

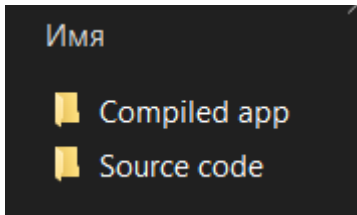
Інструкції з використання програми

Для початку потрібно завантажити програму за посиланням:

https://drive.google.com/file/d/1Wey8jPdrYPNuT7kqSprLsXBSX91T46H4/view?usp=share_link

В zip архіві ви побачите дві папки:

1. Compiled app
2. Source code

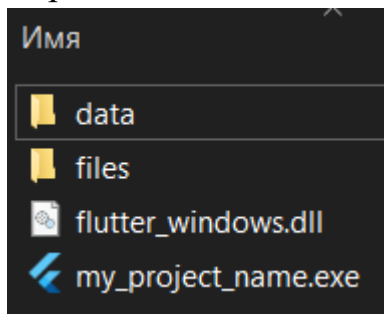


В папці “Compiled app” знаходиться вже скомпільована і готова до використання програма.

В папці “Source code” знаходиться вихідний код програми.

Розглянемо варіант використання вже скомпільованої програми з папки “Compiled app”.

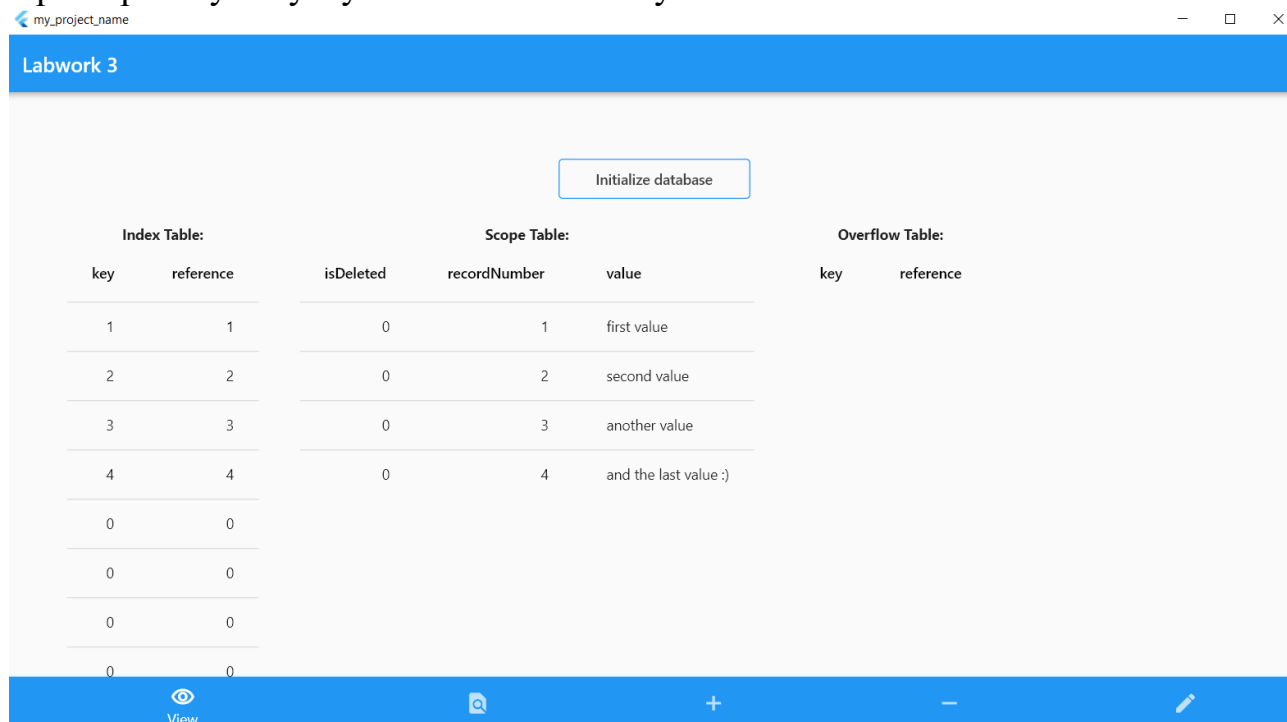
Усередині папки знаходяться наступні файли:



1. data – файли, необхідні для функціонування програми
2. files – файли індексної області, основної області та області переповнення
3. flutter_windows.dll — бібліотека, необхідна для функціонування програми
4. my_project_name.exe — програма

Запустимо програму. Вона має зручний та зрозумілий інтерфейс.

