

Movie watchlist app written in python and make a interface with graphical user interface (GUI).

- Introduction-the project's aim
- Scope of functionalities
- Tkinter
- Tester
- Backlog

Introduction-the project's aim:

The project I have built is a movie watchlist app. This app allows users to:

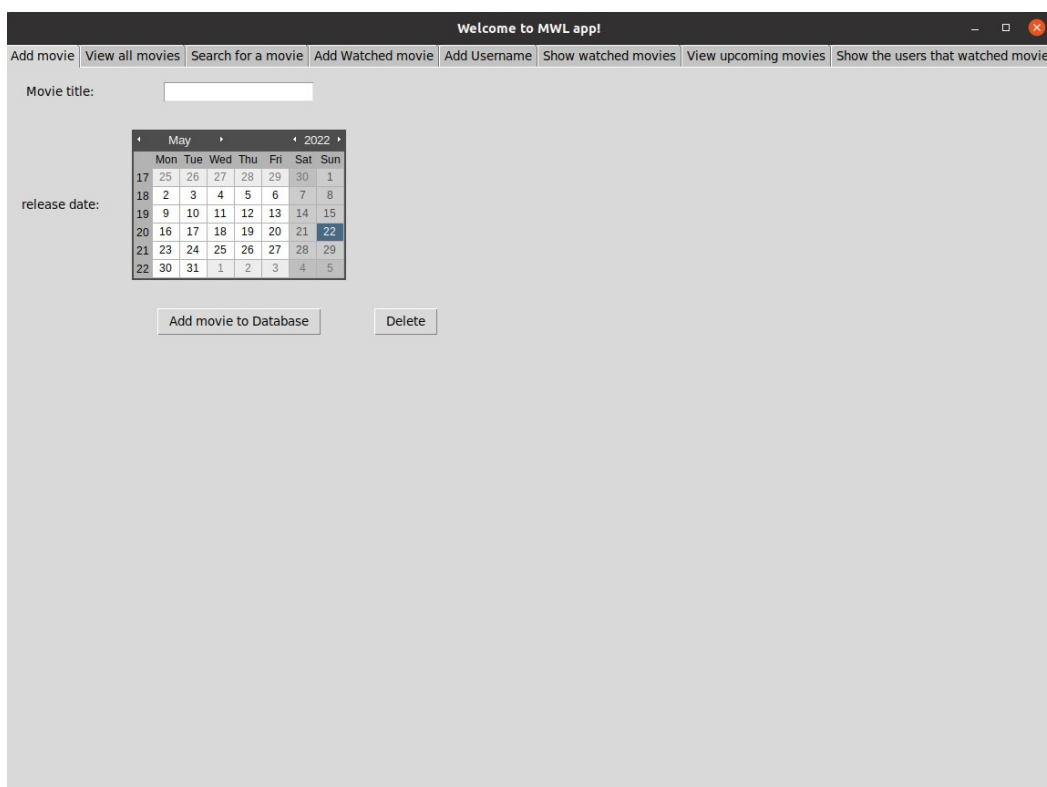
- Keep track of movies they're interested in and their release dates.
- Store the movies a user has seen out of all movies in the database.
- Add a new user to keep track of their watched movies.

There are so many streaming services to watch movies and series like Netflix and HBO, witch makes it hard to choose a movie when several people want to watch a movie that none of them have seen. That's why I decided to make this app.

Scope of functionalities:

In this project I am using Python and SQLite as a database and using python's sqlite3 module.

The project menu consists of eight sections(TAB) which you can see below.



The project contains a python file called Main that must be executed to use this application.

