

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

**РОЗРОБКА ПЛАТФОРМИ ДЛЯ ПІДТРИМКИ ТА ПРОСУВАННЯ МИТЦІВ  
З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ FLUTTER ТА DART**  
Текстова частина до кваліфікаційної роботи  
за спеціальністю «Інженерія програмного забезпечення» 121

Керівник курсової роботи  
с.в. Борозенний С.О.

\_\_\_\_\_ (підпис)

« \_\_\_\_ » \_\_\_\_\_ 2024 р.

Виконала студентка Махиня А.О.

« \_\_\_\_ » \_\_\_\_\_ 2024 р.

Київ 2024

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

ЗАТВЕРДЖУЮ  
Зав. кафедри мультимедійних систем,  
к.ф.-м.н.  
\_\_\_\_\_ О. П. Жежерун  
(підпис)  
« \_\_\_\_ » \_\_\_\_\_ 2023 р.

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ**

на кваліфікаційну роботу

студентці Махині Анастасії Олександрівні факультету інформатики 4 курсу  
ТЕМА «Розробка платформи для підтримки та просування митців з використанням технологій Flutter та Dart»

Зміст ТЧ до кваліфікаційної роботи:

Зміст

Анотація

Вступ

1. Аналіз предметної області
2. Обґрунтування вибору засобів розробки
3. Розробка застосунку

Висновки

Список літератури

Додатки

Дата видачі « \_\_\_\_ » \_\_\_\_\_ 2023 р.

Керівник \_\_\_\_\_  
(підпис)

Завдання отримав \_\_\_\_\_  
(підпис)

Тема: Розробка платформи для підтримки та просування митців з використанням технологій Flutter та Dart

Календарний план виконання роботи:

№ п/п	Назва етапу	Термін виконання етапу	Примітка
1.	Отримання теми кваліфікаційної роботи	25.10.2023	
2.	Огляд технічної літератури за темою роботи	01.11.2023 – 26.12.2023	
3.	Аналіз предметної області	27.11.2023 – 05.01.2024	
4.	Розробка платформи	05.01.2024 – 20.03.2024	
5.	Написання текстової частини кваліфікаційної роботи	20.03.2024 – 15.05.2024	
10.	Створення слайдів для доповіді та написання доповіді.	13.05.2024 – 16.05.2024	
11.	Остаточне оформлення роботи та слайдів	17.05.2024	
12.	Захист кваліфікаційної роботи	28.05.2024	

Студент \_\_\_\_\_

Керівник \_\_\_\_\_

“ \_\_\_\_\_ ” \_\_\_\_\_ 2023 р.

## Зміст

Зміст.....	4
АНОТАЦІЯ .....	5
ВСТУП.....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	7
1.1 Опис предметної області.....	7
1.2 Огляд існуючих альтернатив.....	7
1.2.1 Behance .....	8
1.2.2 ArtStation .....	9
1.2.3 Facebook.....	10
1.2.4 Instagram.....	12
1.2.5 Висновки .....	13
1.3 Формування вимог до застосунку.....	14
2 ОБҐРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РОЗРОБКИ.....	16
2.1 Flutter .....	16
2.1.1 Аналіз переваг фреймворку Flutter .....	16
2.1.2 Акуальність Flutter.....	17
2.2 Firebase.....	21
2.3 Android.....	22
3 РОЗРОБКА ЗАСТОСУНКУ .....	24
3.1 Структура проекту.....	24
3.2 Демонстрація роботи платформи.....	31
ВИСНОВКИ.....	41
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	42
Додаток А.....	43

## АНОТАЦІЯ

У роботі описано створення платформи для підтримки та просування митців з використанням технологій Flutter та Dart.

Кінцевий програмний продукт надає користувачам можливість створювати власні профілі та ділитись своїми мистецькими роботами і виставками, також спілкуватися з іншими користувачами та отримувати відгуки на роботи.

## ВСТУП

У сучасному світі, де цифрові технології займають провідну роль у всіх сферах життя, традиційні методи демонстрації власної творчості стають не такими ефективними, як онлайн-платформи, що надають доступ до великої аудиторії. Разом з цим у митців з'являється потреба в комунікації з аудиторією та взаємодією з іншими творчими людьми, проте застосунки, які наразі існують не спеціалізуються на мистецькій діяльності або не мають усі потрібні властивості.

Метою дослідження є розробка платформи для підтримки та просування митців з використанням технологій Flutter та Dart, що передбачає створення застосунку, який надасть користувачам інструменти для демонстрації своїх робіт, поширення інформації про виставки, спілкування з іншими митцями та отримання відгуків на власні роботи. Також дослідження фокусується на інтеграції сучасних технологічних рішень, а саме фреймворку Flutter, для покращення користувацького досвіду та забезпечення високої продуктивності платформи.

Робота складається з трьох розділів.

У першому розділі проведено аналіз вже існуючих альтернатив, а також сформовано вимоги до застосунку.

У другому розділі проаналізовано переваги фреймворку Flutter, мови програмування Dart та сервісу Firebase, а також досліджено актуальність використання цих технологій.

Третій розділ присвячено розробці платформи для митців. Продемонстровано структуру проекту, фрагменти коду, а також роботу застосунку.

## **1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ**

### **1.1 Опис предметної області**

Предметною областю кваліфікаційної роботи є розробка платформи для підтримки та просування митців з використанням технологій Flutter та Dart.

Соціальні мережі в сучасному світі стали не тільки місцем для спілкування, але і потужним інструментом для просування і сприяння розвитку різних галузей, включаючи мистецтво. Розробка платформи спрямована на створення місця, де митці можуть спілкуватися, ділитися своїми роботами та отримувати підтримку від однодумців.

Застосунок надасть користувачам простий і зрозумілий інтерфейс, можливість створювати особисті профілі, коментувати та лайкати пости, надсилати приватні повідомлення іншим користувачам. Одним із ключових аспектів розробки соціальної мережі для митців є підтримка творчого розвитку. Застосунок надасть можливість для публікації творчих робіт, інформації про виставки, а також портфоліо художника.

Розробка застосунку «Artsty» передбачає використання фреймворку Flutter, що є популярною платформою для розробки мобільних додатків з відкритим кодом. Flutter пропонує єдину кодову базу для створення додатків, які працюють на різних платформах. Використання цього фреймворку дозволить створити швидкий та якісний продукт з привабливим та адаптивним інтерфейсом, що приверне увагу широкого кола користувачів.

### **1.2 Огляд існуючих альтернатив**

Проведено аналіз існуючих сервісів та застосунків зі схожим чи частково схожим функціоналом. Огляд існуючих альтернатив здійснено з метою виявлення

їхніх переваг та недоліків. Для огляду було обрано застосунки з такими функціями як: публікація власних робіт та інформації про виставки. Аналіз спрямований на визначення того, які функції можуть бути важливими для цільової аудиторії, та на виявлення проблем, які можуть бути вирішені за допомогою нового додатку.

### 1.2.1 Behance

Behance – це платформа для творчих людей, особливо дизайнерів, яка надає можливість розміщувати свої творчі проекти у вигляді інтерактивних портфоліо, отримувати зворотній зв'язок від інших учасників та використовувати її для пошуку натхнення. Платформа відзначається великим розмаїттям візуального мистецтва, охоплюючи широкий спектр жанрів від фотографій і малюнків до архітектурних проектів та 3D-анімації. Користувачі можуть створювати власні портфоліо, демонструвати свої роботи, отримувати оцінку від спільноти, стати джерелом натхнення для інших, рекламувати свої творчі зусилля та здійснювати пошук потенційних клієнтів. Behance була заснована Матіасом Коріа та Скоттом Бельскі в листопаді 2005 року.[1]

Серед переваг платформи можна виділити:

- Простий та сучасний інтерфейс(Див. рисунок 1.1);
- Митці можуть отримувати фідбек та коментарі від інших користувачів Behance, що допомагає їм розвиватися та вдосконалювати свої навички;

- Платформа є безкоштовною для використання, що робить її доступною для широкого кола митців, незалежно від їхнього фінансового стану.

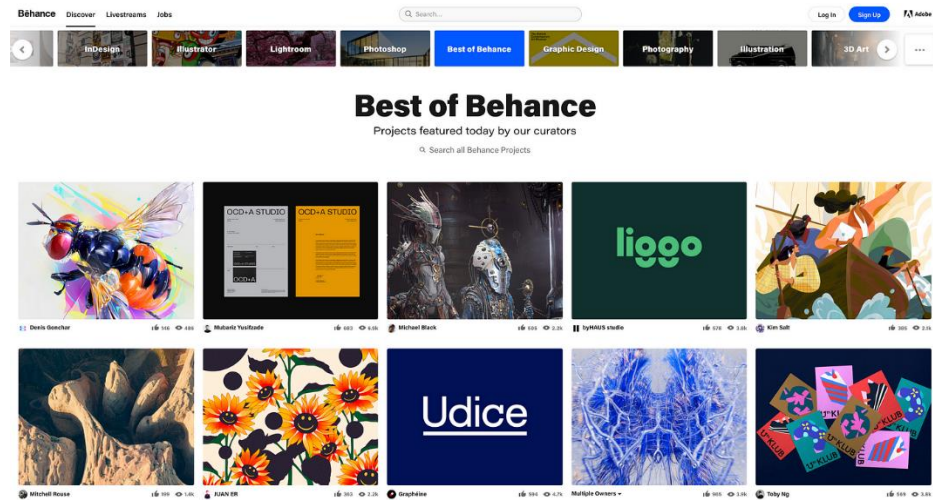


Рисунок 1.0.1 - Інтерфейс Behance

Недоліки:

- Behance більше розраховано на пошук натхнення та роботи для дизайнерів, немає функціоналу для того, щоб поділитись, наприклад, інформацією про виставки.