При першому запуску ми побачимо наступне:



Це меню перегляду БД. Бачимо, що деякі дані вже занесені в нашу базу даних.

В меню “Find”, ми можемо знайти потрібний нам запис за ключем індексної області.

До знаходження:

my_project_name

Labwork 3

Index table key

Find

To find a record press a corresponding button

Database

Index Table:

key	reference
1	1
2	2
3	3

Scope Table:

isDeleted	recordNumber	value
0	1	first value
0	2	second value
0	3	another value

Overflow Table:

key	reference
-----	-----------

Find

+

-

Після знаходження:

my_project_name

Labwork 3

Index table key2

Find

Found record:
0 : 2 : second value

Database

Index Table:

key	reference
1	1
2	2

Scope Table:

isDeleted	recordNumber	value
0	1	first value
0	2	second value

Overflow Table:

key	reference
-----	-----------

Find

+

-

В меню “Add” ми можемо додати запис в нашу БД.
До додавання запису:

my_project_name

Labwork 3

Value

new record

Add

Database

Index Table:

block number	key	reference
1	1	1
1	2	2
1	3	3
1	4	4
1	0	0

Scope Table:

isDeleted	recordNumber	value
0	1	first value
0	2	second value
0	3	another value
0	4	and the last value :)

Overflow Table:

key	reference
-----	-----------

+

Add

—

Після додавання запису:

my_project_name

Labwork 3

Value

Add

Database

Index Table:

block number	key	reference
1	1	1
1	2	2
1	3	3
1	4	4
1	5	5

Scope Table:

isDeleted	recordNumber	value
0	1	first value
0	2	second value
0	3	another value
0	4	and the last value :)
0	5	new record

Overflow Table:

key	reference
-----	-----------

+

Add

—

В меню “Delete” ми можемо видалити запис з БД за номером запису в основній області.

До видалення запису:

my_project_name

Labwork 3

Scope table record number

5

Delete

Database

Scope table record number:

key	reference
1	1
2	2
3	3
4	4
5	5

Scope Table:

isDeleted	recordNumber	value
0	1	first value
0	2	second value
0	3	another value
0	4	and the last value :)
0	5	new record

Overflow Table:

key	reference
-----	-----------

+

Remove

Після видалення запису:

my_project_name

Labwork 3

Scope table record number

Delete

Database

Scope table record number:

key	reference
1	1
2	2
3	3
4	4
0	0

Scope Table:

isDeleted	recordNumber	value
0	1	first value
0	2	second value
0	3	another value
0	4	and the last value :)
1	5	new record

Overflow Table:

key	reference
-----	-----------

+

Remove

В меню “Edit” ми можемо змінити запис в БД за номером запису в основній області.

До зміни запису:

my_project_name

Labwork 3

Scope table index number

1

New value

changed value

Edit

Database

Scope table record number:

key	reference
1	1
2	2
3	3
4	4
0	0

Scope Table:

isDeleted	recordNumber	value
0	1	first value
0	2	second value
0	3	another value
0	4	and the last value :)
1	5	new record

Overflow Table:

key	reference
-----	-----------

👁

🔍

+

-

Edit

Після зміни запису:

my_project_name

Labwork 3

Scope table index number

New value

Edit

Database

Scope table record number:

key	reference
1	1
2	2
3	3
4	4
0	0

Scope Table:

isDeleted	recordNumber	value
0	1	changed value
0	2	second value
0	3	another value
0	4	and the last value :)
1	5	new record