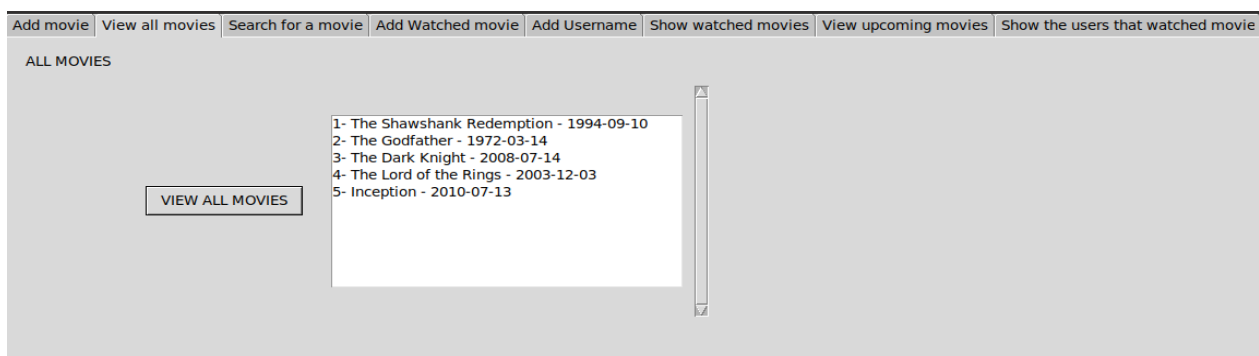
```
(venv) arash@lenovo ~/Downloads/u06_app main ± python main.py
```

Another Python file you see in this project is called Database.py, which contains all the queries needed to build the database, as well as all the functions used to connect to the database.

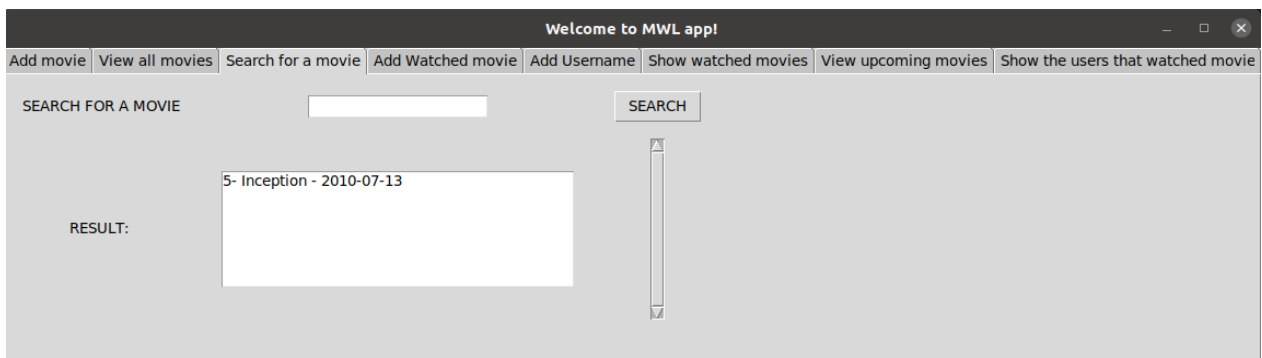
The project contains a directory called Tab, which contains eight Python files. Each tab of the user menu is located separately in these files.

Tab1: To add a movie and the release date of the movie or delete movie from database.

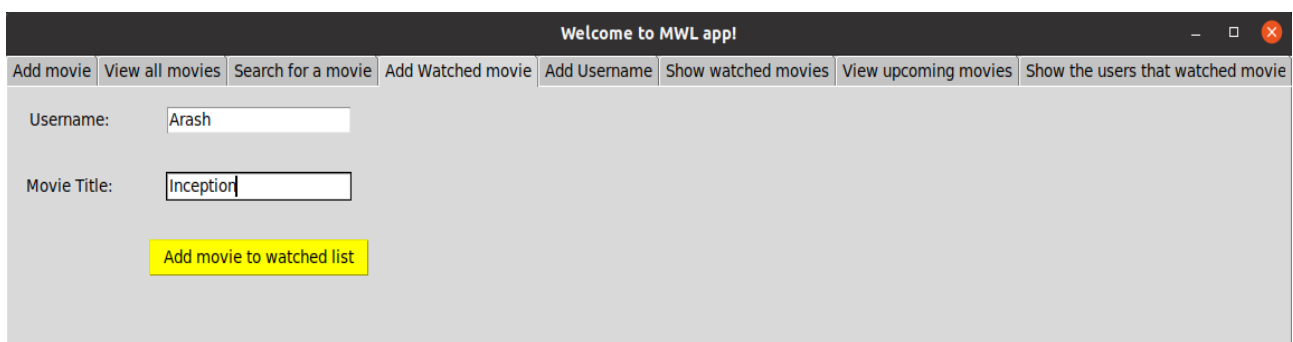
Tab2: To see a list of all the movies in this database.



