Spooling up the Django app from git up repo on mac.

1. Install PostgreSQL to have a database for Django to run on - [Reference](https://medium.com/keeping-code/running-multiple-postgresql-versions-simultaneously-on-macos-linux-90b3d7e08ffd#:~:text=You%20have%20to%20install%20the,the%20things%20installed%2C%20let's%20advance!&text=You%20have%20to%20specify%20the,the%20server%20will%20be%20running.) or [This](https://ogirginc.github.io/en/multi-version-postgresql-macos)
   1. brew tap petere/postgresql
   2. brew install petere/postgresql/postgresql-common
   3. brew install petere/postgresql/postgresql@17
   4. pg\_createcluster 17 logbook
   5. pg\_lsclusters – see clusters and get the port number for your cluster
      1. add to the .env file
   6. pg\_ctlcluster 17 logbook start
      1. will probably throw an error. Just navigate to the config file in the error and go all the way to the bottom and comment out the config file. Then try starting it again.
   7. createdb logbook
   8. psql logbook
   9. ALTER USER username WITH PASSWORD ‘passwork’;
      1. Your username will be your mac username and you can change the password to whatever you want.
      2. This will need to be added to the .env file
2. Clone repo – git clone git@github.com:cloudflying87/logbook.git
3. brew install pyenv-virtualenv - Reference: [Doc](https://faun.pub/how-to-install-multiple-python-on-your-mac-d20713740a2d)
   * 1. this allows you to run multiple versions of python if desired
   1. pyenv install 3.13.1
   2. pyenv virtualenv 3.13.1 logbook
   3. pyenv activate logbook
4. CD into logbook folder
   1. Run this command in the logbook folder from github
   2. mkdir aircraft/migrations && touch aircraft/migrations/\_\_init\_\_.py && mkdir airline/migrations && touch airline/migrations/\_\_init\_\_.py && mkdir airport/migrations && touch airport/migrations/\_\_init\_\_.py && mkdir logbook/migrations && touch logbook/migrations/\_\_init\_\_.py && mkdir reports/migrations && touch reports/migrations/\_\_init\_\_.py && mkdir user/migrations && touch user/migrations/\_\_init\_\_.py
   3. python manage.py makemigrations && python manage.py migrate
   4. Then you will need to create a Django super user
      1. Python manage.py createsuperuser
      2. Then login with that username and password

Installing via Docker

1. Cd into whatever directory you would like the code to live in on your docker machine
   * 1. For the first time you will need to clone it afterwards you can just pull it
   1. git clone [git@github.com:cloudflying87/logbook.git](mailto:git@github.com:cloudflying87/logbook.git)
   2. sudo docker compose up -d
   3. sudo docker compose -f docker-compose.yaml exec app python manage.py collectstatic
   4. sudo docker compose -f docker-compose.yaml exec app python manage.py makemigrations
   5. sudo docker compose -f docker-compose.yaml exec app python manage.py migrate
      1. Note: If you are not importing an exsisting database then you will need to add a superuser to be able to login to the logbook to get started
   6. sudo docker compose -f docker-compose.yaml exec app python manage.py createsuperuser

.env file example

# set this to something secure for django

SECRET\_KEY = ‘unsecure’

#used to hide or show the django errors

DEBUG = False

#used if you want to be able to hit google calendar to import your delta schedule will need to setup a google developer account and setup a project for the project id and client secret

googlecredintials = {"installed":{"client\_id":"","project\_id":"","auth\_uri":"https://accounts.google.com/o/oauth2/auth","token\_uri":"https://oauth2.googleapis.com/token","auth\_provider\_x509\_cert\_url":"https://www.googleapis.com/oauth2/v1/certs","client\_secret":"","redirect\_uris":["http://localhost"]}}

# if you want to be able to import flights from flightaware

flightawareapi =

# future functionality. have not been able to get this to work even though it is in settings

emailhost =

emailport = 465

emailuser =

emailpassword =

# this should be all of local postgres info. For the docker image with docker compose the port will be 5432 and the host is DB

DB\_NAME=logbook

DB\_USER= username

DB\_PASSWORD=password

DB\_HOST=localhost or db in docker

DB\_PORT=5432

remove - pyenv virtualenv-delete 3.13.1/envs/logbook