Overflow Table:

key	reference
-----	-----------

👁

🔍

+

-

Edit

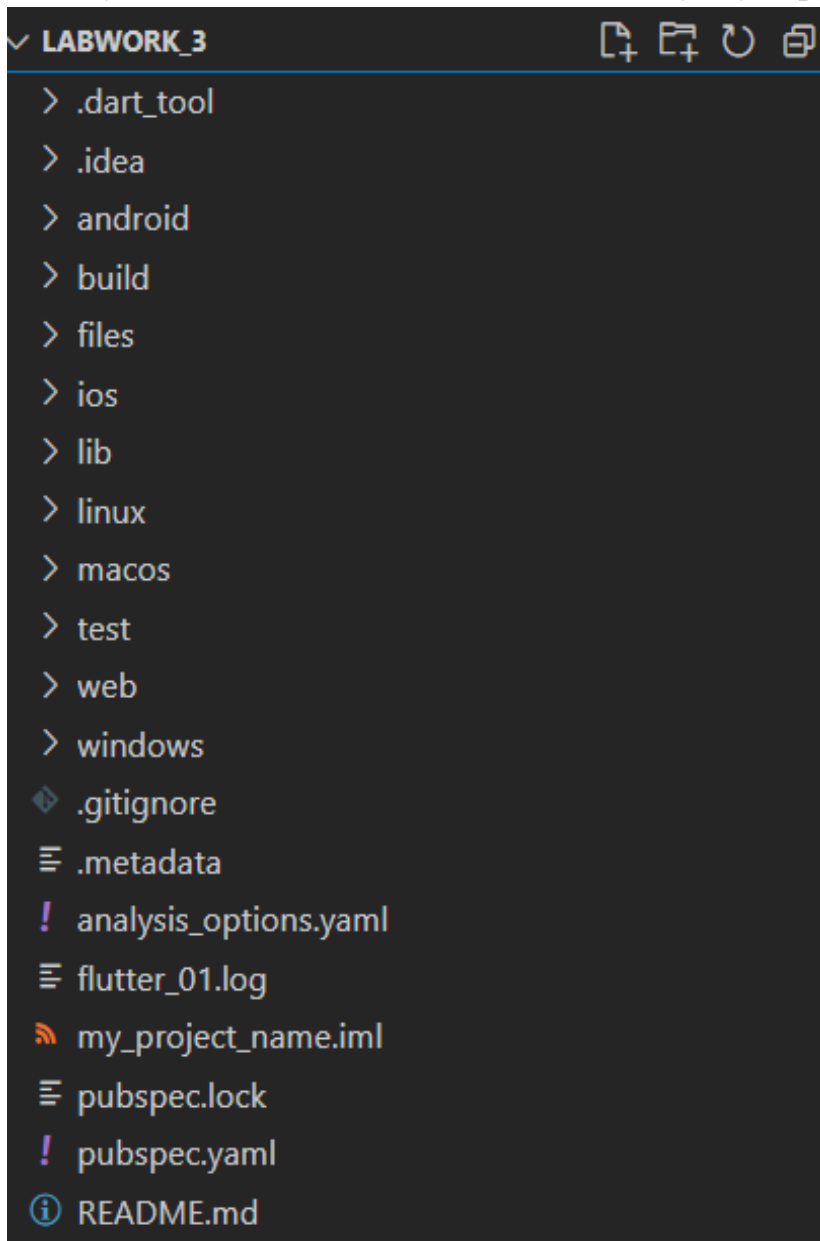
Для того, щоб зручніше переглядати і змінювати вихідний код програми, потрібно встановити Dart і Flutter.

Зробити це можна декількома способами:

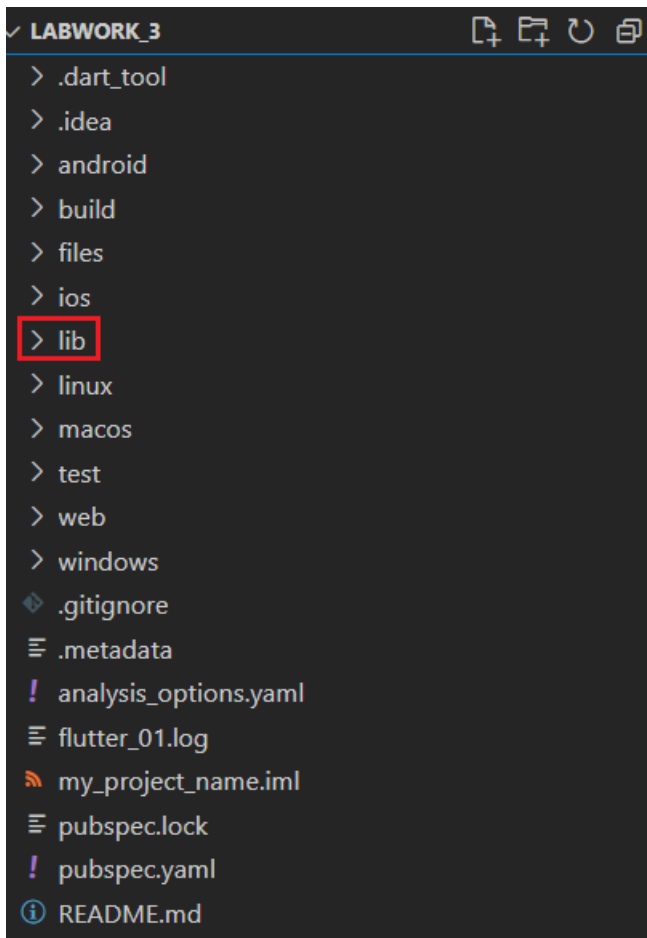
1. За допомогою офіційної інструкції
<https://docs.flutter.dev/get-started/install/windows>
Тоді в якості основної IDE буде використовуватись Android Studio
2. За допомогою неофіційних інструкцій
<https://www.youtube.com/watch?v=nILDkRhILVM>
<https://www.youtube.com/watch?v=k7vCccuDIzc>
Тоді в якості основної IDE буде використовуватись Visual Studio Code