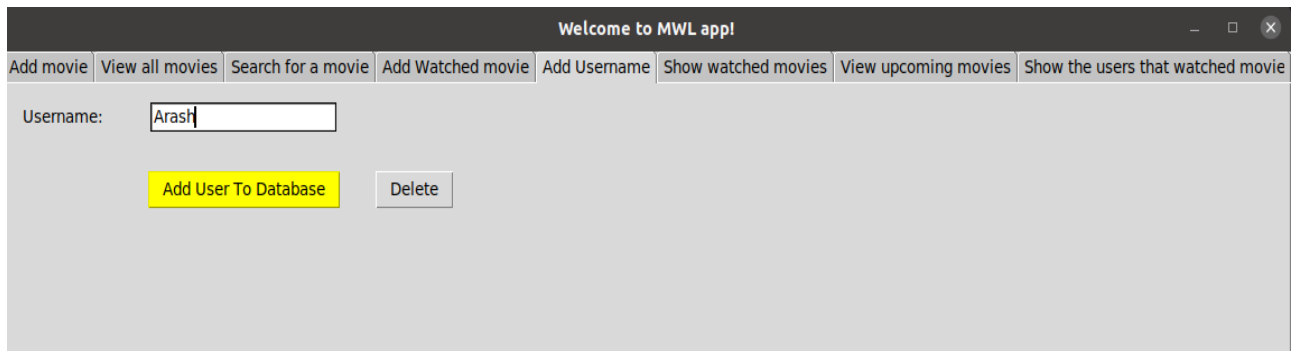
Tab3: To search by movie name.



Tab4: To specify movies that watched by the user.



Tab5: To add a new user or delete user from database.



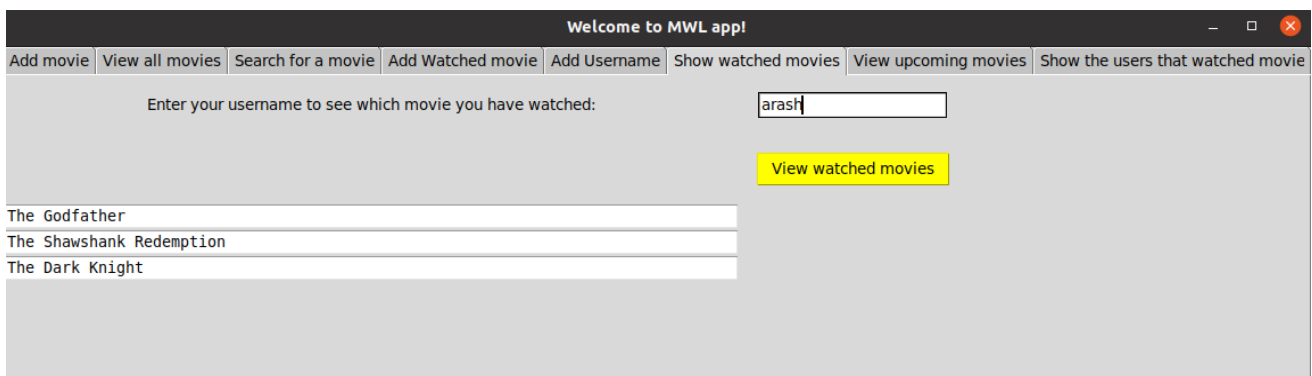
Welcome to MWL app!

Add movie View all movies Search for a movie Add Watched movie Add Username Show watched movies View upcoming movies Show the users that watched movie

Username:

Add User To Database Delete

Tab6: To see a list of movies that a user has watched.



Welcome to MWL app!

