

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем факультету інформатики



РОЗРОБКА ЗАСТОСУНКУ (МІНІ ПРОГРАМИ) ДЛЯ WESCHAT

Текстова частина до курсової роботи
за спеціальністю «Комп'ютерні науки» 122

Керівник курсової роботи
старший викладач *Вовк Н. Є.*

(підпис)

Виконав студент 3 курсу

Процан Д.І.

« ____ » _____ 2024 р.

Київ 2024

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем факультету інформатики

ЗАТВЕРДЖУЮ
Зав. кафедри мультимедійних
систем, доцент, к.ф.-м.н.
_____ О.П.Жежерун
«_____» _____ 2024 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студенту 3-го курсу, факультету інформатики
Процапа Дмитрія Ігоровичу

Тема: Розробка застосунку (міні програми) для WeChat

Зміст ТЧ до курсової роботи:

Зміст

Анотація

Вступ

1. Опис проєкту та дослідження
2. Загальні відомості про WeChat
3. Створення серверної частини застосунку
4. Дослідження
5. Реалізація проєкту

Висновки

Список використаної літератури

Додатки

Дата видачі «__» _____ 2024 р. Керівник _____
(підпис)

Завдання отримав _____
(підпис)

Календарний план

№	Назва етапу курсового проєкту (роботи)	Термін виконання етапу
1.	Отримання завдання на курсову роботу	11.10.2023
2.	Ознайомлення із літературою для розробки додатку для WeChat	07.04.2024-17.04.2024
3.	Дослідження підходів до розробки та постачальників хмарних сервісів	17.04.2024-23.04.2024
4.	Розробка серверної частини додатку	23.04.2024-27.04.2024
5.	Розробка клієнтської частини додатку	27.04.2024-30.04.2024
6.	Розгортання застосунку	30.04.2024
7.	Написання текстової частини курсової роботи	01.05.2024-10.05.2024
8.	Створення презентації	11.05.2024-13.05.2024
9.	Подання роботи на кафедру для перевірки на плагіат	14.05.2024
10.	Захист курсової роботи	з 24.05.2024

Зміст

Індивідуальне завдання	2
Календарний план	3
Зміст	4
Анотація.....	5
Вступ	6
1 Опис проєкту	7
2 Загальні відомості про WeChat	8
2.1 Що таке WeChat	8
2.2 Міні програми в WeChat	9
2.3 QR коди та їх використання у WeChat.....	11
2.4 Файрвол Китаю	12
3 Дослідження	13
3.1 Функціонал застосунку	13
3.2 Серверна частина.....	14
3.3 Клієнтська частина	15
3.4 Розгортання та постачальники хмарних послуг.....	20
4 Реалізація проєкту	25
4.1 Використані технології	25
4.2 Серверна частина застосунку.....	25
4.3 WeChat застосунок	30
5 Висновки	32
6 Список використаної літератури.....	33

Анотація

У даній курсовій роботі розглянуто процес розробки застосунку для WeChat для створення карток-постерів.

Досліджено архітектуру клієнт-серверного застосунку для WeChat, та розробку серверної частини застосунку на мові JavaScript з використанням фреймворку Next.js, яка виконує роль графічного (UI) та програмного інтерфейсу (API) для взаємодії з клієнтською частиною застосунку. Для дослідження використовувалася документація для WeChat без перекладу, адже в англійському перекладі присутня значна кількість помилок, як у коді, так і в тексті.

Результатом роботи є розроблений застосунок для WeChat для створення карток-постерів, який дозволяє користувачам створювати, налаштовувати та поширювати картки-постери та панель адміністратора для керування застосунком.

Вступ

Картки-постери є популярними серед користувачів месенджерів, особливо серед користувачів WeChat у Китаї та країнах Азії. Картки-постери дозволяють створювати та поширювати інформацію серед друзів та знайомих у вигляді невеликих зображень. Вони можуть бути використані для реклами, промоцій, подій, новин та багато чого іншого.

Проте, створення карток-постерів вручну може бути складним та займати багато часу. Тому розробка застосунку для WeChat для створення карток-постерів може значно спростити цей процес та зробити його більш зручним, адже за допомогою застосунку у користувачів буде можливість створювати та налаштовувати картки-постери за декілька кліків, навіть не виходячи з месенджера.

1 Опис проєкту

Метою даного проєкту є розробка застосунку для WeChat для створення карток-постерів. Картки-постери — це невеликі зображення, які складаються із QR коду та певного тексту, іноді вони можуть містити зображення (рамки) або інші декоративні елементи, та дозволяють користувачам ділитися інформацією у зручному та презентабельному форматі.

Генерація карток-постерів відбувається в месенджері за допомогою створеного застосунку. Налаштування карток-постерів дозволяє змінювати рамку (перелік доступних рамок знаходиться та налаштовується на серверній частині застосунку) або колір постеру. Поширення карток-постерів відбувається шляхом надсилання їх іншим користувачам месенджера за допомогою вбудованого у WeChat функціоналу.

Серверна частина застосунку відповідає за аналітики та отримання інформації про доступні рамки та їх зображення.

2 Загальні відомості про WeChat

2.1 Що таке WeChat

WeChat — це китайський месенджер, який належить компанії Tencent із понад 1 мільярдом користувачів[1] (Рис 1.2) по всьому світу. WeChat має велику популярність в Китаї та країнах Азії. Створений у 2011 році, WeChat швидко став одним з найпопулярніших месенджерів у Китаї, завдяки своїм функціям та можливостям. WeChat має багато функцій, які дозволяють користувачам спілкуватися, обмінюватися повідомленнями, фотографіями та відео, використовувати групові чати, викликати таксі, замовляти їжу, покупати товари, бронювати готелі та квитки на транспорт, грати в ігри, слухати музику, дивитися відео та інше. WeChat також має функцію голосового та відео дзвінків, яка дозволяє користувачам спілкуватися в реальному часі.



(Рис 2.1) Логотип WeChat

Country	# of WeChat Users ▾
China	810M
Malaysia	12M
India	10M
Russia	9.5M
Japan	5.5M
South Korea	5M
United States	4M
Indonesia	3M
Mexico	2.5M
Thailand	2.5M

(Рис. 2.2) Користувачі WeChat у різних країнах [1]

2.2 Міні програми в WeChat

У 2017 році WeChat запустив міні програми, які дозволяють розробникам створювати невеликі застосунки для месенджера. Міні програми в WeChat стали дуже популярні серед користувачів, оскільки вони дозволяють швидко та зручно використовувати різноманітні сервіси та функції. Міні програми в WeChat можуть бути використані для пошуку, покупок, бронювання, ігор, новин та іншого. Міні програми в WeChat дозволяють розробникам створювати невеликі застосунки для месенджера, які можуть бути використані користувачами для різних цілей.

Для розробки міні програм в WeChat використовуються мови WXML [2] (WeiXin Markup Language), WXSS [3] (WeiXin Style Sheet) та JavaScript.