Я обрав другий варіант, тож використовую для розробки Visual Studio Code. Подальша інструкція також буде актуальна для Visual Studio Code.

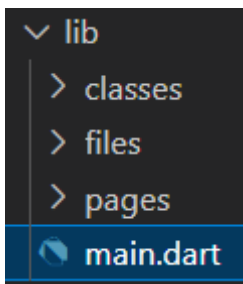
Після встановлення необхідних компонентів, відкриваємо папку з вихідним кодом у Visual Studio Code. Побачимо наступну структуру:



Для нас цікавою є лише папка “**lib**”, тому що весь необхідний нам код знаходиться там.

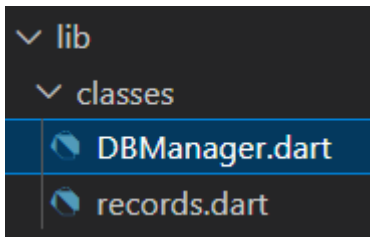


При її відкритті бачимо наступну структуру:



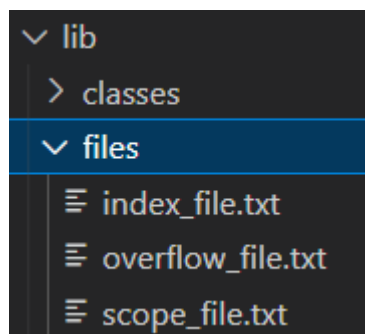
1. classes – папка з класами, які використовуються у програмі
2. files – файли індексної області, основної області та області переповнення
3. pages – меню програми
4. main.dart – основний файл програми. Контролює перехід між меню програми.

Розглянемо класи програми в папці “classes”:



1. DBManager.dart – основний клас нашої програми. За допомогою нього відбувається перегляд БД, знаходження запису, додавання, видалення, зміна записів, та всі інші операції над базою даних.
2. Records.dart – класи записів програми. Містить записи індексної області та основної області.

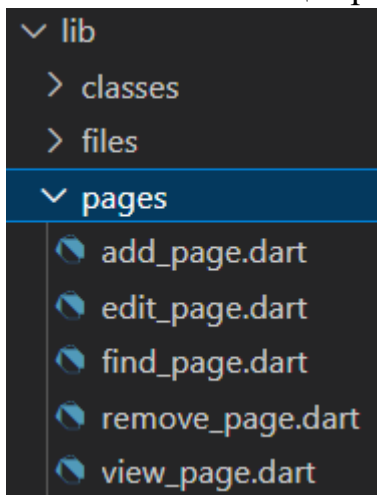
Далі розглянемо папку “files”:



Вона містить

1. файл індексної області (index_file.txt)
2. файл основної області (scope_file.txt)
3. файл області переповнення (overflow_file.txt)