## 1.2.2 ArtStation

ArtStation створена як професійне середовище для цифрових художників, ілюстраторів і концепт-художників, спрямована на надання можливостей для демонстрації портфоліо, пошуку кар'єрних можливостей і взаємодії зі спільнотою професіоналів у цих сферах.[2] Платформа дозволяє використовувати зображення високої роздільної здатності, відео, короткі відеокліпи, 3D-сцени Marmoset та Sketchfab, панорами 360 градусів та інші елементи для ефективного представлення творчих робіт. Функціонал аналітики надає можливість відстежувати аудиторію, що переглядає їхні портфоліо. Крім того, користувачі можуть використовувати платформу для продажу своїх цифрових активів та принтів.

Перваги:

- Зручний інтерфейс(Див. рис. 1.2);
- Платформа є чудовим ресурсом для встановлення зв'язків із професіоналами галузі та отримання відгуків для подальшого вдосконалення вашої майстерності;
- Мобільний додаток є як на Android так і на iOS;

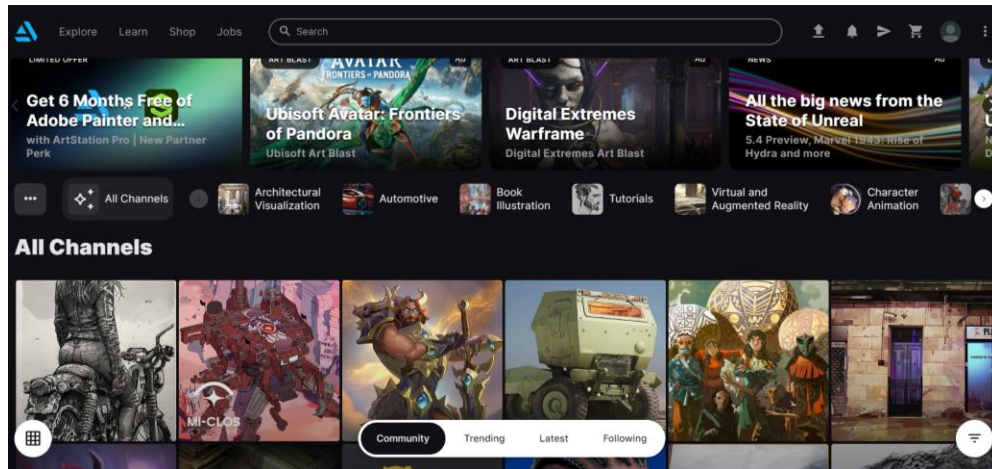


Рисунок 1.0.2 – Інтерфейс ArtStation

Недоліки:

- Платформа орієнтована на цифрових та концептуальних художників у галузі цифрового мистецтва, ігор та розваг, що може не підходити для всіх видів мистецтва.
- Деякі функції доступні лише за підпискою.
- Платформа також більше орієнтована на пошук роботи та продаж робіт.

### 1.2.3 Facebook

Facebook - це соціальна мережа, яка полегшує спілкування та обмін інформацією з родиною та друзями в Інтернеті. Спочатку розроблений для студентів коледжів, Facebook був створений у 2004 році Марком Цукербергом, коли він навчався в Гарвардському університеті. Сьогодні Facebook є найбільшою

соціальною мережею у світі, яка налічує понад 1 мільярд користувачів по всьому світу. Facebook дозволяє надсилати повідомлення і публікувати оновлення статусу, щоб підтримувати зв'язок з друзями та родиною. Також надає можливість ділитися різними типами контенту, наприклад, фотографіями та посиланнями.[3]

#### Переваги:

- Facebook має велику користувацьку базу, що дозволяє митцям залучити більше глядачів та потенційних клієнтів до своєї творчості;
- Facebook надає можливість спілкуватися з фанатами, отримувати відгуки та взаємодіяти з аудиторією;
- Створення власних сторінок, подій та рекламних кампаній дозволяє митцям ефективно просувати свої роботи та залучати увагу.

#### Недоліки:

- Платформа не створена спеціально для митців і різноманітний контент відволікає від робіт митців;
- Facebook має безліч можливостей, які не завжди підходять під потреби митців(Див. рис. 1.3).

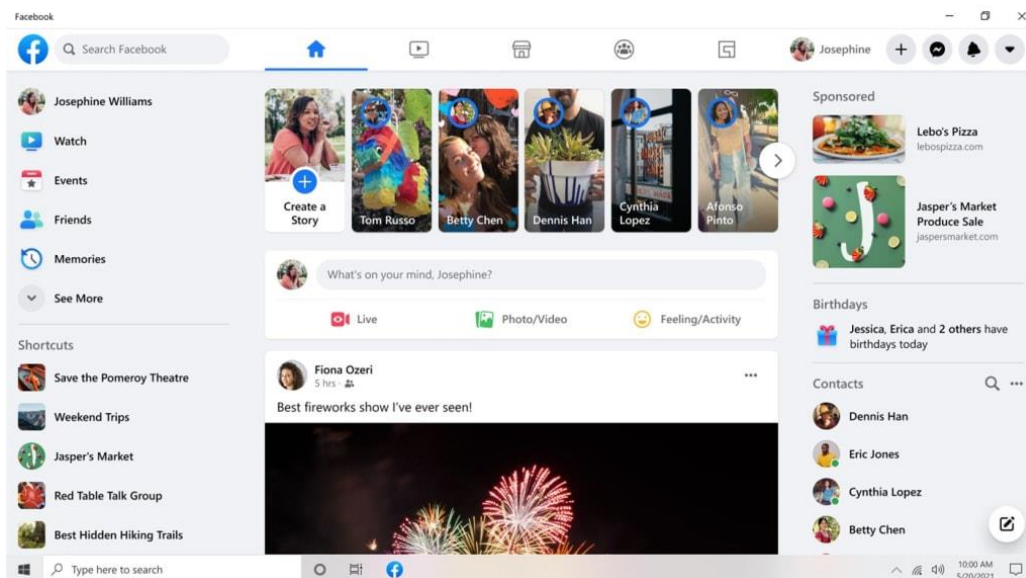


Рисунок 1.0.3 – Інтерфейс Facebook

### 1.2.4 Instagram

Instagram - це безкоштовний застосунок для обміну фотографіями та відео, доступний на iPhone та Android. Користувачі можуть завантажувати фотографії та відео і ділитися ними зі своїми підписниками або з обраною групою друзів. Вони також можуть переглядати, коментувати та вподобати публікації, якими поділилися їхні друзі в Instagram. Спочатку запущений для iOS Кевіном Сістром і Майком Крігером у жовтні 2010 року Instagram швидко набрав популярності: за два місяці було зареєстровано один мільйон користувачів, за рік – 10 мільйонів, а до червня 2018 року – 1 мільярд. [4]

Переваги:

- Застосунок зосереджений на візуальному контенті, що робить його ідеальною платформою для взаємодії митців зі своєю аудиторією(Див. рис. 1.4);
- Велика користувацька база Instagram надає митцям можливість збільшити зацікавленість до своєї творчості, отримуючи велику кількість фоловерів та взаємодіючи зі спільнотою.
- Інструменти просування Instagram надають митцям можливість розширити свою аудиторію та залучити нових фоловерів.

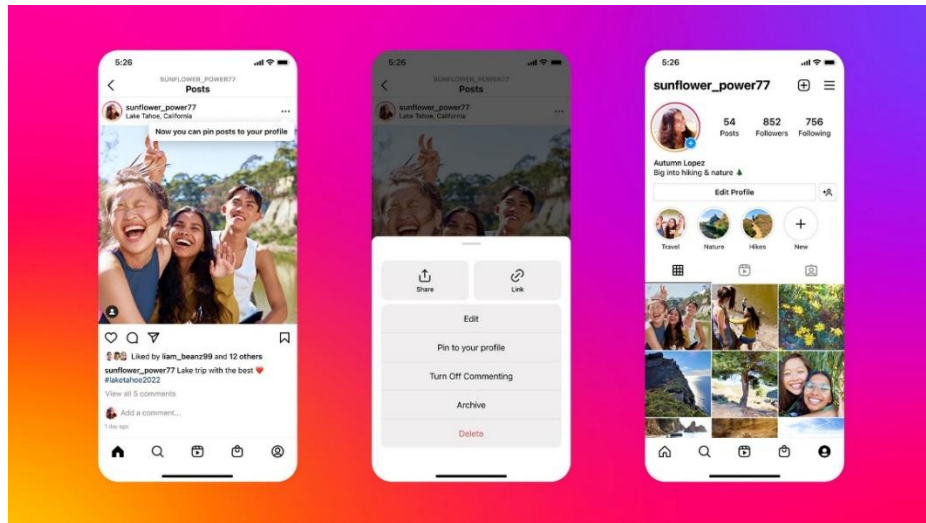


Рисунок 1.0.4 – Інтерфейс Instagram

#### Недоліки:

- Подібно до Facebook платформа не розроблена для митців і контент з різною тематикою відволікає;
- Різноманітні інструменти, що не завжди підходять під потреби митців;

### 1.2.5 Висновки

За результатами аналізу виявлено, що обрані альтернативи застосунків мають свої переваги та недоліки. Behance відзначається своїм розмаїттям візуального мистецтва та безкоштовним доступом, але не має функціоналу для публікації інформації про виставки, що може бути важливим для митців. ArtStation, зі зручним інтерфейсом і можливістю взаємодії з професіоналами галузі, спрямована переважно на цифрове мистецтво, а також має обмеження доступу до деяких функцій в залежності від підписки. Facebook і Instagram, як соціальні мережі, мають велику користувачську базу, що може привернути увагу до творчості митців. Проте, їхній різноманітний функціонал і різноманіття контенту можуть відволікати від робіт митців.