WXML — це мова розмітки, подібна до HTML, яка використовується для створення структури сторінок міні програм, проте, на відміну від HTML, WXML підтримує прив'язку даних (data binding), цикли, умови та компоненти із синтаксисом, схожим на Vue. За допомогою WXML можна створювати динамічні сторінки застосунку, реалізуючи логіку сторінки на JavaScript, використовуючи систему подій та велику кількість вбудованих компонентів.

Для стилізації сторінок та компонентів використовується мова WXSS, яка має синтаксис та більшість функцій CSS із додатковим функціоналом:

- **gpx (responsive pixel)** - адаптується до ширини екрана пристрою, за замовчуванням ширина – 750 гpx.
- **Імпорт стилів** – у файлах WXSS, на відміну від CSS можна використовувати `@import` для імпорту зовнішньої таблиці стилів у файл.
- **Ізоляція стилів** – стилі у файлі компоненту або сторінки будуть застосовуватися лише для елементів самого компоненту і не будуть впливати на глобальні стилі. Глобальні стилі застосовуються до усіх елементів застосунку.

Логіка застосунку або сторінок/компонентів реалізується на JavaScript у відповідному до WXML файлі із структурою Page або Component (Рис. 1.3).

JavaScript (JS) - це високорівнева, динамічна, об'єктно-орієнтована прототипна мова програмування, яка використовується для створення веб-сайтів та веб-застосунків. JavaScript є однією з найпопулярніших мов програмування у світі та використовується для розробки веб-сайтів, веб-застосунків, мобільних додатків, десктопних додатків, серверних застосунків та інших програмних продуктів.

```

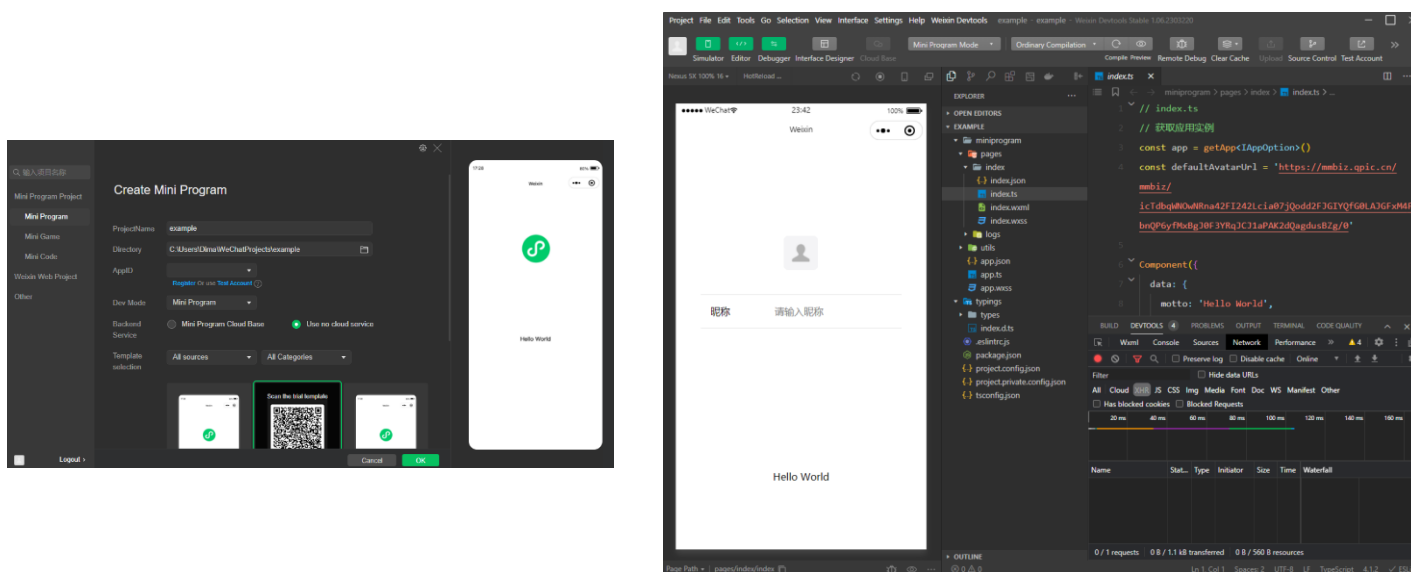
// index.js
Page({
  data: {
    text: "This is page data."
  },
  Component({
    properties: {
      // 这里定义了innerText属性, 属性值可以在组件使用时指定
      innerText: {
        type: String,
        value: 'default value',
      }
    },
  },

```

(Рис. 2.3) Структура сторінки та компоненту

Також, WeChat має обмеження для міні програм, які розробляються розробниками. Обмеження включають обмеження на розмір міні програм, доступ по пристрою користувача, обмеження на використання зовнішніх ресурсів (API, зображення) (усі зовнішні ресурси, які використовуються мають бути зазначені заздалегідь і використовувати https протокол, при цьому ресурс має бути дозволений, якщо використовується вже наявний сервіс або сертифікований та затверджений WeChat у випадку використання власного сервісу). Обмеження для міні програм у WeChat дозволяють забезпечити безпеку та приватність користувача застосунку, а також забезпечити якість та швидкодію міні програм.

Для створення, розробки, дебагу, тестування та розгортання міні програм використовується IDE (Integrated Development Environment) Weixin DevTools (Рис. 1.4), яке можна завантажити, створивши акаунт розробника у WeChat. Середовище складається із симулятора та редактора коду із вбудованими плагінами для перевірки синтаксису та автозавершенням коду для WXML, WXSS та JavaScript. Також, IDE підтримує віддалений дебаг на пристрої розробника: за допомогою функції “Remote Debug” може протестувати на власному пристрої повноцінну версію застосунку, усі помилки і повідомлення якої будуть відображатися в середовищі розробки.



(Рис. 2.4) Зліва зображено інтерфейс створення нового застосунку, а праворуч – симулятор та редактор коду

2.3 QR коди та їх використання у WeChat

QR-код (Рис. 1.5) - зображення, яке містить інформацію, яка може бути зчитана за допомогою камери смартфона або іншого пристрою зі спеціальним програмним забезпеченням. Він швидко став невід'ємною частиною сучасного життя, особливо в Азії, де платформи, такі як WeChat, активно використовують його для різних цілей.



(Рис 2.5) Приклади QR-кодів, зліва у кодї заковано слово “Hello”, а праворуч – “https://example.com/”

QR-коди стали неодмінною частиною цифрової культури в Азії, особливо в Китаї, де платформи, такі як WeChat, активно використовують їх для різноманітних операцій. У WeChat, найпопулярнішому месенджері та платіжній системі в Китаї, QR-коди є основним інструментом для здійснення платежів, аутентифікації, обміну контактною інформацією та взаємодії зі спеціальними послугами та промоакціями. Користувачі можуть просто сканувати QR-коди за допомогою камери свого смартфона, щоб здійснити покупки, підписатися на різноманітні сервіси або вступити в спілкування з іншими користувачами. Використання QR-кодів у WeChat допомагає спростити процеси та забезпечити зручну та ефективну взаємодію між користувачами та бізнесами. Одним із популярних використань QR-кодів є створення карток-постерів, які дозволяють замість звичайного QR-коду використовувати зображення з текстом та іншими елементами для привертання уваги користувачів, вони значно частіше використовуються для реклами, промоцій, подій та іншого.