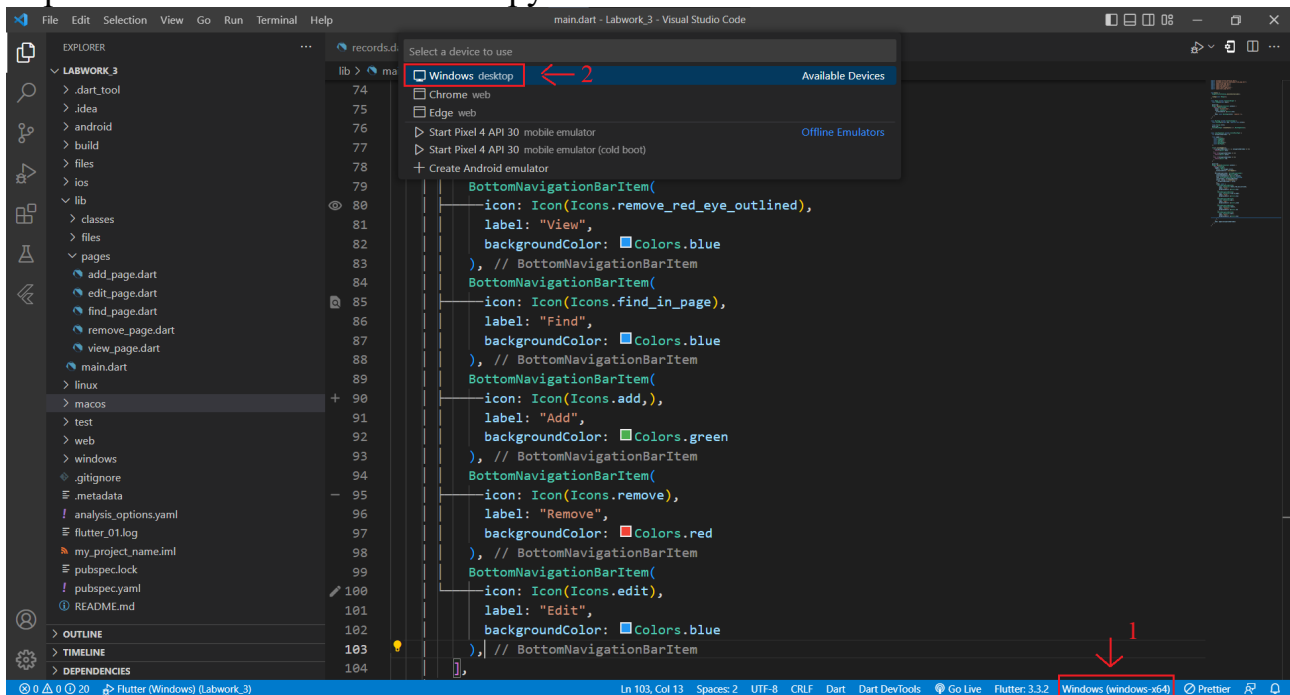
І остання папка - це “pages”



