Manga entity:

```
constructor(
  @InjectRepository(Manga) private readonly mangaRepository:
  Repository<Manga> ) {}
```

Функція для пошуку коміксу для не авторизованих користувачів:

- Метод приймає id комікса;

					<b>КР.ІІЗ – 04.00.000 ІЗ</b>	Арк.
						48
Змн.	Арк.	№ докум.	Підп.	Дата		

- асинхронна константа яка поченає стукатися до сховища, а саме до таблиці манг;
- методом об'єднання у вибірку входить розділ
- сортування по спадвнню

Пошук комікса для неавторизованого користувача :

```
public async findDetail(komiksId: string):Promise<Manga> {
const manga = await this.mangaRepository
    .createQueryBuilder("manga")
    .where({id: komiksId })
    .leftJoinAndSelect("manga.chapters", "chapter")
    .addOrderBy('chapter.chapter', 'DESC')
    .getOne();
manga.viewsCount++;
await this.mangaRepository.createQueryBuilder('komiks')
    .update(Manga)
    .set({viewsCount: komiks.viewsCount})
    .where('id =:id', {id: komiksId })
    .execute()
return manga
```

Після отримання манги йде стадія збільшення на одиницю кількості поля viewsCount тобто перегляди. Якщо операція пройшла успішного збільшення комікс повертається на рівень API.

Алгоритм роботи методу для пошуку манги для користувача який є авторизований:

- на вхід приходять ід користувача;
- пошук у таблиці user по полю id;
- отримання юзера відбувається пошук комікса по ід користувача та ід комікс;

Пошук авторизованого користувача:

```
public async findDetailAboutUserData(mangaId, userId): Promise<any | Manga>
{
    const user = await this.userRepository.createQueryBuilder()
    .where({ id: userId })
    .getOne();
    const manga = await this.findDetail(mangaId)
    const bookmark = await this.bookmarkRepository
    .findOne({ userId, mangaId })
    return {
```

					<b>КР.ІІЗ – 04.00.000 ІЗ</b>	Арк.
						49
Змн.	Арк.	№ докум.	Підп.	Дата		

```
...manga, subscribe: !!bookmark }
```

При успішному виконанні коду, результат відправляється на верхній шар PL та API. В клас NoveltyController потрібно впровадити в конструктор MangaService після чого ми можемо стукатися до методів класу.

Контролер манги з новинками:

```
@Controller('manga')
export class NoveltyController
{
  constructor(private readonly mangaService: MangaService) {
  }
}
```

Код який буде відпрацьовувати на localhost:3000/manga/{ mangaId}.

Тут використовується спеціальний метод OptionalJwtAuthGuard суть якого перевірити чи є в cookies токен доступу, сам метод findMangaById приймає в собі param який є частиною url і поле req який є сам запит.

З поля запиту витягується id, після відповідної перевірки на те чи користувач авторизований викликається один з двох запитів, findDetailWithUserData з mangaService якщо перевірка не пройшла тоді findDetail.

Метод контролера для пошуку конкретної манги:

```
@Get('/:id')
@UseGuards(OptionalJwtAuthGuard)
findMangaById(@Param() param,@Request() req):Promise<any>
{
  const userId = req.user.user?.id
  if (userId)
  {
    return this.mangaService.findDetailWithUserData(param.id, userId)
  }
  return this.mangaService.findDetail(param.id)
}
```

Робота з сервером для візуалізації HTML сторінки відбувається завдяки ізоморфним методам життєвого циклу із Next.js. Один із них є

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
						50
Змн.	Арк.	№ докум.	Підп.	Дата		

“getServerSideProps” який дозволяє отримати данні через API. На основі цих даних зробити візуалізацію.

Метод також виступає у ролі контролера даних на проксі сервері.

Приклад використання ізоморфного методу:

```
export const getServerSideProps: GetServerSideProps = async (context) =>
{
  const id:string | string[] = context.params.mid
}
```

В момент виклику методу він отримує поле context з якого отримує доступ до params. У параметрі поле по якому ми можемо вилучити те що нам потрібно з URL, у випадку з мангою витягується mid (manga id) тому, що в Next.js маршрутизація працює на рівні файлової системи, тому і назва файлу не повинна семантично показувати зміст mid.tsx.

При виклику методу getOne класу MangaService який передає id, і в залежності від того було щось знайдено повертається результати.

Робота методу з API:

```
export const getServerSideProps: GetServerSideProps = async (context) =>
{
  const id: string | string[] = context.params.cid
  const payload = await MainMangaService.getChapterById(id)
  if (!payload) {
    return {
      notFound: true,
    }
  }
  return {
    props: {
      chapter: payload
    },
  }
}
```

Також в класі MainMangaService метод getOne містить в собі інтерцептор, який є спеціальною функцією, яка вставляє в заголовки поле Authorization з токеном доступу користувача. Інтерцептор впроваджується в програму через бібліотеку axios.

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		51

Axios надає обгортку над xmlhttprequest яка надає великі можливості для роботи та взаємодії з серверними API, через те, що на платформі node.js метод fetch ще не підтримується, крім того, він ще й повільніший.

