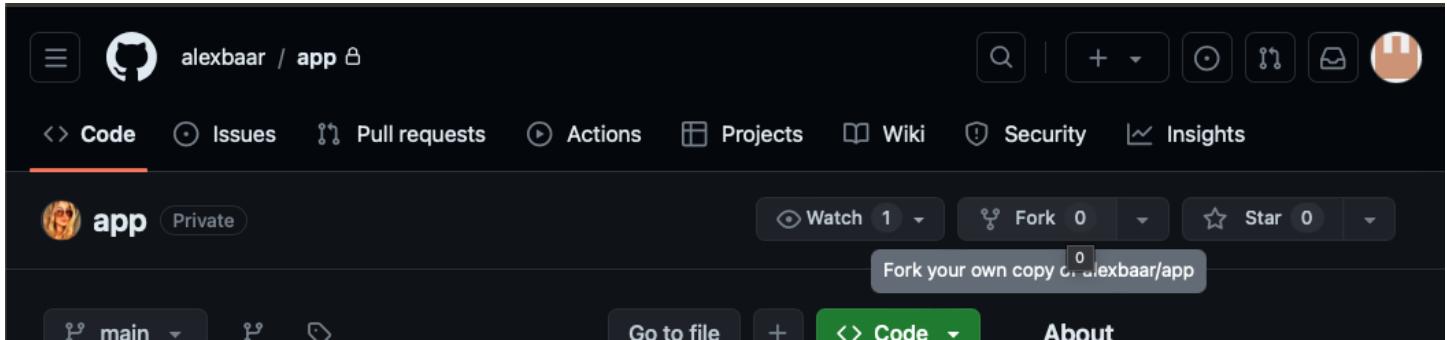


Github Version Control : Capstone team

FIRST TIME SETUP : FLUTTER APP

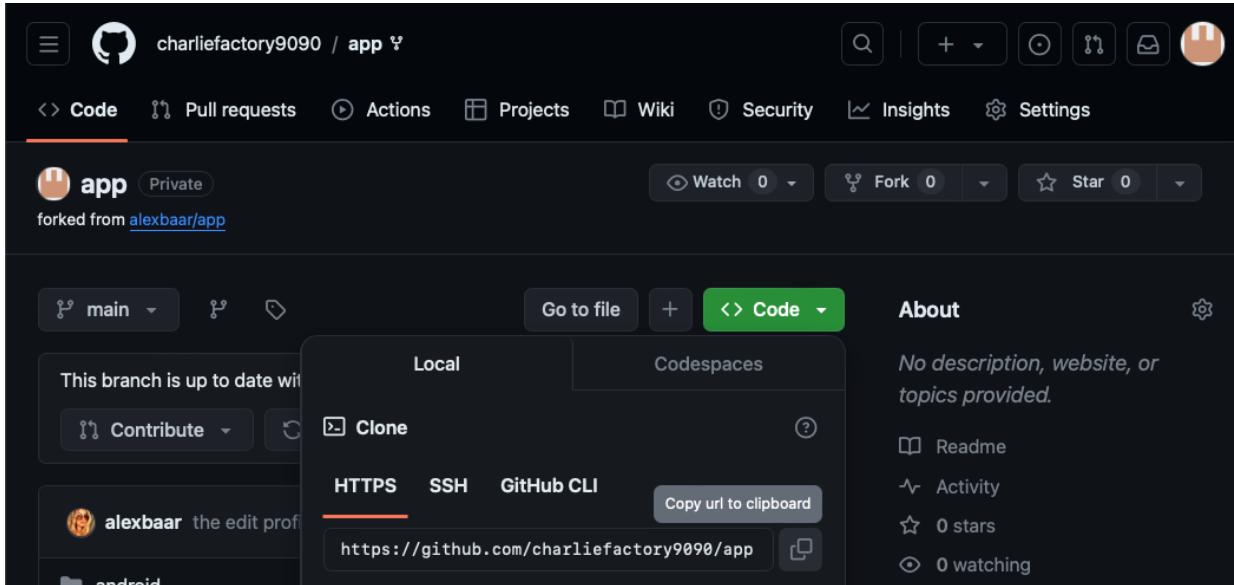
1. Fork the repo <https://github.com/alexbaar/app>

Click on fork



2. In your github navigate to that repository

3. Copy the repo's url :



4. Open terminal

5. Type: `git clone <repository_url>`

so for me it is: `git clone https://github.com/charliefactory9090/app.git`

6. Navigate to that cloned repo : `cd app`

The repo name is the last name that comes after / in the url you copied earlier

7. Set your github details :

```
git config --global user.name "your_github_username"
```

```
git config --global user.email your_github_email@gmail.com
```

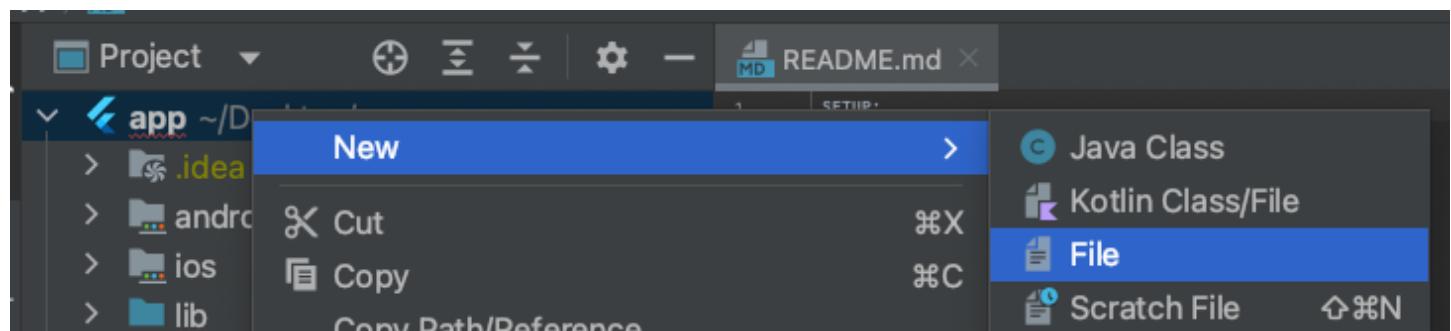
8. Make sure you are in main branch

```
type git checkout main
```