Отже, при розробці нового застосунку для публікації творчих робіт важливо врахувати потреби цільової аудиторії і проблеми, які можуть виникнути з використанням існуючих платформ. Новий додаток повинен пропонувати простий та сучасний інтерфейс, сприяти взаємодії між митцями та отриманню зворотного зв'язку, а також надавати можливості публікації інформації про виставки, що може стати конкурентною перевагою в порівнянні з існуючими альтернативами.

### **1.3 Формування вимог до застосунку**

Успіх платформи залежить від її можливості задовольнити потреби та очікування цільової аудиторії, тому для досягнення цієї мети необхідно чітко сформулювати вимоги, які охоплюють функціональність, зручність використання, продуктивність та технічні аспекти додатку.

Функціональні вимоги застосунку “Artsty” включають реєстрацію та вхід користувача, створення власного профілю, публікацію мистецьких робіт та виставок, можливість отримувати коментарі та лайки, а також спілкуватися з іншими користувачами.

Зручність у використанні застосунку – це ще одна важлива вимога. Тобто застосунок має мати інтуїтивно зрозумілий інтерфейс, за допомогою якого можна швидко зрозуміти, як ним користуватися. Інформація повинна бути структурована, щоб користувачі могли швидко знайти потрібну інформацію за кілька натискань, а також налаштована навігація між вікнами. Крім того, кольорова тема додатку повинна бути приємною для очей і не викликати дискомфорту.

Продуктивність також є важливою вимогою до застосунку. Платформа повинна працювати швидко та ефективно, має бути сумісна з різними розмірами

екранів мобільних пристроїв, щоб забезпечити комфортну роботу на будь-яких пристроях.

Виконання цих вимог грає важливу роль в успішному функціонуванні платформи, спрямованої на підтримку й просування митців. Вони визначають умови, необхідні для залучення користувачів та забезпечення їхнього задоволення від використання програми.

## 2 ОБҐРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РОЗРОБКИ

### 2.1 Flutter

#### 2.1.1 Аналіз переваг фреймворку Flutter

Наразі існує безліч фреймворків для розробки застосунків, такі як React Native, Flutter, Xamarin, Swiftic, Ionic та багато інших. Кожен з цих фреймворків має свої унікальні переваги та обмеження. Задоволення потреб користувачів, швидкість розробки і ефективність – ключові вимоги до сучасних мобільних застосунків. Вибір правильного фреймворку для розробки має вирішальне значення, і одним з найбільш перспективних варіантів є Flutter.

Flutter – це фреймворк з відкритим вихідним кодом, розроблений компанією Google, який дозволяє розробляти естетичні, нативно скомпільовані додатки для різних платформ з використанням єдиної кодової бази. Він був вперше представлений у 2015 році, але здобув значну увагу, коли Google випустив свій перший комерційний додаток Flutter у 2017 році. У 2019 році Google розширила фреймворк поза мобільні платформи, оголошуючи про можливість розробки веб застосунків за допомогою Flutter. З 2021 року Flutter продовжує розвиватися та закріплювати свою позицію як стабільний та надійний фреймворк розробки. [5]

Однією з головних переваг переваг Flutter є кросплатформеність, можливість розробки застосунків для різних операційних систем, таких як Android та iOS, а також для вебу, десктопу (Windows, macOS, Linux). Завдяки цьому розробники можуть створити один код і використовувати його на різних платформах, що заощаджує час і ресурси.

Ще однією перевагою є висока продуктивність. Проект Flutter написаний за допомогою Dart – це динамічна, об'єктно-орієнтована мова програмування загального призначення з відкритим вихідним кодом, розроблена компанією Google. Вона підтримує розробку додатків як на стороні клієнта, так і на стороні сервера і синтаксично Dart схожа на Java, C та JavaScript. Завдяки тому, що Dart

компілює код під нативну архітектуру платформи за допомогою AOT, програми виконуються швидше та зменшує затримки[6]

Також головним лозунгом Flutter є «Everything's a Widget», тобо віджет є ключовим компонентом інтерфейсу користувача у Flutter. [5] Це є однією з революційних особливостей цього продукту – здатність створювати користувацький інтерфейс за допомогою цих будівельних блоків. Порівняно з іншими підходами, які використовують різні об'єкти, у Flutter є послідовна та уніфікована модель об'єктів. Кожен віджет визначає різні структурні елементи, такі як кнопка, елементи стилю, розміщення об'єкту та інші. Завдяки двовимірному рендеринговому двигуну та вже наявним віджетам, можна легко реалізувати будь-який дизайн застосунку. До того ж, Flutter надає віджети, які відповідають дизайну матеріалу для Android і вигляду Cupertino для Apple. Налаштування користувацького інтерфейсу, яке зазвичай займає найбільше часу у кросплатформенній розробці, вимагає мінімального часу з використанням Flutter. [6]

Ще однією перевагою Flutter є гаряче перезавантаження – це функція, яка дозволяє розробникам бачити відображення змін у кодї в режимі реального часу без перезапуску програми. Flutter автоматично вносить зміни у запущену віртуальну машину Dart, щойно розробник зберігає свою роботу. [6] Функція гарячого перезавантаження значно прискорює процес розробки, дозволяючи ефективніше відлагоджувати і тестувати код у найкоротший час. Це робить процес розробки більш приємним і продуктивним для розробників.

### **2.1.2 Актуальність Flutter**

У сучасному світі швидкого розвитку та постійній зміні технологій для успішного запуску застосунку важливо також розглянути актуальність технології, у данному випадку фреймворку Flutter.

У рамках дослідження актуальності Flutter переглянуто статистику Google Trends, де можна побачити значний ріст інтересу до цього фреймворку протягом останніх кількох років. (Див. рис. 2.1) Наприклад, з 2017 року і надалі, можна відзначити поступове збільшення зацікавленості у Flutter, що свідчить про його зростаючу популярність серед розробників. Індекс зацікавленості у цьому періоді відображає стійке зростання, що може бути пов'язане з розвитком та вдосконаленням функціоналу фреймворку та з ростом кількості проектів, розроблених з його використанням. Також можна помітити, що з 2020 року зацікавленість у Flutter переважає зацікавленість у інших фреймворках.

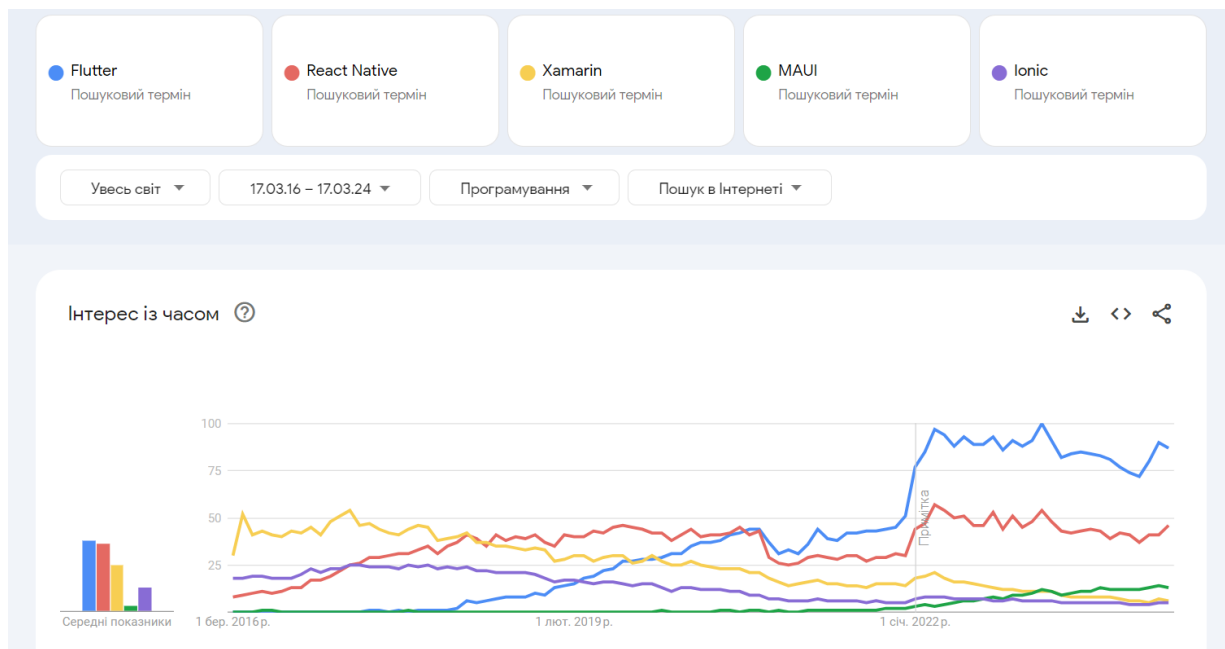


Рисунок 2.1 – Статистика Google Trends

Згідно зі статистикою Stack Overflow (Див. рис. 2.2) – одного з найбільших ресурсів для розробників, де вони обмінюються знаннями, задають та відповідають на питання, а також діляться досвідом у різних технологічних галузях, Flutter знову таки продемонстрував стабільне зростання популярності серед розробників за останні роки. Кількість пошукових запитів та тегів, пов'язаних з Flutter постійно збільшується і переважає кількість запитів пов'язаних з іншими фреймворками.