Приклад роботи метода з запитом до основного сервера:

```
getOne = async (id:string|string[]) => {
  return authInterceptor.get(` ${SERVER_BASE_URL}manga/${id}` )
  .then(({data}) => {
    return data
  })
}
```

Функції які будуть повторюватися кілька разів:

- useState() – повертає кортеж. Першим параметром у масиві є значення поточного стану. Другим параметром є функція, яка дозволить оновити значення стану.

- useEffect – функція, яка приймає два аргументи. Перший аргумент, який передається useEffect – це функція, називається effect тому що працює з нативними методами javascript, а другий аргумент (який не є обов'язковим) являє собою масив залежностей, який якраз і говорить якщо в масиві буде залежність то useEffect буде викликатися тоді коли змінюється залежність, якщо масив пустий то useEffect буде відпрацьовувати що разу при будь-яких змінах.

- useRef() – функція яка приймає один аргумент як початкове значення і повертає посилання. Посилання, що має особливу властивість current, це може використати як getElementById щоб збергти силку на елемент.

- useContext – дозволить передавати дані у всій програмі, не передаючи вручну параметри по дереву компонентів. Він є частиною контекстного API React (інші частини – це компоненти Provider та Consumer).

Алгоритм роботи після отримання даних з ізоморфного методу:

- перевірка чи в параметрі props не undefined,
- якщо перевірка пройшла успішно з props через деструктуризацію витягується поле payload;

					<b>КР.ІІЗ – 04.00.000 ІЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		52

– передача в компоненту MangaDetail, показ сторінки mangaId:

```
const mid = (props) =>
{
  if (!props)
  {
    return <div>Loading</div>
  }

  const {setCurrentManga} = useCurrentManga()
  setCurrentManga(props.payload)
  return (
    <div>
      <MangaDetail {...props.payload}/>
    </div>
  )
}
```

Для стилізації компонентів був використаний спосіб CSS-in-JS та бібліотеку Material-ui, в якому стилі описуються у вигляді JavaScript об'єктів.

Поля в об'єкті виступають як селектори CSS, а значення об'єкта є значеннями селектору. Стилізація CSS-in-JS:

```
const useStyles = makeStyles((theme: Theme) =>
createStyles({
  root: {
    position: 'relative'
  },
  paper: {
    padding: theme.spacing(2),
    textAlign: 'center',
    color: theme.palette.text.secondary,
  },
}
```

Перед тим як зображати комікс та всю інформацію про нього спочатку треба перевірити чи користувач підписаний на оновлення коміксів.

Тому в React потрібно оголосити стан і в методі життєвого циклу у момент візуалізації перевірити відправити запит до API та виконати перевірку є.

Стан підписок на закладки:

```
const [bookmark, setBookmark] = useState<boolean>(false)
useEffect(()=>{
  if(props.subscribe){
    setBookmark(true)
  }
  console.log(props.subscribe)
```

					<b>КР.ІІЗ – 04.00.000 ІЗ</b>	Арк.
						53
Змн.	Арк.	№ докум.	Підп.	Дата		



```
},[])
```

Для центрування та надання певної ширини використовуються компонента `Container`. Для розташування елементів на сторінці використовується технологія `Grid`. Вивід тегових елементів відбувається завдяки методу `map`.

Вивід фотографії з коміксом, та теги:

```
<Container className={classes.root}>
  <Grid container spacing={6}>
    <Grid item sm={3}>
      <CardMedia
        className={classes.media}
        image={imageUrl}
        title="Contemplative Reptile"
      />
      <Box className={classes.tagWrapper}>
        {tags.map((tag: string, index: number) =>
          {
```

Для того, щоб показати авторизованому користувачу елемент кнопки «додати в закладки» використовується тернарний оператор, який є аналогом оператора «if», перевірка чи знаходиться щось в заданому полі.

Вивід кнопки в закладки:

```
{ subscribe
  ? <Button onClick={subscribeToManga} >
  : <BookmarksIcon style={subscribe ? {color: '#4e6baf'}: {color: 'black'}} />
  </Button>
}
```

В розташуванні елементів була використана технологія `Grid`. Вивід інформації про комікс:

```
<Typography component='span' className={classes.title}>
  {title + '/' + englishTitle + '/' + originalTitle}
</Typography>
<MangaPopUp icon='GradeIcon'>
<Rating />
```

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
						54
Змн.	Арк.	№ докум.	Підп.	Дата		

Результат виконаного коду, тобто елемент який містить фото, заголовок та рейтинг, можна побачити на рисунку (рис. 3.2).