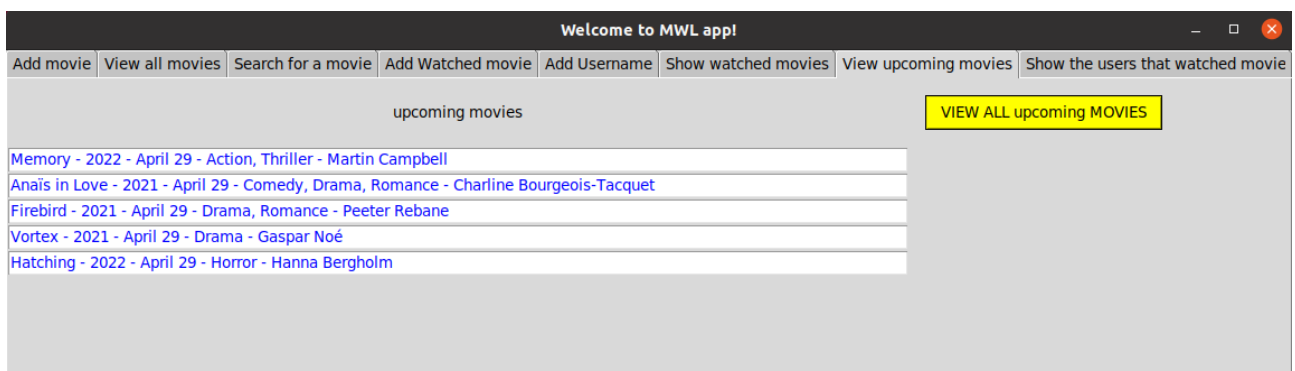
Add movie View all movies Search for a movie Add Watched movie Add Username Show watched movies View upcoming movies Show the users that watched movie

Enter your username to see which movie you have watched:

View watched movies

The Godfather
The Shawshank Redemption
The Dark Knight

Tab7: To see upcoming movies using the Imdb's API.
(Link: <https://imdb-api.com/api/#ComingSoon-header>).



Welcome to MWL app!

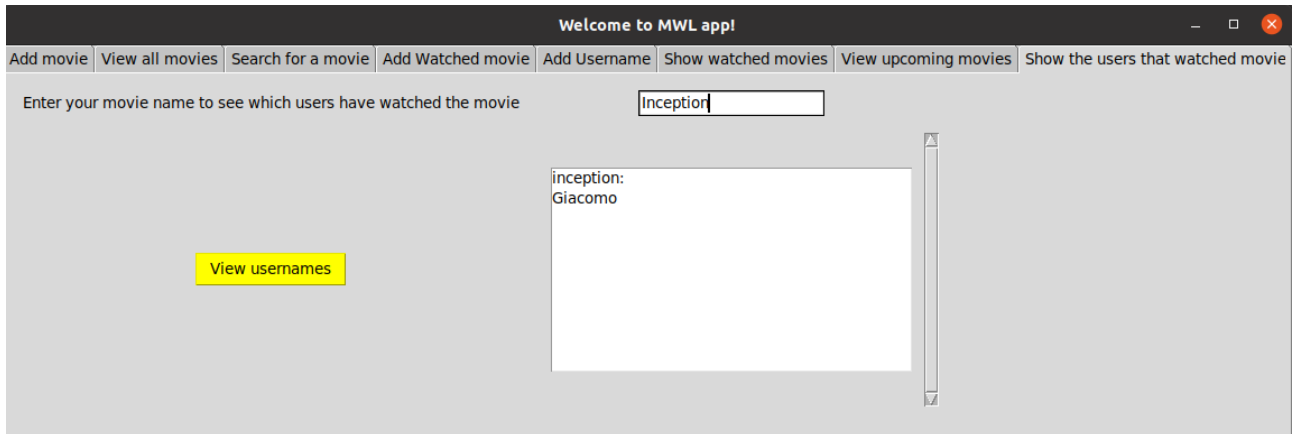
Add movie View all movies Search for a movie Add Watched movie Add Username Show watched movies View upcoming movies Show the users that watched movie

upcoming movies

VIEW ALL upcoming MOVIES

Memory - 2022 - April 29 - Action, Thriller - Martin Campbell
Anaïs in Love - 2021 - April 29 - Comedy, Drama, Romance - Charline Bourgeois-Tacquet
Firebird - 2021 - April 29 - Drama, Romance - Peeter Rebane
Vortex - 2021 - April 29 - Drama - Gaspar Noé
Hatching - 2022 - April 29 - Horror - Hanna Bergholm

