

FINAL PROJECT

The Dog House Web Application

INTRODUCTION

I built a web application called **The Dog House** that collects data from users about their pet. The data is used to profile the pet as well collect emergency contact information. In the next iteration the site will allow the user to schedule their pet for dog sitting. It is a site I plan to use for my small business. It is a POC website that taught me a lot about the latest web technologies while using Python. The site was built using a scalable architecture while implementing the MVC Flask framework. I took an online Udemy Flask course that walked me through creating a scalable site. I have deployed the site to this URL: <https://the-dog-house.herokuapp.com/>

REQUIREMENTS

PostgreSQL 10

The following requirements list was produced by using the: *pip freeze > requirements.txt* command

awsebcli==3.12.4	itsdangerous==0.24
bcrypt==3.1.4	Jinja2==2.10
bitstring==3.1.5	jmespath==0.9.3
blessed==1.14.2	MarkupSafe==1.0
botocore==1.10.3	pathspec==0.5.5
cement==2.8.2	psycpg2==2.7.4
certifi==2017.7.27.1	pycparser==2.18
click==6.7	python-dateutil==2.6.1
colorama==0.3.7	pytz==2018.3
dockerpty==0.4.1	PyYAML==3.12
docopt==0.6.2	requests==2.9.1

docutils==0.14	semantic-version==2.5.0
dominate==2.3.1	six==1.11.0
Flask==0.12	SQLAlchemy==1.2.6
Flask-Bcrypt==0.7.1	tabulate==0.7.5
Flask-Bootstrap==3.3.7.1	termcolor==1.1.0
Flask-Login==0.4.1	visitor==0.1.3
Flask-SQLAlchemy==2.3.2	wcwidth==0.1.7
Flask-WTF==0.14.2	websocket-client==0.47.0
gunicorn==19.7.1	Werkzeug==0.14.1
	WTForms==2.1

PROGRAM DESCRIPTIONS

run.py

This program is used to launch the application. It is the entry point for creating the flask application. An application factory pattern is used to create the application instance, the factory takes a configuration name as an argument. The database instance is created in this script too.

dev.py

This script is a configuration script used by the application factory in the run.py script. It defines globals that get used in the application. dev.py is setup to run a local version of the Postgres database.

routes.py

This script can be thought of as the ‘V and C’ in the MVC pattern. It is the Controller for the application it helps the views by passing application data to them. All requests are serviced from this script and routed to the appropriate view The application logic is found here. The important thing to remember about this script is that it services both **get** and **put** requests. The routes.py script implements the assignment requirements, **try/except, importing external modules, Classes, Decorators and Functions.**

models.py

This script is considered the “M” in the MVC pattern. It is the Model for the application. It is where the data is stored and retrieved. It uses the ORM technology discussed in our class. The ORM allowed me to abstract the data requests from the underlying data base.

forms.py

This script is part of the View of MVC. This is where the web forms code exist. The controller (routes.py) will instantiate the form build the HTML using the template HTML files and jinja and render it to the browser. Once the user submits the form, it returns to the routes.py as a post request for processing.

auth directory

Contains the same script names as above but these scripts perform the user authentication functions of the application.

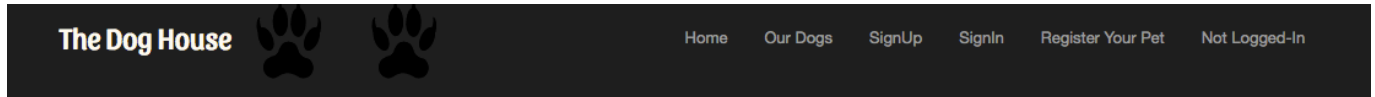
__init__.py

This script implements the Blueprint for the scalable architecture. [Flask uses a concept of *blueprints* for making application components and supporting common patterns within an application or across applications. Blueprints can greatly simplify how large applications work and provide a central means for Flask extensions to register operations on applications. A [Blueprint](#) object works similarly to a [Flask](#) application object, but it is not actually an application. Rather it is a *blueprint* of how to construct or extend an application.]

Conclusion

The problem: A new client needs to fill out several paper forms that provide critical information about their pet. I take the data from the form and add it to a spreadsheet. Now with the web app I can have the user supply the data electronically which eliminates my involvement. This save time and paper and reduces human error. The next release will have a scheduling form that will allow the users to supply Drop Off and Pick up dates and times. These dates and times will automatically be added to my gCal once accepted by using the gCal API.

Screen Shots



Register Here

Name

E-mail

Password

Confirm

Register



Welcome to the Dog House

At the Dog House you can be assured your dog will get the best personalized attention.





Sparky



Owner: Scott Barth

Contact Number: 888-888-8888

Breed: Catahoula

Vet Info: Dr. Bains

Sex: male

Neutered/Spade: yes

Feeding Instructions: Twice daily

Exercise Instructions: Twice daily

Age: 1

Notes: Likes to fetch the ball

Rating: 6.0

[create](#) [edit](#) [delete](#)



Pet Profile Form

Register your pet

Owner Name

Phone

Pet Name

Breed

☐ Male☐ Female☒ Neutered/Spade (Yes)☐ Neutered/Spade (No)

Vet Contact Information

Feeding Instructions

Exercise Instructions

Other information I should know

Pets Date Of Birth