

Міністерство освіти і науки України
Кам'янець-Подільський національний університет імені Івана Огієнка
Фізико-математичний факультет
Кафедра комп'ютерних наук

Дипломна робота

бакалавра

**з теми: «РОЗРОБКА КРОСПЛАТФОРМЕННОГО ЗАСТОСУНКУ
“ПОМІЧНИК РИБАЛКИ”»**

Виконав: студент 4 курсу,
групи КН1-В18
спеціальності 122 Комп'ютерні науки
Стасюк Валерій Русланович

Керівник: Слободянюк Олександр
Васильович
старший викладач кафедри
комп'ютерних наук, доцент,
кандидат технічних наук

Рецензент: Поведа Руслан
Анатолійович
кандидат фізико-математичних наук,
доцент, доцент кафедри фізики

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1. Розробка мобільних додатків за допомогою фреймворку Flutter на мові Dart	4
1.1 Знайомство з Dart.....	4
1.1.1 Мова Dart.....	5
1.1.2 Dart - Бібліотеки.....	6
1.1.3 Dart - Платформи	7
1.1.4 Dart Native (машинний код JIT і AOT).....	8
1.2 Вступ до Flutter	9
1.2.1 Що таке Flutter.	9
1.2.2 Flutter 2.....	10
1.2.3 Плюси розробки Flutter	11
1.2.4 Слабкі сторони Flutter	14
РОЗДІЛ 2. РОЗРОБКА КРОСПЛАТФОРМЕННОГО ЗАСТОСУНКУ «ПОМІЧНИК РИБАЛКИ».....	15
2.1 Постановка задачі	15
2.2 Розробка домашньої сторінки додатку.....	16
2.3 Розробка сторінки з відео.....	24
2.4 Реалізація сторінки «Чат»	33
2.5 Розробка сторінки «Лайфхаки».....	41
ВИСНОВКИ.....	50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	51
ДОДАТКИ.....	52

ВСТУП

Актуальність роботи. Відома істина говорить, що можна безкінечно дивитися на три речі: як тече вода, як горить вогонь і як працює інша людина. Якщо перефразувати вислів про воду і працю, то можна констатувати, що справжній рибалка може безкінечно дивитися, як ловиться риба. З давніх часів було помічено позитивний вплив риболовлі на здоров'я та настрій. І наукою доведено, що коли людина у вихідні вирушає на риболовлю, то їй гарантоване хороше самопочуття протягом наступного робочого тижня. Перегляд фотографій з видатними трофеями багато років викликатиме лише позитивні спогади та емоції.

Іноді, не володіючи всією інформацією, риболови можуть потрапити у халепу. Наприклад, розмита дорога, ті чи інші непередбачувані ситуації, або ж, через впровадження різноманітних заборон на вилов водних біоресурсів – порушити закон. Саме через це було прийнято рішення розробити додаток, зайшовши в який ви зможете отримувати різноманітну корисну інформацію.

Метою роботи – створення повноцінного кроссплатформеного додатку з корисною та необхідною інформацією для риболовів.

Об'єктом дослідження – технології створення кроссплатформеного застосунку, підтримки серверної частини та збір необхідної корисної інформації.

Предмет дослідження – flutter та dart в якості кроссплатформеної розробки.

Структура роботи. Дипломна робота бакалавра складається зі вступу, двох розділів, списку використаних джерел та додатків.

РОЗДІЛ 1. РОЗРОБКА МОБІЛЬНИХ ДОДАТКІВ ЗА ДОПОМОГОЮ ФРЕЙМВОРКУ FLUTTER НА МОБІ DART

1.1 Знайомство з Dart

Dart — це оптимізована для клієнта мова для розробки швидких програм на будь-якій платформі. Його мета — запропонувати найпродуктивнішу мову програмування для багатоплатформної розробки в поєднанні з гнучкою платформою виконання для фреймворків додатків. На *Рис.1.* представлений логотип мови Dart.



Рис. 1. Логотип мови Dart

Мови визначаються їх технічними можливостями — виборами, зробленими під час розробки, які формують можливості та їхні сильні сторони. Dart розроблено для технічного пакета, який особливо підходить для розробки і надає високоякісний виробничий досвід в широкому діапазоні цілей компіляції (веб, мобільний і настільний).

Dart також є основою Flutter. Dart це мова програмування, що забезпечує роботу програм Flutter.

1.1.1 Мова Dart

Мова Dart безпечна для типізації. Вона використовує статичну перевірку типу, щоб переконатися, що значення змінної завжди збігається зі статичним типом змінної. Хоча типи є обов'язковими, анотації типу є необов'язковими через їхнє виведення. Система введення Dart також гнучка, що дозволяє використовувати динамічні типи у поєднанні з перевітками під час виконання, що може бути корисно під час експериментів або для коду, який має бути особливо динамічним.

Dart пропонує надійну NULL-безпеку, тобто значення не можуть бути NULL, якщо ви не задасте, що вони можуть бути такими. Завдяки надійній NULL-безпеці Dart може захистити вас від NULL винятків під час виконання за допомогою статичного аналізу коду.

1.1.2 Dart - Бібліотеки

Dart має багатий набір основних бібліотек, які забезпечують необхідне для багатьох щоденних завдань програмування:

- Вбудовані типи, колекції та інші основні функції для кожної програми Dart (dart:core)
- Більш багаті типи колекцій, такі як черги, зв'язані списки, хеш-карти та двійкові дерева (dart:collection)
- Кодери та декодери для перетворення між різними представленнями даних, включаючи JSON та UTF-8 (dart:convert)
- Математичні константи, функції та генерація випадкових чисел (dart:math)
- Підтримка файлів, сокетів, HTTP та іншого введення-виведення для не веб-програм (dart:io)
- Підтримка асинхронного програмування з такими класами, як Future і Stream (dart:async)
- Списки, які ефективно обробляють дані фіксованого розміру (наприклад, беззнакові 8-байтові цілі числа) і числові типи SIMD (dart:typed_data)
- Інтерфейси сторонніх функцій для взаємодії з іншим кодом, який представляє інтерфейс у стилі C (dart:ffi)
- Одночасне програмування з використанням ізоляторів (dart:isolate)
- Елементи HTML та інші ресурси для веб-додатків, які повинні взаємодіяти з браузером і об'єктною моделлю документа (DOM) (dart:html)
- Крім основних бібліотек, багато API надаються через повний набір пакетів.

1.1.3 Dart - Платформи

Технологія компілятора Dart дозволяє запускати код різними способами:

- Вбудована платформа: для додатків, орієнтованих на мобільні та настільні пристрої, Dart включає в себе, як віртуальну машину Dart з компіляцією «just-in-time» (JIT), так і компілятор «ahead-on-time» (AOT) для створення машинного коду.

- Веб-платформа: для програм, орієнтованих на Інтернет, Dart включає як компілятор часу розробки (dartdevc), так і компілятор часу виробництва (dart2js). Обидва компілятори перекладають Dart на JavaScript.

Детальне представлення кросплатформенної роботи Dart можна розглянути на *Рис.2*.

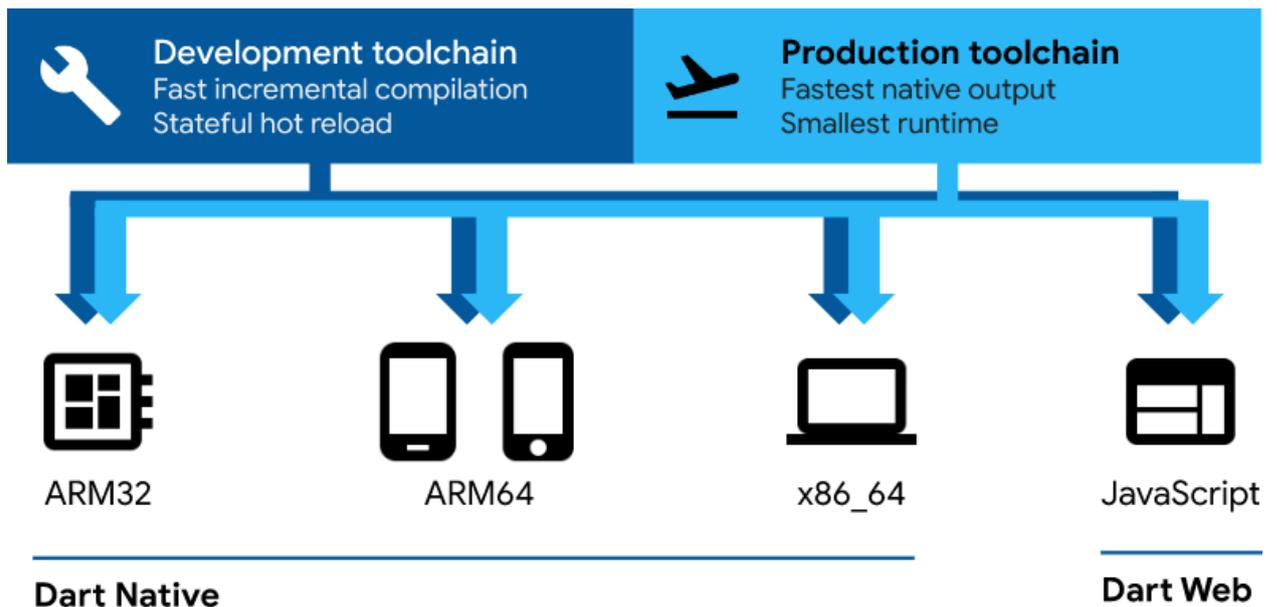


Рис. 2. Кросплатформенність Dart.

