Python

.gitignore* *in root folder add—>* ***venv/*

flask run -p 5001
python -m flask run -p 5001

python3 -m venv venv

source venv/bin/activate

pip3 install "Flask<3" "Werkzeug<3"

pip3 install flask-wtf

pip3 freeze > requirements.txt
pip install -r requirements.txt

install git+https://github.com/pallets-eco/flask-debugtoolbar
-pip install flask-debugtoolbar (alternate**)**

pip3 install packaging

pip3 install psycopg2-binary

pip install python-dotenv
>
touch .env
(add **.env** to the .gitignore)

touch .gitignore

(How to install ipython in a virtual environment: **pip3 install ipython)**
hash -r (makes computer forget last place it looked to reset the ipython location for venv issue)
which ipython3

get out of venv: **deactivate

- from **flask import** Flask, render_template, request, redirect, jsonify, flash
- from flask_debugtoolbar import DebugToolbarExtension

 - from models import db, connect_db, Pet

- from models import db, connect_db, Pet
    - app = Flask(__name__)
    - app.config['SECRET_KEY'] = "secret"
    - debug = DebugToolbarExtension(app)

app.config['SQLALCHEMY_DATABASE_URI'] = os.environ.get(
"DATABASE_URL", 'postgresql:///sqla_intro')
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = **False**
app.config['SQLALCHEMY_ECHO'] = **True**

connect_db(app)

put all static files in **static folder
put all html files in **templates folder

- {% extends 'base.html' %}
- **for loop example:** {% for prompt in website_prompts %}   {% endfor %}
-
- {% block content %}   {% endblock %}
- **{{}}** — insert our variables from python inside
- <int:INDEX> — use this to **add a variable into route parameter** (include this variable in def argument to use inside def)

** **stop redirect intercepting for the debug toolbar**—>
app.config['DEBUG_TB_INTERCEPT_REDIRECTS'] = **False**

app.config['TESTING'] = True —> put this in the testing page
app.config['DEBUG_TB_HOSTS'] = ['dont-show-debug-toolbar'] —> this is test file as well

loop.index —> jinja (this starts at index 1)
flash('message here')—>get_flashed_messages()

TESTS:
run a doctest **(example):** python -m doctest -v currency_calc.py
run an integration test **(example)**: python3 -m unittest -v testing_app

IN MODELS:
- from flask_sqlalchemy import SQLAlchemy
-
- db = SQLALchemy()

-def connect_db(app):
“””connect to database.””

```
app.app_context().push()
db.app = app
db.init_app(app)
```

PSQL

```
$ psql
=# CREATE DATABASE vehicles;
=# (control-d)
$ psql -f vehicles.sql vehicles
```

```
\c your_database_name

dropdb database_name
psql —> CREATE DATABASE database_name;

In [1]: %run app.py

In [2]: db.create_all()
```

POPULATE EDIT INFORMATION

```
 form = EditNoteForm()
 note = Note.query.get_or_404(note_id)

 form = EditNoteForm(obj=note)
```

API: AIzaSyBdqp0e0Ziq5ssTvgw2JwW40k0sAPn7DYE