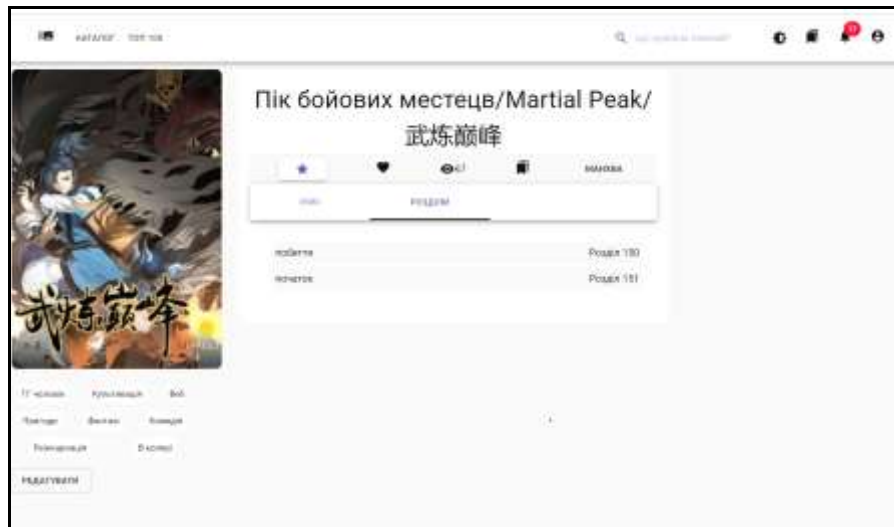


Рисунок 3.2 – Сторінка вибраного комікса

### 3.2 Розробка інтерактивних речей.

Для завантаження зображень була використана бібліотека `multer`, яка дозволяє з буфера витягнути файл.

Алгоритм роботи збереженню файлу:

- створення пустого масиву для фотографій;
- змінна для генерації унікального ім'я;
- змінна з маршрутом де файл повинен зберігатися на диску;
- рекурсивна функція яка шукає директорію;
- якщо директорія буде знайдена, то файл буде записаний туди;
- якщо директорії немає, створюється директорія;
- записується файл;
- на рівень вище повертається об'єкта зі шляхом до манги та його індекс;

Частина коду по завантаженню файлу на диск сервера:

					КР.ПЗ – 04.00.000 ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підп.	Дата		

```

fileName = mangaName + '-' + chapterNumber + uuid.v4() + '.' + fileExtension
const filePath = path.resolve(__dirname, '..', '..', 'static', mangaName chapterNumber)
fs.access(filePath, function (err) {
  if (err && err.code === 'ENOENT') {
    fs.mkdir(filePath, {recursive: true}, (err) => {
      if (err) throw err;
      fs.writeFile(path.resolve(filePath, fileName), file.buffer, (err) => {
        if (err) {

```

Одним із найвищих покладених завдань було – розробити можливість отримувати конкретний розділ з даними на попередній розділ та на наступний.

Алгоритм роботи:

- отримання по Id розділу;
- пошук коміксах якому належить розділ;
- з масиву манг витягується розділи;
- пошук розділа в масиві коміксами по Id;
- пошук індексу розділ в масиві манг;
- створення об’єкту з наступним та попереднім розділом по індексу;
- якщо індекс рівний нулю, тоді посилання на наступний розділ буде пустим;
- якщо індекс рівний «- 1» тоді посилання на попередній не буде.

Кодова реалізація:

```

async findChapterById(chapterId) {
const chapter = await this.chapterRepository.createQueryBuilder('chapter')
  .where({id:chapterId})
  .leftJoinAndSelect('chapter.chapterImgs', 'chapter_imgs') .getOne();
const manga = await this.mangaRepository.createQueryBuilder("manga")
  .where({id: chapter.mangaId}) .leftJoinAndSelect("manga.chapters",
  "chapter").addOrderBy('chapter.chapter', 'DESC').getOne()
if (chapter.id === chapterFromManga.id ){
  return true } })
(findIdx >= 1 && findIdx < listChapter.length - 1){

```

Реалізація клієнтської частини відбувається за таким алгоритмом:

- Зчитується id розділу який потрібно знайти;
- Відправляється запит до API;
- Перевіряється чи є розділ;

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		56

- При умові якщо є розділ перевіряється чи є посилання на наступний
- Показ кнопки вперед;
- При умові якщо є розділ перевіряється чи є посилання на попередній;
- Показ кнопки Назад;
- Маппінг масиву фотографій;

Результатом виконаного коду дивитися на рисунку 3.3



Рисунок 3.3 – Відображення розділу та перемикання між ними

Для завантаження манги на сервер було використано вікно де модератор, чи адміністратор може додати всю потрібну інформацію. В кодї був реалізовані методи відкрити та закрити. Було оголошено стани з допомогою useStata із бібліотеки React, були оголошені обробники подій які в собі використовували спеціальний оптимізаційний хук useCallback який допоможе нам усунути зайві рейдери в virtual DOM.

Для завантаження фотографії було використано тег input. На тег input було поставлено хук useRef. Це робилося задля того, щоб сховати input, тому, що стилізація самого input є досить складною, а залишати стандартний стиль було не прийнято для користувачів, тому було використано тег div, щоб при натисканні на div відкривалася файлова система де можна було завантажити фотографі. Для посилання форми був використаний метод handleClose в якому формувався спеціальний об'єкт FormData, щоб відправляти файли на сервер, де він міг це вчитати з буфера.

Стан та обробник для вікна:

```
const [open, setOpen] = React.useState(false);
const handleClickOpen = () => {setOpen(true);
const onCancel = () => { setOpen(false)
```

Частина станів для поля, так функції обробники щоб цей стан змінювати:

```
const [imgUrl, setImg] = useState<any>()
const [imgForServer, setImgForServer] = useState<any>()
const [title, setTitle] = useState<string>("")
const [englishName, setEnglishName] = useState<string>("")
```

Метод `coverHandler` служить для читання сайту.