1.1.4 Dart Native (машинний код JIT і AOT)

Dart VM пропонує компілятор «just-in-time» (JIT) з інкрементальною перекомпіляцією (з можливістю гарячого перезавантаження), колекціями поточних показників (забезпечення DevTools) і багатою підтримкою налагодження.

Коли програми готові до використання — незалежно від того, чи публікуєте ви їх в магазині додатків чи використовуєте їх у виробничій серверній частині — компілятор Dart AOT дозволяє завчасно компілювати в код у машинний код ARM або x64. Ваш скомпільований AOT додаток запускається з постійним коротким часом запуску.

Код, скомпільований AOT, виконується в ефективному середовищі виконання Dart, яке забезпечує надійну систему і керує пам'яттю за допомогою швидкого розподілу об'єктів.

1.2 Вступ до Flutter

1.2.1 Що таке Flutter.

Flutter – безкоштовний open-source мобільний фреймворк для побудови інтерфейсу користувача від компанії Google, яка вийшла в травні 2017 року. Однією з переваг є можливість написання єдиної кодової бази для платформ IOS і Android, також нещодавно з'явилася підтримка Flutter Web для веб браузерів. Це означає наявність можливості використання однієї мови програмування для декількох додатків.

Flutter складається з двох важливих частин:

- SDK (Software Development Kit) – колекція інструментів, які допомагають в розробці застосунків. Це включає інструменти для компіляції коду в нативний машинний код.
- Framework – колекція віджетів для багаторазового використання UI елементів, які можна використати в своєму застосунку.

Висока продуктивність застосунків і швидкість розробки досягається за рахунок декількох технік:

- На відмінну від багатьох відомих сьогодні мобільних платформ, Flutter не використовує Javascript взагалі. В якості мови програмування для Flutter вибрали Dart, який компілюється в бінарний код, завдяки якому досягається швидкість виконання операцій, яку можна порівняти з нативними технологіями.

- Flutter не використовує нативні компоненти, тому нічого не потрібно розробляти для праці з ними. Замість цього інтерфейс будується як в ігрових програмах. Фон, текст, медіа-елементи, кнопки – весь рендер відбувається в Flutter.

- Для побудови інтерфейсу користувача використовується декларативний підхід, що схоже на шлях ReactJs, на основі віджетів. Для ще більшого приросту до швидкості роботи інтерфейсу віджети малюються за необхідністю – тільки якщо в моделі відбулися зміни (подібно до того як відбувається у світі FrontEnd).

1.2.2 Flutter 2

Спочатку Flutter був розроблений як SDK, орієнтований на мобільні пристрої, для створення нативних додатків для Android та iOS з єдиною кодовою базою. Однак Google наполегливо працював над розробкою та вдосконаленням свого дітища, і тепер, з їх останнім великим оновленням, з'явилася давно обіцяна можливість створювати програми практично для будь-якої платформи. Ось про що йдеться в офіційному повідомленні:

«З Flutter 2 можна використовувати ту саму кодову базу для доставки рідних програм до п'яти операційних систем: iOS, Android, Windows, macOS і Linux; а також веб-переглядач, націлений на такі веб-переглядачі, як Chrome, Firefox, Safari або Edge. Flutter можна навіть вбудувати в автомобілі, телевізори та розумну побутову техніку, забезпечуючи найбільш поширений і портативний досвід для навколишнього світу комп'ютерів».

Також, Google хвалиться своєю співпрацею з Microsoft, Toyota і Canonical, спрямованою на те, щоб Flutter використовувати на настільних комп'ютерах, складаних і вбудованих пристроях.

Щоб продемонструвати цю адаптивність, вони створили демонстраційний додаток, створений для того, щоб продемонструвати кросплатформеність Flutter на iOS, Android, MacOS, Windows, Linux та Web — Flutter Folio, що представлений на *Рис.3*.



Рис. 3. Flutter folio

1.2.3 Плюси розробки Flutter

Що інноваційного, відмінного та просто краще виконаного у Flutter? Давайте розглянемо особливості, які змусять вас задуматися про відмову від традиційних інструментів для використання нових.

Готові та користувацькі віджети для швидкого кодування інтерфейсу. Раніше згадувалось, що Flutter використовує готові віджети. Можна навіть сказати, що Flutter — це віджети. Однією з революційних речей продукту є те, як він допомагає створити інтерфейс користувача з використанням блоків. Різноманітність блокових віджетів Flutter подано на *Рис. 4.*

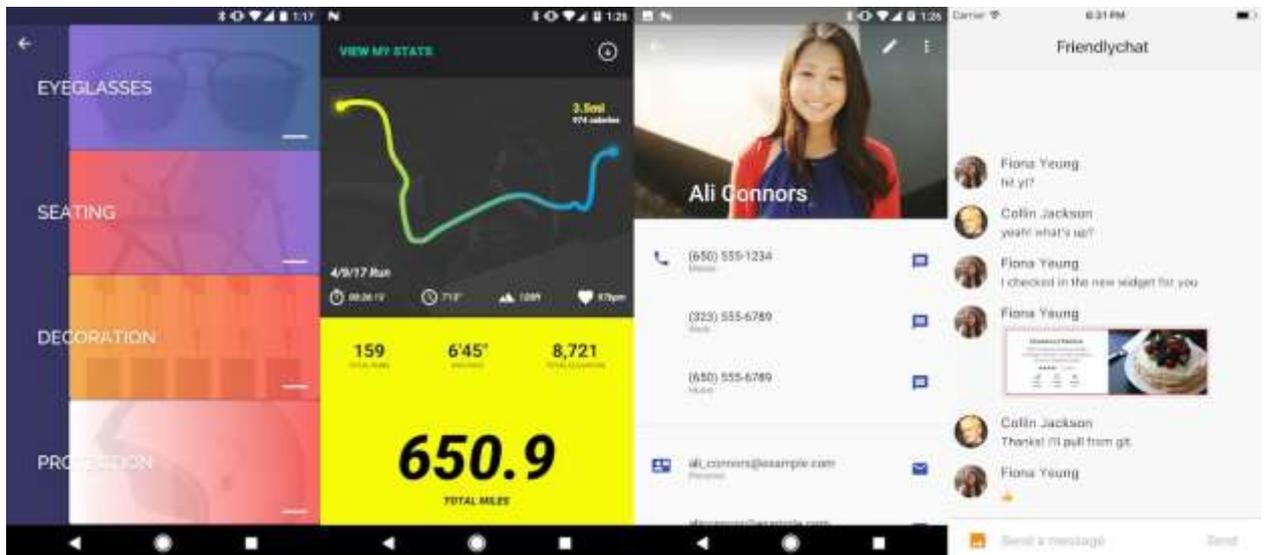


Рис. 4. Різноманітність віджетів Flutter

Будь-який об'єкт у Flutter є віджетом, від кнопки до відступу чи шрифту. Віджети можна комбінувати для створення макетів, і ви можете використовувати віджети на будь-якому рівні налаштування – від існуючих будівельних блоків до найнижчого рівня, коли ви створюєте власні віджети за допомогою тих самих інструментів, що й команда Flutter.

Віджети у Flutter організовані в деревах, що зручно для візуалізації, але може призвести до надмірного ускладнення всієї структури. Великим додатком може знадобитися до 10 шарів коду для створення базового об'єкта, тому вам доведеться спланувати структуру заздалегідь.

Dart – проста та ефективна мова, орієнтована на програмістів Java.

Dart — сучасна об'єктно-орієнтована мова, яка своїм синтаксисом нагадує Java або C++. Він підтримує різноманітні стилі набору тексту, що робить його легким для початківців.

Flutter використовує найкраще з обох сторін, використовуючи JIT-компіляцію під час розробки та перемикаючись на AOT для випуску програми.

Спільнота розробників Flutter для обміну знаннями. Враховуючи його коротку історію, можна припустити, що Flutter не має підтримки спільноти. Однак для такої молоді технології вона стрімко розвивається. Щорічне дослідження Stack Overflow, зображене на *Рис. 5*, показало, що Flutter є 3-м найбільш популярним і 4-м затребуваним фреймворком 2020 року.



Рис. 5. Результати щорічного дослідження Stack Overflow

Безліч шанувальників Flutter на Medium, Stack Overflow, Discord, Reddit, Gitter та у всіх основних соціальних мережах допоможуть вам вирішити будь-які проблеми, які можуть виникнути.

Google як гарантія довгострокової підтримки. Оскільки Flutter підтримується Google, а сам Google є великим користувачем цієї технології,

будьте впевнені, що він матиме постійну підтримку. Усі помилки будуть негайно виправлені, а продукт буде розвиватися і надалі.

Функція гарячого перезавантаження для миттєвих оновлень.

Інструмент гарячого перезавантаження вже вбудовано в архітектурі Flutter і не потребує жодних плагінів для роботи. Гаряче перезавантаження дозволяє вам бачити оновлення в режимі реального часу.

Гаряче перезавантаження підвищує продуктивність програмістів, допомагає виконувати швидкі ітерації та дозволяє експериментувати без довгих затримок.

Висока продуктивність. На оцінку продуктивності програми впливає багато факторів: використання центрального процесора (ЦП), кількість запитів в секунду, середній час відповіді, кількість кадрів в секунду тощо. Команда Flutter обіцяє постійну швидкість 60 кадрів в секунду (або продуктивність 120 кадрів в секунду на пристроях, які можуть оновлювати 120 Гц), що є швидкістю, з якою сучасні екрани відображають плавне зображення.

