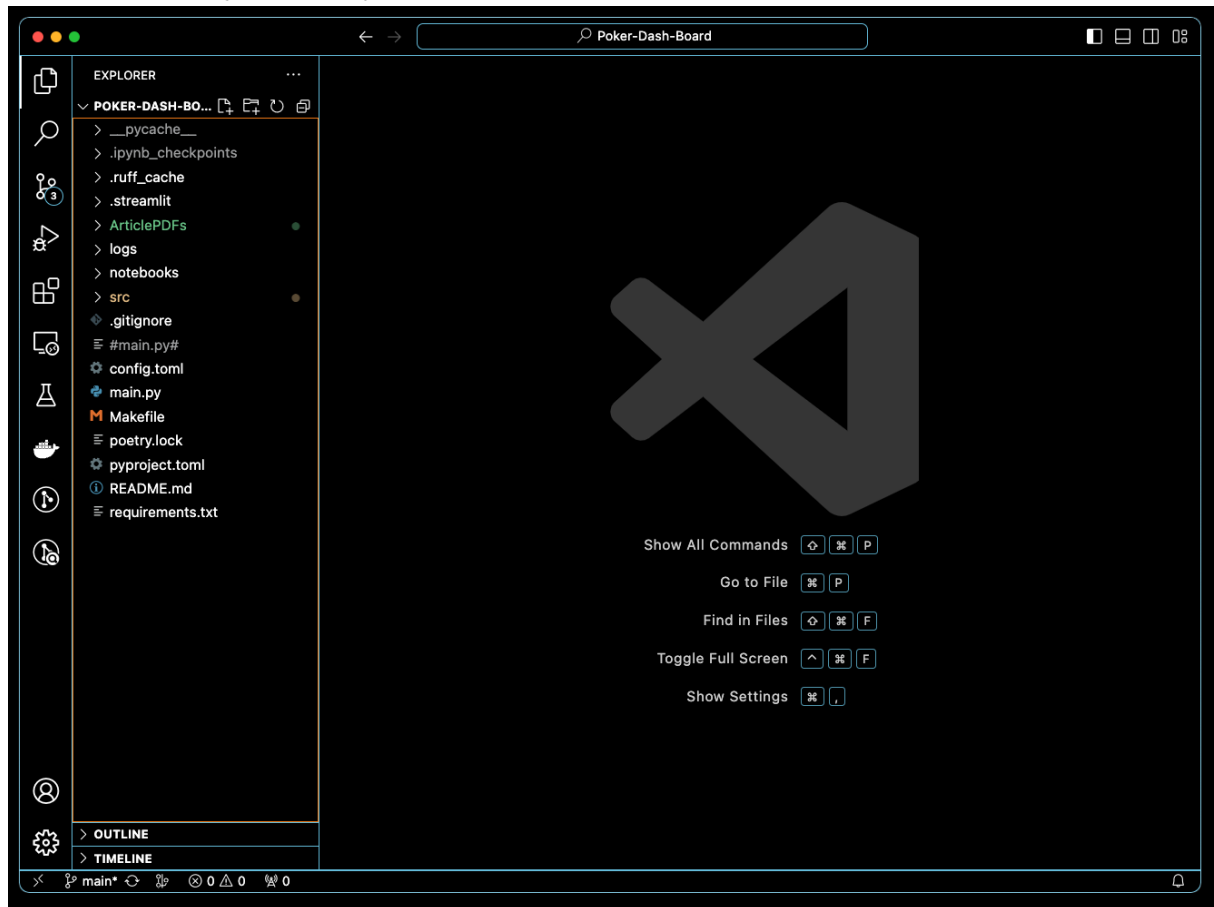


How To Add to Poker Dash

1. Open the Poker Dash project folder in your favourite text editor or IDE...



2. You will notice the following structure in the project folder:

```
.
├── ArticlePDFs
├── Makefile
├── README.md
├── config.toml
├── main.py
├── poetry.lock
├── pyproject.toml
├── requirements.txt
├── src
│   ├── __init__.py
│   ├── collection.py
│   ├── colorhash.py
│   ├── lib.py
│   ├── objects.py
│   ├── pages
│   │   ├── awards.py
│   │   ├── game_comparison.py
│   │   ├── history.py
│   │   ├── home.py
│   │   ├── multiplayer.py
│   │   ├── player_comparison.py
│   │   └── upload_data.py
│   ├── processing.py
│   ├── rating.py
│   └── streamlit.py
```

3. We will go through some of these files/folder and explain what they do:

- `ArticlePDFs`
 - Folder
 - Where I have saved some webpages and articles that have aided in the development of the project.
- `Makefile`
 - File
 - On your computer simply type in `make` in your shell (given it is Unix-like I believe) and instructions to run this application locally on your device will begin to execute. You will find the app's web interface on `http://localhost:8501/` (side note the deployed version is: <https://poker-dash-t579yhzser.streamlit.app/>).
- `config.toml`
 - File
 - Where we put the unique key Google gives us for our Google Sheet where all the data on the games are stored and input into by the user.
- `main.py`
 - File
 - This is our main entry point into the program and where we initialise and run all other files from.
- `poetry.lock` and `pyproject.toml`
 - Files
 - These files define the virtual environment created to run the app.
- `requirements.txt`
 - File
 - Lists Python packages and specific versions needed for Streamlit to deploy and make our app available on the web @ <https://poker-dash-t579yhzser.streamlit.app/>
- `src/collection.py`
 - Where we grab the data from Google Sheet and process it into a Pandas Dataframe
- `src/colorhash.py`
 - Functions for giving players unique colours (visible on dark backgrounds) based on a hashing of the player names.
- `src/lib.py`
 - A collection of useful functions that don't fit in specifically anywhere
- `src/objects.py`
 - Where our main classes are defined which allow us to more easily create informative and interesting visualisations. Our main classes are `Player` and `Game`.
 - A `Player` is assigned to one player from the game logs, and vice versa for `Games`.
 - We also want to add `PlayerGroup` and `GameGroup` later to provide collective analytics on multiple players/users respectively.
- `src/processing.py`
 - Here we have the functions which do the work of transforming our tabular data we get from the `collection.py` file into our `Player` and `Game` objects.
- `src/rating.py`
 - The GELO system logic
- `src/streamlit.py`
 - Where our Web frontend gets initialised and which then runs all the files inside of the `pages` folder. Where our page menu is defined.
- `src/pages/home.py`
 - Home page and where the snadwhic
- `src/pages/player_comparison.py`
 - Players page
- `src/pages/game_comparison.py`
 - Games page
- `src/pages`
 - Where pages on the streamlit website go. A couple of place holder files have been created in preparation for future development.

4. You can think of the execution of files when the program starts as roughly going...

1. `src/collection.py`
2. `src/processing.py`
3. `src/streamlit.py`
4. `src/pages/home.py`

5. We make heavy use of Streamlit, a Python web graphics library for data science. You will find all the relevant information in the very good documentation they keep as well as the streamlit forums (through endless googling).

- [Streamlit documentation](#)
- [Streamlit discussion forum](#)

6. I have also made a lot of use of Pandas, Matplotlib, and NumPy.

7. In order to Add to this project I would go through the code and see if I could follow the logic laid out above and look through the code to see where I want to add some functionality.