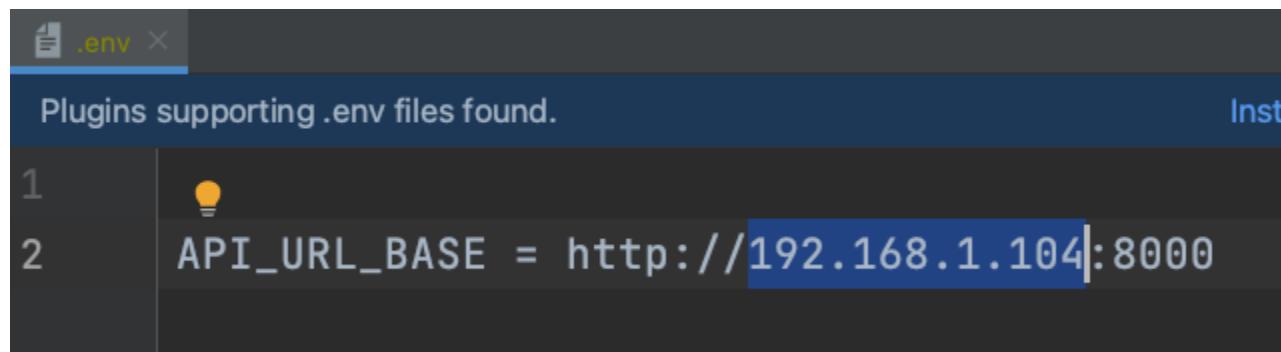
9. Being inside the app folder, open the project in Android studio :

```
aleksandrabartosiak@Aleksandras-MacBook-Air ~ % open -a "Android Studio" .
```

10. Create the .env file at project root:



Add your network address:

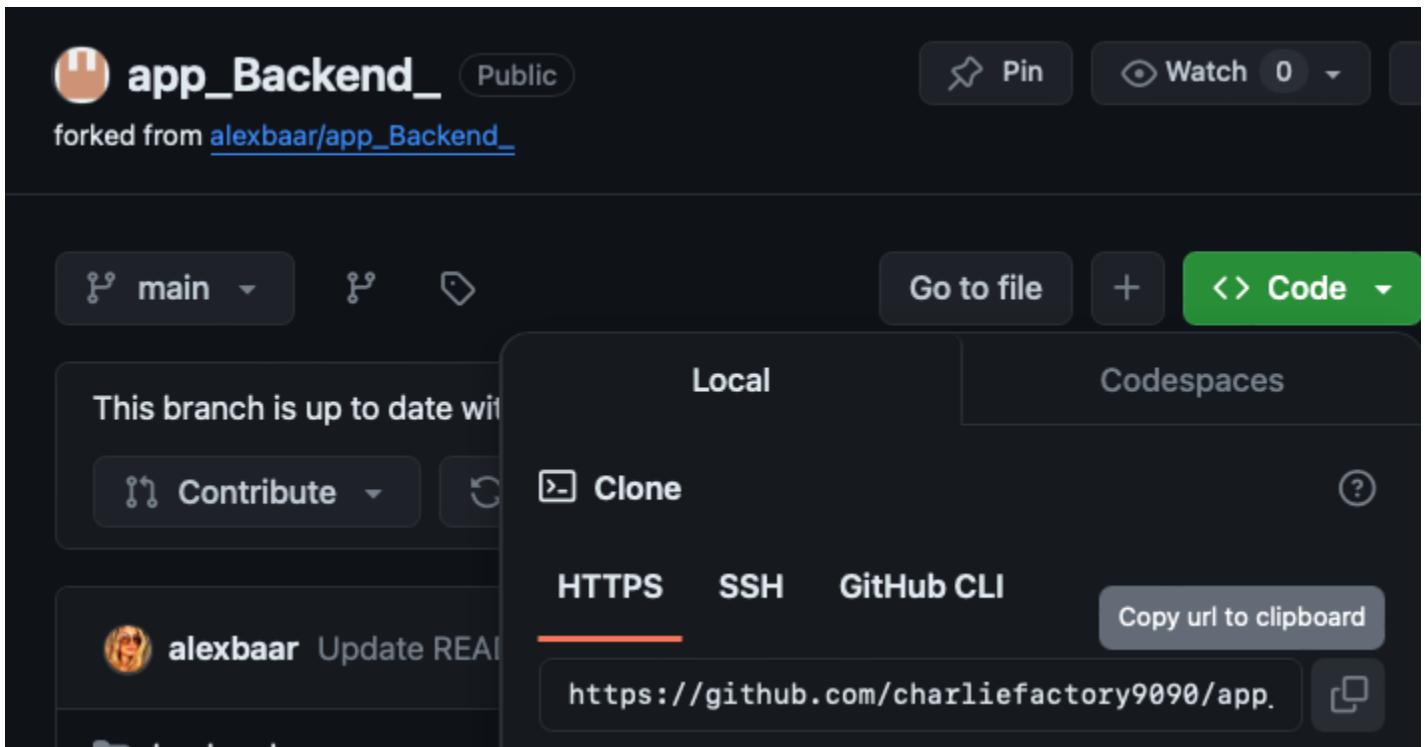


Go to pubspec.yaml and click on pub get:

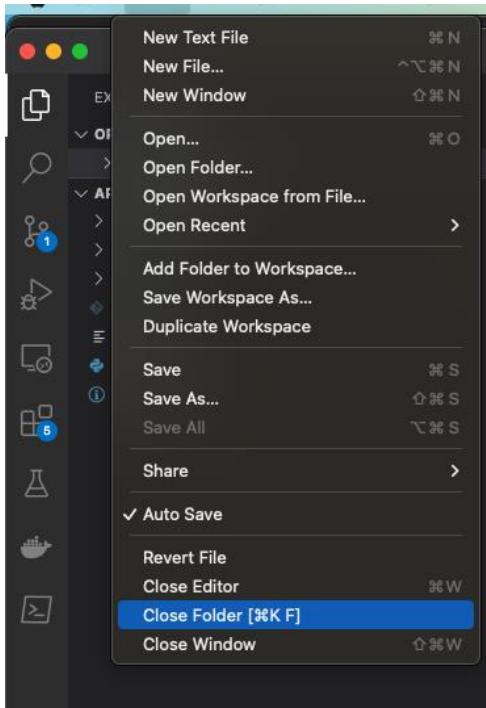
The screenshot shows the VS Code interface with the 'pubspec.yaml' file open in the editor. The status bar at the bottom right displays the text 'Pub get'. This indicates that the user has run the 'pub get' command in their terminal or IDE, which is a common step in setting up a Flutter project.