1.2.4 Слабкі сторони Flutter

Будь-яка технологія має свої слабкі сторони, особливо на початку свого існування. Продовжуючи слідкувати за еволюцією Flutter, спостерігається ліквідування деяких початкових проблем. Наприклад, перші сумніви щодо впровадження Flutter для iOS більше не є проблемою через постійно оновлювані віджети Купертіно. Крім того, його надійність і ефективність вже перевірена часом такими гігантами, як Alibaba і Google Ads. Розберемо детальніше недоліки Flutter.

Відсутність сторонніх бібліотек. Сторонні бібліотеки та пакети відіграють важливу роль в автоматизації розробки програмного забезпечення для програмістів і позбавляють від необхідності кодувати все з нуля. Ці бібліотеки здебільшого відкриті, легкодоступні та попередньо протестовані – хто б не захотів спробувати інструмент, який використовувався та тестувався в різних середовищах раніше?

Для багатьох старих і більш популярних технологій знайти потрібний пакет не є великою проблемою, тоді як для нових – це часто проблема. Проте станом на сьогодні на офіційному ресурсі Flutter є понад 15 тис. пакетів, і кількість продовжує зростати. Звичайно, це менше, ніж у його головного конкурента, React Native, але більш ніж достатньо. Крім того, як вже обговорювалось раніше, у Flutter є багато попередньо встановлених віджетів, які дозволяють охопити більшу частину циклу розробки без будь-яких додаткових інструментів сторонніх розробників.

Більший розмір програми. Розробники докладають всіх зусиль, щоб мінімізувати розмір програми. Користувачі мають обмежений обсяг пам'яті на своїх телефонах, тому набагато краще випустити програму, яка не змусить їх видаляти її на користь цінних фотографій або музичної бібліотеки. Щоб зменшити розмір програми, програмісти прагнуть уникати анімації, звести кількість бібліотек і пакетів до мінімуму або стискати зображення.

РОЗДІЛ 2. РОЗРОБКА КРОСПЛАТФОРМЕНОГО ЗАСТОСУНКУ «ПОМІЧНИК РИБАЛКИ».

2.1 Постановка задачі

Розглянемо поставлене завдання — «Розробка кросплатформеного застосунку “Помічник рибалки”». Опитавши кілька знайомих риболовів щодо їхніх побажань, було прийнято рішення розробки наступних сторінок:

- 1) Домашня сторінка;
- 2) Сторінка з корисними, пізнавальними та цікавими відео;
- 3) Сторінка із корисними лайфхаками.
- 4) Чат для комунікації.

2.2 Розробка домашньої сторінки додатку

Розробка головної сторінки додатку – найскладніше завдання у створенні будь якого застосунку. Адже саме з цієї сторінки починається весь функціонал та знайомство з додатком. Було прийнято рішення, реалізувати домашню сторінку у декількох файлах, для зручності використання та редагування коду. Файли під назвою: main.dart, home_page.dart та info.json.

Також, було завчасно створено файл colors.dart для зручності використання кольорів та зменшення кількості коду.

Вміст файлу main.dart

```
import 'package:diplom/home_page.dart';
import 'package:diplom/video_info.dart';
import 'package:diplom/chats.dart';
import 'package:get/get.dart';
import 'package:flutter/material.dart';
import 'package:page_transition/page_transition.dart';
import 'package:diplom/lifehacks.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return GetMaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(

        primarySwatch: Colors.blue,
      ),
      home: HomePage(),
    );
  }
}
```

Вміст файлу home_page.dart

```
// ignore_for_file: prefer_const_constructors,
prefer_const_literals_to_create_immutables

import 'dart:convert';
import 'package:diplom/video_info.dart';
import 'package:flutter/cupertino.dart';
import 'package:diplom/chats.dart';
import 'package:flutter/material.dart';
import 'package:get/get.dart';
```

```

import 'colors.dart' as color;

class HomePage extends StatefulWidget {
  const HomePage({Key? key}) : super(key: key);

  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  List info = [];
  _initData() {
    DefaultAssetBundle.of(context).loadString("json/info.json").then((value) {
      setState(() {
        info = jsonDecode(value);
      });
    });
  }

  @override
  void initState() {
    super.initState();
    _initData();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: color.AppColor.homePageBackground,
      body: Container(
        padding: const EdgeInsets.only(top: 70, left: 20, right: 20),
        child: Column(
          children: [
            Row(
              children: [
                Text(
                  "Помічник рибалки",
                  style: TextStyle(
                    fontSize: 30,
                    color: color.AppColor.homePageTitle,
                    fontWeight: FontWeight.w700),
                ),
              ],
            ),
            SizedBox(
              height: 30,
            ),
            Row(
              children: [
                Text(
                  "Більше про нас",
                  style: TextStyle(
                    fontSize: 20,
                    color: color.AppColor.homePageSubtitle,
                    fontWeight: FontWeight.w700),
                ),
                Expanded(child: Container()),
                Text(
                  "Details",
                  style: TextStyle(

```



```

width: MediaQuery.of(context).size.width,
child: Stack(
  children: [
    Container(
      width: MediaQuery.of(context).size.width,
      margin: const EdgeInsets.only(top: 30),
      height: 120,
      decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(20),
        image: DecorationImage(
          image: AssetImage("assets/card.jpg"),
          fit: BoxFit.fill),
      // ignore:
prefer_const_literals_to_create_immutables
      boxShadow: [
        BoxShadow(
          blurRadius: 40,
          offset: Offset(8, 10),
          color: color.AppColor.gradientSecond
            .withOpacity(0.3)),
        BoxShadow(
          blurRadius: 10,
          offset: Offset(-1, -5),
          color: color.AppColor.gradientSecond
            .withOpacity(0.3))
      ]),
    Container(
      height: 200,
      width: MediaQuery.of(context).size.width,
      margin: const EdgeInsets.only(
        right: 200, left: 30, bottom: 30),
      decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(20),
        image: DecorationImage(
          image: AssetImage("assets/rubolov2.png"),
          //fit:BoxFit.fill
        ),
      // ignore:
prefer_const_literals_to_create_immutables
    ),
    Container(
      width: double.maxFinite,
      height: 100,
      margin: const EdgeInsets.only(left: 150, top: 50),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Text(
            "Цікавий факт",
            style: TextStyle(
              fontSize: 16,
              fontWeight: FontWeight.bold,
              color: color.AppColor.homePageDetail),
          ),
          SizedBox(
            height: 10,
          ),
          RichText(
            text: TextSpan(
              text: "З усіх істот швидше за всіх зростає
риба\n",
              style: TextStyle(

```



```

        offset: Offset(-5, -5),
        color:
color.AppColor.gradientSecond
        .withOpacity(0.1))
    ]),
    child: Center(
      child: Align(
        alignment: Alignment.bottomCenter,
        child: Text(
          info[a]["title"],
          style: TextStyle(
            fontSize: 20,
            color: color
.AppColor.homePageDetail),
        )),
    )),
  Container(
    width:
      (MediaQuery.of(context).size.width -
90) /
        2,
    height: 140,
    margin: EdgeInsets.only(
      left: 30,
      bottom: 30,
    ),
    padding: EdgeInsets.only(bottom: 5),
    decoration: BoxDecoration(
      color: Colors.white,
      borderRadius:
BorderRadius.circular(15),
      image: DecorationImage(
        image:
AssetImage(info[b]['img']),
        boxShadow: [
          BoxShadow(
            blurRadius: 3,
            offset: Offset(5, 5),
            color:
color.AppColor.gradientSecond
              .withOpacity(0.1)),
          BoxShadow(
            blurRadius: 3,
            offset: Offset(-5, -5),
            color:
color.AppColor.gradientSecond
              .withOpacity(0.1))
        ]),
    child: Center(
      child: Align(
        alignment: Alignment.bottomCenter,
        child: Text(
          info[b]["title"],
          style: TextStyle(
            fontSize: 20,
            color: color
.AppColor.homePageDetail),
        )),
    )),
  ],

```

```
        );  
    })))  
    ],  
    ),  
    ),  
    );  
}
```

Вміст файлу info.json

```
[  
  {  
    "title": "Лайфхаки",  
    "img": "assets/ryb1.png"  
  },  
  {  
    "title": "Чат",  
    "img": "assets/ryb2.png"  
  },  
]
```

2.3 Розробка сторінки з відео

Розробку цієї сторінки також було поділено на декілька файлів, а саме: `videoinfo.dart` та `videoinfo.json`. У першому файлі знаходиться дизайн сторінки та реалізація функцій відео програвача, а у другий файл було створено задля зручної зміни/додавання відео, щоб постійно не шукати необхідний шматок коду.