Алгоритм роботи:

- в константу поміщається об'єкт `FileReader`;
- в сеттер хуку стану поміщається посилання на файл;
- в констату змінну вставляється сам файл;
- виклик в створеному об'єкті `FileReader` методу `onloadend`, в який поміститься сам файл і де його можна буде прочитати через спеціальне посилання;

```
function coverHandler(e:ChangeEvent<HTMLInputElement>) {
const reader = new FileReader();
setImgForServer(e.target.files[0]
const file = e.target.files[0]
reader.onloadend = () =>{
setImg(reader.result
} reader.readAsDataURL(file }
```

Метод завантаження файлу на сервер. Алгоритм роботи:

- відбувається перевірка на те що поля є не пустими;
- виклик `http` методу для створення манги;
- коли відповідь прийде успішною виконається редірект;

Реалізація методу завантаження:

```
const handleClose = () => {
if (imgUrl &&title&&englishTitle&&originalTitle && description &&year&&author){
```

					<b>КР.ІІЗ – 04.00.000 ІЗ</b>	Арк.
						58
Змн.	Арк.	№ докум.	Підп.	Дата		

```
axios.post(`${SERVER_BASE_URL}manga`,formData).then(res=>console.log(res)).catch(e=>console.log(e))
setOpen(false);
```

Результато написаного коду можна подивитися на рисунку 3.4

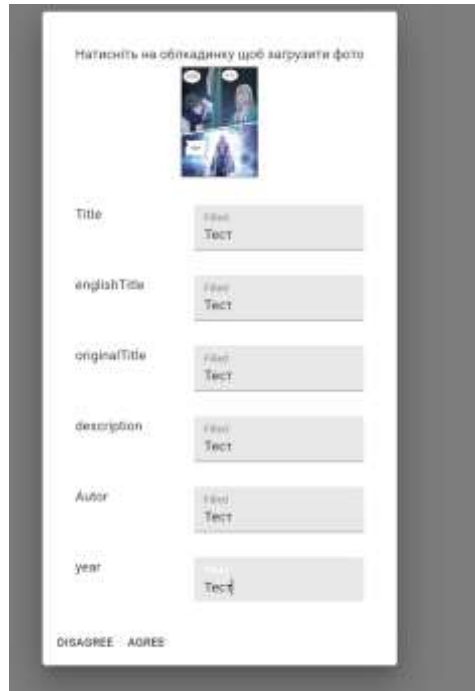


Рисунок 3.4 – Результат роботи коду

### Висновок до розділу 3

Було розроблено ізоморфний веб додаток для перегляду та публікацій комксів. Розроблено серверну частину вебдодатку. Імплементований та показаний принцип виклику ізоморфних методів. Проведено аналіз алгоритму роботи інтерактивних речей. Базуючись на проведеному аналізі аналогів був побудований клієнтській інтерфейс.

## РОЗДІЛ 4. ОХОРОНА ПРАЦІ

### 4.1 Умови роботи на робочому місці з комп'ютером

Програмування з використанням комп'ютера, відбувається в середовищі, яке в певній мірі впливає на функціональний стан програмістів, які перебувають у комп'ютерному класі. Найважливішими несприятливими факторами середовища при роботі за комп'ютером є:

- фізичні параметри мікроклімату, освітлення електромагнітні випромінювання різних частотних діапазонів; виробничий шум та вібрація; іонізація повітря; статична електрика;
- психофізіологічні перенапруження зорового аналізатора, недостатня рухома діяльність, нервово–емоційне напруження;
- аналіз умов праці виконується для приміщення площею 50м<sup>2</sup> яке знаходиться, на 3 поверсі п'ятиповерхової будівлі.

Відповідно до ДСТУ 0.00-1.28-10 є неприпустимим розташування приміщень, призначених для роботи з ВДТ у підвалах та цокольних поверхах. Також забороняється розташування вибухонебезпечних приміщень категорії А і Б (ОНТП 24-86) та виробництв з мокрими технологічними процесами поряд з приміщенням, де розташовуються ЕОМ (ПЕОМ), а також над такими приміщеннями, або під ними. Окрім того, виробничі приміщення для роботи з ВДТ не повинні межувати з приміщеннями, у яких рівень шуму та вібрації перевищує допустимі значення.

У процесі роботи з комп'ютером необхідно дотримувати правильний режим праці і відпочинку. У іншому випадку в працюючого за ПК відзначаються значна напруга зорового апарата з появою скарг на незадоволеність роботою, дратівливість, порушення сну, утома і хворобливі відчуття в очах, у попереку, в області шиї і руках та головні болі.

Розташування робочого місця особи, що працює з ПК, в даному випадку

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
						60
Змн.	Арк.	№ докум.	Підп.	Дата		



студента, повинне відповідати вимогам ДСТУ 22.269-79 «Робоче місце оператора. Взаємне розташування елементів робочого місця».

Облаштування робочого місці повинно забезпечувати:

- правильне розміщення робочого місця у виробничому приміщенні;
- належні умови освітлення приміщення і робочого місця, відсутність відблисків;
- належні ергономічні характеристики основних елементів робочого місця;
- характер та особливості трудової діяльності.