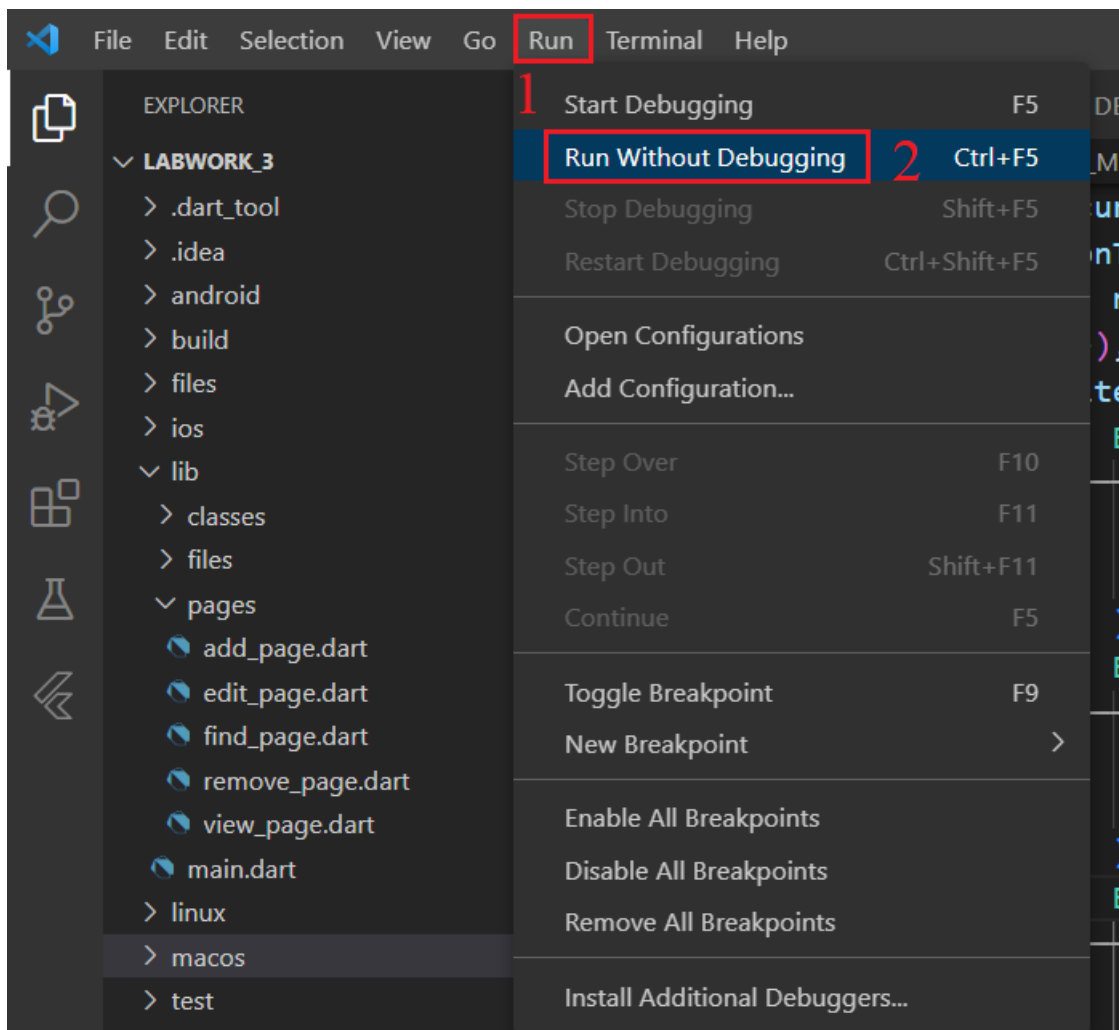
Вона містить:

1. `add_page.dart` – меню додавання елементу в бд
2. `edit_page.dart` – меню зміни елементу в бд
3. `find_page.dart` – меню знаходження елементу в бд
4. `remove_page.dart` – меню видалення елементу з бд
5. `view_page.dart` – меню перегляду елементів в бд

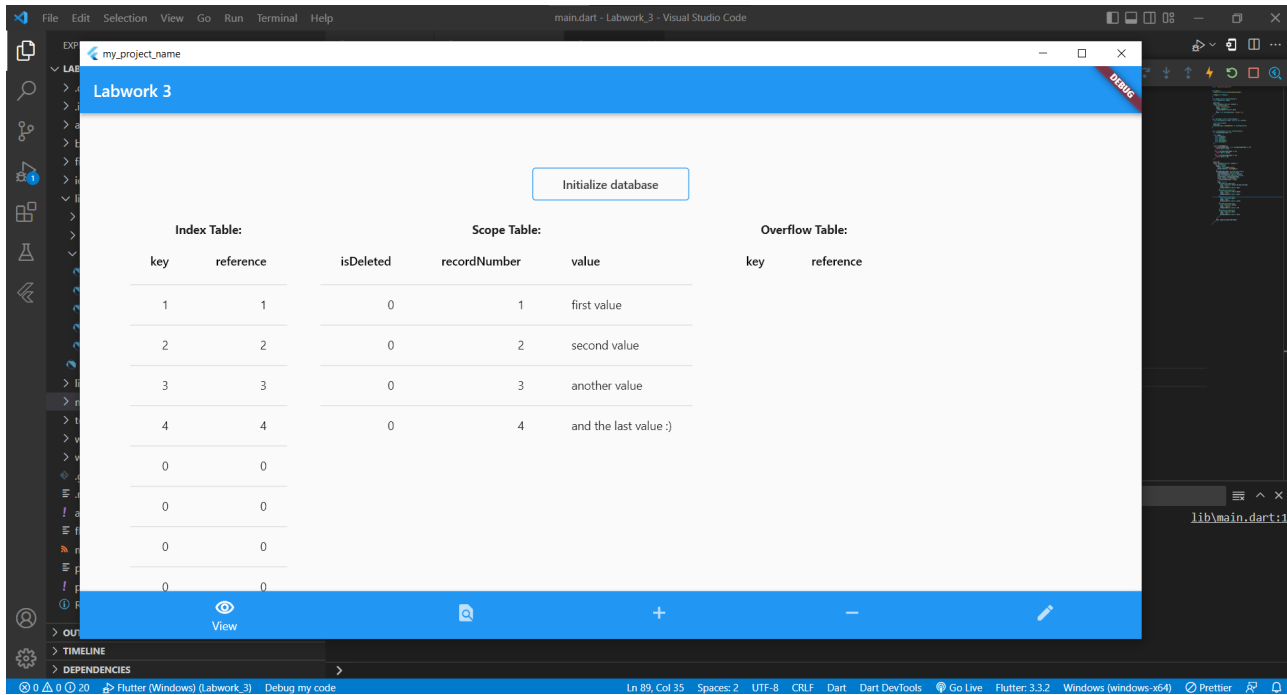
Запустимо наш код. Програма створювалась під ОС Windows, то ж потрібно обрано саме його в меню вибору ОС:



Після вибору ОС, запустимо програму (Run → Run Without Debugging):

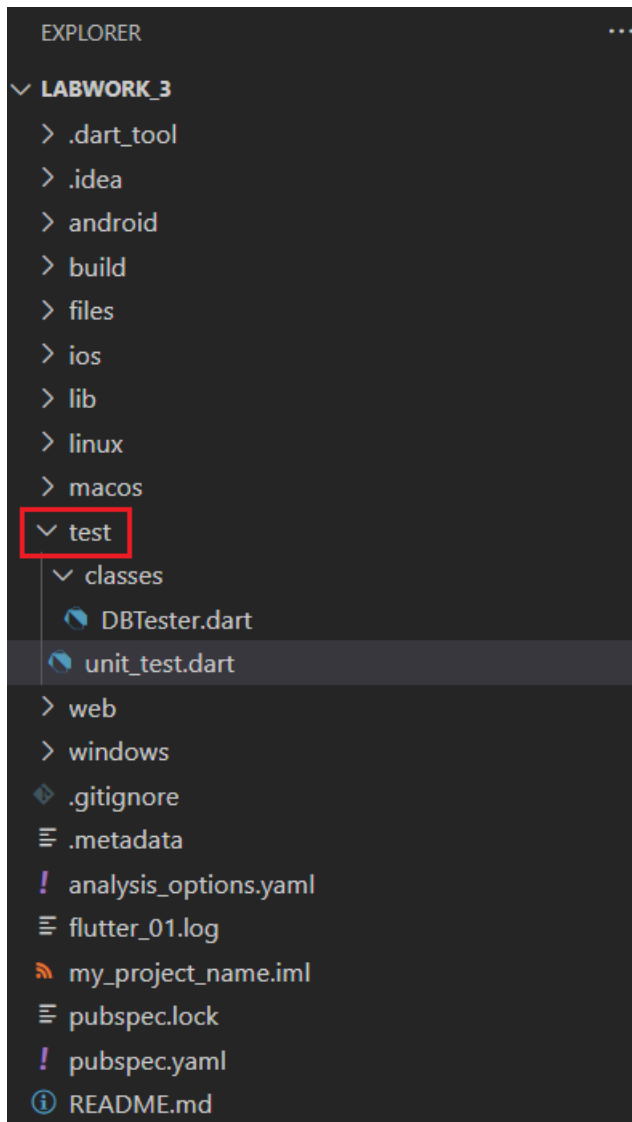


Програма успішно запустилась і функціонує:



Тестування програми

Усе, що пов'язано з тестами знаходиться в папці **“Test”**:

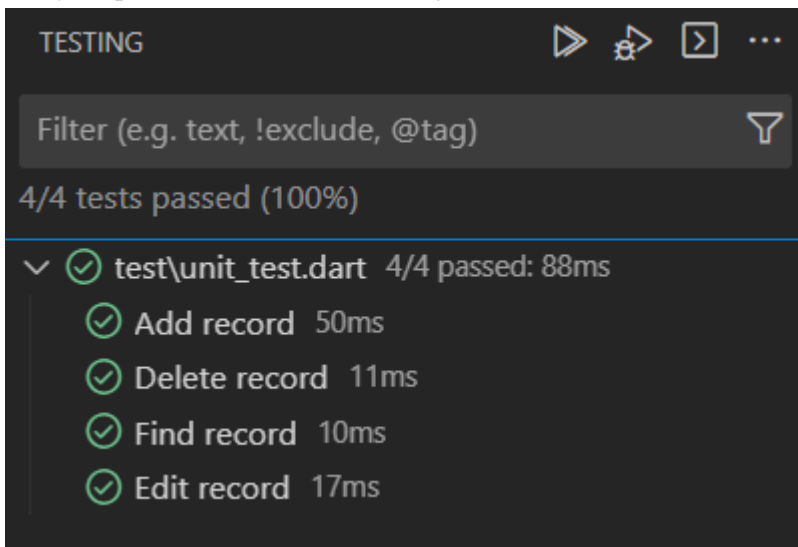


Ця папка має таку структуру:

1. `unit_test.dart` – безпосередньо файл в якому знаходяться усі unit тести
2. `classes` - папка з допоміжними класами для коректного тестування

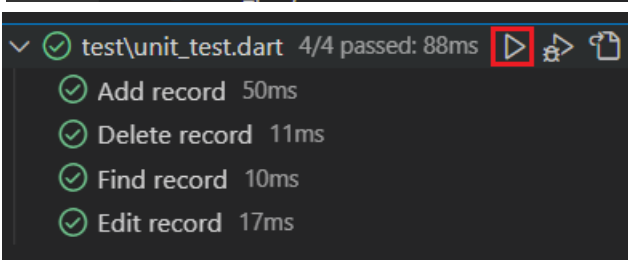
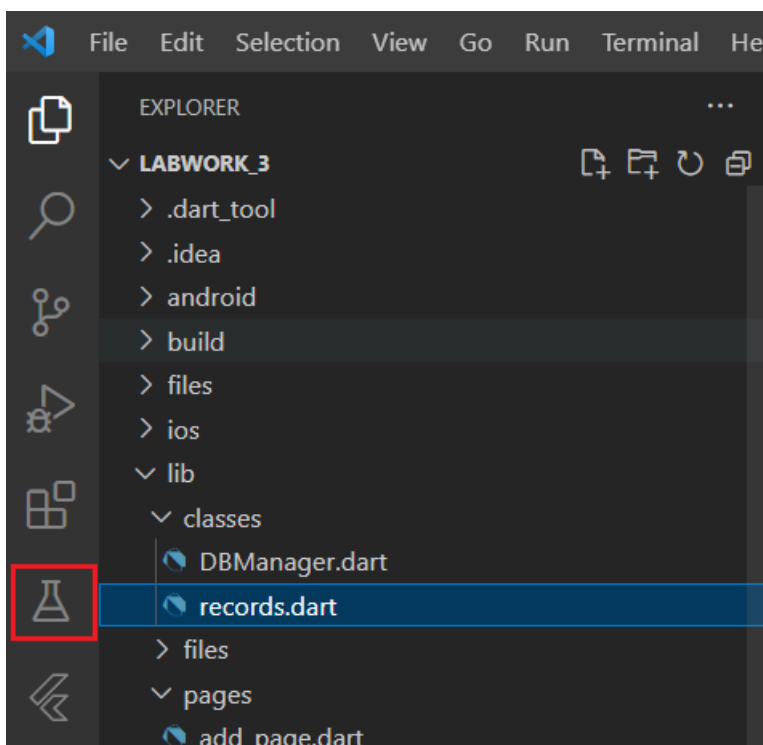
Розглянемо файл unit_test.dart:

Цей файл детально задокументований і має 4 тести:



1. Add record – тест для додавання запису в базу даних
2. Delete record – тест для видалення запису в базу даних
3. Find record – тест для знаходження запису в базу даних
4. Edit record – тест для зміни запису в базі даних

Запустимо усі unit тести, натиснувши на відповідну кнопку:



Бачимо, що всі тести були виконані успішно:

```
4/4 tests passed (100%)  
✓ test\unit_test.dart 4/4 passed: 74ms  
  ✓ Add record 47ms  
  ✓ Delete record 10ms  
  ✓ Find record 7.0ms  
  ✓ Edit record 10ms
```

Розглянемо папку “classes”.

Вона має тільки один файл: DBTester.dart. В цьому файлі знаходиться клас DBTester. Це допоміжний клас, який використовується для unit тестів, він не бере участі в основній програмі.