Вміст файлу `videoinfo.json`

```
[
  {
    "title": "Риболовля 1",
    "time": "12:04",
    "thumbnail": "assets/pic1.jpg",
    "videoUrl": "https://www.4shared.com/s/f3f_2krAt"
  },
  {
    "title": "Риболовля 2",
    "time": "00:55",
    "thumbnail": "assets/pic2.jpg",
    "videoUrl":
"https://img.bslmeiyu.com/uploads/20170706/The%20Bananas%20Song%20%20Count
ing%20Bananas%20%20Super%20Simple%20Songs.mp4"
  },
  {
    "title": "Риболовля 3",
    "time": "01:00",
    "thumbnail": "assets/pic3.jpg",
    "videoUrl": "https://flutter.github.io/assets-for-api-
docs/assets/videos/butterfly.mp4"
  },
  {
    "title": "Риболовля 4",
    "time": "02:00",
    "thumbnail": "assets/pic4.jpg",
    "videoUrl": "https://youtu.be/bgCHfyOzeEc"
  }
]
```

Вміст файлу `videoinfo.dart`

```
// ignore_for_file: prefer_const_literals_to_create_immutables,
prefer_const_constructors, avoid_single_cascade_in_expression_statements

import 'dart:convert';

import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:video_player/video_player.dart';
import 'colors.dart' as color;

class VideoInfo extends StatefulWidget {
  const VideoInfo({Key? key}) : super(key: key);
```

```

@override
State<VideoInfo> createState() => _VideoInfoState();
}

class _VideoInfoState extends State<VideoInfo> {
  List videoInfo = [];
  bool _playArea = false;
  bool _isPlaying = false;
  bool _disposed = false;
  int _isPlayingIndex = -1;

  VideoPlayerController? _controller;
  _initData() async {
    await DefaultAssetBundle.of(context)
      .loadString("json/videoinfo.json")
      .then((value) {
        setState(() {
          videoInfo = jsonDecode(value);
        });
      });
  }

  @override
  void initState() {
    super.initState();
    _initData();
  }

  @override
  void dispose(){
    _disposed=true;
    _controller?.pause();
    _controller?.dispose();
    _controller=null;
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        decoration: _playArea==false?BoxDecoration(
          gradient: LinearGradient(
            colors: [
              color.AppColor.gradientFirst.withOpacity(0.9),
              color.AppColor.gradientSecond
            ],
            begin: const FractionalOffset(0.0, 0.4),
            end: Alignment.topRight,
          ):BoxDecoration(
            color:color.AppColor.gradientSecond,
          ),
        child: Column(
          children: [
            _playArea==false?Container(
              padding: const EdgeInsets.only(top: 70, left: 30, right: 30),
              width: MediaQuery.of(context).size.width,
              height: 300,
              child: Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [

```

```

    Row(
      children: [
        InkWell(
          onTap: () {
            Get.back();
          },
          child: Icon(Icons.arrow_back_ios,
            size: 20, color:
color.AppColor.secondPageIconColor),
        ),
        Expanded(child: Container()),
        Icon(Icons.info_outline,
          size: 20, color:
color.AppColor.secondPageIconColor),
      ],
    ),
    SizedBox(
      height: 30,
    ),
    Text(
      "Корисні відео",
      style: TextStyle(
        fontSize: 25, color:
color.AppColor.secondPageTitleColor),
    ),
    SizedBox(
      height: 5,
    ),
    Text(
      "та просто для душі",
      style: TextStyle(
        fontSize: 25, color:
color.AppColor.secondPageTitleColor),
    ),
    SizedBox(
      height: 50,
    ),
    Row(
      children: [
        Container(
          width: 220,
          height: 30,
          decoration: BoxDecoration(
            borderRadius: BorderRadius.circular(10),
            gradient: LinearGradient(
              colors: [
                color.AppColor
                  .secondPageContainerGradient1stColor,
                color.AppColor
                  .secondPageContainerGradient2ndColor
              ],
            begin: Alignment.bottomLeft,
            end: Alignment.topRight,
          )),
        child: Row(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Icon(
              Icons.handyman_outlined,
              size: 20,
              color: color.AppColor.secondPageIconColor,

```

```

        ),
        SizedBox(
          width: 5,
        ),
        Text(
          "Valeriy Stasiuk, KPNU",
          style: TextStyle(
            fontSize: 16,
            color:
color.AppColor.secondPageIconColor,
          ),
        ),
      ]))
    ],
  ),
  // SizedBox(width: 20,),
],
),
):Container(
  //123
  child:Column(
    children: [
      Container(
        height:100,
        padding: const EdgeInsets.only(top: 50, left: 30, right:
30),
        child:Row(children: [
          InkWell(
            onTap: () {
              Get.back();
            },
            child:Icon(Icons.arrow_back_ios, size: 20, color:
color.AppColor.secondPageIconColor)
          ),
          Expanded(child: Container()),
          Icon(Icons.info_outline,
            size: 20, color: color.AppColor.secondPageTopIconColor)
        ],)
      ),
      _playView(context),
      _controlView(context),
    ],
  ),
  Expanded(
    child: Container(
      decoration: BoxDecoration(
        color: Colors.white,
        borderRadius: BorderRadius.only(topRight:
Radius.circular(70))),
      child: Column(
        children: [
          SizedBox(
            height: 30,
          ),
          Row(
            children: [
              SizedBox(
                width: 30,
              ),
              Text("Приемного просмотра",

```

```

        style: TextStyle(
          fontSize: 20,
          fontWeight: FontWeight.bold,
          color: color.AppColor.circuitsColor)),
Expanded(child: Container()),
Row(
  children: [
    Icon(Icons.loop,
      size: 30, color: color.AppColor.loopColor),
    SizedBox(
      width: 10,
    ),
    Text("",
      style: TextStyle(
        fontSize: 15,
        color: color.AppColor.setsColor,
      ))
  ],
),
SizedBox(
  width: 20,
),
]),
SizedBox(height: 20,),
Expanded(child: _listView(),)
    ],
  ),
),
));
}

var _onUpdateControllerTime;
void _onControllerUpdate() async{
  if(!_disposed){
    return;
  }

  _onUpdateControllerTime = 0;
  final now = DateTime.now().millisecondsSinceEpoch;
  if(_onUpdateControllerTime>now){
    return;
  }

  _onUpdateControllerTime=now+500;

  final controller = _controller;
  if(controller==null){
    debugPrint("controller is null");
    return;
  }
  if(!controller.value.isInitialized){
    debugPrint("controller is not initialized");
    return;
  }
  final playing = controller.value.isPlaying;
  _isPlaying=playing;
}

_onTapVideo(int index) {

```



```

        if(index<=videoInfo.length-1){
            _onTapVideo(index);
        }else{
            Get.snackbar("Video List", "No more video i the list");
        }
    },
    child: Icon(Icons.fast_forward,size:36,color:Colors.white,)
  ),
],
),
);
}

```

```

Widget _playView(BuildContext context) {
  final controller = _controller;
  if(controller!=null && controller.value.isInitialized) {
    return AspectRatio(
      aspectRatio: 16/9,
      child: VideoPlayer(controller),
    );
  } else {
    return AspectRatio(
      aspectRatio: 16/9,
      child: Center(child: Text("Preparing...",
        style: TextStyle(
          fontSize: 20,
          color: Colors.white60
        ),
      ),
    ));
  }
}

```

```

_listView() {
  return ListView.builder(
    padding: EdgeInsets.symmetric(horizontal: 30,
vertical: 8),
    itemCount: videoInfo.length,
    itemBuilder: (_, int index) {
      return GestureDetector(
        onTap: () {
          _onTapVideo(index);
          debugPrint(index.toString());
          setState(() {
            if (_playArea==false){
              _playArea = true;
            }
          });
        },
        child: _buildCard(index)
      );
    },
  );
}
_buildCard (int index) {
  return Container(
    height: 135,
    child: Column(
      children: [
        Row(
          children: [
            Container(
              width: 80,

```



```

    ),
    Row(children: [
      for(int i = 0; i<70;i++)
      i.isEven?Container(
        width: 3,
        height: 1,
        decoration: BoxDecoration(
          color:Color(0xFF839fed),
          borderRadius: BorderRadius.circular(2),
        ),
      ):Container(
        width: 3,
        height: 1,
        color:Colors.white,
      ),
    ],),
  ],),
),
);
}
}

```

2.4 Реалізація сторінки «Чат»

Комунікація між риболовами – дуже важлива річ. Спілкування одне з одним це найкращий спосіб взнати ту чи іншу інформацію, саме через це було прийнято рішення в реалізації даної функції.

Вміст файлу chats.dart

```
import 'dart:ui';
import 'package:flutter/material.dart';

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Chat UI',
      debugShowCheckedModeBanner: false,
      home: RybChatPage(),
    );
  }
}

class RybChatPage extends StatefulWidget {
  @override
  _RybChatPageState createState() => _RybChatPageState();
}

class _RybChatPageState extends State<RybChatPage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.indigo,
      body: SafeArea(
        child: Column(
          children: [
            _top(),
            _body(),
          ],
        ),
      ),
    );
  }

  Widget _top() {
    return Container(
      padding: const EdgeInsets.only(top: 30, left: 30),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          const Text(
            'Chat with \nyour friends',
            style: TextStyle(
              fontSize: 28, fontWeight: FontWeight.bold, color:
Colors.white),
          ),
          Row(
            children: [
```

```

    Container(
      padding: const EdgeInsets.all(14),
      decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(50),
        color: Colors.black12,
      ),
      child: const Icon(
        Icons.search,
        size: 30,
        color: Colors.white,
      ),
    ),
    const SizedBox(
      width: 15,
    ),
    Expanded(
      child: Container(
        height: 100,
        child: ListView.builder(
          physics: const BouncingScrollPhysics(),
          scrollDirection: Axis.horizontal,
          shrinkWrap: true,
          itemCount: 8,
          itemBuilder: (context, index) {
            return Avatar(
              margin: const EdgeInsets.only(right: 15),
              image: 'assets/${index + 1}.jpg',
            );
          },
        ),
      ),
    ),
  ],
),
],
);
}

Widget _body() {
  return Expanded(
    child: Container(
      padding: const EdgeInsets.symmetric(horizontal: 30),
      width: double.infinity,
      decoration: const BoxDecoration(
        borderRadius: BorderRadius.only(
          topLeft: Radius.circular(45), topRight: const
Radius.circular(45)),
        color: Colors.white,
      ),
    ),
    child: ListView(
      padding: const EdgeInsets.only(top: 35),
      physics: const BouncingScrollPhysics(),
      children: [
        _itemChats(
          avatar: 'assets/2.jpg',
          name: 'Смутьський Вадим',
          chat:
            'Надсилає допис користувача..',
          time: '17:15',
        ),
        _itemChats(