Для зменшення втоми, місця користувачів ЕОМ мають бути зручними. Конструкція робочого місця користувача ПК, (при роботі сидячи) має забезпечувати підтримання оптимальної робочої пози з такими ергономічними характеристиками:

- ступні ніг на підлозі або на підставці для ніг;
- стегна в горизонтальній площині;
- передпліччя вертикально;
- лікті під кутом 70-90 до вертикальної площини;
- зап'ястя зігнуті під кутом не більше 20 відносно горизонтальної площини;
- нахил голови 15-20 відносно вертикальної площини.

Робоче місце користувача ПК, обладнується робочим столом, стільцем і підставкою для ніг. Висота робочого стола має бути в межах від 0,65 до 0,8 м, а ширина повинна забезпечувати можливість виконання операцій в зоні досяжності моторного поля.

Клавіатуру слід розташовувати на поверхні столу на відстані 200 мм від краю, звернутого до працюючого. У конструкції клавіатури має передбачатися опорний пристрій (виготовлений із матеріалу з високим коефіцієнтом тертя, що перешкоджає його переміщенню), який дає змогу змінювати кут нахилу поверхні клавіатури у межах 5...15.

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		61

## 4.2 Забезпечення правильного мікроклімату у приміщенні

Як один з факторів виробничого середовища, мікроклімат впливає на теплообмін організму людини з цим середовищем. Необхідною умовою життєдіяльності людини є терморегуляція, тобто здатність організму регулювати віддачу тепла в оточуюче середовище. Цей процес визначається параметрами мікроклімату. Метеорологічні умови визначаються такими параметрами:

- температурою повітря в приміщенні, С;
- відносною вологістю повітря, %;
- рухливістю повітря, м/с;
- тепловим випромінюванням, Вт/м<sup>3</sup>.

Принцип нормування мікроклімату створення оптимальних умов для теплообміну тіла людини з навколишнім середовищем.

Параметри мікроклімату, вміст шкідливих речовин на робочих місцях, оснащених моніторами, відповідають вимогам ДСН 3.3.6.042-99 «Державні санітарні норми мікроклімату виробничих приміщень», ДСТУ 12.1.005-88 «ССБТ. Повітря робочої зони. Загальні санітарно– гігієнічні вимоги до повітря робочої зони».

Обчислювальна техніка є джерелом істотних тепловиділень, що може привести до підвищення температури і зниження відносної вологості в приміщенні.

В приміщеннях, де встановлені комп'ютери, повинні дотримуватися оптимальні параметри мікроклімату, які визначають комфортні умови. Ці параметри залежать від періоду року, категорії робіт за важкістю, і від теплових характеристик виробничого приміщення, (табл. 4.1).

Робота за комп'ютером характеризується малими фізичним навантаженнями, цей вид діяльності належить до категорії легких робіт за критерієм загальних енерговитрат організму (ДСН 3.3.6.042-99).

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		62

Таблиця 4.1

## Параметри мікроклімату для приміщень

Період року	Категорія робіт	Параметр мікроклімату	Величина
Холодний	Роботи легкі I а	Температура повітря в приміщенні	22÷24°C
		Відносна вологість	40÷60%
		Швидкість руху повітря	до 0,1м/с
Теплий		Температура повітря в приміщенні	23÷25°C
		Відносна вологість	40÷60%
		Швидкість руху повітря	0,1.0,2м/с

Системи опалення та кондиціонування повітря в приміщеннях з ПК повинні відповідати СНиП 2.04.05–91 «Опалення, вентиляція і кондиціонування». Під час роботи комп'ютерної техніки в повітряному середовищі відбувається суттєва трансформація іонного складу, істотно знижується концентрація легких, середніх та важких негативно зарядних частинок. Така зміна балансу іонного складу призводить до негативного впливу на здоров'я працюючих.

Рівні іонізації повітря приміщень при роботі на персональних комп'ютерах визначені в (таблиці 4.2), відповідно до НПАОП 0.03-3.06-80.

Таблиця 4.2

## Рівні іонізації повітря в приміщенні з ПК

Рівні	Кількість іонів в 1 см <sup>3</sup> повітря	
	n+	n-
Мінімально необхідні	400	600
Оптимальні	1500– 3000	3000– 5000
Максимально допустимі	50000	50000

Для підтримки оптимальних значень мікроклімату та підтримання нормальної концентрації позитивних та негативних іонів в приміщенні аудиторії пропонується удосконалити системи опалення, природної вентиляції та встановити кондиціонер.

Дисплеї на основі ЕПТ є потенційним джерелом випромінювання кількох діапазонів електромагнітного спектра: рентгенівського, оптичного, радіочастотного.

Кожний вид випромінювання відрізняється своїми особливими характеристиками впливу на організм людини.

Рентгенівське випромінювання. Дослідження показують, що відео термінал не несе небезпеки для користувача ПЕОМ, оскільки інтенсивність такого випромінювання нижча за гранично допустимі норми (ГДН).