FIRST TIME SETUP : Backend

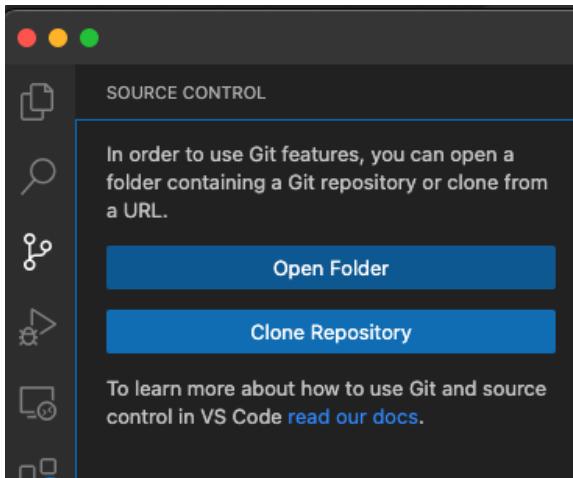
1. Fork the backend repo to your github: https://github.com/alexbaar/app_Backend_
2. Copy the URL



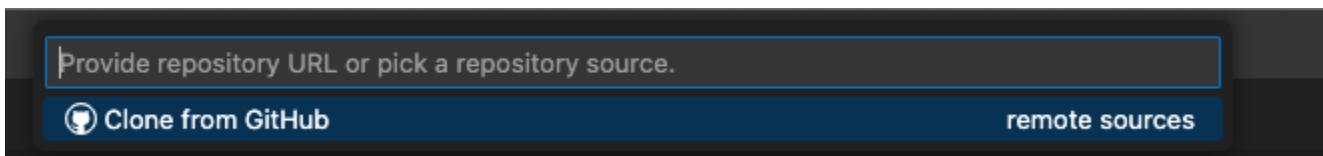
3. Open VS Code (with no project open, just empty). If you have sth open just close the folder:



4. Click on source control -> Clone repository



Click on clone from github



Paste the url:

https://github.com/charliefactory9090/app_Backend_git

charliefactory9090/app_Backend_ https://github.com/charliefactory9090/app_Backend_.git

https://github.com/charliefactory9090/app_Backend_.git

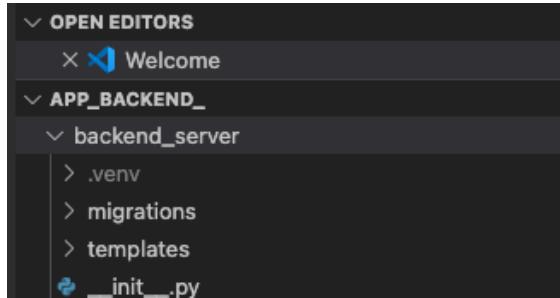
Follow the instructions, choose where to save it.

5. Now that you have the project open, you need to add a virtual environment:
6. Open terminal in VS Code
7. Type: cd backend_server

then type: python3 -m venv .venv (if you work with python, not python3, then type: python)

Then either: .venv\Scripts\activate (on Windows) or source .venv/bin/activate (on mac)

8. You can see a new file called .venv:



9. then type: pip install Django djangorestframework (see the pic with the work so far)

```
alexandrabartosiak@Aleksandras-MacBook-Air app_Backend_ % cd backend_server
alexandrabartosiak@Aleksandras-MacBook-Air backend_server % ls
__init__.py           asgi.py          migrations           serializers.py      templates        views.py
admin.py              auth_form_serializers.py  models.py         serializer_logger.py test.py
apps.py               forms.py          signals.py
alexandrabartosiak@Aleksandras-MacBook-Air backend_server % python3 -m venv .venv
alexandrabartosiak@Aleksandras-MacBook-Air backend_server % source .venv/bin/activate
(.venv) alexandrabartosiak@Aleksandras-MacBook-Air backend_server % pip install Django djangorestframework
Collecting Django
  Using cached https://files.pythonhosted.org/packages/a1/b9/5adf3f78e4c7b762eb6f1140057bd128c978c3f85c08e412f951aef65f9e/Django-4.2.11-py3-none-any.whl
Collecting djangorestframework
  Using cached https://files.pythonhosted.org/packages/c0/7e/8c45ea7f85dd5d52ceddbacc6f56ecaca21ecbf0e8c34c95618a14d5082/djangorestframework-3.15.1-py3-none-any.whl
Collecting backports.zoneinfo; python_version < "3.9" (from Django)
  Using cached https://files.pythonhosted.org/packages/4a/6d/eca004eadcbfb8bd64cc96feb9e355536147f0577420b44d80c7cac70767/backports.zoneinfo-0.2.1-cp38-cp38-macosx_10_14_x86_64.whl
Collecting sqlparse>=0.3.1 (from Django)
  Using cached https://files.pythonhosted.org/packages/98/5a/66d7c9305baa9f11857f247d4ba761402cea75db6058ff850ed7128957b7/sqlparse-0.4.4-py3-none-any.whl
Collecting asgiref<4,>=3.6.0 (from Django)
  Using cached https://files.pythonhosted.org/packages/39/e3/893e8757be2612e6c266d9bb58ad2e3651524b5b40cf56761e985a28b13e/asgiref-3.8.1-py3-none-any.whl
Collecting typing_extensions>=4; python_version < "3.11" (from asgiref<4,>=3.6.0->Django)
  Downloading https://files.pythonhosted.org/packages/01/f3/936e209267d6ef7510322191003885de524fc48d1b43269810cd589ceaf5/typing_extensions-4.11.0-py3-none-any.whl
Installing collected packages: backports.zoneinfo, sqlparse, typing_extensions, asgiref, Django, djangorestframework
Successfully installed Django-4.2.11 asgiref-3.8.1 backports.zoneinfo-0.2.1 djangorestframework-3.15.1 sqlparse-0.4.4 typing_extensions-4.11.0
WARNING: You are using pip version 19.2.3, however version 24.0 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
(.venv) alexandrabartosiak@Aleksandras-MacBook-Air backend_server %
```