```

```

        avatar: 'assets/4.jpg',
        name: 'Задорожний',
        chat: 'Шо ти там?',
        time: '16:13',
      ),
      _itemChats(
        avatar: 'assets/3.jpg',
        name: 'AspazX',
        chat: 'Привіт ☐',
        time: '11:21',
      ),
      _itemChats(
        avatar: 'assets/6.jpg',
        name: 'Козаков',
        chat: 'Скинь титулку дипломної',
        time: '09:54',
      ),
      _itemChats(
        avatar: 'assets/7.jpg',
        name: 'Задворний',
        chat:
          'Як це має проходити?',
        time: '21:51',
      ),
      _itemChats(
        avatar: 'assets/8.jpg',
        name: 'Каріна',
        chat:
          'Duis aute irure dolor in reprehenderit in voluptate
          velit esse cillum dolore eu fugiat nulla pariatur',
        time: '00.09',
      ),
    ],
  ),
);
}

```

```

Widget _itemChats(
  {String avatar = '', name = '', chat = '', time = '00.00'}) {
  return GestureDetector(
    onTap: () {
      Navigator.of(context).push(
        MaterialPageRoute(
          builder: (context) => ChatPage(),
        ),
      );
    },
  ),
  child: Card(
    margin: const EdgeInsets.symmetric(vertical: 20),
    elevation: 0,
    child: Row(
      children: [
        Avatar(
          margin: const EdgeInsets.only(right: 20),
          size: 60,
          image: avatar,
        ),
        Expanded(
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [

```

```

        Row(
          mainAxisAlignment: MainAxisAlignment.spaceBetween,
          children: [
            Text(
              '$name',
              style: const TextStyle(
                fontSize: 17, fontWeight: FontWeight.bold),
            ),
            Text(
              '$time',
              style: const TextStyle(
                color: Colors.grey, fontWeight:
FontWeight.bold),
            ),
          ],
        ),
        const SizedBox(
          height: 10,
        ),
        Text(
          '$chat',
          overflow: TextOverflow.ellipsis,
          style: const TextStyle(
            color: Colors.black54,
            fontWeight: FontWeight.bold,
          ),
        ),
      ],
    ),
  ],
),
);
}

class ChatPage extends StatefulWidget {
  @override
  _ChatPageState createState() => _ChatPageState();
}

class _ChatPageState extends State<ChatPage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.indigo,
      body: SafeArea(
        child: Stack(
          children: [
            Column(
              children: [
                _topChat(),
                _bodyChat(),
                const SizedBox(
                  height: 120,
                )
              ],
            ),
            _formChat(),
          ],
        ),
      ),
    );
  }
}

```

```

    ),
  );
}

_topChat() {
  return Container(
    padding: const EdgeInsets.symmetric(horizontal: 30, vertical: 25),
    child: Row(
      mainAxisAlignment: MainAxisAlignment.spaceBetween,
      children: [
        Row(
          children: [
            GestureDetector(
              onTap: () => Navigator.of(context).pop(),
              child: const Icon(
                Icons.arrow_back_ios,
                size: 25,
                color: Colors.white,
              ),
            ),
            const Text(
              'Валерій',
              style: TextStyle(
                fontSize: 28,
                fontWeight: FontWeight.bold,
                color: Colors.white),
            ),
          ],
        ),
        Row(
          children: [
            Container(
              padding: const EdgeInsets.all(12),
              decoration: BoxDecoration(
                borderRadius: BorderRadius.circular(50),
                color: Colors.black12,
              ),
              child: const Icon(
                Icons.call,
                size: 25,
                color: Colors.white,
              ),
            ),
            const SizedBox(
              width: 20,
            ),
            Container(
              padding: const EdgeInsets.all(12),
              decoration: BoxDecoration(
                borderRadius: BorderRadius.circular(50),
                color: Colors.black12,
              ),
              child: const Icon(
                Icons.videocam,
                size: 25,
                color: Colors.white,
              ),
            ),
          ],
        ),
      ],
    ),
  ),
}

```

```

    );
}

Widget _bodyChat() {
  return Expanded(
    child: Container(
      padding: const EdgeInsets.only(left: 25, right: 25, top: 25),
      width: double.infinity,
      decoration: const BoxDecoration(
        borderRadius: BorderRadius.only(
          topLeft: Radius.circular(45), topRight:
Radius.circular(45)),
        color: Colors.white,
      ),
    ),
    child: ListView(
      physics: const BouncingScrollPhysics(),
      children: [
        _itemChat(
          avatar: 'assets/5.jpg',
          chat: 1,
          message:
'typesetting industry.',
          time: '18.00',
        ),
        _itemChat(
          chat: 0,
          message: 'Okey ☐',
          time: '18.00',
        ),
        _itemChat(
          avatar: 'assets/5.jpg',
          chat: 1,
          message: 'It has survived not only five centuries, ☐',
          time: '18.00',
        ),
        _itemChat(
          chat: 0,
          message:
random text. ☐',
          time: '18.00',
        ),
        _itemChat(
          avatar: 'assets/5.jpg',
          chat: 1,
          message:
repetition, injected humour, or non-characteristic words etc.',
          time: '18.00',
        ),
        _itemChat(
          avatar: 'assets/5.jpg',
          chat: 1,
          message: '☐ ☐ ☐',
          time: '18.00',
        ),
      ],
    ),
  );
}

```

```

// 0 = Send
// 1 = Recieved
_itemChat({chat, avatar, message, time}) {
  return Row(
    mainAxisAlignment:
      chat == 0 ? MainAxisAlignment.end : MainAxisAlignment.start,
    crossAxisAlignment: CrossAxisAlignment.end,
    children: [
      avatar != null
        ? Avatar(
            image: avatar,
            size: 50,
          )
        : Text(
            '$time',
            style: TextStyle(color: Colors.grey.shade400),
          ),
      Flexible(
        child: Container(
          margin: const EdgeInsets.only(left: 10, right: 10, top: 20),
          padding: const EdgeInsets.all(20),
          decoration: BoxDecoration(
            color: chat == 0 ? Colors.indigo.shade100 :
Colors.indigo.shade50,
            borderRadius: chat == 0
              ? const BorderRadius.only(
                  topLeft: const Radius.circular(30),
                  topRight: const Radius.circular(30),
                  bottomLeft: const Radius.circular(30),
                )
              : const BorderRadius.only(
                  topLeft: Radius.circular(30),
                  topRight: const Radius.circular(30),
                  bottomRight: Radius.circular(30),
                ),
          ),
        child: Text('$message'),
      ),
    ],
  );
}

Widget _formChat() {
  return Positioned(
    child: Align(
      alignment: Alignment.bottomCenter,
      child: Container(
        height: 120,
        padding: const EdgeInsets.symmetric(vertical: 30, horizontal:
20),
        color: Colors.white,
        child: TextField(
          decoration: InputDecoration(
            hintText: 'Type your message...',

```

```

        suffixIcon: Container(
          decoration: BoxDecoration(
            borderRadius: BorderRadius.circular(50),
            color: Colors.indigo),
          padding: const EdgeInsets.all(14),
          child: const Icon(
            Icons.send_rounded,
            color: Colors.white,
            size: 28,
          ),
        ),
        filled: true,
        fillColor: Colors.blueGrey[50],
        labelStyle: const TextStyle(fontSize: 12),
        contentPadding: const EdgeInsets.all(20),
        enabledBorder: OutlineInputBorder(
          borderSide: BorderSide(color: Colors.blueGrey),
          borderRadius: BorderRadius.circular(25),
        ),
        focusedBorder: OutlineInputBorder(
          borderSide: BorderSide(color: Colors.blueGrey),
          borderRadius: BorderRadius.circular(25),
        ),
      ),
    ),
  ),
),
);
}

class Avatar extends StatelessWidget {
  final double size;
  final image;
  final EdgeInsets margin;
  Avatar({this.image, this.size = 50, this.margin = const
EdgeInsets.all(0)});
  @override
  Widget build(BuildContext context) {
    return Padding(
      padding: margin,
      child: Container(
        width: size,
        height: size,
        decoration: new BoxDecoration(
          shape: BoxShape.circle,
          image: new DecorationImage(
            image: AssetImage(image),
          ),
        ),
      ),
    );
  }
}

```

2.5 Розробка сторінки «Лайфхаки»

Будь яка людина, у будь якій сфері праці – не проти взнати якусь певну хитрість, щоб полегшити свою працю або зробити щось якісне. Риболовля не виключення, адже багато риболовів так чи інакше користуються певними лайфхаками. Саме через це, було прийнято рішення розробки сторінки «Лайфхаки». Цю сторінку також було поділено на два файли, а саме: `lifehucks.dart` та `details.dart`