Відповідно до «Норм радіаційної безпеки України» (НРБУ-97), гранично допустима потужність експозиційної дози рентгенівського випромінювання на відстані 5 см від поверхні екрана відео терміналу становить  $7,74 \cdot 10^{-12}$  Кл/кг, що відповідає еквівалентній дозі 0,1 мбер/год. (100 мкР/год.).

Оптичне випромінювання. Оптичні види випромінювання виникають завдяки взаємодії електронів з шаром люмінофору, нанесеного на екран ВДТ. Область оптичного випромінювання включає ультрафіолетове (УФ), світлове та інфрачервоне (ІЧ) випромінювання.

УФ випромінювання впливає на шкіру та очі людини. Такий вплив на шкірі проявляється досить швидко, а для очей характерним є період прихованої дії. Рівень УФ випромінювання, який був виявлений, досить низький і становить 1 середньому  $0,001 \text{ Вт/м}^2$ .

Світлове випромінювання впливає в основному на око і призводить до втоми очей, запалення райдужної оболонки. Однак ці симптоми швидко минають і не викликають патологічних змін.

ІЧ випромінювання довжина хвиль обмежена від 0,76мм до 1мм. Для більшості біологічних матеріалів випромінювання цього діапазону вважаються непрозорими. Інтенсивність інфрачервоних випромінювана нижча за значення,

					<b>КР.ІІЗ – 04.00.000 ІІЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		64

передбачені ДсанПіН 3.3.2. 007-98.

Електромагнітні випромінювання (ЕМВ) радіочастотного діапазону. Джерелом ЕМВ є відео термінал. Тому, обираючи робоче місце для комп'ютера, необхідно пам'ятати, що його задня і бокові стінки можуть бути джерелом значно більшого ЕМВ, ніж екран.

З метою профілактики несприятливого впливу електромагнітного випромінювання від ВДТ на користувача необхідно:

- встановити на робочому місці відео термінал, що відповідає сучасним вимогам стосовно захисту від випромінювані;
- не переобтяжувати приміщення значною кількістю робочих місць з ВДТ;
- не концентрувати на робочому місці великої кількості радіо електронних пристроїв. Гранично допустимі рівні випромінювань на робочих місцях з ПК наведено в (таблиці 4.3).

Такий вплив на шкірі проявляється досить швидко.

Таблиця 4.3

Гранично допустимі рівні випромінювань

Вид випромінювання	Діапазон хвиль (частот)	Гранично допустимий рівень
Іонізуюче випромінювання		
М'яке рентгенівське випромінювання ( на відстані 0,05 м від екрана та корпусу ВДТ)	0,01 – 1 нм	100мкР/год
Оптичні випромінювання		
Ультрафіолетове випромінювання	315 – 400 нм	10 Вт/м <sup>2</sup>
Видиме випромінювання (яскравість)	400 – 700 нм	1000 кд/м <sup>2</sup>
Інфрачервоне випромінювання	700 нм – 1мм	100 Вт/м <sup>2</sup>

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		65

Продовження таблиці 4.3

Електромагнітні випромінювання (поля радіочастотного та низькочастотного діапазонів)		
Напруженість електромагнітного поля на відстані 0,5 м навколо монітору за електричною складовою	2 кГц – 400 кГц 5 Гц – 2 кГц	2,5 В/м 25 В/м
Щільність магнітного потоку	2 кГц – 400 кГц 5 Гц – 2 кГц	25 нТл 250 нТл
Електростатичні поля		
Поверхневий електростатичний потенціал	–	500 В

Відомо, що шум несприятливо діє на слуховий аналізатор та інші органи та системи організму людини. Визначальне значення щодо такої дії має інтенсивність шуму, його частотний склад, тривалість щоденного впливу, індивідуальні особливості людини, а також специфіка виробничої діяльності. Ті види діяльності, у яких поєднується напружена розумова робота та інтенсивне використання комп'ютера (редагування тексту, верстка оригіналу, "запуск" та від лагодження програм тощо) характеризується відчутним впливом навіть незначних рівнів шуму. Цей вплив виражається у зниженні розумової працездатності, швидкій втомлюваності, послабленні уваги, появі головного болю та ін.

Рівні звукового тиску в октавних смугах частот, рівні звуку та еквівалентні рівні звуку на робочих місцях, обладнаних ВДТ і ПК визначені ДСанПіН 3.3.2-007-98 не повинні перевищувати 65 дБА. Для приміщення аудиторії, рівень шуму не перевищує 60 дБА.

Для забезпечення нормованих рівнів шуму у виробничих приміщеннях та на робочих місцях застосовуються шумопоглинальні засоби, вибір яких обґрунтовується спеціальними інженерно-акустичними розрахунками.

					<b>КР.ІІЗ – 04.00.000 ІЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		66

Близько 90% всієї інформації, що отримується людиною, приходится на органи зору. Організація освітленості робочих місць грає велику роль у житті людини. Недостатнє та нераціональне освітлення веде до стомлення очей, розладу центральної нервової системи, зниженню розумової та фізичної працездатності, а у ряді випадків може бути причиною травматизму (близько 5% травм приходится на частку нераціонального та недостатнього освітлення).