10. After installing Django framework type : cd ..

11. Then install Pillow:

python -m pip install Pillow or python3 -m pip install Pillow (check your python version)

12. Then type :

python3 manage.py makemigrations (if you work with python, not python3, then type: python)

13. After that :

```
python3 manage.py migrate
```

(if you work with python, not python3, then type: python)

Should look like that:

```
● (.venv) aleksandrabartosiak@Aleksandras-MacBook-Air backend_server % cd ..
● (.venv) aleksandrabartosiak@Aleksandras-MacBook-Air app_Backend_ % python3 -m pip install Pillow
Collecting Pillow
  Using cached https://files.pythonhosted.org/packages/af/92/f6e7372b208830fa7b8bf23758588a90ed82bf8b5584ec49f352674f9...
Installing collected packages: Pillow
Successfully installed Pillow-10.3.0
WARNING: You are using pip version 19.2.3, however version 24.0 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
● (.venv) aleksandrabartosiak@Aleksandras-MacBook-Air app_Backend_ % python3 manage.py makemigrations
No changes detected
● (.venv) aleksandrabartosiak@Aleksandras-MacBook-Air app_Backend_ % python3 manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, backend_server, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007.Alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying backend_server.0001_initial... OK
  Applying backend_server.0002_rename_name_users_username... OK
  Applying backend_server.0003_warehouse... OK
  Applying backend_server.0004_alter_warehouse_email... OK
  Applying backend_server.0005_alter_warehouse_password_alter_warehouse_username... OK
  Applying backend_server.0006_acc_details... OK
  Applying backend_server.0007_acc_details_email_acc_details_username... OK
  Applying backend_server.0008_acc_details_image... OK
  Applying backend_server.0009_alter_acc_details_image_alter_acc_details_name... OK
  Applying backend_server.0010_alter_acc_details_image... OK
  Applying sessions.0001_initial... OK
○ (.venv) aleksandrabartosiak@Aleksandras-MacBook-Air app_Backend_ []
```

14. Finally you can run the backend. Type : python3 manage.py runserver 0.0.0.0:8000

You should see:

```
PROBLEMS    OUTPUT    TERMINAL    DEBUG CONSOLE    PORTS

○ (.venv) aleksandrabartosiak@Aleksandras-MacBook-Air app_Backend_ % python3 manage.py runserver 0.0.0.0:8000
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
April 06, 2024 - 00:32:21
Django version 4.2.11, using settings 'backend_server.settings'
Starting development server at http://0.0.0.0:8000/
Quit the server with CONTROL-C.
```

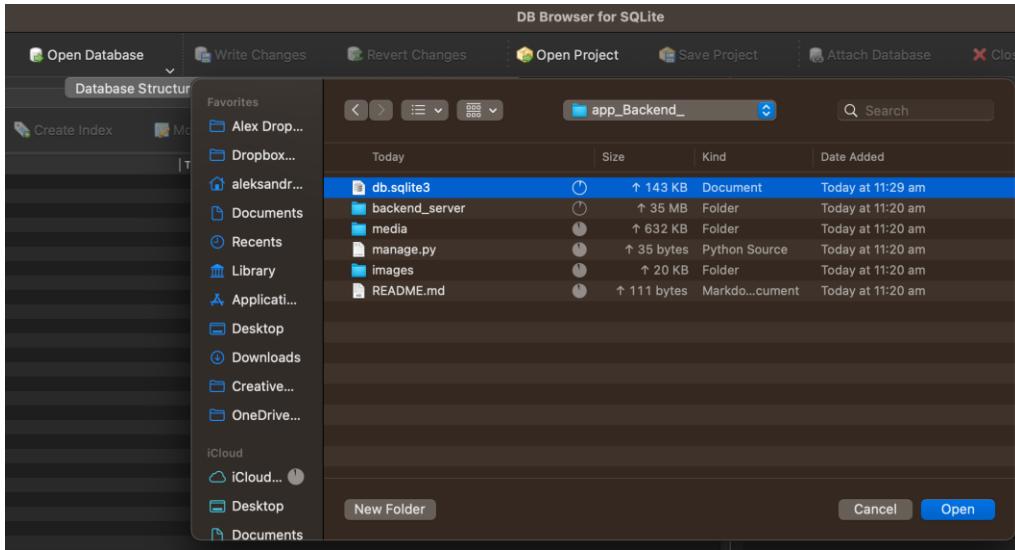
SQLite and running the project together: Android Studio + VS CODE:

Ok, so now you can run the backend and the app and see it in work. I would advise you to also download the SQLite here : <https://sqlitebrowser.org/dl/>.

It is a visual representation of your backend, so that you can easily see the tables and data.