Вміст сторінки `lifehucks.dart`

```
// ignore_for_file: prefer_const_constructors

import 'package:diplom/details.dart';
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:page_transition/page_transition.dart';
import 'package:page_transition/page_transition.dart';
/*import 'package:diplom/gradient_text.dart';
import 'package:diplom/page_transition/page_transition.dart';*/

class MyApp extends StatelessWidget {
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.blue,
        visualDensity: VisualDensity.adaptivePlatformDensity,
      ),
      home: lifehacks(),
    );
  }
}

class lifehacks extends StatefulWidget {
  @override
  _lifehacksState createState() => _lifehacksState();
}

class _lifehacksState extends State<lifehacks> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Padding(
        padding: EdgeInsets.only(left: 10),
        child: Column(
          children: <Widget>[
            Expanded(
              flex: 1,
              child: Row(
                mainAxisAlignment: MainAxisAlignment.spaceBetween,
```

```

children: <Widget>[
  Container(
    decoration: BoxDecoration(
      borderRadius: BorderRadius.circular(50),
      boxShadow: [
        BoxShadow(offset: Offset(3, 3), blurRadius: 10),
      ],
    ),
    child: CircleAvatar(
      backgroundImage: AssetImage("assets/pic1.jpg"),
    ),
  ),
  Text(
    "Лайфхаки для риболовів",
    style: TextStyle(fontSize: 22, fontWeight:
FontWeight.bold),
  ),
  IconButton(
    icon: Icon(
      Icons.search,
      size: 25,
    ),
    onPressed: () {},
  )
],
),
Expanded(
  flex: 2,
  child: ListView(
    physics: BouncingScrollPhysics(),
    scrollDirection: Axis.horizontal,
    padding: EdgeInsets.only(left: 10),
    children: <Widget>[
      makeItem(image: "assets/pic1.jpg"),
      makeItem(image: "assets/pic2.jpg"),
      makeItem(image: "assets/pic3.jpg"),
    ],
  ),
),
Expanded(
  flex: 3,
  child: Column(
    children: <Widget>[
      Padding(
        padding: EdgeInsets.symmetric(horizontal: 10),
        child: Row(
          mainAxisAlignment: MainAxisAlignment.spaceBetween,
          children: <Widget>[
            Text(
              "Популярні",
              style: TextStyle(
                color: Colors.black87,
                fontSize: 25,
                fontWeight: FontWeight.bold,
              ),
            ),
            OutlineButton(
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(20)),
              borderSide:
BorderSide(width: 3, color:
Colors.blue.shade300),

```

```

        child: Text(
          "Показати все",
          style: TextStyle(color: Colors.blue.shade300),
        ),
        onPressed: () {},
      ),
    ],
  ),
),
Expanded(
  child: ListView.separated(
    physics: BouncingScrollPhysics(),
    itemBuilder: (context, index) {
      return listItem(
        image: "assets/pic$index.jpg", tag:
"tag$index");
    },
    separatorBuilder: (context, index) {
      return Divider(
        height: 20,
        color: Colors.black26,
      );
    },
    itemCount: 4),
  ),
),
),
),
),
),
);
}

```

```

Widget listItem({image, tag}) {
  return InkWell(
    onTap: () {
      Navigator.push(
        context,
        PageTransition(
          child: DetailsPage(
            image: image,
            tag: tag,
          ),
          type: PageTransitionType.fade));
    },
    child: Container(
      height: 100,
      width: 400,
      child: Row(
        mainAxisAlignment: MainAxisAlignment.start,
        children: <Widget>[
          Hero(
            tag: tag,
            child: Container(
              height: 100,
              width: 60,
              decoration: BoxDecoration(
                image: DecorationImage(
                  image: AssetImage(image), fit: BoxFit.cover),
                borderRadius: BorderRadius.only(
                  topRight: Radius.circular(20),

```



```

    ),
  );
}

Widget makeItem({image}) {
  return Container(
    width: 330,
    margin: EdgeInsets.only(right: 20, bottom: 40),
    decoration: BoxDecoration(
      borderRadius: BorderRadius.circular(20),
      image: DecorationImage(image: AssetImage(image), fit:
BoxFit.cover),
      boxShadow: [
        BoxShadow(offset: Offset(5, 5), blurRadius: 10),
      ],
    ),
    child: Padding(
      padding: EdgeInsets.only(left: 30, top: 30),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: <Widget>[
          Text("Лайфхак",
            style: TextStyle(
              color: Colors.white,
              fontSize: 20,
              fontWeight: FontWeight.bold,
              letterSpacing: 2)),
          SizedBox(
            height: 10,
          ),
        ],
      ),
    ),
  );
}
}

```

Вміст сторінки details.dart

```

import 'package:flutter/material.dart';
import 'package:page_transition/page_transition.dart';
class DetailsPage extends StatefulWidget {
  final String image;
  final String tag;

  const DetailsPage({Key? key, required this.image, required this.tag}) :
super(key: key);

  @override
  State<DetailsPage> createState() => _DetailsPageState();
}

class _DetailsPageState extends State<DetailsPage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: PreferredSize(
        preferredSize: Size.fromHeight(200),
        child: AppBar(
          shape:

```

```

        RoundedRectangleBorder(borderRadius:
BorderRadius.circular(20)),
        flexibleSpace: Hero(
            tag: widget.tag,
            child: Container(
                height: 600,
                decoration: BoxDecoration(
                    borderRadius: BorderRadius.circular(20),
                    image: DecorationImage(
                        image: AssetImage(widget.image), fit:
BoxFit.cover)),
            ),
        ),
        actions: <Widget>[
            IconButton(
                icon: Icon(
                    Icons.more_vert,
                    color: Colors.white,
                ),
                onPressed: () {},
            )
        ],
    ),
    body: SingleChildScrollView(
        child: Padding(
            padding: EdgeInsets.all(20),
            child: Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: <Widget>[
                    Row(
                        mainAxisAlignment: MainAxisAlignment.spaceBetween,
                        children: <Widget>[
                            Text(
                                "Лайфхак",
                                style: TextStyle(
                                    color: Colors.black87,
                                    fontSize: 25,
                                    fontWeight: FontWeight.bold),
                            ),
                            OutlineButton(
                                shape: RoundedRectangleBorder(
                                    borderRadius: BorderRadius.circular(20)),
                                borderSide:
                                    BorderSide(width: 3, color: Colors.blue.shade300),
                                child: Text(
                                    "View All",
                                    style: TextStyle(
                                        color: Colors.blue.shade300,
                                    ),
                                ),
                                onPressed: () {},
                            )
                        ],
                    ),
                    Text("KPNU Valeriy",
                        style: TextStyle(
                            color: Colors.black54,
                        )),
                    SizedBox(
                        height: 5,
                    ),
                ],
            ),
        ),
    ),

```

```

Row(
  children: <Widget>[
    Icon(Icons.star, color: Colors.orange.shade700, size:
20),
    Icon(Icons.star, color: Colors.grey, size: 20),
    SizedBox(
      width: 5,
    ),
    Text(
      "4.0",
      style: TextStyle(
        color: Colors.orange.shade700,
        fontWeight: FontWeight.bold,
        fontSize: 15),
    ),
    SizedBox(
      width: 5,
    ),
    Text(
      "(2460)",
      style: TextStyle(
        color: Colors.grey,
        fontSize: 15,
        fontWeight: FontWeight.bold),
    )
  ],
),
SizedBox(
  height: 20,
),
Divider(
  thickness: 1.5,
),
Padding(
  padding: EdgeInsets.symmetric(vertical: 5),
  child: Row(
    children: <Widget>[
      Icon(
        Icons.favorite,
        color: Colors.red,
      ),
      SizedBox(
        width: 10,
      ),
      Text(
        "22",
        style:
          TextStyle(fontWeight: FontWeight.bold, fontSize:
15),
      ),
      Text(
        " Likes",
        style: TextStyle(fontSize: 15, color:
Colors.black54),
      ),
      SizedBox(

```

```

        width: 30,
      ),
      Icon(Icons.message),
      SizedBox(
        width: 10,
      ),
      Text(
        "14",
        style:
          TextStyle(fontWeight: FontWeight.bold, fontSize:
15),
      ),
      Text(" Comments",
        style: TextStyle(fontSize: 15, color:
Colors.black54)),
      SizedBox(
        width: 30,
      ),
      Icon(Icons.share),
      Text(" Share",
        style: TextStyle(
          fontSize: 15,
          color: Colors.black54,
        ))
    ],
  ),
  Divider(
    thickness: 1.5,
  ),
  SizedBox(
    height: 10,
  ),
  Text(
    "Детальніше про лайфхак",
    style: TextStyle(
      color: Colors.black87,
      fontSize: 22,
      fontWeight: FontWeight.bold),
  ),
  SizedBox(
    height: 10,
  ),
  Text(
    "Як правило, при заготівлі наживки у вигляді хробаків
рибалки використовують невеликі пластмасові відра. Черв'яки ж у свою чергу
люблять збиратися на дні відерця. Вони сплутуються в грудку, а щоб їх
витягнути та розплутати, доводиться витратити чимало часу та сил. Тому для
того, щоб цього не відбувалося, можна скористатися звичайними контейнерами
для їжі. Нам потрібно розрізати дно кожного контейнера та з'єднати їх за
допомогою клею. У кришці одного з контейнерів необхідно зробити невеликі
отвори, щоб черв'якам було чим дихати. Хробаки будуть накопичуватися на
звороті кришки з отворами. Щоб їх дістати, достатньо просто перевернути
контейнер та відкрити кришку.",
    style: TextStyle(color: Colors.grey, height: 1.5),
  ),
  SizedBox(
    height: 30,
  ),
  Row(
    mainAxisAlignment: MainAxisAlignment.spaceEvenly,
    children: <Widget>[

```

```

        createButton(
            icon: Icon(
                Icons.book,
                color: Colors.white,
            ),
            color: Colors.blue,
            text: "READ NOW"),
        createButton(
            icon: Icon(Icons.shopping_basket, color:
Colors.white),
            text: "PURCHASE",
            color: Colors.pink,
        )
    ],
)
),
),
),
),
);
}

```

```

Widget createButton({icon, color, text}) {
    return ButtonTheme(
        minWidth: 160,
        height: 45,
        child: FlatButton.icon(
            onPressed: () {},
            icon: icon,
            label: Text(
                text,
                style: TextStyle(color: Colors.white),
            ),
            color: color,
            shape: RoundedRectangleBorder(borderRadius:
BorderRadius.circular(20)),
        ),
    );
}
}

```

ВИСНОВКИ

У процесі написання дипломної роботи було дослідження нову технологію розробки мобільних додатків за допомогою фреймворку Flutter від компанії Google. Було досліджено принцип роботи інструменту у поєднанні з мовою програмування Dart.