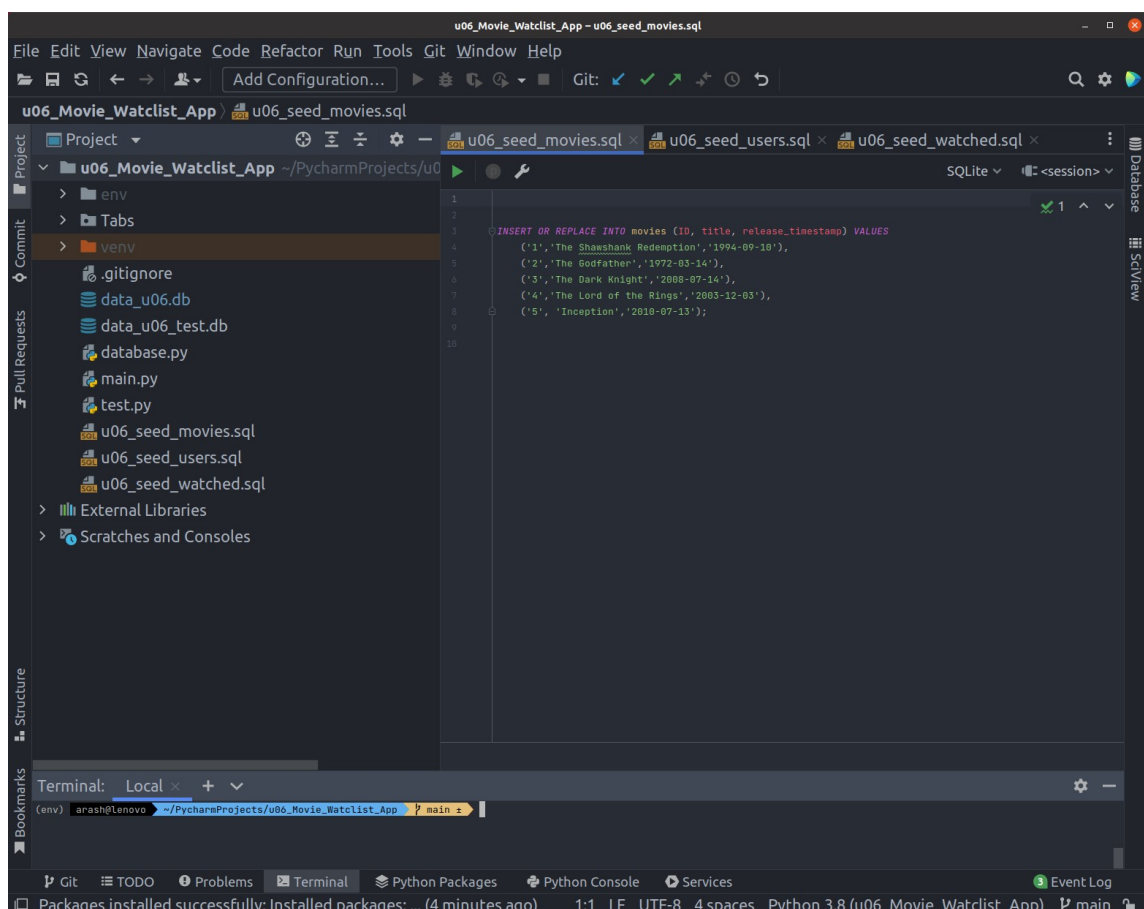
Tab8: To view movies that watched by a specific user.