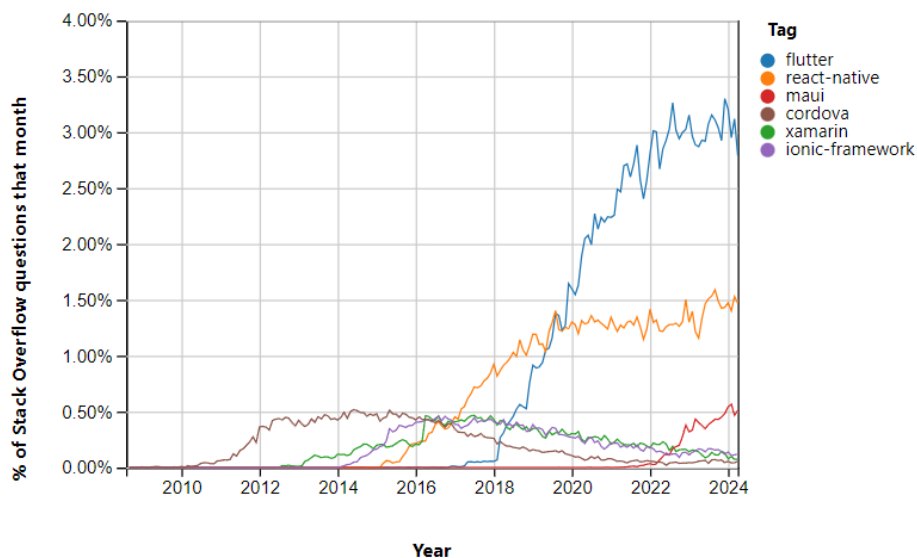


Рисунок 2.2 – Статистика Stack Overflow

Також за результатами опитування, проведеного серед розробників на платформі Stack Overflow у 2023 році Flutter займає 9 місце за популярністю і випереджає React Native.(Див. рис. 2.3)

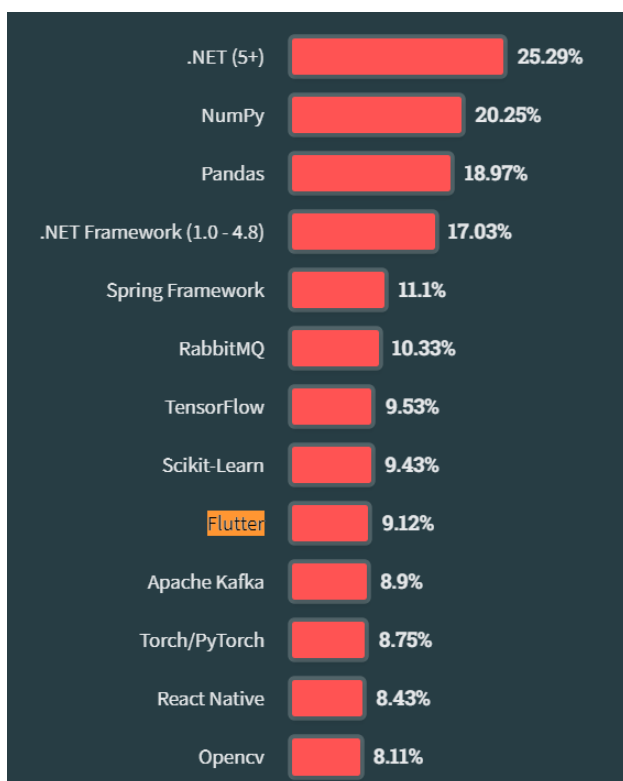


Рисунок 2.3 – Результат опитування Stack Overflow у 2023 (усі відповіді)

Серед професійних розробників Flutter займає 8 місце за популярністю і знову ж таки трохи випереджає React Native.(Див. рис. 2.4)

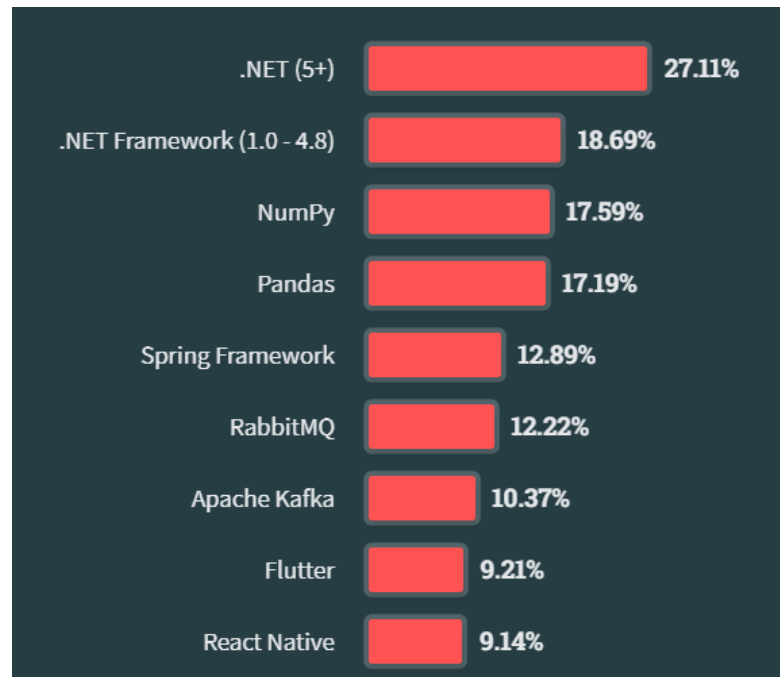


Рисунок 2.4 - Результат опитування Stack Overflow у 2023 (відповіді професіоналів)

А також входить у топ 5 фреймворків для вивчення.(Див.рис. 2.5)

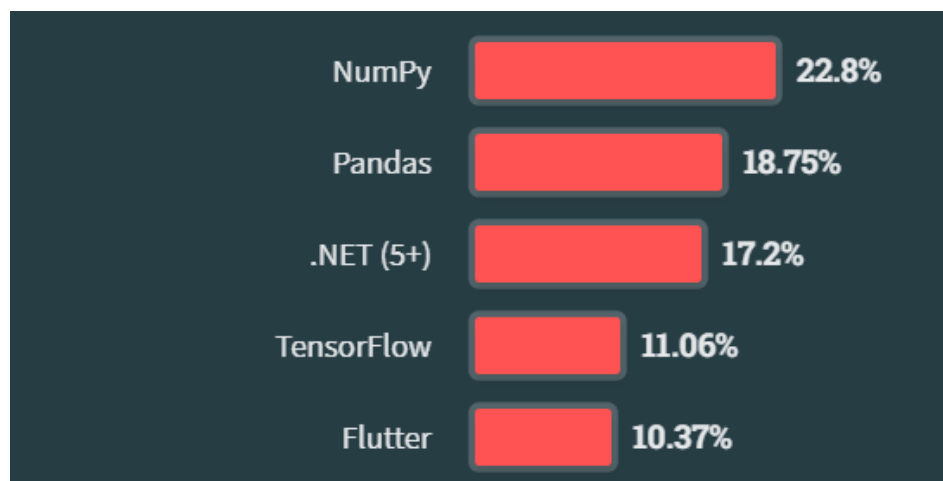


Рисунок 2.5 - Результат опитування Stack Overflow у 2023 (відповіді стосовно вивчення коду)

Таким чином, наведене дослідження підтверджує, що фреймворк Flutter є актуальним та перспективним інструментом для розробки кросплатформних додатків у сучасній індустрії програмного забезпечення. Його популярність та

широке використання відображають рівень довіри та зацікавленості спільноти розробників, а також підтверджують його здатність задовольняти вимоги різних сегментів ринку програмного забезпечення.

## 2.2 Firebase

Firebase - це комплексний набір інструментів і сервісів, які пропонуються як платформа BaaS, що дозволяє розробникам легко створювати, запускати та розширювати мобільні та веб-додатки. Вона надає базу даних у реальному часі, аутентифікацію, аналітику, хостинг та інші функції, всі керовані з єдиної платформи.[7]

За своєю суттю Firebase є базою даних, яка синхронізує дані на всіх підключених пристроях в режимі реального часу і зберігає їх у форматі JSON, який не використовує запити для вставки, оновлення, видалення або додавання даних. Розробники отримують більшу користь від використання такої платформи, оскільки їм не потрібна підтримка складних серверних служб. Firebase надає набір інструментів для роботи з базами даних у реальному часі для різних платформ, включаючи Android, iOS та Web, що дозволяє розробникам зберігати інформацію у гнучкий та масштабований спосіб. [8]

До того ж Firebase надає служби автентифікації, що дозволяють розробникам з легкістю впроваджувати безпечну автентифікації користувачів у своїх додатках. Ці служби підтримують різні методи автентифікації, такі як електронна пошта/пароль, номер телефону, а також популярні сторонні сервіси, зокрема Google, Facebook і Twitter. [9]

Також платформа надає набір інструментів для аналізу використання та продуктивності застосунків, включаючи моніторинг продуктивності, звіти про збої та віддалене налаштування. Ці інструменти допомагають у вирішенні проблем з продуктивністю та покращенню загального користувацького досвіду. [9]

Додатковою перевагою Firebase є безпечний та швидкий хостинг. Хостинг Firebase підтримує всі типи контенту, включаючи веб-додатки, динамічний та статичний контент. Щоб захистити свій власний домен від зовнішніх загроз, також корисно використовувати безкоштовну SSL-сертифікацію Firebase. Крім того, Firebase CLI допомагає програмістам запустити свій додаток за лічені секунди.

Підсумовуючи, Firebase пропонує потужну та універсальну платформу для створення та розвитку сучасних застосунків. Великий набір функцій, включаючи базу даних у реальному часі, автентифікацію, хостинг і аналітику, що робить цю платформу популярним вибором серед розробників.

## 2.3 Android

Хоча Flutter – це кросплатформенна платформа для розробки обрано створити застосунок саме для Android. Таке рішення зумовлене тим, що Android є однією з найпопулярніших мобільних платформ у світі з великою базою користувачів. За статистикою StatCounter(Див. Рис. 2.6) , приблизно 71% мобільних пристроїв по всьому світу використовують операційну систему Android, приблизно 28% використовують IOS, інші операційні системи займають менше 1%.

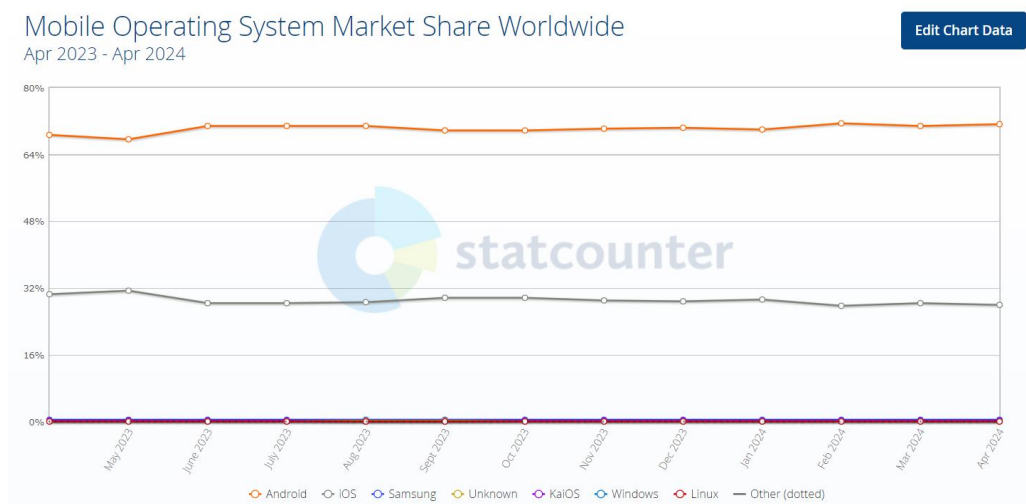


Рисунок 2.6 – Статистика StatCounter використання операційних системи на мобільних пристроях по всьому світу

Розробка застосунку для платформи Android дозволить охопити максимально велику аудиторію.

## 3 РОЗРОБКА ЗАСТОСУНКУ

### 3.1 Структура проекту

З метою зробити проект більш читабельним, а також мінімізувати непередбачувану поведінку реалізовано архітектуру MVVM.

MVVM — це архітектурний шаблон, який чітко відокремлює бізнес-логіку програми від інтерфейсу користувача. Винайдений MVVM архітекторами Microsoft Кеном Купером і Тедом Пітерсом.

Архітектура MVVM ділить програму на три рівні: Model, View та ViewModel(Див. Рис. 3.1):

- Model відображає дані та бізнес-логіку застосунку. Вона представляє основні джерела даних, такі як бази даних або веб-сервіси, і виконує такі операції, як пошук, обробка та перевірка даних. Таким чином дані залишаються відокремленими від користувацького інтерфейсу, що сприяє гнучкості та масштабованості.
- ViewModel діє як посередник між Model та View. Вона надає дані та поведінку для View, дозволяючи йому безпосередньо зв'язуватися з властивостями та командами ViewModel. ViewModel абстрагується від складнощів моделі і надає спрощений інтерфейс до View.
- View відповідає за відображення користувацького інтерфейсу кінцевому користувачеві. Він отримує вхідні дані від користувача і представляє дані, надані ViewModel. У контексті MVVM, View повинен бути якомога легшим, з мінімальною логікою. Такий підхід полегшує супровід і дозволяє дизайнерам і розробникам одночасно працювати над різними аспектами програми.

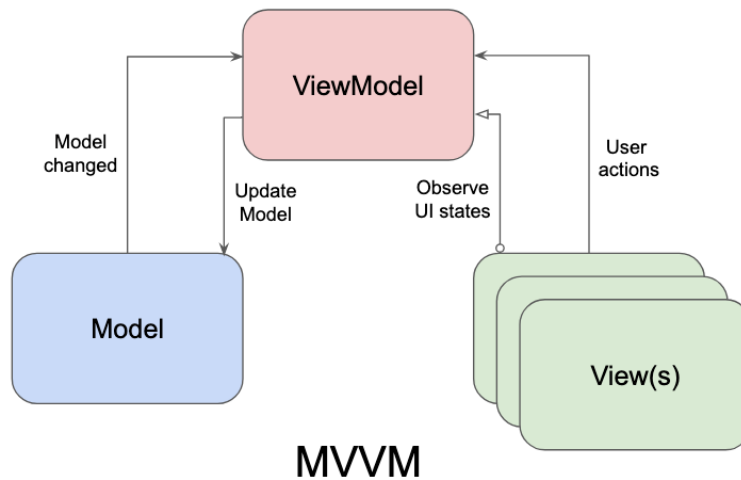


Рисунок 3.1 – Архітектура MVVM

У папці Models знаходяться файли, що відповідають за представлення моделі в базі даних Firebase. У Firebase таблиці називаються колекціями, а записи в них документами. Документи містять поля з різними типами даних. В проєкті використовуються такі моделі: «UserModel», «PostModel», «ActivityModel», «MessageModel», «ExhibitionModel», «CommentModel».

Модель «UserModel» містить інформацію про користувача. Поля до цієї моделі наведено в таблиці нижче.

Назва поля	Тип поля	Опис
username	String	ім'я користувача
email	String	електронна адреса користувача
photoUrl	String	посилання на зображення профілю користувача
country	String	країна, у якій проживає користувач
donateURL	String	посилання на сайт, де можна надіслати кошти користувачу
portfolioURL	String	посилання на сайт з портфолію користувача
id	String	id користувача
signedUpAt	Timestamp	дата та час, коли користувач був зареєстрований

lastSeen	Timestamp	дата та час, коли користувач був онлайн останній раз
isOnline	bool	перевірка чи користувач онлайн

Таблиця 3.1 – Поля UserModel

Модель «PostModel» містить інформацію про пости, тобто мистецькі роботи.

Поля до цієї моделі наведено в таблиці 3.2.

Назва поля	Тип поля	Опис
postId	String	id посту
ownerId	String	id акаунту, який виклав допис
username	String	ім'я акаунту, який виклав допис
location	String	локація, з якої виклали допис
nameOfArtwork	String	назва мистецької роботи
typeOfArt	String	тип мистецтва
style	String	стиль мистецької роботи
materials	String	матеріали, які використані при створенні роботи
size	String	розмір мистецької роботи
mediaUrl	String	посилання на зображення, у базі даних
year	int	рік створення мистецької роботи
timestamp	Timestamp	час, коли допис виклали

Таблиця 3.2 – Поля PostModel

Тут потрібно виділити поля, які відповідають саме за опис мистецьких робіт, обрано саме назва, тип, стиль мистецької роботи, матеріали, які використані при створенні роботи, а також розмір та рік створення мистецької роботи, оскільки саме такі короткі містять корисну інформацію для колекціонерів, галерей, а також звичайних покупців. Подібну характеристику роботи можна побачити на сайті онлайн-архіву PinchukArtCentre[10], а також на сайті Центральний державний архів-музей літератури і мистецтва України[11].

Наступна модель «ActivityModel» містить інформацію про активність користувача для сповіщень, а саме підписка на іншого користувача, коментар або лайк допису. Всі поля до цієї моделі наведено у таблиці 3.3.

Назва поля	Тип поля	Опис
type	String	тип активності
userId	String	id акаунту
username	String	ім'я акаунту
userDp	String	посилання на зображення профілю користувача в firebase
postId	String	id допису
mediaUrl	String	посилання на зображення допису в firebase
commentData	String	текст коментаря
timestamp	Timestamp	час створення активності

Таблиця 3.3 – Поля ActivityModel

Модель «MessageModel» містить в собі інформацію про повідомлення, які надсилають користувачі один одному. Поля до цієї моделі наведено нижче.

Назва поля	Тип поля	Опис
content	String	текст повідомлення
senderUid	String	Uid відправника
messageId	String	id повідомлення
time	Timestamp	час створення повідомлення

Таблиця 3.4 – Поля MessageModel

Також модель «ExhibitionModel» містить інформацію про виставки, які додають користувачі. Поля моделі наведено у таблиці 3.5.

Назва поля	Тип поля	Опис
exhibitionId	String	id виставки
ownerId	String	id користувача
username	String	ім'я користувача

exhibitionname	String	назва виставки
location	String	місце проведення
description	String	коротки опис
startDate	String	дата початку
endDate	String	дата кінця
moreInfoUrl	String	посилання, де можна знайти більше детальної інформації
timestamp	Timestamp	час створення

Таблиця 3.5 – Поля ExhibitionModel

Остання модель – це «CommentModel», яка містить інформацію про коментарі.

Поля цієї моделі наведено у таблиці 3.6.

Назва поля	Тип поля	Опис
username	String	ім'я користувача
comment	String	текст коментаря
timestamp	Timestamp	час створення
userDp	String	посилання на зображення профілю користувача в firebase
userId	String	id користувача

Таблиця 3.6 – Поля CommentModel

У папці View розміщені файли, які відповідають за відображення інтерфейсу користувача, тобто код для віджетів та екранів застосунку. Для прикладу розглянемо код меню навігації у застосунку на рисунку 3.2 зображено його візуалізацію.

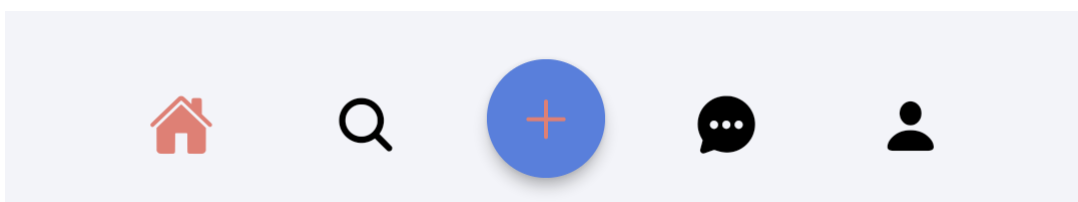


Рисунок 3.2 – Меню навігації

Row(

*mainAxisSize: MainAxisSize.max,*

```

mainAxisAlignment: MainAxisAlignment.spaceBetween,
crossAxisAlignment: CrossAxisAlignment.center,
children: [
  SizedBox(width: 5),
  for (Map item in pages)
    item['index'] == 2
      ? Container(
        height: 45.0,
        width: 45.0,
        // ignore: missing_required_param
        child: FabContainer(
          icon: Ionicons.add_outline,
          mini: true,
        ),
      )
    : Padding(
        padding: const EdgeInsets.only(top: 5.0),
        child: IconButton(
          icon: Icon(
            item['icon'],
            color: item['index'] != _page
              ? Colors.black
              : Theme.of(context).colorScheme.secondary,
            size: 25.0,
          ),
          onPressed: () => navigationTapped(item['index']),
        ),
      ),
    ),
  ),

```

```

        SizedBox(width: 5),
        |,
        ),

```

Цей код створює рядок (Row) з кнопками для навігації. На початку створено `SizedBox`, який додає горизонтальний поміжок шириною 5 пікселів. Відбувається ітерація по списку сторінок і якщо це не 2 сторінка створюється кнопка іконка, в іншому випадку спеціальний контейнер з плаваючою дією (`FabContainer`). Також змінюється колір обраної сторінки, щоб користувачу було зрозуміло, на якій саме сторінці він зараз перебуває.

У папці `ViewModels` файли, що забезпечують зв'язок між моделлю даних (`Model`) і відображенням (`View`).

Для прикладу розглянемо вхід у додаток. У лістингу написана функція `auth.loginUser`, яка перевіряє чи існує такий користувач за електронною поштою і паролем. Якщо вся інформація підходить, відбувається вхід у додаток і перехід на головний екран `TabScreen`, в іншому випадку, з'являється помилка.

```

try {
  bool success = await auth.loginUser(
    email: email,
    password: password,
  );
  if (success) {
    Navigator.of(context)
      .pushReplacement(CupertinoPageRoute(builder: (_) => TabScreen()));
  }
} catch (e) {
  loading = false;
  notifyListeners();
  print(e);
}

```

```
showInSnackBar(  
    '${auth.handleFirebaseAuthError(e.toString())}', context);  
}
```

### 3.2 Демонстрація роботи платформи

Відкривши додаток вперше перед користувачем з'являється екран з кнопка «Login» та «Sign up». (Див. рис. 3.3)

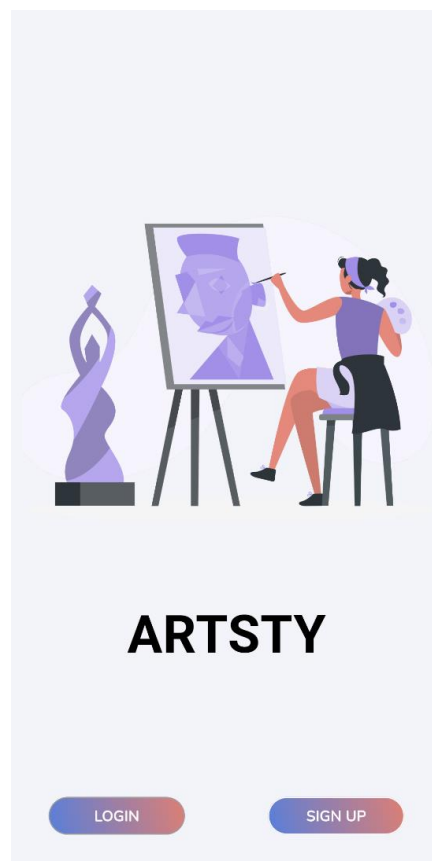


Рисунок 3.3 – Перший екран застосунку

Якщо користувач не зареєстрований, то він може це зробити, натиснувши кнопку «Sign up», після чого з'явиться екран реєстрації. Якщо ж користувач вже зареєстрований, то він може натиснути кнопку «Login» і увійти до свого акаунту, ввівши електронну пошту та пароль. Якщо користувач забув свій пароль для входу,

то він може натиснути на кнопку «Forgot Password?», і йому на електронну адресу прийде лист для скидання паролю. (Див. рис. 3.4)

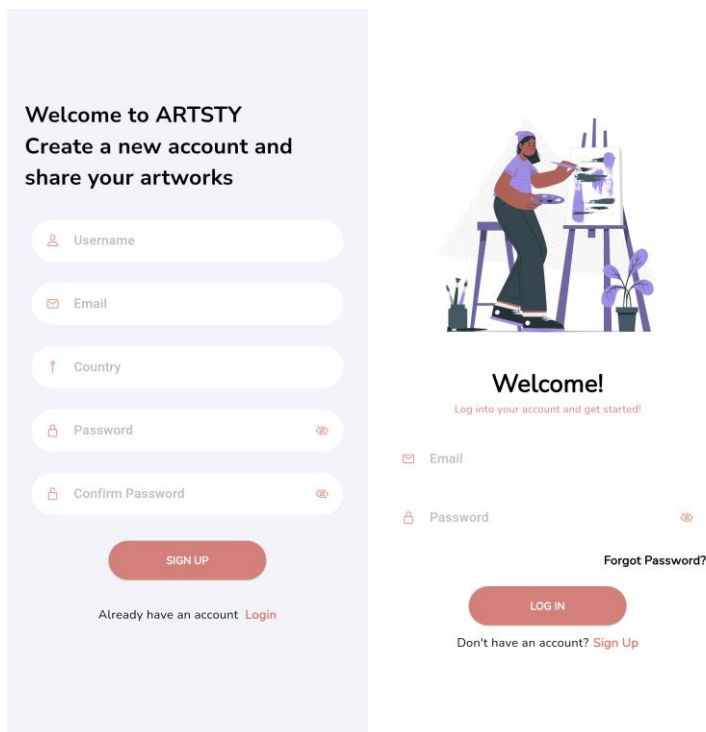


Рисунок 3.4 – Екран реєстрації(зліва) та екран входу(справа)

Після входу першим відкривається вікно з усіма дописами та доступними виставками. (Див. рис. 3.5)

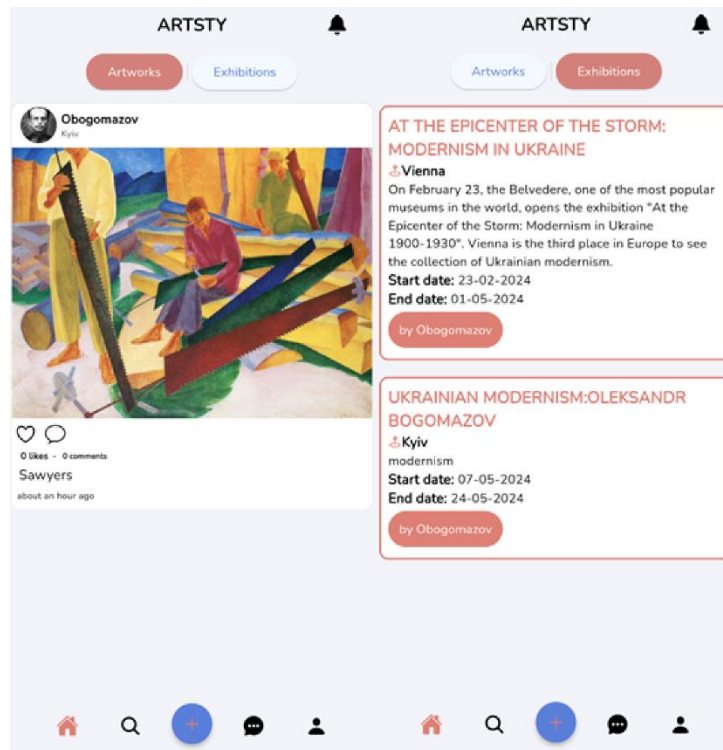


Рисунок 3.5 – Екран, де відображаються усі мистецькі роботи та виставки

Якщо натиснути на картку виставки відкриється браузер і посилання з більш детальною інформацією, якщо натиснути на допис, тоді відкриється екран з більш детальною інформацією про мистецький твір. Також можна вподобати мистецьку роботу, натиснувши на сердечко, і відкрити екран коментарів натиснувши на іконку повідомлення. (Див. Рис. 3.6)