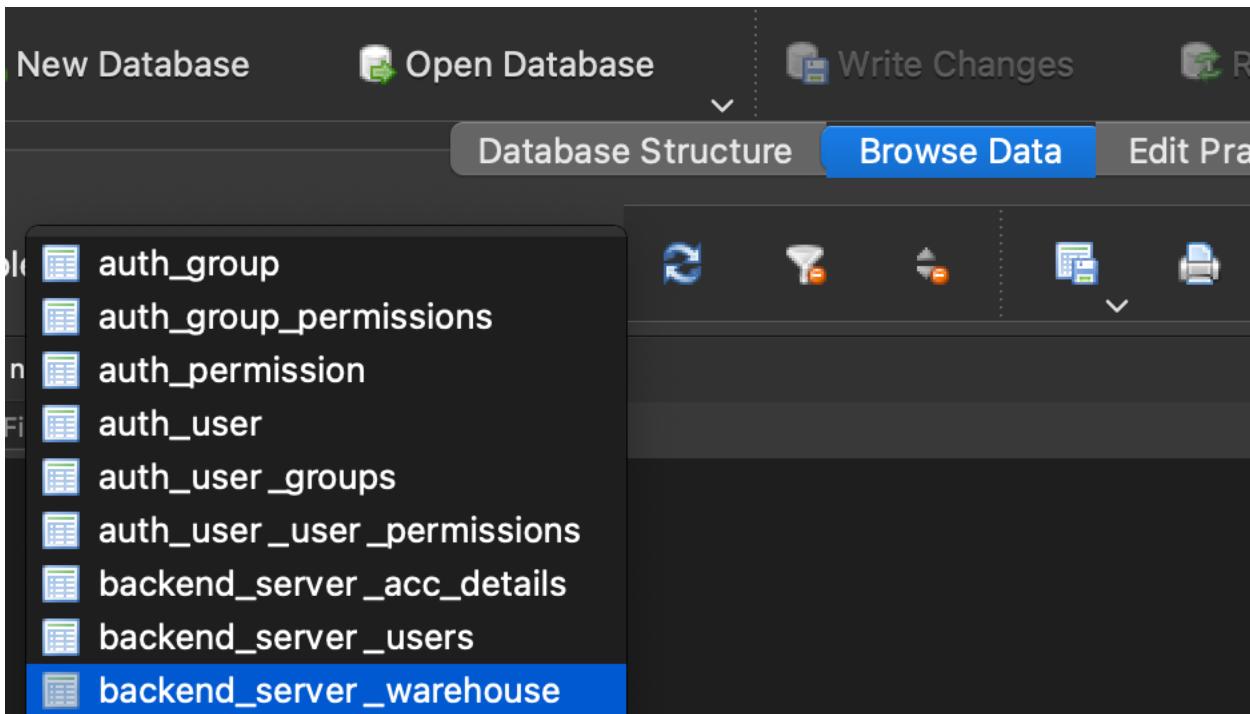
After installing open it . Click on open a database -> navigate to where you cloned the backend project from Github before, inside, double click on the 'db.sqlite3' file.

Make sure your backend is running in VS Code to see the data.

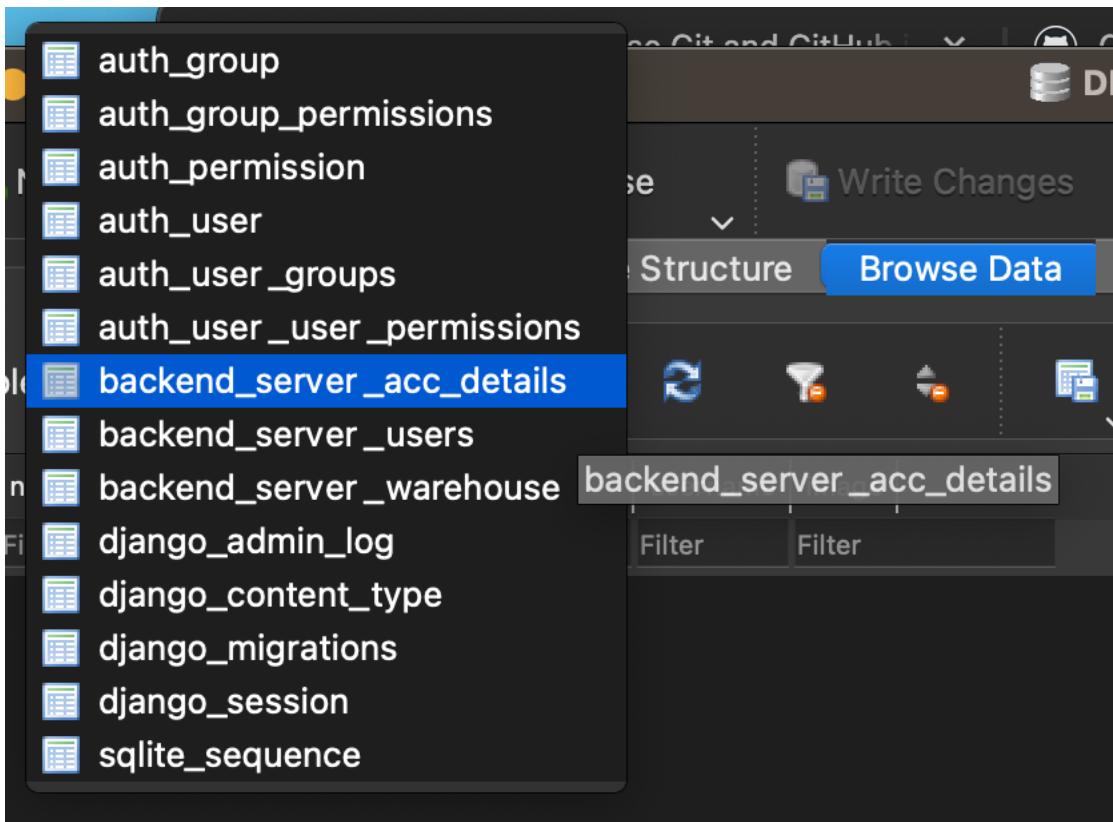


To see the tables navigate to Browse Data. From dropdown menu chose:

- **backend_server_warehouse** – to see the current login data for registered users



- **acc_details** – for all personal data associated with users, based on their email.



When you run the app 1st time they will be empty. When in Flutter you register an account, the data: username, email and password will be saved in backend_server_warehouse table.

Table: backend_server_warehouse			
id	email	password	username
...	Filter	Filter	Filter

At the same time a record will be created in acc_details table with that email and username, but not password. It is done by signals.py file from backend. All the other details will be empty as for now you have not edited the profile and not provided any additional information for your profile:

Table: **backend_server_acc_data**

id	name	surname	dob	phone_number	email	username	image
...	Filter	Filter	Filt...	Filter	Filter	Filter	Filter

So email value and username have to be unique, and email can act as a Primary Key, like a unique identifier that we can filter all information with.

So now when you run the app 1st time, register a user the SQLite tables will look like that:



Sign Up

Alexa

alexb@gmail.com

.....



.....



Password Requirements

Sign Up

Log In

```
W/dynamite: Example: soap, 0x101, background connection  
I/flutter ( 6261): Sign-up successful!
```

Go to SQLite:

Table: **backend_server_warehouse**

	id	email	password	username
...	Filter	Filter	Filter	Filter
1	1	alexb@gmail.com	Double07!	Alexa

Table: **backend_server_acc_details**

The screenshot shows a table with the following columns and data:

	id	name	surname	dob	phone_number	email	username	image
1	1		N...	NULL		alexb@gmail.com	Alexa	

So Log in with the details you set, you will see backend response in terminal:

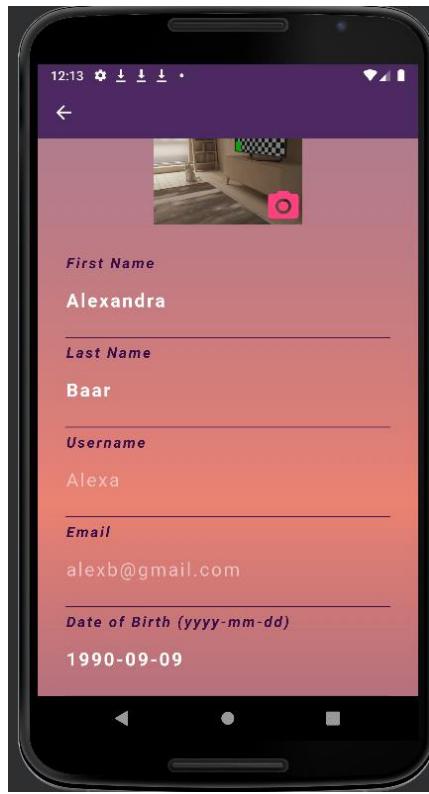
```
12:10 I/flutter ( 6261): User details: {email: alexb@gmail.com, username: Alexa, name: , surname: , dob: null, phone_number: null, image: null}
```

Go to -> Account -> Profile

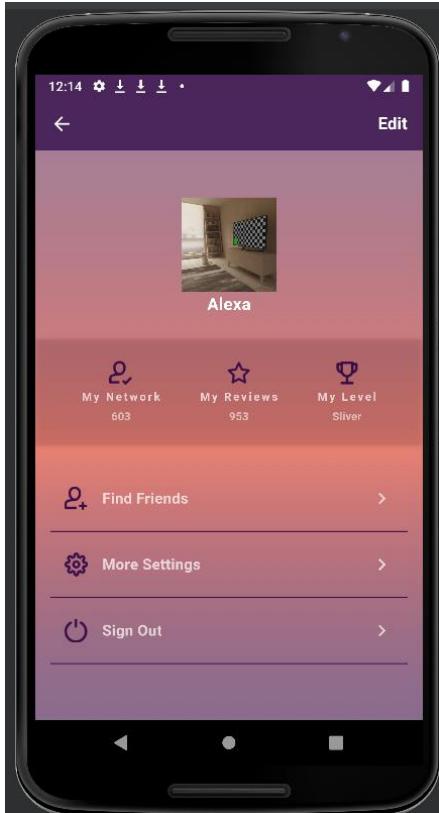
You will see a default profile picture and your username:



Edit your profile: provide all fields with N/A and a picture. If there are no pictures in the device's gallery, use camera and take a pic like that:



Save the changes: and you have now have an updated user profile:



Check the SQLite again: (hit cmd + R to refresh)

Table: [backend_server_acc_details](#) 














id	name	surname	dob	phone_number	email	username	image
...	Filter						
1	Alexandra	Baar	1990-09-09	00097655	alexb@gmail.com	Alexa	images/0a0f9a81-564c-4171-ab3e-...