Щоб уникнути перевтоми, а також для профілактики професійних захворювань та виробничого травматизму потрібно дотримуватись наступних вимог:

- створювати на робочій поверхні освітленість, що відповідає характеру зорової роботи і не є нижчою за встановлені норми;
- забезпечити достатню рівномірність та постійність рівня освітленості у виробничих приміщеннях, щоб уникнути частоті преадаптації органів зору;
- не створювати засліплюваної дії як від самих джерел освітлення, так і від інших предметів, що знаходяться в полі зору;
- не створювати на робочій поверхні різких та глибоких тіней (особливо рухомих);
- повинен бути достатній для розрізнення деталей контраст поверхонь, що освітлюються;
- не створювати небезпечних та шкідливих виробничих чинників (шум, теплові випромінювання, небезпека ураження струмом, пожежо та вибухонебезпека світильників);
- повинно бути надійним і простим в експлуатації, економічним та естетичним .

Приміщення аудиторії №330 має природну та штучну системи освітлення, які повинні відповідати вимогам ДБН В.2.5-28-2996 «Інженерне обладнання будинків та споруд. Природне та штучне освітлення».

Штучне освітлення в приміщенні з комп'ютеризованим робочим місцем здійснюється системою загального освітлення.

					<b>КР.ІІЗ – 04.00.000 ІЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		67



Як джерела світла в разі штучного освітлення застосовуються світильники серії LED-353. Світильники укомплектовані високочастотними пускорегулювальними апаратами (ВЧ ПРА). Яскравість світильників загального освітлення в зоні кутів випромінювання від 50 до 90 град. з вертикаллю в повздовжній та поперечній площинах становить не більше 200 кд/м, захисний кут світильників не менше ніж 40 град. Рівень освітленості у аудиторії,  $E = 300$ лк.

Електробезпека система організаційних та технічних заходів і засобів, що забезпечують захист людей від шкідливого та небезпечного впливу електричного струму, електричної дуги, електромагнітного поля і статичної електрики.

Приміщення аудиторії де знаходяться комп'ютери відноситься до приміщень з підвищеною безпекою.

ЕОМ, периферійні пристрої ЕОМ та устаткування для обслуговування, ремонту та налагодження ЕОМ, електропроводи та кабелі за виконанням та ступенем захисту мають відповідати класу зони за ПВЕ, мати апаратуру захисту від струму короткого замикання та інших аварійних режимів.

Використання нульового робочого провідника як нульового захисного провідника забороняється.

Нульовий захисний провід прокладається від склейки групового розподільчого щита до розеток живлення.

Не допускається підключення на щиті до одного контактного затискача нульового робочого та нульового захисного провідників.

Площа перерізу нульового робочого та нульового захисного провідника в груповій три провідній мережі повинна бути не менше площі перерізу фазового провідника. Усі провідники повинні відповідати номінальним параметрам мережі та навантаження, умовам навколишнього середовища, умовам розподілу провідників, температурному режиму та типам апаратури захисту, вимогам ПВЕ.

З метою підвищення рівня електробезпеки в приміщенні аудиторії де

					<b>КР.ІІЗ – 04.00.000 ІЗ</b>	Арк.
						68
Змн.	Арк.	№ докум.	Підп.	Дата		

знаходяться комп'ютери не припустимим є:

- експлуатація кабелів та проводів з пошкодженою або такою, що втратила захисні властивості за час експлуатації, ізоляцією;
- залишення під напругою неізольованих кабелів та проводів;
- застосування саморобних подовжувачів, які не відповідають вимогам ПВЕ до переносних електропроводок;
- користування пошкодженими розетками, розгалужувальними та з'єднувальними коробками, вимикачами та іншими електровиробами;
- застосування для опалення приміщення нестандартного (саморобного) електронагрівального обладнання.

ВДТ є джерелом електростатичних зарядів. Тривале перебування в електростатичному полі, створеному цими зарядами, негативно впливає на здоров'я працюючих: бронхо– легеневі захворювання, порушення серцево-судинної та нервової систем, ураження шкіри тощо .

Напруженість електростатичного поля на робочих місцях, в тому числі й з ВДТ, не повинна перевищувати 20 кВ/м відповідно до ДСТУ 12.1.045-84 «ССБТ. Електростатичні поля. Допустимі рівні на робочих місцях і вимоги до проведення контролю», поверхнево електростатичній Потенціал відео термінала не винних перевищувати 500В.

Для запобігання створенню значної напруженості поля та захисту від статичної електрики необхідно:

- встановити нейтралізатори статичної електрики;
- підтримувати в приміщенні з ВДТ відносну вологість повітря не нижче 45-50% (чим сухіше повітря тим більше електростатичних зарядів), можна для цього використати навіть побутові зволожувачі;
- застелити підлогу в приміщеннях з ВДТ антистатичним лінолеумом і проводити щоденне вологе прибирання;
- складати всі полімерні покриття (чохли) ВДТ у найбільш віддаленому від користувачів місці розміщення;

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		69

– протирати екран та робоче місце спеціальною антистатичною серветкою або зволоженою тканиною.

### 4.3 Вимоги до безпечної експлуатації комп'ютерної техніки

Пожежна безпека при роботі за комп'ютером повинна забезпечуватись у відповідності з вимогами Закону України «Про пожежну безпеку» та НПАОП 0.01-1.01-95 «Правил пожежної безпеки в Україні» та інших нормативних документів.