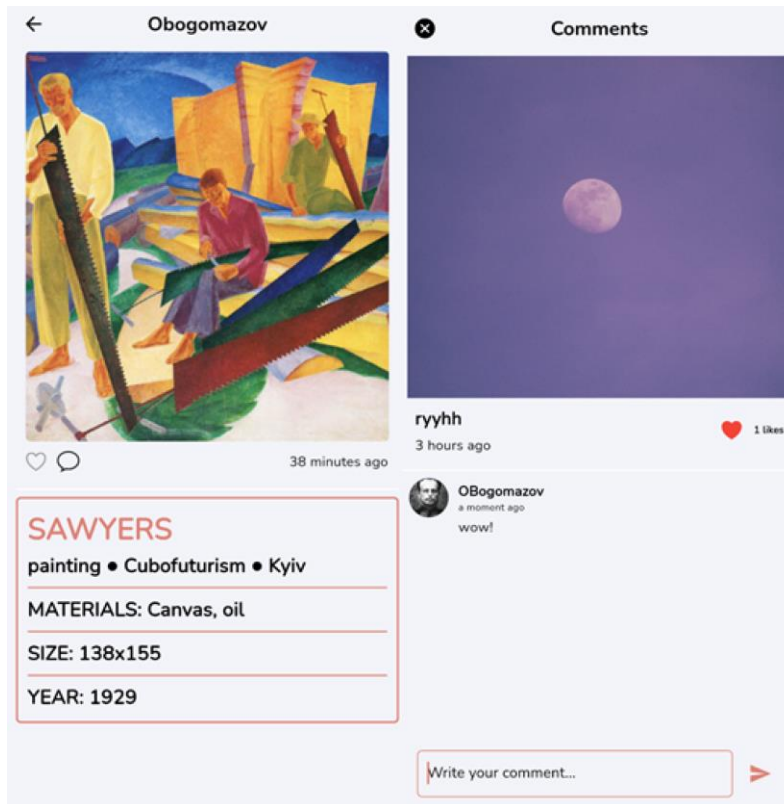


Рисунок 3.6 – Екран з детальною інформацією про мистецьку роботу(зліва), екран коментарів(справа)

При натисненні на кнопку лупи, в меню навігації, відкривається екран пошуку, де можна шукати користувачів, мистецькі роботи та виставки. Пошук користувачів відбувається за іменем, пошук мистецьких робіт за назвою, стилем, типом мистецтва, використаними матеріалами та локацією, а пошук виставок за локацією та назвою. Також поруч з користувачами під час пошуку є кнопка «Message», яка відкриває чат з обраним користувачем.(Див. рис. 3.7)

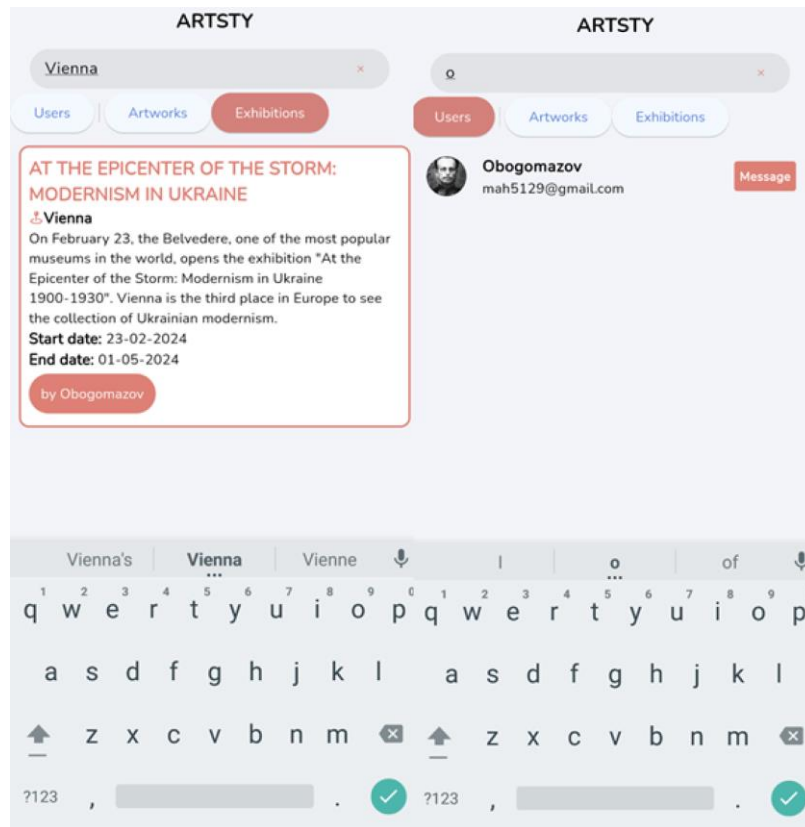


Рисунок 3.7 – Екран пошуку (зліва продемонстровано пошук виставок, справа пошук користувачів)

У чаті користувачі можуть обмінюватися текстовими повідомленнями. Також можна знайти всі чати користувача, натиснувши на іконку повідомлення у меню навігації.(Див. рис. 3.8)

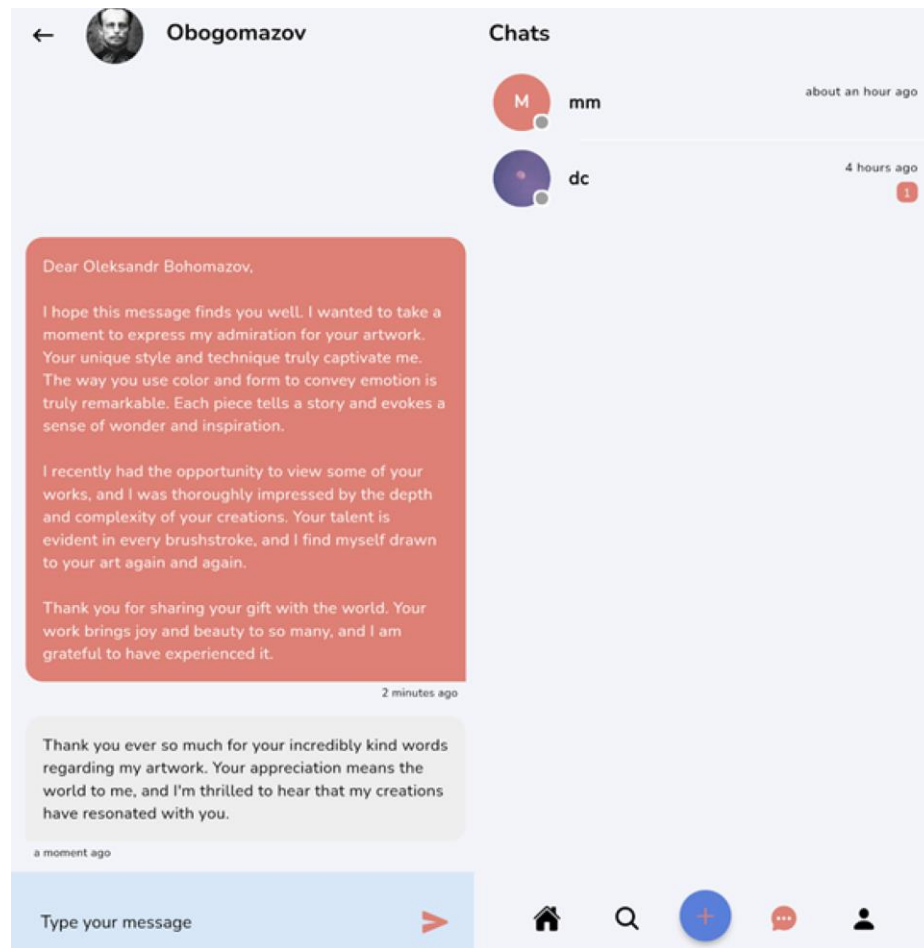


Рисунок 3.8 – Зліва екран чату з користувачем, справа екран нещодавніх чатів

У профілі користувача відображається імя користувача, країна, фото профілю і посилання на портфолію, за яким можна перейти, натиснувши «Portfolio». Також можна побачити кількість підписок та підписників, мистецькі роботи і виставки, опубліковані користувачем. Якщо митець додав посилання у своєму профілі на сервіс, де можна надіслати кошти для підтримки, при підписці на користувача з'явиться кнопка «Donate».(Див.рис. 3.9)

