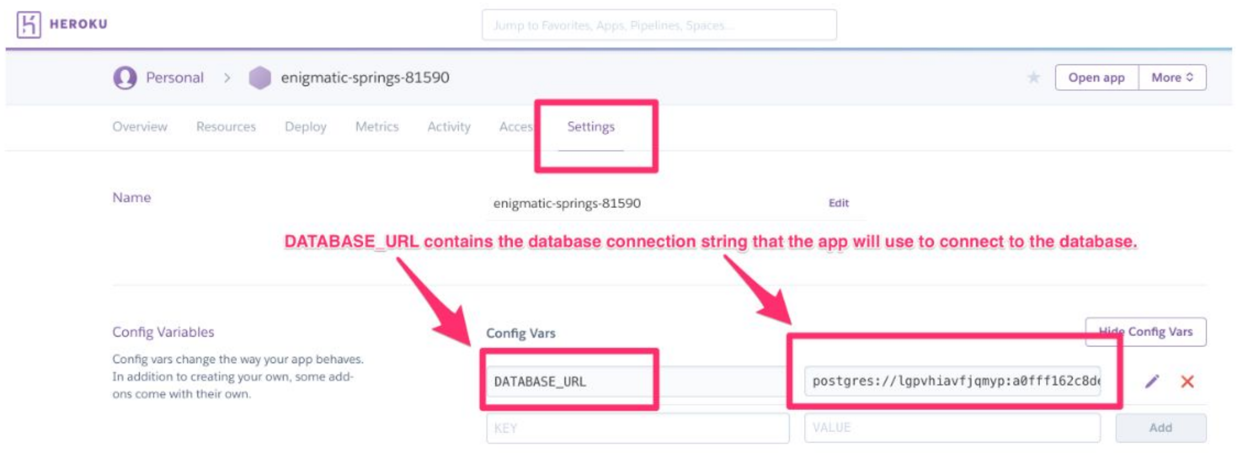
- **Create a virtual environment for your project**
  - Install all dependencies in the virtual environment in addition to gunicorn and psycopg2
- **Create initdb.py file and make sure it works by typing in:**
  - python initdb.py
- **Run the flask app by running:**
  - FLASK_APP=BeerApp/app.py flask run
  - You may need to copy all folders from BeerApp to the folder with initdb.py (other than the database)
  - Alternatively you may also create a “run.sh” file and type the following into command
    - ./run.sh
- **Create requirements.txt by running the following:**
  - pip freeze > requirements.txt
- **Create Procfile by running:**
  - touch Procfile
  - Add the following to the Procfile” web: gunicorn BeerApp.app:app
- **Deploy the app using the instructions on Heroku’s main page**
- **Go to the resources page and install postgres as an Add-On**
  - Click on the add on, then navigate to settings and click on Reveal Config Variables.
  - The connection string to the database should now be available



Name enigmatic-springs-81590

DATABASE_URL contains the database connection string that the app will use to connect to the database.

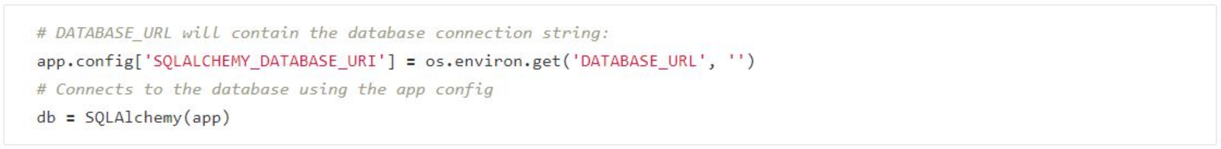
Config Variables

Config vars change the way your app behaves. In addition to creating your own, some add-ons come with their own.

Config Vars

DATABASE_URL postgres://lgpvhiavfjqmyp:a0fff162c8d

- Heroku will automatically assign this URI string to the DATABASE_URL environment variable that is used within app.py. The code that is already in app.py will be able to use that environment variable to connect to the Heroku database.