CONGRATS! You have FINISHED the setup now 😊

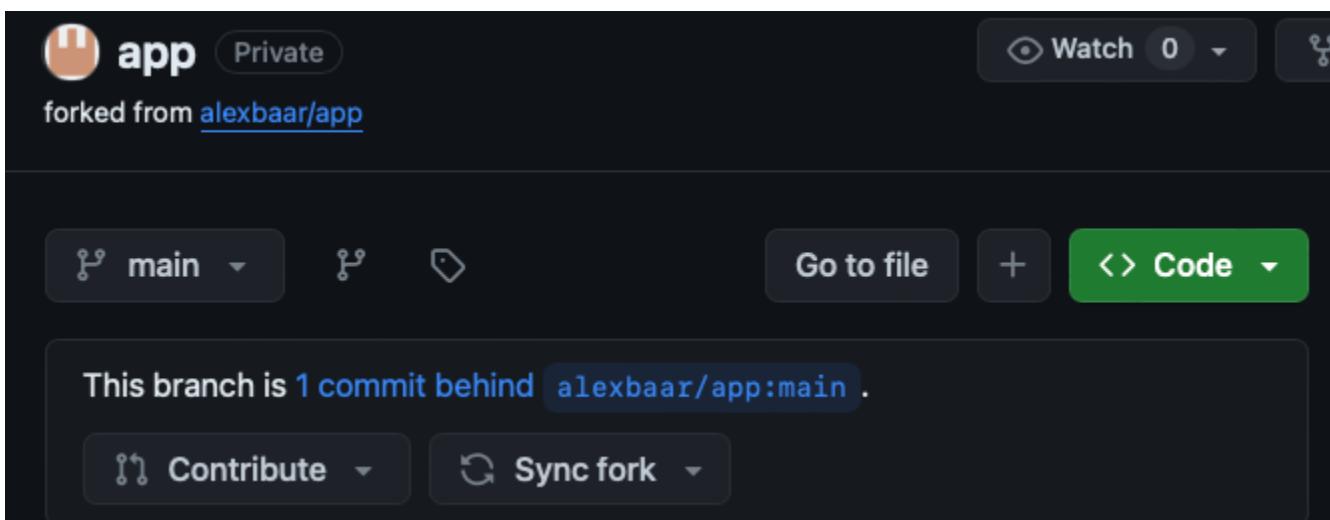
SUBMITTING A COMPLETE FEATURE : FLUTTER APP

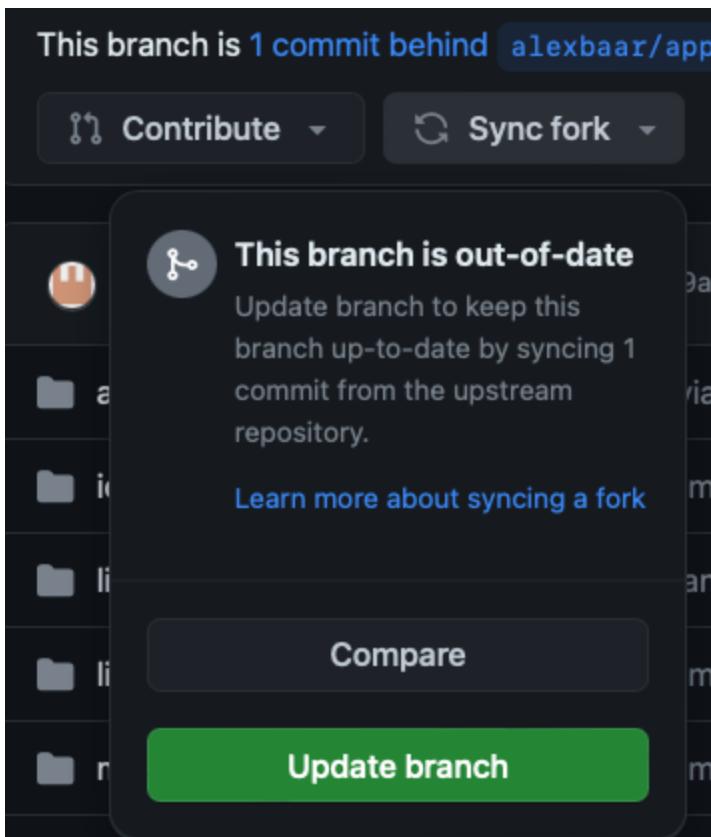
Now lets say you have worked on a feature and completed a full task. You made sure it works and want to sent those updates.

Steps here will be:

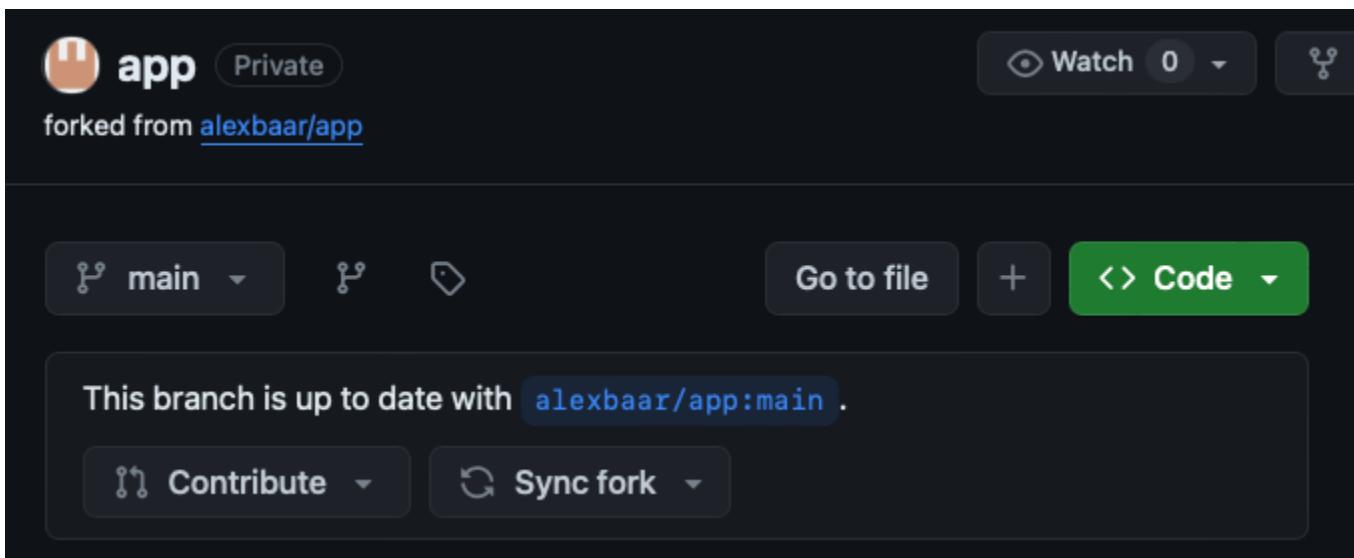
1. Check for any changes in the owners repo while you didnt work

Go to your forked repo and look for messages like ‘1 commit behind...’ like this:





If any changes were made you need to click on 'Sync fork' -> update branch



If there are no conflicts you should see that your fork is up to date. If there are conflicts please contact me as it involves going through code and reviewing changes.

2. Lets say you worked on a feature and during that time some changes were made. So your forked repo is up to date, but in your Android studio you cant see changes yet. You also have added code and other changes already. This is what you do:

- In Android Studio type:

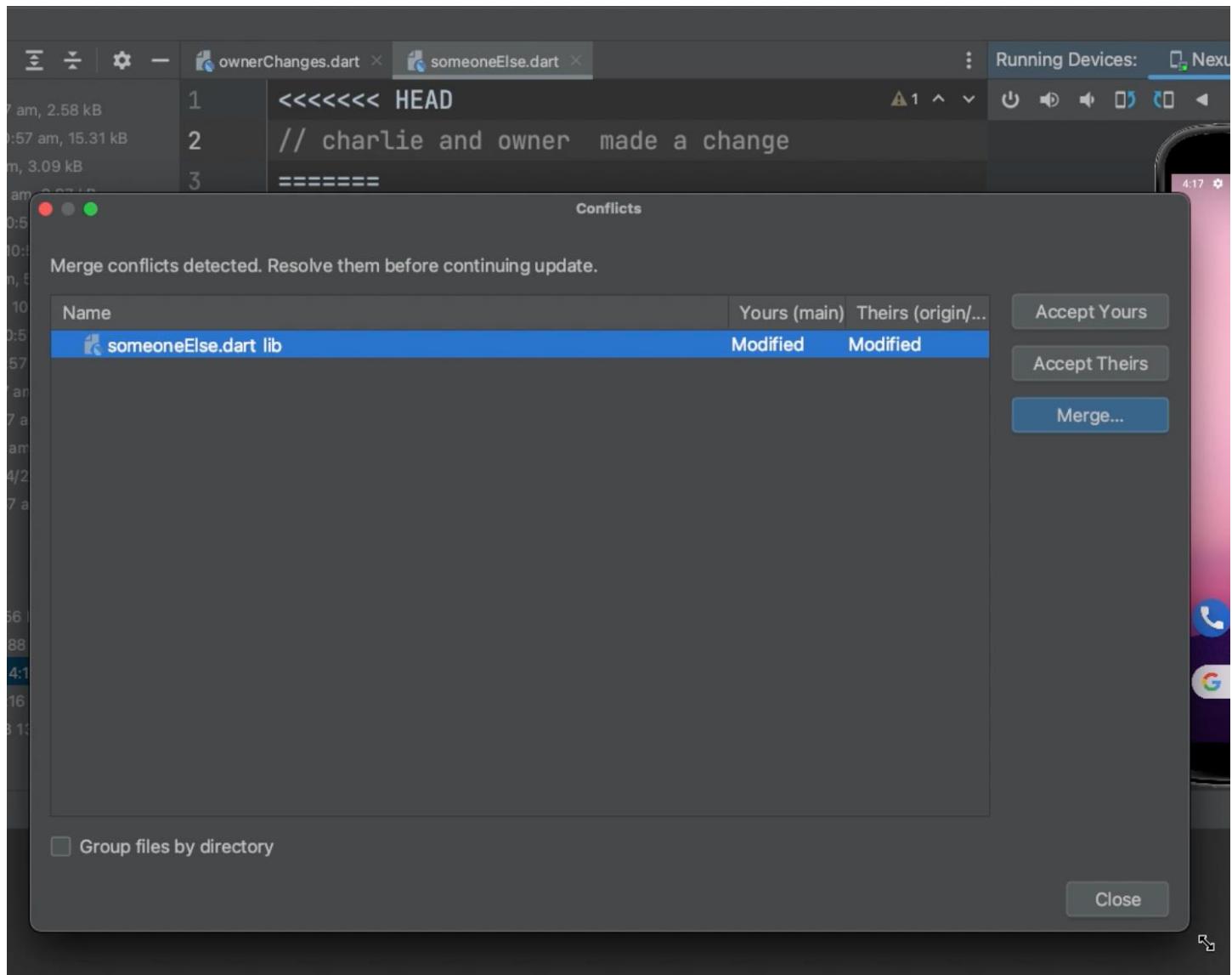
Type : Git -> commit (to prepare to send your changes)