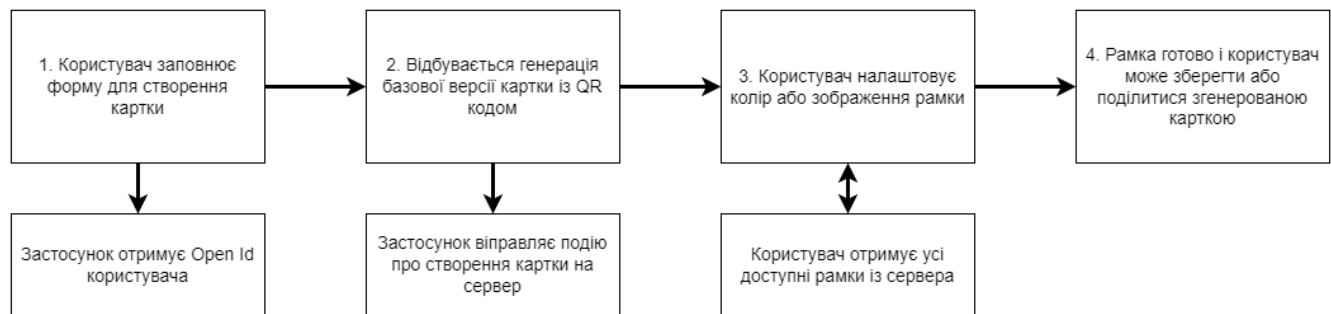
2.4 Файрвол Китаю

Одним із важливих факторів розробки застосунків для WeChat є обмеження, які накладає файрвол Китаю, який накладає значні обмеження на розробників, оскільки він обмежує доступ до значної кількості сайтів та сервісів, таких як Google, Facebook, Twitter, YouTube та інші. Це значно ускладнює розробку застосунків для WeChat, бо розробники при створенні застосунків повинні враховувати обмеження файрволу Китаю та шукати шляхи для обходу заборонених сервісів або використовувати доступні альтернативи.

3 Дослідження

3.1 Функціонал застосунку

Для розуміння необхідного функціоналу застосунку необхідно визначити основні функції, які має виконувати застосунок та як користувачі будуть взаємодіяти з ним. Користувачів можна розділити на дві групи: звичайних користувачів та адміністраторів. Звичайні користувачі мають можливість створювати, налаштовувати та поширювати картки-постери, адміністратори мають можливість керувати рамками та переглядати статистику застосунку. Для звичайного користувача взаємодію із застосунком можна зобразити за допомогою наступної схеми (Рис. 3.1):



(Рис 3.1) Схема взаємодії користувача із застосунком

Для адміністратора застосунку необхідно реалізувати панель адміністратора, яка буде відповідати за керування рамками та перегляд статистики застосунку, для цього необхідно реалізувати наступний функціонал:

- Аутентифікація адміністратора
- Додавання, редагування та видалення рамок-постерів
- Перегляд статистики застосунків
- Можливість керувати доступом застосунків до API серверу
- Створення, редагування та видалення застосунків (ключів та секретів застосунків для доступу до API серверу)

3.2 Серверна частина

Визначивши функціонал застосунку, можна приступити до розробки серверної частини застосунку, яка буде відповідати за веб інтерфейс для адміністратора та інтерфейсу для взаємодії з клієнтською частиною застосунку, базу даних для зберігання рамок-постерів, статистики застосунку та доступу до WeChat API.

Для взаємодії клієнтської частини застосунку з серверною частиною необхідно реалізувати Rest API.

API (Application Programming Interface) — програмний інтерфейс, що надає можливість різним програмам або сервісам взаємодіяти одна з одною. API визначає набір правил та протоколів, за якими програми можуть обмінюватися даними та взаємодіяти одна з одною.

REST API (Representational State Transfer Application Programming Interface) — стиль організації ендпоінтів серверу, який широко використовується для створення веб-сервісів. Основною ідеєю REST є використання стандартних протоколів HTTP для взаємодії з ресурсами (дані, функції) на веб-сервері.

Основні принципи REST API:

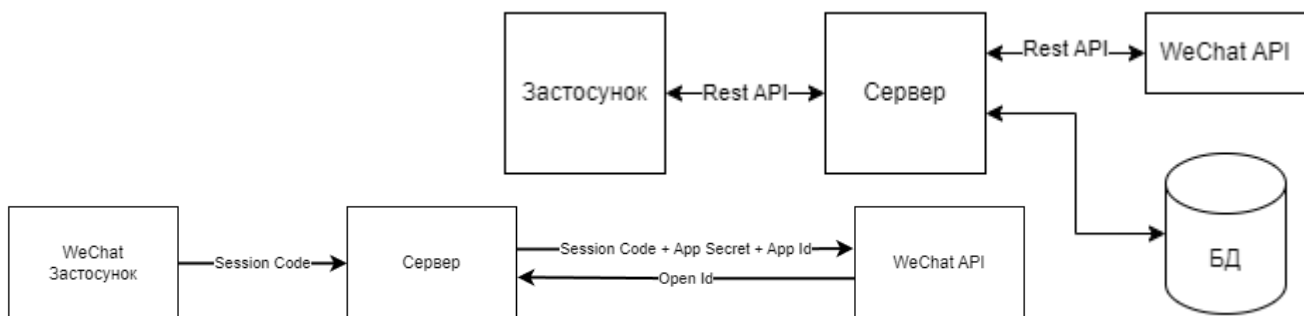
- Клієнт-серверна архітектура — взаємодія між клієнтом та сервером здійснюється через HTTP протокол, сервер відповідає за зберігання та обробку даних, а клієнт — за відображення та взаємодію з ними.
- Відсутність стану — кожен запит до сервера має містити всю необхідну інформацію для обробки запиту, сервер не зберігає стан клієнта між запитами.
- Зосередженість на ресурсах — уся інформація в REST API представлена у вигляді ресурсів, кожен ресурс має унікальний ідентифікатор URL (Uniform Resource Locator) та може бути доступний за допомогою HTTP методів. В

залежності від методу запиту (GET, POST, PUT, DELETE) та URL сервер виконує різні операції над вказаним ресурсом. Наприклад, GET запит використовується для отримання даних, POST — для створення нового ресурсу, PUT — для оновлення існуючого ресурсу, DELETE — для видалення ресурсу.

- Форматування даних — REST API використовує стандартні формати даних для обміну інформацією між клієнтом та сервером, такі як JSON (JavaScript Object Notation) або XML (eXtensible Markup Language).

Цей підхід дозволяє створювати веб-сервіси, з якими зручно взаємодіяти і які можуть бути легко розширені та інтегровані з іншими застосунками та сервісами.