У результаті виконання дипломної роботи було розроблено кросплатформенний застосунок «Помічник рибалки».

Для досягнення поставленої мети було:

- Проаналізовано літературні джерела на відповідну тематику;
- Розроблено архітектуру застосунку;
- Обрано методи, засоби та інструменти розробки.

В розробленому кросплатформенному застосунку реалізовано:

- Головна сторінка застосунку;
- Відео програвач та його функціонал;
- Сторінка з можливістю переглядати відео;
- Сторінка з різноманітними лайфхаками;
- Сторінка «Чат».

У результаті дипломної роботи, отримали кросплатформенний застосунок «Помічник рибалки», який може стати у нагоді, як починаючим так і досвідченим риболовам.

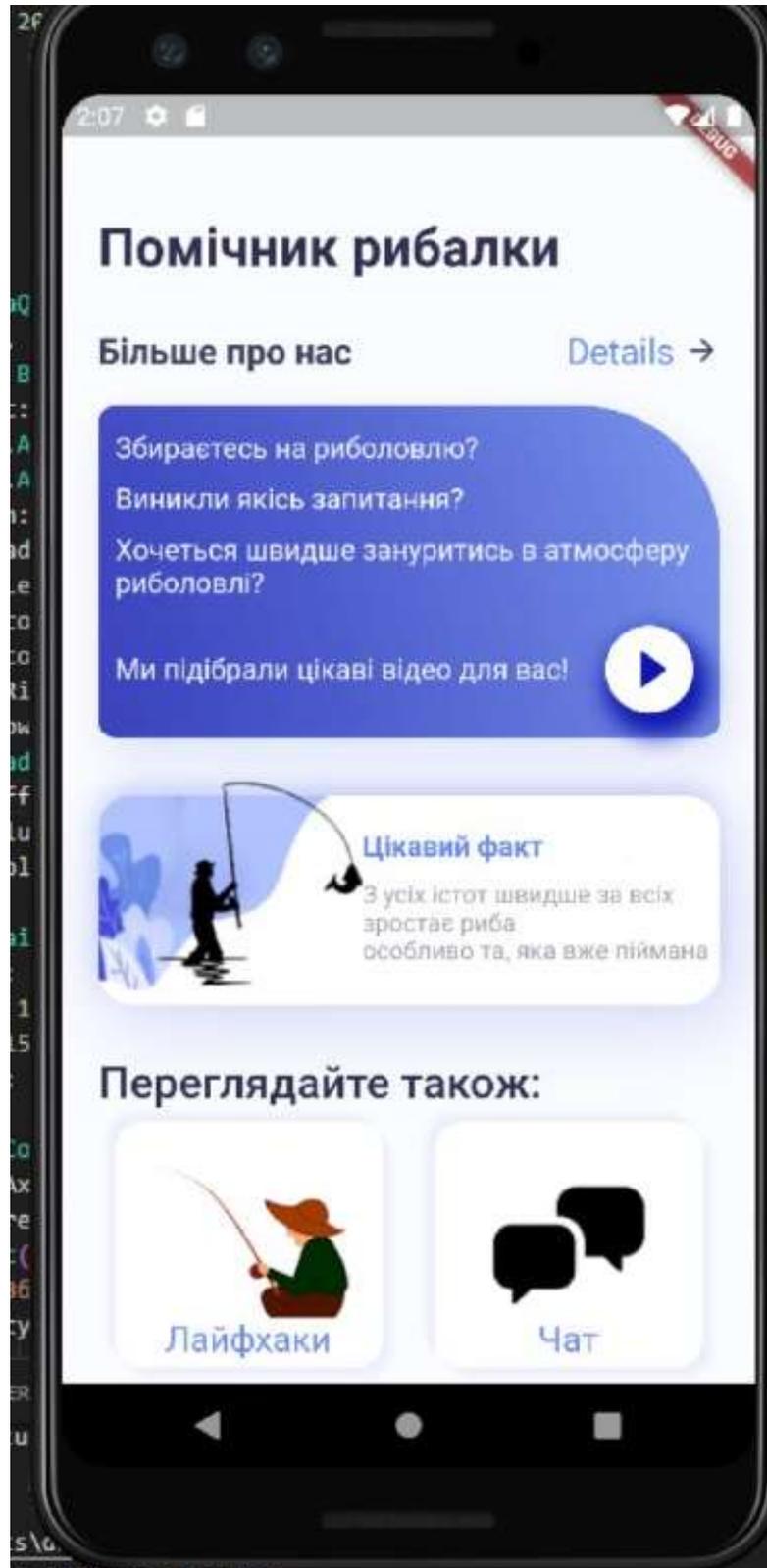
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Термоса С. . Про Flutter, кратко, основы / Станіслав Термоса., 2018.
2. Feoktistov I. Top 8 Flutter Advantages and Why You Should Try Flutter on Your Next Project [Електронний ресурс] / Ihor Feoktistov. – 2018. – Режим доступу до ресурсу: <https://relevant.software/blog/top-8-flutter-advantages-and-whyyou-should-try-flutter-on-your-next-project/>.
3. What is Flutter? Benefits and limitations. [Електронний ресурс] // Code Magic. – 2018. – Режим доступу до ресурсу: <https://blog.codemagic.io/what-is-flutter-benefits-and-limitations/>.
4. Marco L. Beginning Flutter: A Hands On Guide to App Development / L. Marco.
5. Dart Overview [Електронний ресурс] – Режим доступу до ресурсу: <https://dart.dev/overview>.
6. Введення у Flutter [Електронний ресурс] – режим доступу до ресурсу: <https://metanit.com/dart/flutter/1.1.php>.
7. Rap P. Beginning App Development with Flutter / Payne Rap., 2019. – 334 с.
8. Flutter - Beautiful native apps in record time [Електронний ресурс]/Режим доступу: <https://flutter.dev/>
9. Dart programming language [Електронний ресурс]/Режим доступу: <https://dart.dev/>
10. Marco L.N. Beginning Flutter: A Hands On Guide to App Development / Marco L. Napoli, 2019. – 528 с.
11. Windmill E. Flutter in Action / Eric Windmill, 2019. – 368 с.
12. Gilad B. The Dart Programming Language / Gilad Bracha, 2015. – 224 с.

ДОДАТКИ

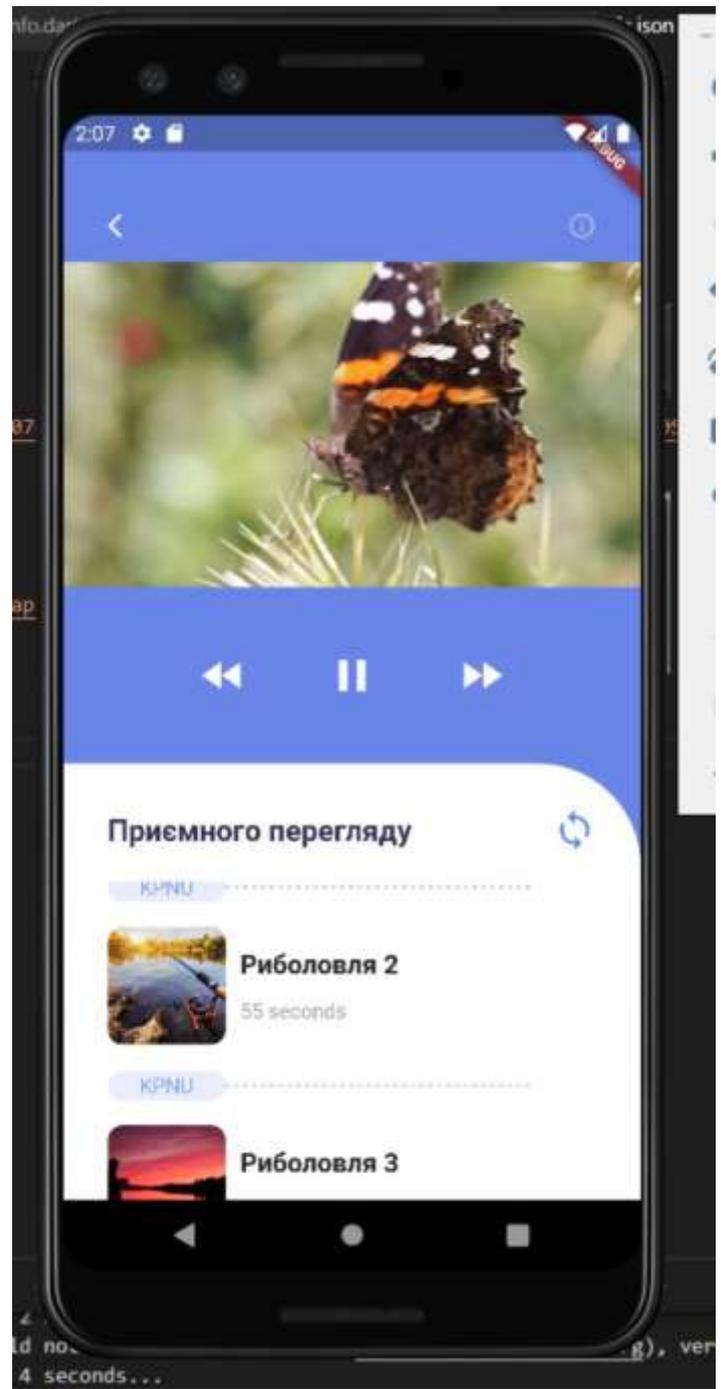
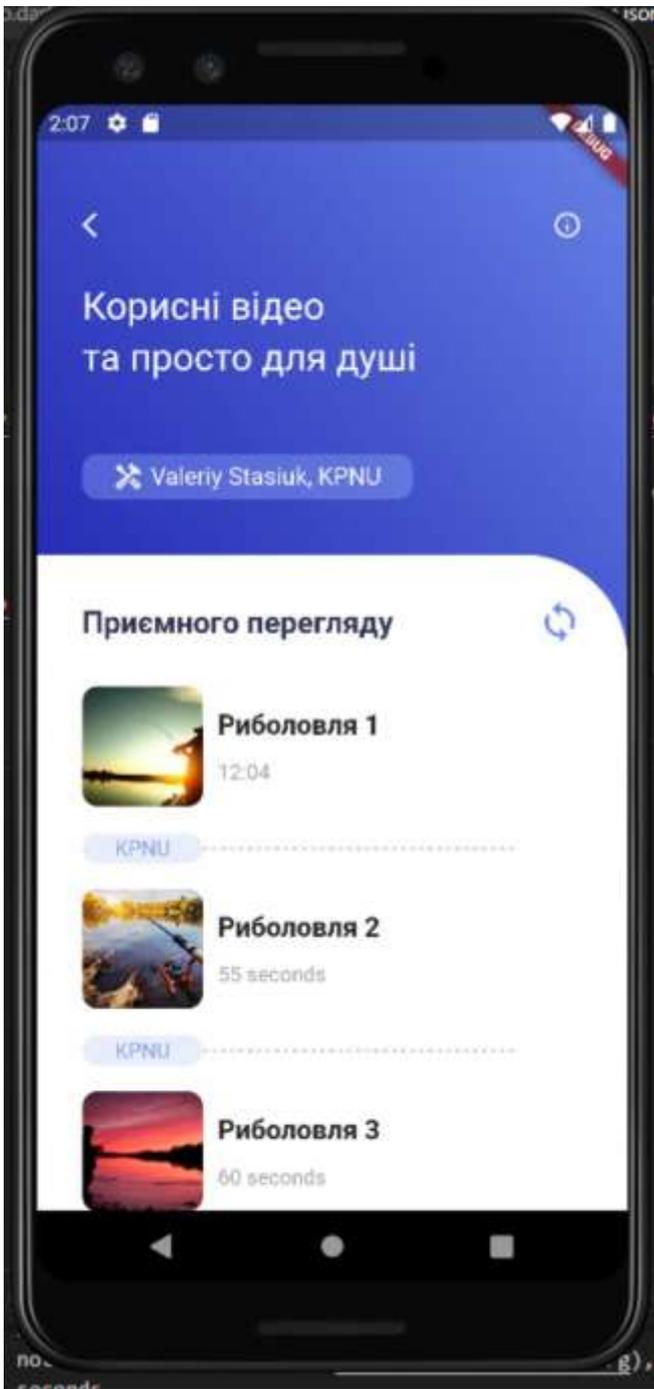
Додаток А