Finally : Git -> push (send your changes to forked repo)

There might be a pop up asking if you want to merge -> click **merge**

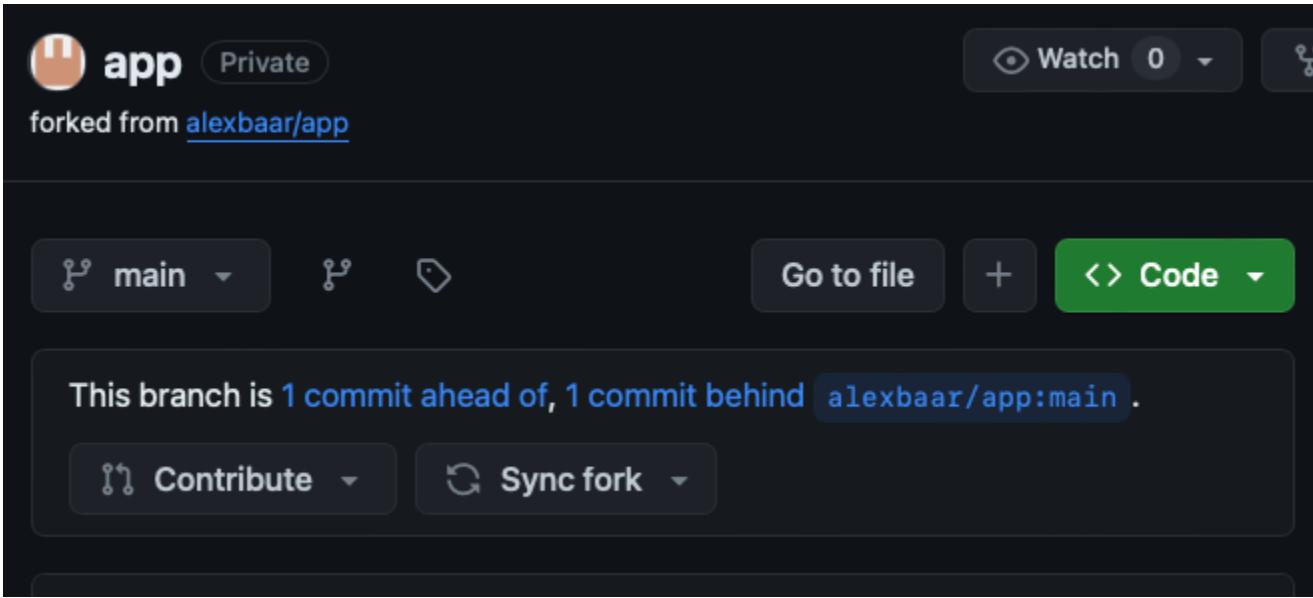
If your changes do not conflict with other changes then the **merge** will be successful, not impacting any other work.

But if meantime someone pushed a change and after pressing **merge** a screen like below will come up, please see the video to learn how to try and resolve it:



Sometimes you can resolve the conflicts simply by moving your lines of code below / over but sometimes there might be a more complicated conflict. If you cannot figure out how to handle such conflict, just send me a message.

- Finally go to your forked repo, you should see the confirmation that you sent the changes:



So now you want to send those changes to the Owner (here the owner is alexbaar):

Click on the 1 commit ahead -> create pull request

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#) or [learn more about diff comparisons](#).

base repository: alexbaar/app ▾ base: main ▾ ← head repository: charliefactory9090/app ▾ compare: main ▾

✓ Able to merge. These branches can be automatically merged.

Discuss and review the changes in this comparison with others. [Learn about pull requests](#) [Create pull request](#)

And again create pull request

The screenshot shows a GitHub pull request interface. At the top, it displays the base repository as "alexbaar/app" and the base branch as "main". It also shows the head repository as "charliefactory9" and the head branch as "charlie changes 2". A green checkmark indicates that the branches are "Able to merge". Below this, there's a section to "Add a title" with the placeholder "charlie changes 2". Under "Add a description", there's a rich text editor toolbar with options like Write, Preview, H, B, I, etc., and a large text area placeholder "Add your description here...". Below the editor, it says "Markdown is supported" and "Paste, drop, or click to add files". At the bottom, there's a checkbox for "Allow edits by maintainers" and a green button labeled "Create pull request".

If it says that you can merge automatically, great, do it: merge -> confirm merge

The fact that it merged automatically, means that the owner doesn't need to review anything. The

Conflicts when merging:

When working on a project, ideally each team member will change just a part of the project. But sometimes the work of team members can overlap on the same part.