Схему взаємодії застосунку з серверною частиною можна зобразити наступним чином (Рис. 3.2):



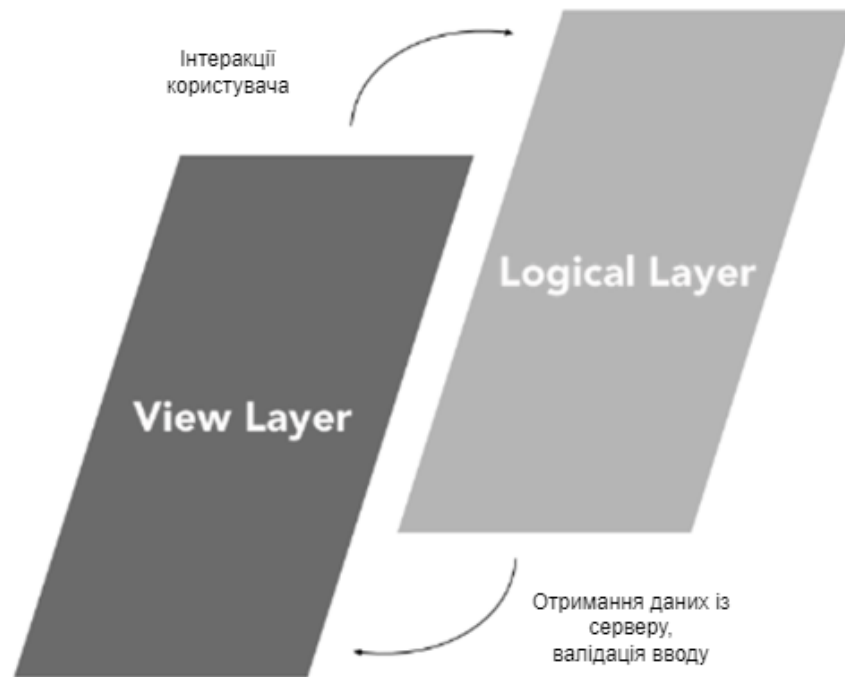
(Рис. 3.2) Праворуч зображена загальна схема взаємодії, а ліворуч – механізм отримання Open Id користувача.

3.3 Клієнтська частина

Клієнтська частина застосунку відповідає за відображення та взаємодію з користувачем, вона має бути зручною та інтуїтивно зрозумілою для користувачів.

Реалізація клієнтської частини застосунку відбувається за допомогою міні програм в WeChat. Застосунки в WeChat реалізуються за допомогою мов WXML, WXSS і JavaScript [2] та використовують розділення сторінок та компонентів на Logic Layer (логіка сторінки) та View Layer (відображення сторінки) (Рис. 3.3).

Такий підхід забезпечує належне розділення логіки та відображення сторінки, що дозволяє зберігати код чистим та зрозумілим, а також забезпечує можливість повторного використання компонентів та сторінок у різних частинах застосунку.



(Рис. 3.3) Зображення взаємодії логіки та відображення застосунку із прикладами [4]

Система компонентів/сторінок, синтаксис WXML та організація коду та даних у логічному шарі майже повністю відповідає розробці застосунку із використанням UI фреймворку Vue (Табл 3.).

Також, для розробки логіки застосунку можливо використовувати зовнішні JS бібліотеки із реєстру npm. Npm - це пакетний менеджер для JavaScript, що дозволяє легко управляти залежностями проєкту. З його допомогою можна швидко встановлювати, оновлювати та видаляти пакети, а також керувати їх версіями. Використання зовнішніх JS бібліотек з npm розширює можливості розробки,

дозволяючи використовувати готові рішення для розв'язання різних завдань без необхідності написання коду з нуля.

Проте на зовнішні бібліотеки теж накладаються певні обмеження:

- У бібліотеці не має використовуватися HTML DOM (Document Object Model), адже у WXML використовується інший механізм взаємодії зі сторінкою.
- Не можна використовувати `fetch`. `Fetch` [5] — це вбудована в JavaScript функція, яка надає можливість здійснювати асинхронні запити до сервера. Натомість у WeChat застосунку використовується `wx.request`, який є повною заміною `fetch`.
- Обмеження на використання Canvas API. Canvas API — це інтерфейс програмування застосунків (API) в JavaScript, який дозволяє малювати графіку на веб-сторінках за допомогою елемента HTML `<canvas>`.

Приклад	Vue	WXML + JS
Відображення інформації	<pre> <script> export default { data() { return { msg: "Hello" } } } </script> <template> Message: {{ msg }} </template> </pre>	<pre> Page({ data: { msg: "Hello" } }) <view> Message: {{ msg }} </view> </pre>
Динамічні зміни	<pre> <script> export default { data() { return { counter: 0 } } } </script> <template> <button @click="counter++"> Counter: {{ counter }} </button> </template> </pre>	<pre> Page({ data: { counter: 0 }, plus() { const counter = this.data.counter + 1; this.setData({ counter }); } }) <button bindtap="plus"> Counter {{ counter }} </button> </pre>

Табл. 3.1

<p>Відображення списків</p>	<pre><script> export default { data() { return { array: [1, 2, 3] } } } </script> <template> <div v-for="item in array"> {{ item }} </div> </template></pre>	<pre>Page({ data: { array: [1, 2, 3] } }) <view wx:for="{{array}}"> {{ item }} </view></pre>
<p>Умови</p>	<pre><script> export default { data() { return { num: 5 } } } </script> <template> <div v-if="num > 10"> 10+ </div> <div v-else-if="num > 1"> 1+ </div> <div v-else> 0- </div> </template></pre>	<pre>Page({ data: { num: 5 } }) <view wx:if="{{num > 10}}"> 10+ </view> <view wx:elif="{{num > 1}}"> 1+ </view> <view wx:else> 0- </view></pre>

Продовження Табл. 3.1

3.4 Розгортання та постачальники хмарних послуг

Постачальники хмарних послуг (також відомі як хмарні провайдери) [6] є ІТ-компаніями, які надають клієнтам віртуалізовані ресурси на вимогу. Ці ресурси побудовані на базі фізичних компонентів ІТ-інфраструктури, таких як сервери, сховища та мережі.

Основні переваги хмарних сервісів:

- Масштабованість — можливість збільшувати або зменшувати обсяг ресурсів в залежності від потреб користувача.
- Простота використання — при використанні хмарних послуг користувачам не потрібно витратити час на налаштування та підтримку власної ІТ-інфраструктури.
- Безпека та надійність — хмарні послуги забезпечують високий рівень безпеки та надійності для даних користувачів.
- Економія коштів — користувачі платять лише за фактично використані ресурси.

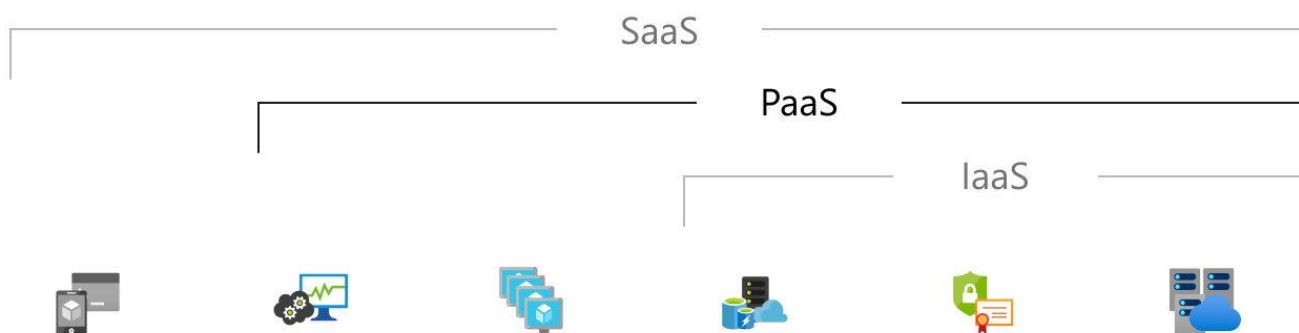
Види хмарних послуг (Рис. 3.4):

- Інфраструктура як сервіс, IaaS (Infrastructure as a Service) — надання віртуалізованих ресурсів, таких як сервери, сховища та мережі.
- Платформа як сервіс, PaaS (Platform as a Service) — хмарний провайдер надає платформи для розробки, тестування та розгортання застосунків. Розробники можуть використовувати ці платформи для створення власних застосунків без необхідності налаштування та підтримки інфраструктури.
- Програмне забезпечення як сервіс (SaaS) — надає користувачам доступ до програмного забезпечення через хмару. Користувачі можуть

використовувати програмне забезпечення без необхідності встановлення та оновлення на своїх пристроях.

- Функції як сервіс (FaaS) — надання можливості виконання функцій без необхідності управління інфраструктурою. Наприклад, реалізація функцій застосунку, які виконуються відповідно до викликів користувачів.

(Рис 3.4) Графічне порівняння різних видів хмарних сервісів [7]



Популярні постачальники хмарних послуг:

- Amazon Web Services (AWS) — найбільший постачальник хмарних послуг у світі, який надає широкий спектр послуг для розробки та розгортання веб-сервісів та застосунків.
- Microsoft Azure — хмарний сервіс від компанії Microsoft, який надає послуги для розробки, тестування та розгортання веб-сервісів та застосунків.
- Google Cloud Platform (GCP) — хмарний сервіс від компанії Google, який надає послуги для розробки, тестування та розгортання веб-сервісів та застосунків.
- IBM Cloud — хмарний сервіс від компанії IBM, який надає послуги для розробки, тестування та розгортання веб-сервісів та застосунків.
- Alibaba Cloud — хмарний сервіс від компанії Alibaba, який надає послуги для розробки, тестування та розгортання веб-сервісів та застосунків.

- Netlify — хмарний сервіс для розгортання статичних веб-сайтів та застосунків.
- Vercel — хмарний сервіс, побудований на основі AWS із спеціалізацією на розгортанні веб-сайтів та покращеним DX (Developer Experience).

У якості потенційного постачальника хмарних послуг для розгортання серверної частини застосунку будуть розглядатися Google Cloud Platform (GCP) та Vercel (Табл 3.2) [9].

Критерій	Google Cloud Platform	Vercel
Ціна	Вартість послуг Google Cloud Platform залежить від використаних ресурсів та обсягу трафіку.	Vercel надає безкоштовний тариф для невеликих проєктів та платний тариф для підприємств. При великому обсязі трафіку стає дорожчим за Google Cloud Platform. [10]
Простота використання	Google Cloud Platform вимагає налаштування та підтримки інфраструктури.	Vercel надає простий інтерфейс для розгортання веб-сайтів та застосунків, при використанні мета фреймворків, таких як Next.js, Nuxt.js, Gatsby, розгортання відбувається автоматично.
Масштабованість	Google Cloud Platform надає можливість масштабування ресурсів в залежності від потреб користувача.	Vercel керує масштабуванням ресурсів автоматично, в залежності від навантаження на застосунок.
HTTPS	Google Cloud Platform надає можливість налаштування HTTPS для веб-сайтів та застосунків.	Vercel надає безкоштовний SSL сертифікат та HTTPS домен для розгортання веб-сайтів та застосунків.
Сховище	Google Cloud Storage надає можливість зберігання даних на власних серверах або в хмарі.	Vercel Blob надає можливість зберігання незначних об'ємів даних на серверах Vercel.

(Табл 3.2)

За рахунок простоти використання та безкоштовного тарифу для невеликих проєктів, безкоштовного SSL сертифікату та автоматичного масштабування ресурсів, Vercel є найбільш підходящим варіантом для розгортання серверної частини застосунку даного проєкту.

Vercel надає послуги Postgres бази даних для зберігання даних, проте для зберігання даних серверної частини застосунку буде використаний сервіс Turso, який надає не лише розміщення SQLite бази даних в хмарі, а й зручний інтерфейс для роботи з базою даних, резервне копіювання та реплікацію даних. Сервіс Turso був обраний через його простоту використання та можливість розширення в майбутньому.

4 Реалізація проєкту

4.1 Використані технології

Серверна частина застосунку реалізована за допомогою Node.js, середовища виконання JavaScript, яке дозволяє створювати веб-сервери та веб-застосунки на мові JavaScript. Для створення веб-сервера та панелі адміністратора використовувався фреймворк Next.js, який надає можливість створення веб-серверів та веб-застосунків на основі React.js, а також надає можливість створення статичних та динамічних сторінок. Одним із важливих факторів для вибору цього фреймворку стало те, що Vercel підтримує розгортання Next.js застосунків в один клік. Усі дані серверної частини застосунку зберігаються в SQLite базі даних Turso, а за зберігання зображення для рамок-постерів відповідає Vercel Blob.

Клієнтська частина застосунку реалізована у вигляді міні програми в WeChat, із використанням мов WXML, WXSS та JavaScript. Для генерації QR-кодів використовувався бібліотека QR Code generator library [8], яка надає можливість генерації QR-кодів за допомогою JavaScript.

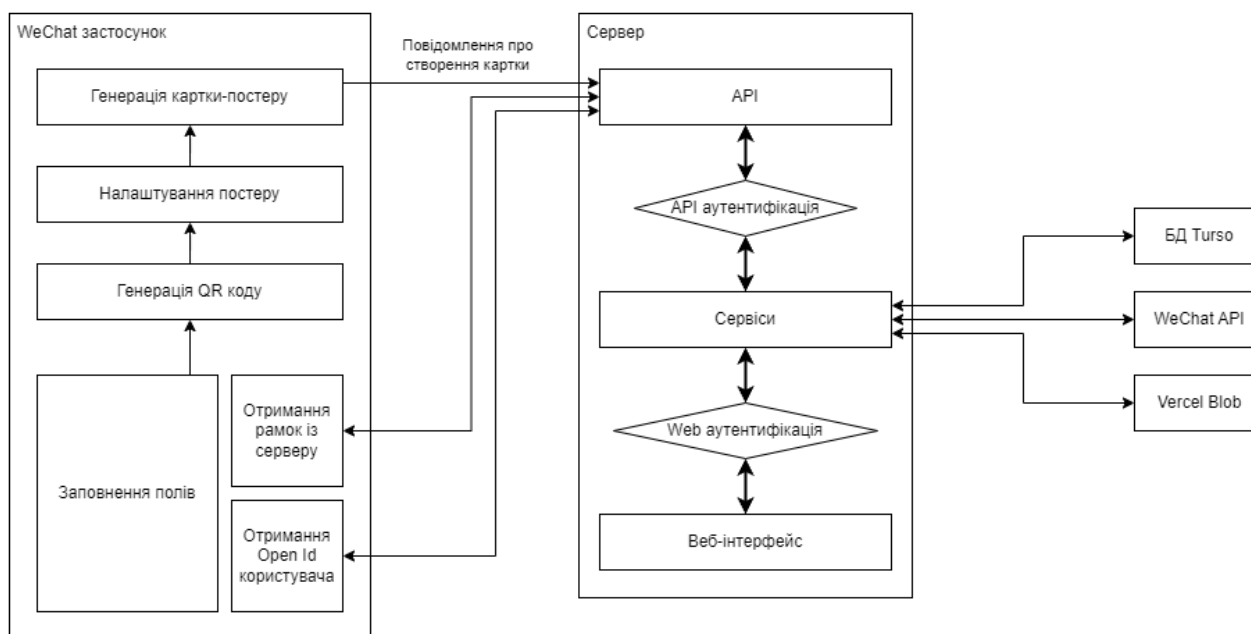
4.2 Серверна частина застосунку

Реалізація серверної частини застосунку побудована за наступною схемою (Рис. 4.1).

Серверна частина застосунку має наступні API ендпоінти:

- POST /api/auth/login — аутентифікація адміністратора
- GET /api/v1/mini/frames — отримання списку рамок-постерів
- GET /api/v1/mini/frames/:id — отримання рамки-постеру за ідентифікатором
- POST /api/v1/mini/analytics/report — створення запису аналітики
- POST /api/v1/mini/openid — отримання ідентифікатора користувача

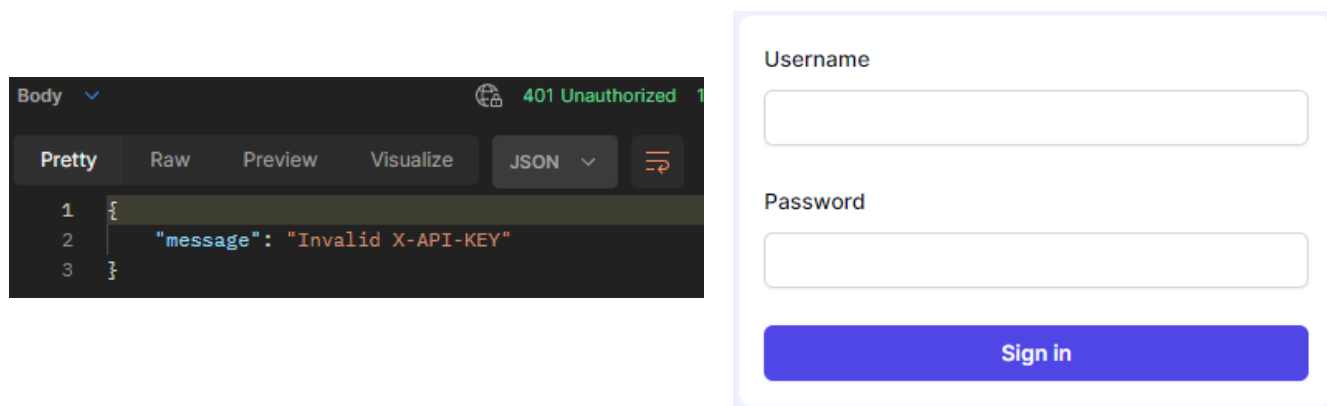
Усі інші CRUD (Create, Read, Update, Delete) операції над рамками-постерами виконуються через веб-інтерфейс панелі адміністратора за допомогою механізму `server actions` у фреймворку `Next.js`.



(Рис. 4.1) Схема проєкту

Для панелі адміністратора було реалізовано аутентифікацію адміністратора за допомогою JWT (JSON Web Token), який надає можливість створення та перевірки токенів для аутентифікації користувачів без використання сесій, що дозволяє зберігати стан аутентифікації на клієнтській стороні та використовувати `serverless` архітектуру для реалізації аутентифікації. Для обробки JWT токенів використовувалася бібліотека `NextAuth.js`, яка надає можливість реалізації аутентифікації за допомогою різних провайдерів, таких як Google, Facebook, GitHub, а також можливість реалізації власного провайдера аутентифікації, як у даному випадку.

При спробі використання API ендпоінтів без аутентифікації адміністратора сервер повертатиме помилку 401 Unauthorized (Рис. 4.2а), а при спробі перегляду сторінок панелі адміністратора без аутентифікації сервер перенаправлятиме користувача на сторінку аутентифікації (Рис. 4.2б).

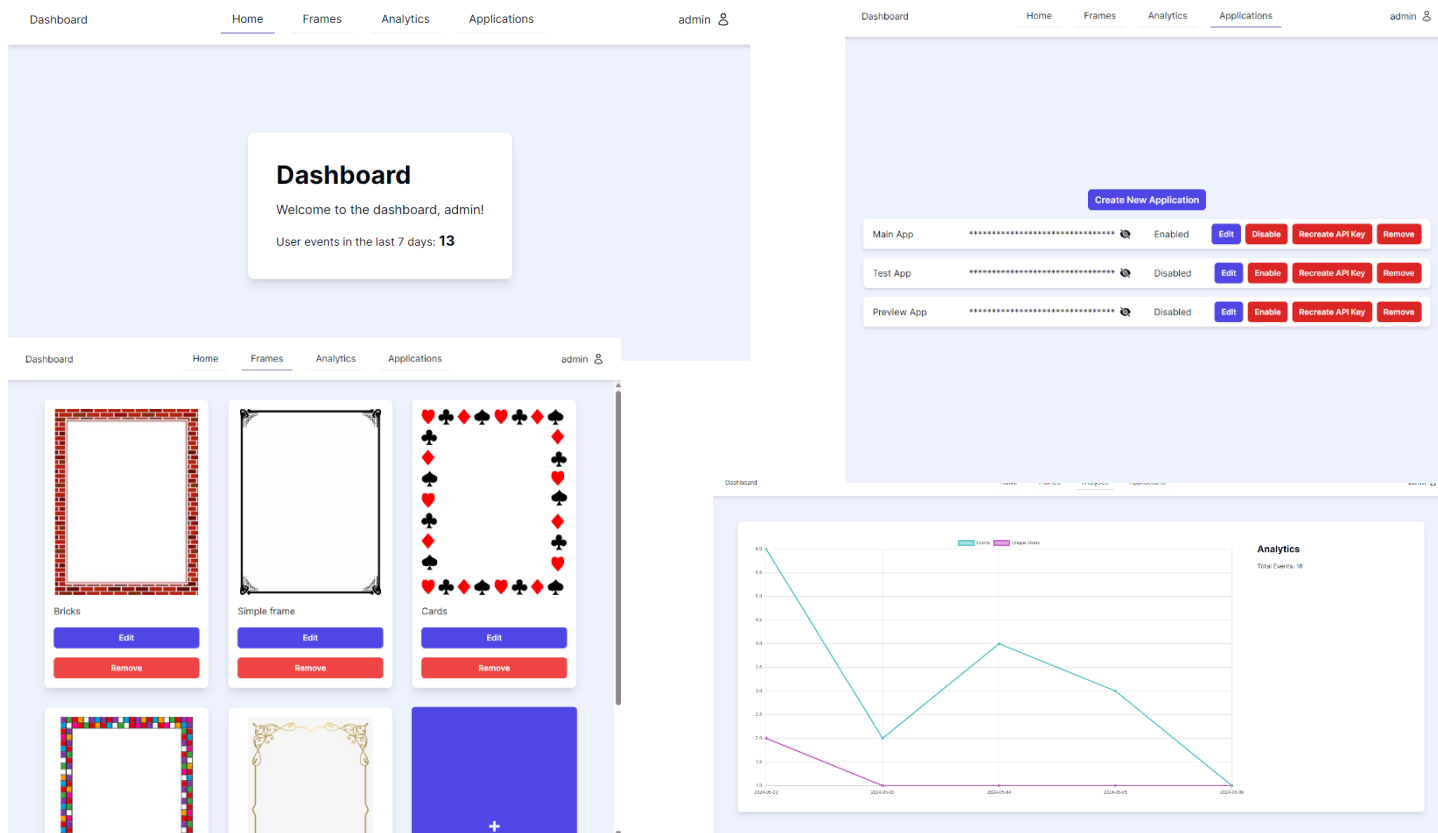


(Рис. 4.2) (а) Повідомлення при спробі використання API без ключа аутентифікації або із невірним ключем. (б) Форма логіну для користувача

У панелі адміністратора доступні наступні сторінки:

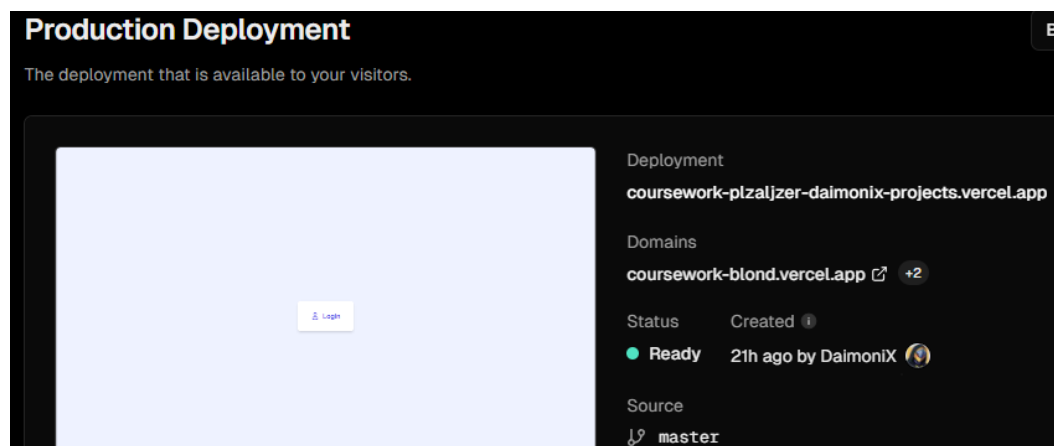
- /dashboard — основна сторінка панелі адміністратора, на якій відображається кількість подій за останній тиждень.
- /dashboard/frames — сторінка для керування рамками-постерами, на якій відображається список рамок-постерів та можливість додавання, редагування та видалення рамок-постерів.
- /dashboard/analytics — сторінка для перегляду статистики застосунку, на якій відображається кількість подій за останній місяць та графік активності користувачів.
- /dashboard/applications — сторінка для керування застосунками, на якій відображається список застосунків та можливість додавання, редагування, видалення, активації і деактивації застосунків та можливість генерації API ключів для застосунків.

Інтерфейс сторінок (Рис. 4.3):



(Рис. 4.3) Зліва згорі - /dashboard, праворуч згорі - /dashboard/applications, зліва знизу - /dashboard/frames, праворуч знизу - /dashboard/analytics

Після завершення серверної частини було використано сервіс Vercel для розгортки застосунку (Рис 4.4), за рахунок використання фреймворку Next.js (який розроблений Vercel) розгортання застосунку відбулося дуже швидко і без додаткових налаштувань.



Trusted Domains

Optional, Each trusted domain name can be configured with 5 domain names.

Modify

Server Configuration	Domain Name
Request Legal Domain Name	https://coursework-blond.vercel.app
DownloadFile Legal Domain Name	https://coursework-blond.vercel.app
UploadFile Legal Domain Name	
Socket Legal Domain Name	

MiniProgram Sandbox Profile Info

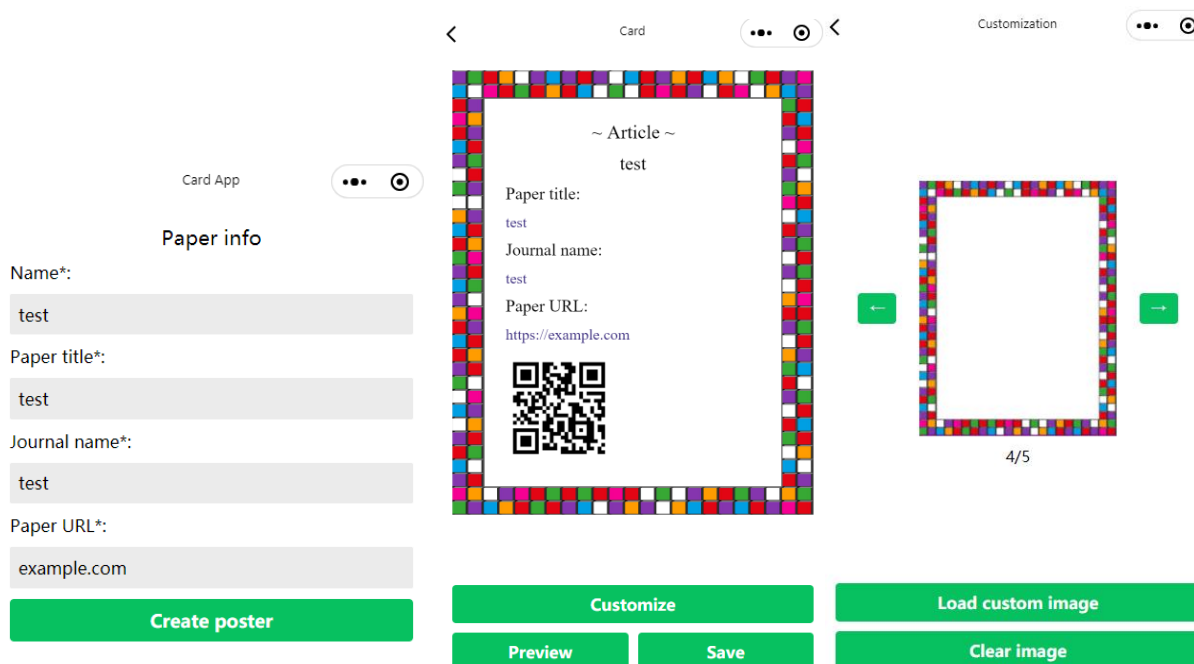
AppID wx437081 [REDACTED]
AppSecret 81bd [REDACTED]

(Рис 4.4) Результат розгортання застосунку та використання отриманого домену для WeChat застосунку

4.3 WeChat застосунок

WeChat застосунок складається із трьох сторінок (Рис. 4.5):

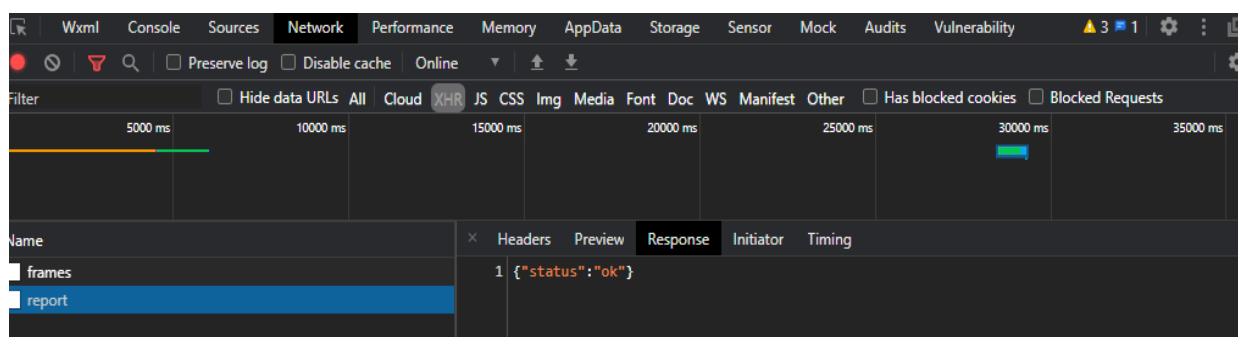
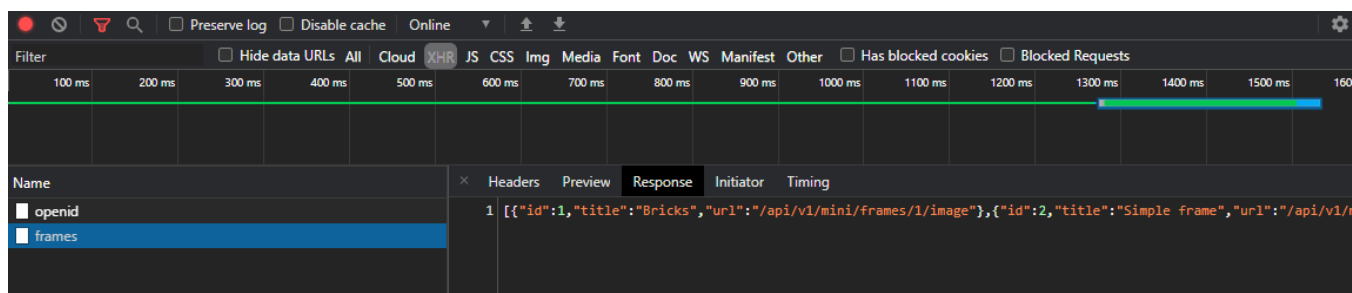
- /index — головна сторінка застосунку, на якій відображається форма із полями для введення тексту.
- /display - сторінка для відображення вже згенерованого рамки-постеру.
- /customize - сторінка для налаштування рамки-постеру.



(Рис. 4.5) Сторінка для заповнення форми, сторінка із згенерованою рамкою, сторінка для налаштування рамки.

При відкритті головної сторінки застосунку користувачу відображається форма із полями для введення тексту, після введення тексту користувач може натиснути кнопку "Create poster" для генерації постеру. Після генерації постеру користувачу відображається сторінка для відображення рамки-постеру, на якій відображається згенерований постер. Користувач може натиснути кнопку "Preview" для перегляду постеру у великому розмірі або кнопку "Save" для збереження постеру на пристрої. При відкритті меню "Preview" користувач також може одразу поділитися постером у WeChat.

Якщо користувач натисне кнопку "Customize", він буде перенаправлений на сторінку для налаштування рамки-постеру, на якій він зможе обрати рамку-постеру із списку рамок-постерів із серверу, завантажити власне зображення для рамки-постеру або обрати кольору рамки. При відкритті застосунку автоматично відправляється запит на сервер для отримання списку рамок-постерів (Рис. 4.6 а) та запит на отримання Open Id користувача для відстеження активності користувачів, а після генерації постеру відправляється запит на сервер для створення запису аналітики (Рис. 4.6 б).



(Рис. 4.6) Запит для отримання Open Id користувача та список рамок (а), Запит на створення запису аналітики (б)

5 Висновки

У ході виконання даного проєкту було розроблено застосунок для генерації рамок-постерів у WeChat, який дозволяє користувачам створювати та поширювати рамки-постери у мережі WeChat. Для реалізації проєкту були досліджені технології створення веб-застосунків, різні підходи до реалізації серверної та клієнтської частин застосунку, а також проаналізовані можливості розгортання застосунку у хмарних сервісах.

Дослідження проведене у ході виконання даного проєкту дозволило краще зрозуміти принципи розробки клієнт-серверних застосунків, а також навчитися розгортати застосунки у хмарних сервісах.

6 Список використаної літератури

1. WeChat Users by Country 2024 [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://worldpopulationreview.com/country-rankings/wechat-users-by-country>
2. 小程序框架 / 视图层 / WXML [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://developers.weixin.qq.com/miniprogram/dev/framework/view/wxml/>
3. 小程序框架 / 视图层 / WXSS [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://developers.weixin.qq.com/miniprogram/dev/framework/view/wxss.html>
4. WeChat Mini-programs Wiki [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://github.com/apelegri/wechat-mini-program-wiki>
5. Using the Fetch API - Web APIs | MDN [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch
6. Кто такой хмарний провайдер і як його вибрати [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://blog.colobridge.net/uk/2024/02/cloud-service-provider-ua/>
7. What is PaaS? Platform as a Service | Microsoft Azure [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-paas/>
8. QR Code generator library by nayuki [Електронний ресурс] / nayuki. – 2021. – Режим доступу до ресурсу: <https://www.nayuki.io/page/qr-code-generator-library>
9. [Google Cloud vs Vercel - 2024 Comparison - Software Advice](https://www.softwareadvice.com/app-development/google-cloud-platform-profile/vs/vercel/) [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://www.softwareadvice.com/app-development/google-cloud-platform-profile/vs/vercel/>
10. How, When, and Why you should switch from Vercel to a different Hosting Provider (Especially for Next.js) [Електронний ресурс] / Sushrit Pasupuleti. – 2022. – Режим доступу до ресурсу: <https://medium.com/@sushrit.pk21/how-when-and-why-you-should-switch-from-vercel-to-a-different-hosting-provider-especially-for-8ba25e439788>