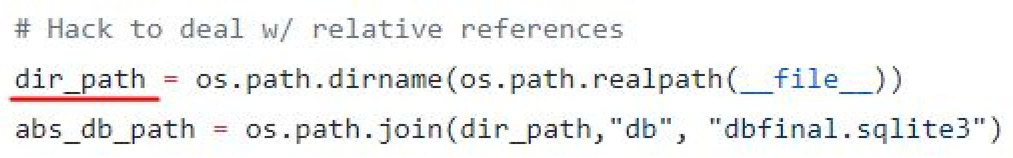


```
# DATABASE_URL will contain the database connection string:
app.config['SQLALCHEMY_DATABASE_URI'] = os.environ.get('DATABASE_URL', '')
# Connects to the database using the app config
db = SQLAlchemy(app)
```

- Initialize the database by entering in:
    - heroku run python initdb.py

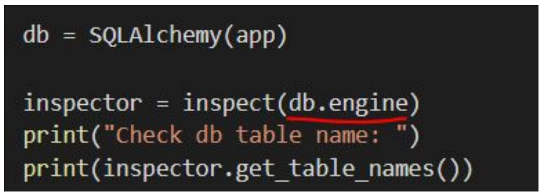
ITEMS TO NOTE:

- If you receive a SQLite3 error “unable to open database file” it could actually stem from the fact that the flas app cannot find the database file itself.
    - Take a look in your app.py file, create a relative reference and then use os.path.join using that relative reference as below



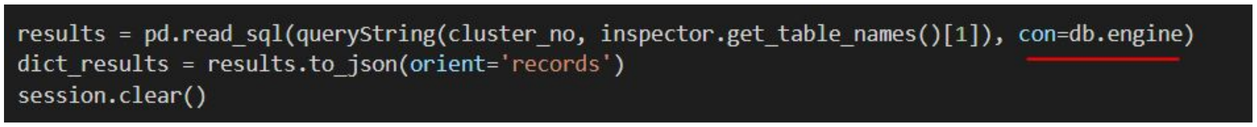
```
# Hack to deal w/ relative references
dir_path = os.path.dirname(os.path.realpath(__file__))
abs_db_path = os.path.join(dir_path,"db", "dbfinal.sqlite3")
```

-
    - When using that relative reference you do NOT need to create an engine - simply type in “db.engine” anywhere you need the engine



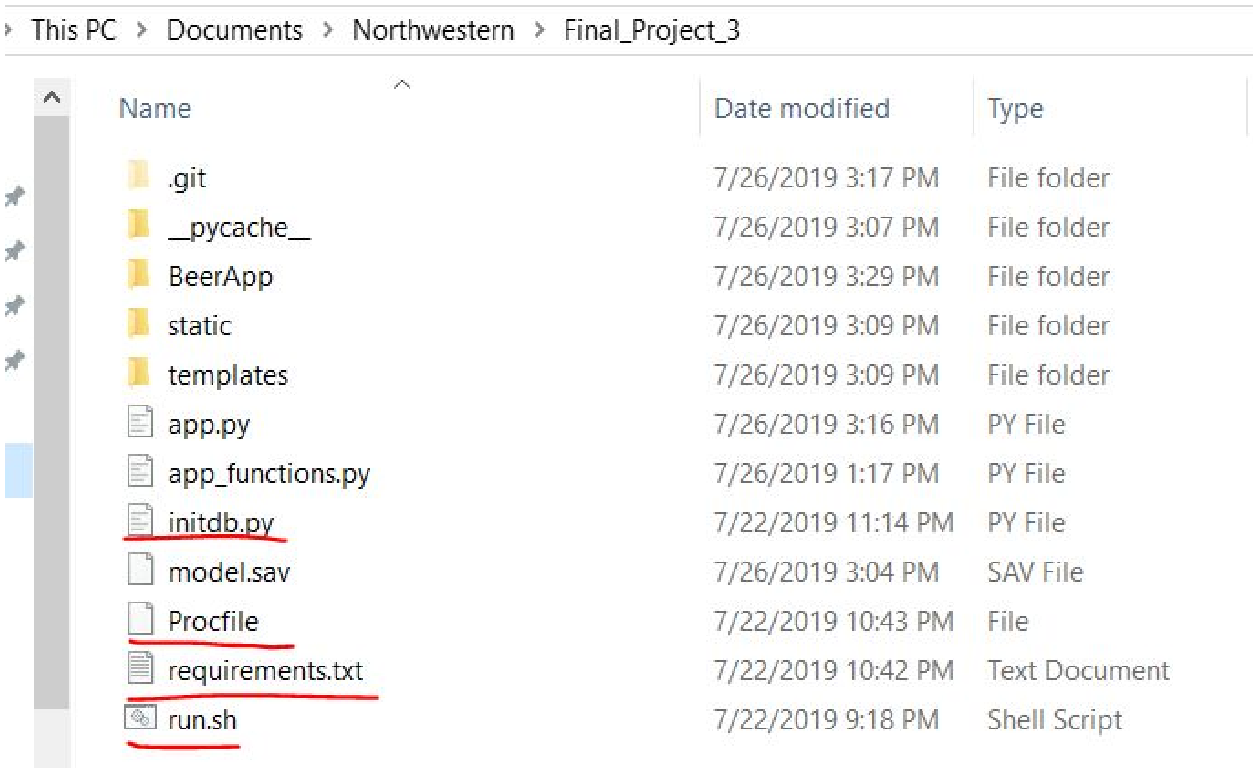
```
db = SQLAlchemy(app)

inspector = inspect(db.engine)
print("Check db table name: ")
print(inspector.get_table_names())
```



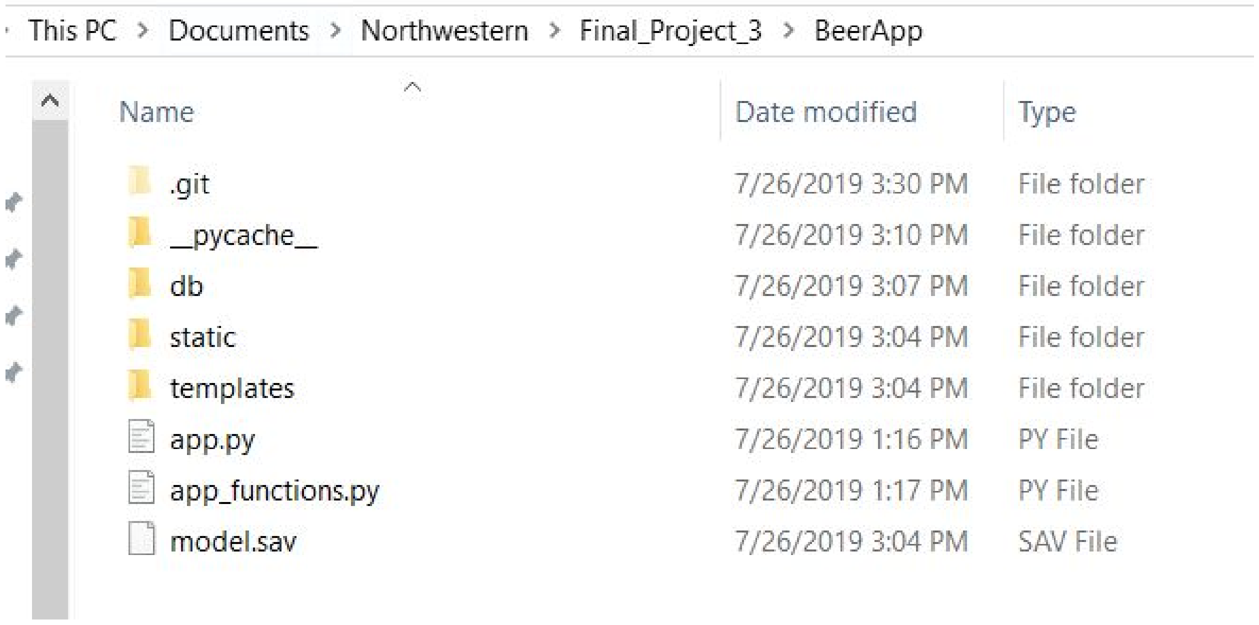
```
results = pd.read_sql(queryString(cluster_no, inspector.get_table_names()[1]), con=db.engine)
dict_results = results.to_json(orient='records')
session.clear()
```

- An “init.py” document is included in the Heroku folder, this may or may not be needed for your app - when I used the relative reference Heroku did not need it
- Always check every page of the heroku app, sometimes if you have file ends in caps such as .PNG heroku will not load those images and you need to change things to .png
- The basic file structure is as follows:



This PC › Documents › Northwestern › Final_Project_3

Name	Date modified	Type
.git	7/26/2019 3:17 PM	File folder
__pycache__	7/26/2019 3:07 PM	File folder
BeerApp	7/26/2019 3:29 PM	File folder
static	7/26/2019 3:09 PM	File folder
templates	7/26/2019 3:09 PM	File folder
app.py	7/26/2019 3:16 PM	PY File
app_functions.py	7/26/2019 1:17 PM	PY File
initdb.py	7/22/2019 11:14 PM	PY File
model.sav	7/26/2019 3:04 PM	SAV File
Procfile	7/22/2019 10:43 PM	File
requirements.txt	7/22/2019 10:42 PM	Text Document
run.sh	7/22/2019 9:18 PM	Shell Script



This PC › Documents › Northwestern › Final_Project_3 › BeerApp

Name	Date modified	Type
.git	7/26/2019 3:30 PM	File folder
__pycache__	7/26/2019 3:10 PM	File folder
db	7/26/2019 3:07 PM	File folder
static	7/26/2019 3:04 PM	File folder
templates	7/26/2019 3:04 PM	File folder
app.py	7/26/2019 1:16 PM	PY File
app_functions.py	7/26/2019 1:17 PM	PY File
model.sav	7/26/2019 3:04 PM	SAV File