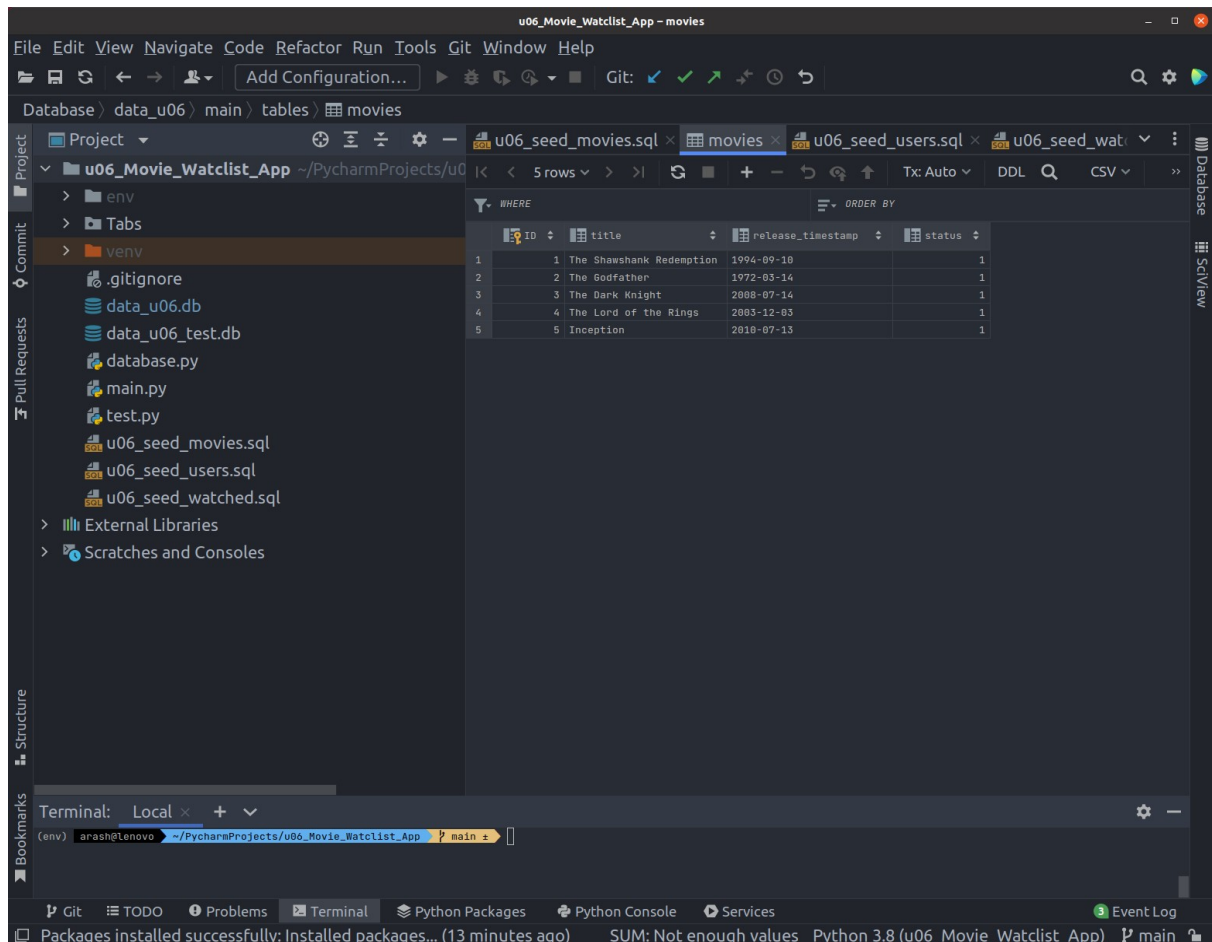
In order not to need to insert data into database for testing this application, I have put three SQL files in the project, which will be inserted into the database by executing the main.py file. You can also use the syntax below to insert data to database.

Cat u06_seed_movies.sql / sqlite3 data_u06.db

Data in the SQL file



The data when inserted inside the table



Tkinter:

To create the user menu I used a python package that called Tkinter, to be able to create a graphical user interface.

(Link: <https://docs.python.org/3/library/tk.html>)

You have to install tkinter package before execute the main.py.

>> *sudo apt-get install python3-tk*

Requirements.txt

To make it easier to use this app, there is a file called requirements.txt that contains all the packages required to run the project. You can simply install them with this script:

\$ *pip install -r requirements.txt*

all of the program's "dependencies" will be downloaded, installed, and ready to go in one fell swoop.

Test file

I wrote the tests in two methods, pytest and unittest, which have 80% coverage, and all the tests are in the test.py file. To run the tests, just run this script in the terminal :

```
>>pytest test.py -cov
```

or

```
>>pytest --cov --cov-report html (To get html file as a output)
```

Backlog for sprint2

you can do what you want with the code. But here is a suggestion for the backlog.

The project I will build in u06 is an app for film surveillance. Finally, I have a project that allows users to:

Keep track of movies and series they are interested in and their release dates.

Store the movies that a user has seen among all the movies in the database.

Add a new user.

GUI (tkinter ← tips)

Below are the features that this app should have but two of them is not complete and you have to add them to the project.

1) Add new movie.

2) Watch upcoming movies. Downloading Movies from API (IMDB)

3) Watch all movies in a list (titles)

4) Add watched movie

5) Add users to the app.

6) Gui or Cli for front end

7) Dynamic to-watch list (based on minimum grade, genre, director / actor, year) at least 3

8) show details for movies (plot, comments / reviews)

9) Divide into two sprints (sprint 1 and sprint 2)

Green: done Orange:incomplete

Below are the other steps that need to be done to complete the project:

