**Setup commands to run**

* Pip install pipenv
  + Create a virual environment so when we install packages, it goes to the virtual environment instead of the global system
* Pip env shell
  + We run the next command inside this virtual environment so that we don’t install it globally
  + Pipenv install flask
    - Flask is our web framework, basically like our package.json for node js
  + Pipenv install psycopg2
    - This is our database adapter used to work with postgres
  + Pipenv install psycopg2-binary
    - Extra one installed just in case?
    - In the video he says the normal one gives him problems
  + Pipenv install flask-sqlalchemy
  + Pipenv install gunicorn
  + Shift-control-P , we type in python and select interpretor
    - The feedback app! There should be an option with feedbackapp.pipenv
* Create two folders, one for template and one for static
  + Work on the two python files, can just refer
  + A class needs to be created to match the ‘Feedback’ table we created in our database!
* Create Heroku and mailtrap account, and also download Heroku cli
* Do git init
* Heroku login
* Heroku create feedbackpython
  + The name we choose here must be (globally) unique, and if we don’t choose a name then a weird one will be chosen for us
* Heroku addons:create heroku-postgresql:hobby-dev --app feedbackpython
  + The –app is a double line!
* Heroku config –-app feedbackpython
  + This will give us our database URL, and we put it into our production database url
* Echo >> Procfile
  + Create our procFile
* Create a runtime.txt
  + Just create manually and add the python version we are running, he uses 3.7.2 but im using my own 3.7.7
* Telling it our python version
* Pip freeze > requirements.txt
  + Create a requirements.txt
* Heroku run python

https://youtu.be/w25ea\_I89iM