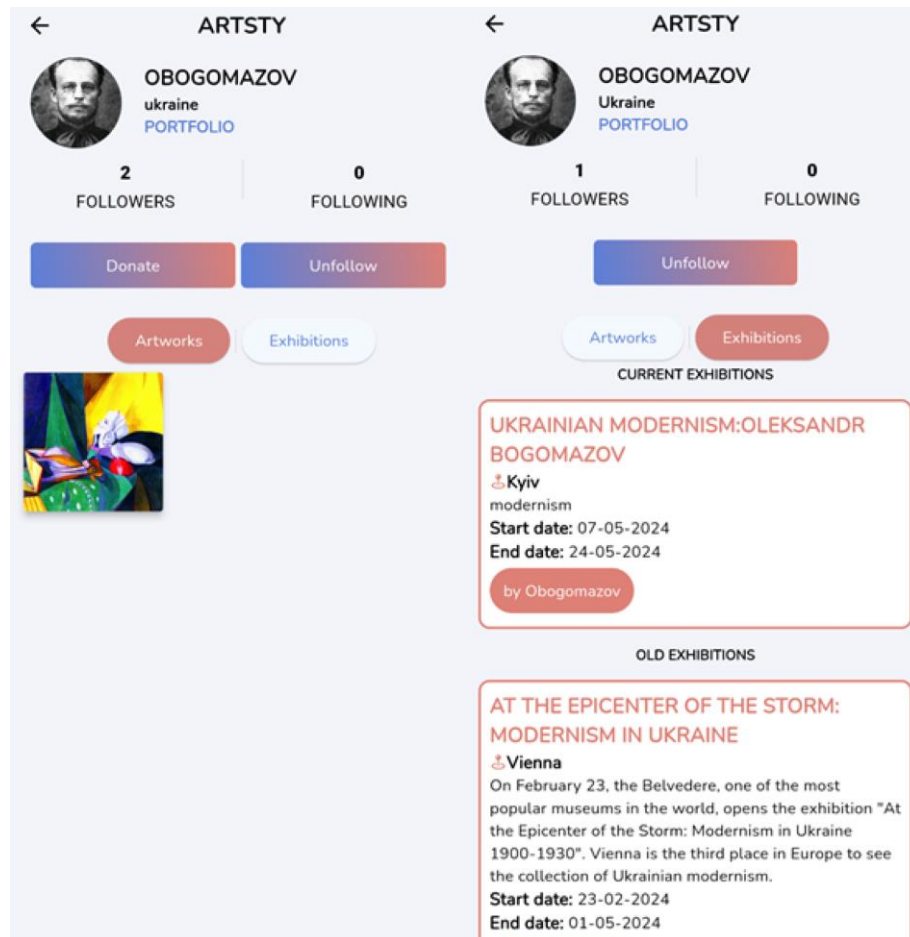


Рисунок 3.9 – Екран профілю користувача

Також виставки користувача поділяються на старі та ті, що проходять зараз, тобто завжди можна переглянути чи виставляє свої роботи зараз десь улюблений митець. При натисненні на кількість підписників та підписок можна переглянути список відповідно підписників та підписок.(Див. Рис. 3.10)

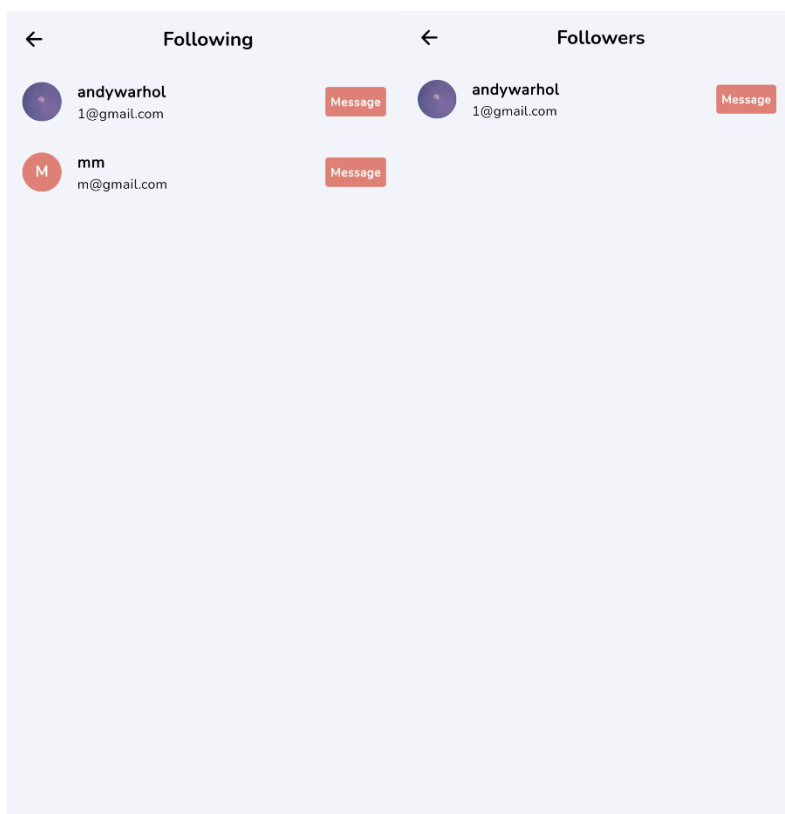


Рисунок 3.10 – Список підписок та підписників відповідно

Профіль користувача можна редагувати і додавати або прибирати посилання на портфолію та сервіс для грошової підтримки.(Див. рис. 3.11)

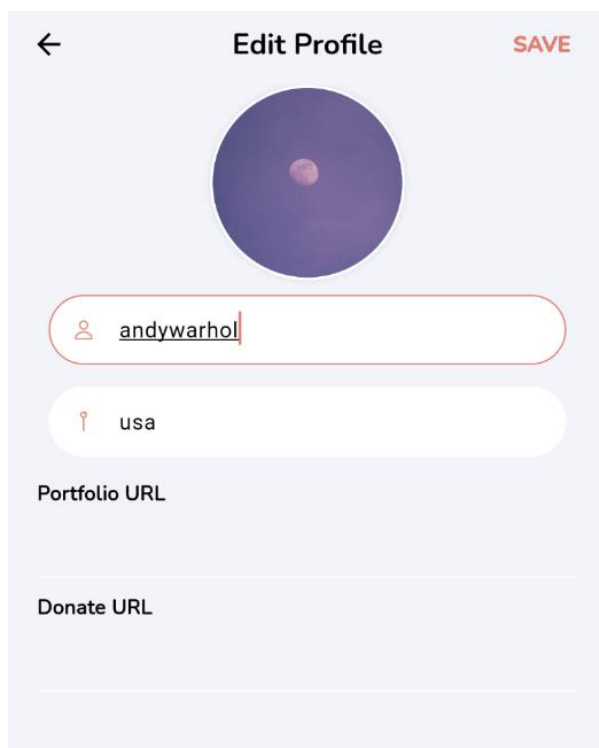


Рисунок 3.11 – Екран редагування профілю користувача

При натисненні на центральну клавішу зі знаком «+» в меню навігаціїї можна створити допис з мистецькою роботою або допис про виставку. Інтерфейс створення мистецької роботи наведено на рисунку 3.12

× ARTSTY POST

andywarhol  
1@gmail.com

Upload a Photo

ARTWORK NAME

TYPE

STYLE

MATERIALS

ARTWORK NAME

Upload a Photo

ARTWORK NAME

TYPE

STYLE

MATERIALS

SIZE  
Ex: 113x150 cm

YEAR

LOCATION

Рисунок 3.12 – Екран додавання мистецької роботи

Інтерфейс створення виставки наведено на рисунку 3.13

× ARTSTY POST

andywarhol  
1@gmail.com

EXHIBITION NAME

DESCRIPTION

URL FOR MORE INFORMATION

📅 Enter Start date

📅 Enter End date

LOCATION

Рисунок 3.13 – Екран додавання виставки

## ВИСНОВКИ

Отже, в роботі проведено аналіз предметної області, проаналізовано існуючі альтернативи, а саме «Behance», «ArtStation», «Facebook», «Instagram», а також сформовано функціональні вимоги до платформи.

Розглянуто фреймворк Flutter, мову програмування Dart та платформу Firebase, їх переваги, а також актуальність використання цих технологій на сьогоднішній день.

Результатом кваліфікаційної роботи є платформа для просування та підтримки митців, яка надає користувачам можливість публікувати власні мистецькі роботи, інформацію про виставки, а також спілкуватися та отримувати відгуки.

Перспективами розвитку платформи є розширення на інші операційні системи такі як IOS, Windows, MacOS, що можна реалізувати завдяки кросплатформеності Flutter, інтеграція онлайн-магазину або аукціонів, де митці зможуть також продавати свої роботи і створення алгоритмів рекомендацій, які покращать користувацький досвід і допоможуть знайти тих, митців, що будуть більш цікаві конкретному користувачу.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] URL: <https://www.behance.net/>
- [2] URL: <https://www.artstation.com/>
- [3] URL: <https://www.facebook.com/>
- [4] Instagram. Instagram. URL: <https://www.instagram.com/>
- [5] Flutter - Build apps for any screen. Flutter - Build apps for any screen. URL: <https://flutter.dev/>
- [6] Dagne, L. (2019). Flutter for cross-platform App and SDK development.
- [7] Osman J. What is Firebase? | AppMaster. AppMaster - The No-Code platform for building web & mobile apps. URL: <https://appmaster.io/blog/what-is-firebase> (дата звернення: 15.05.2024).
- [8] Khawas C., Shah P. Application of Firebase in Android App Development-A Study. International Journal of Computer Applications. 2018. Vol. 179, no. 46. P. 49–53. URL: <https://doi.org/10.5120/ijca2018917200>
- [9] Firebase - Introduction - GeeksforGeeks. GeeksforGeeks. URL: <https://www.geeksforgeeks.org/firebase-introduction/>
- [10] URL: <https://archive.pinchukartcentre.org/>
- [11] ЦДАМЛМ – ЦДАМЛМ України Центральний державний архів-музей літератури і мистецтва України. ЦДАМЛМ – ЦДАМЛМ України Центральний державний архів-музей літератури і мистецтва України. URL: <https://csamm.archives.gov.ua/>

**Додаток А**  
(обов'язковий)

Перелік прийнятих скорочень

AOT – Ahead-of-time

BaaS – Backend-as-a-Service

JSON – JavaScript Object Notation

MVVM – Model-View-ViewModel