Головний екран застосунку



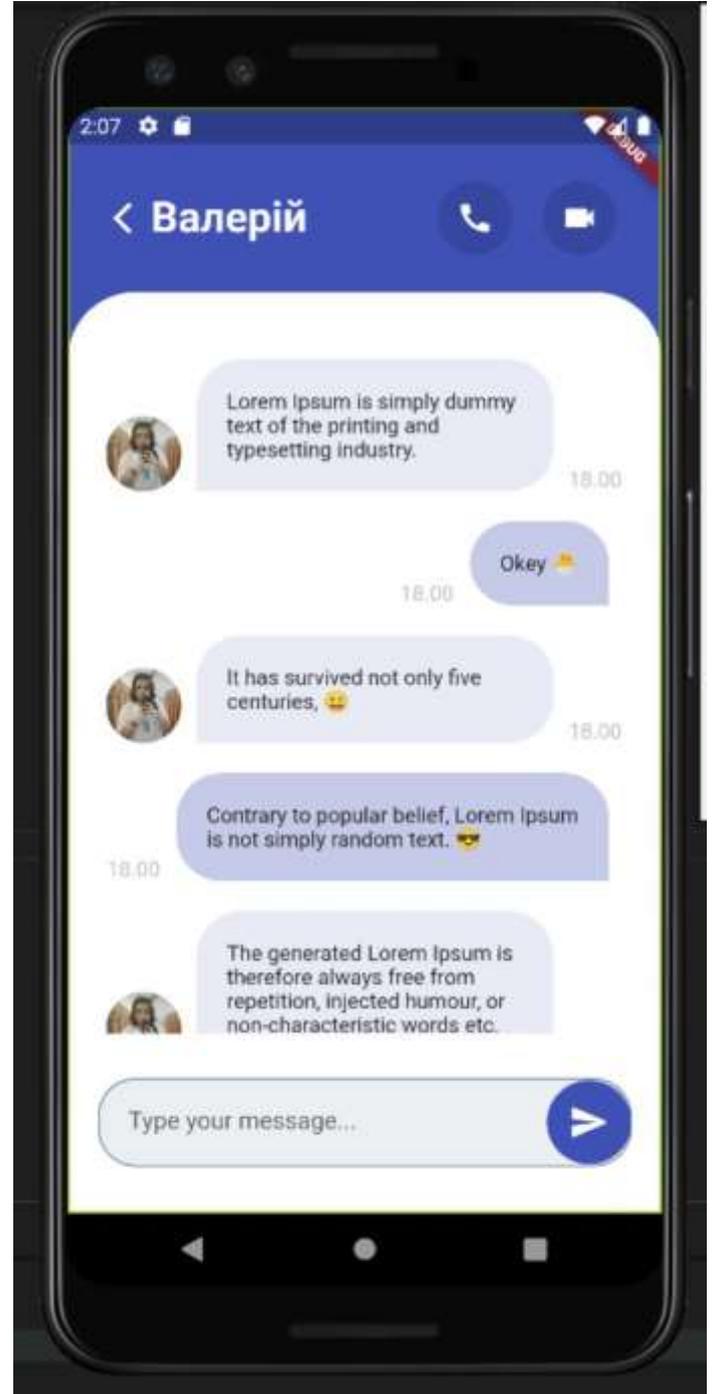
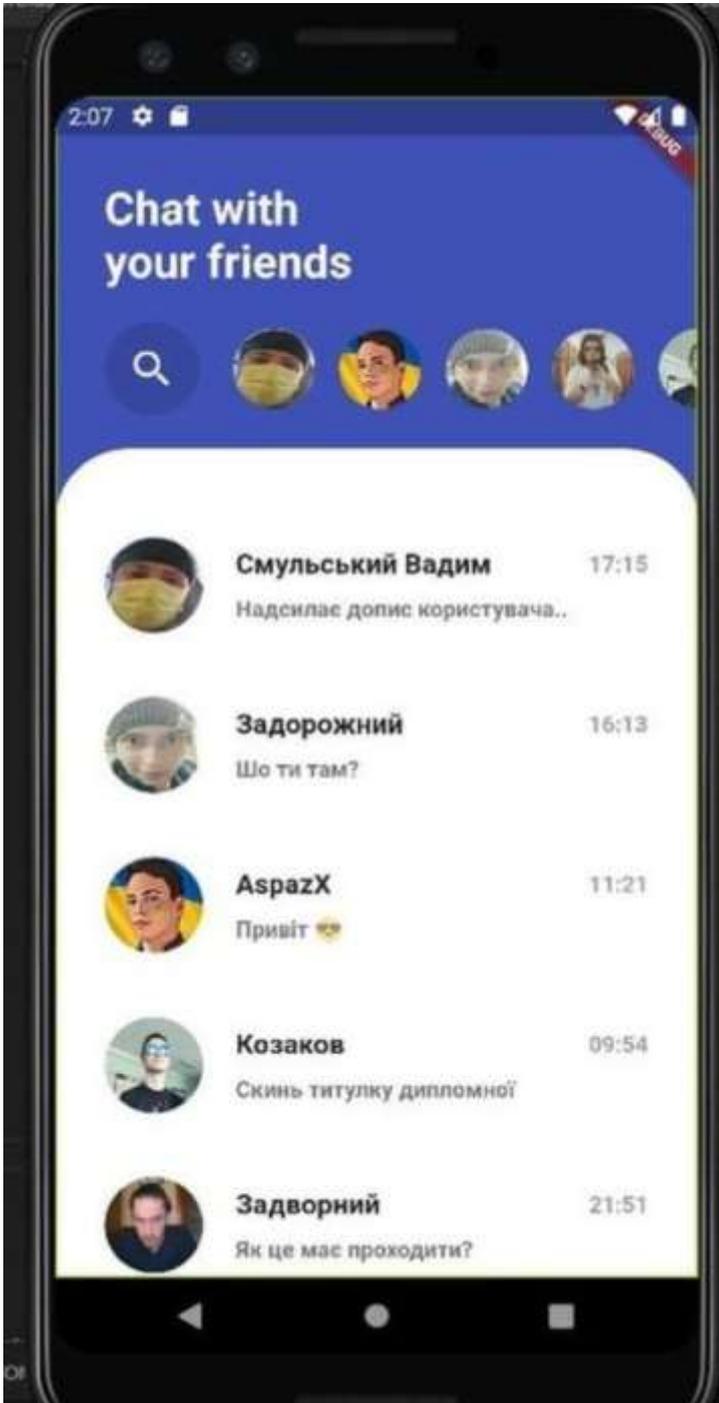
ДОДАТОК Б

Сторінка з відео



ДОДАТОК В

Сторінка «чат»



ДОДАТОК Г

Сторінка з лайфхаками