Відповідно до ОНТП 24-86 «Визначення категорій приміщень і будинків по вибух пожежної і пожежної небезпеки» приміщення аудиторії №330 відноситься до категорії Д.

Найбільш імовірною причиною виникнення пожеж в приміщенні є порушення вимог при експлуатації комп'ютерної техніки та займання електропроводки внаслідок коротких замикань.

Для того щоб уникнути виникнення пожежі, потрібно дотримуватися наступних заходів:

- дотримання правил пожежної безпеки при роботі з комп'ютером,
- електрообладнанням та освітлювальними приладами;
- періодичний контроль цілісності і надійності електроізоляції;
- наявність інструкцій з пожежної безпеки;
- навчання, атестація і переатестація персоналу з пожежної безпеки;
- наявність системи захисту від атмосферної електрики;
- періодичне зняття зарядів статичної електрики;
- заборона куріння в приміщенні;
- застосування будівельних конструкцій із ступенем вогнестійкості не нижче ІІ, а також використання важко горючих або негорючих матеріалів в інтер'єрі виробничого приміщення;
- наявність схеми евакуації;

					<b>КР.ІІЗ – 04.00.000 ІІЗ</b>	Арк.
						70
Змн.	Арк.	№ докум.	Підп.	Дата		

- наявність пристроїв автоматичного вимкнення ПЕОМ та іншого електроустаткування на випадок пожежі;
- наявність первинних засобів пожежогасіння (вогнегасник ВПК-5-2 шт).

#### **Висновки до розділу 4**

Було описано охорону праці, як потрібно себе поводити під час роботи з комп'ютерами. Щоб забезпечити нормальні умови праці на робочому місці з комп'ютерною технікою. Також ми описали що потрібно забезпечення правильного мікроклімату у приміщенні. Не мало важливо є вимоги до безпечної експлуатації комп'ютерної техніки. Будучи на роботі ми повинні виконувати вимоги, для власного забезпечення.

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		71

## ВИСНОВКИ

Результатом виконаної кваліфікаційної роботи:

- розроблено вебдодаток для перегляду та публікацій коміксів;
- проаналізовано ряд аналогів, та виокремлені їх сильні та слабкі сторони;
- спроектовано та розроблено базу даних основувшись на реаліційній системі управління базою даних PostgreSQL;
- створено користувацькі історії
- були імплементовані шаблони проектування GoF, а саме:
  - Singleton
  - Repository
  - Factory
  - Builder
  - Observable
- підчас розробки було використано гібридну систему рендеренгу
- була створена карта вебдодатку;

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк.
						72
Змн.	Арк.	№ докум.	Підп.	Дата		

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. React *uk.reactjs.org*: вебсайт URL: <https://uk.reactjs.org/docs/getting-started.html>
2. Node.js *nodejs.org*: вебсайт URL: <https://nodejs.org/uk/>
3. JavaScript *developer.mozilla.org* : вебсайт URL: <https://developer.mozilla.org/ua/docs/Web/JavaScript>
4. Next.js *nextjs.org* : вебсайт URL: <https://nextjs.org/docs/basic-features/static-file-serving>
5. TyOrm *morioh.com* : вебсайт URL: <https://morioh.com/p/a28e50eac36f>
6. PostgreSQL *postgresql.org* : вебсайт URL: <https://www.postgresql.org/>
7. CJM *wikipedia* : вебсайт URL: [https://en.wikipedia.org/wiki/Convent\\_of\\_Jesus\\_and\\_Mary](https://en.wikipedia.org/wiki/Convent_of_Jesus_and_Mary)
8. MUI *wikipedia* : вебсайт URL: [https://uk.wikipedia.org/wiki/Material\\_design](https://uk.wikipedia.org/wiki/Material_design)
9. Nest.js *nestjs.com* : вебсайт URL: <https://nestjs.com/>
10. TypeScript *typescriptlang.org* : вебсайт URL: <https://www.typescriptlang.org/docs/>
11. User stories *quality-assurance-group.com* : вебсайт URL: <https://www.quality-assurance-group.com/user-stories/>
12. Three-Tier Architecture *techopedia.com* : вебсайт URL: <https://www.techopedia.com/definition/24649/three-tier-architecture>
13. Isomorphic Web Applications Книга, автор Еліз Колкер Гордон *amazon.com* : вебсайт URL: <https://www.amazon.com/Isomorphic-Web-Applications-Universal-Development/dp/161729439X>
14. Nest.js: прогресивна структура Node.js книга Грег Маголан *packtpub.com* : вебсайт URL: <https://www.packtpub.com/programming/nest-js-a-progressive-node-js-framework>

					<b>КР.ПЗ – 04.00.000 ПЗ</b>	Арк. 73
Змн.	Арк.	№ докум.	Підп.	Дата		

15. Full-Stack React Projects: Learn MERN stack development by building modern web apps *www.amazon.in*:вебсайт URL: <https://www.amazon.in/Full-Stack-React-Projects-development-building>

					<b>КР.ІІЗ – 04.00.000 ІЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		74