<https://www.youtube.com/watch?v=Z1RJmh_OqeA>

# install command line tools (if not installed)

$ xcode-select --install

# install pip via easy\_install

$ sudo easy\_install pip

# show current pip version (optional)

$ pip –version

//install virtualenv

su@SUs-MacBook-Pro Flask % pip3 install virtualenv

//created virtual environment

//env is the name,

su@SUs-MacBook-Pro Flask % virtualenv env

//active environment

su@SUs-MacBook-Pro Flask % source env/bin/activate

(env) su@SUs-MacBook-Pro Flask %

//install the requirements

(env) su@SUs-MacBook-Pro Flask % pip3 install flask flask-sqlalchemy

//create a file call app.py

# import flask

from flask import Flask

# set up our application

# reference this file

app = Flask(\_\_name\_\_)

# create an index route

@app.route('/')

#define function for that route

def index():

return "Hello World!"

if \_\_name\_\_ == "\_\_main\_\_":

# if there are any errors they will pop up on the webpage

app.run(debug=True)

// execute this with python 3

(env) su@SUs-MacBook-Pro Flask % python3 python3 app.py

/Users/su/Desktop/Flask/env/bin/python3: can't open file 'python3': [Errno 2] No such file or directory

(env) su@SUs-MacBook-Pro Flask % python3 app.py

\* Serving Flask app "app" (lazy loading)

\* Environment: production

WARNING: This is a development server. Do not use it in a production deployment.

Use a production WSGI server instead.

\* Debug mode: on

\* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

\* Restarting with stat

\* Debugger is active!

\* Debugger PIN: 109-272-396

// open the page on browser

<http://localhost:5000/>

A screenshot of a cell phone

Description automatically generated

App.py

# import flask

from flask import Flask, render\_template

# set up our application

# reference this file

app = Flask(\_\_name\_\_)

# create an index route

@app.route('/')

#define function for that route

def index():

return render\_template('index.html')

if \_\_name\_\_ == "\_\_main\_\_":

# if there are any errors they will pop up on the webpage

app.run(debug=True)

index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Document</title>

</head>

<body>

Hello world 2!

</body>

</html>

// template:

Base.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

{% block head %}{% endblock %}

</head>

<body>

{% block body %}{% endblock %}

</body>

</html>

Index.html

{% extends 'base.html' %}

{% block head %}

<h1>Template</h1>

{% endblock %}

{% block body %}

{% endblock %}

// link to the style sheet

A screenshot of a cell phone

Description automatically generated

Main.css

body {

margin: 0;

font-family: sans-serif;

}

Base.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<!-- link to the style sheet -->

<link

rel="stylesheet"

href="{{ url\_for('static', filename='css/main.css')}}"

/>

{% block head %}{% endblock %}

</head>

<body>

{% block body %}{% endblock %}

</body>

</html>

Database(15:43)

// set the database in terminal

(env) su@SUs-MacBook-Pro Flask % source env/bin/activate

(env) su@SUs-MacBook-Pro Flask % python3

>>> from app import db

>>> db.create\_all() //test.db will appear

>>> exit()

Base.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<!-- link to the style sheet -->

<link

rel="stylesheet"

href="{{ url\_for('static', filename='css/main.css')}}"

/>

{% block head %}{% endblock %}

</head>

<body>

{% block body %}{% endblock %}

</body>

</html>

Index.html

{% extends 'base.html' %} {% block head %}

<title>Task Master</title>

{% endblock %} {% block body %}

<div class="content">

<h1>Task Master</h1>

<table>

<tr>

<th>Task</th>

<th>Added</th>

<th>Actions</th>

</tr>

{% for task in tasks %}

<tr>

<td>{{task.content}}</td>

<td>{{task.date\_created.date()}}</td>

<td>

<a href="">Delete</a>

<br />

<a href="">Update</a>

</td>

</tr>

{% endfor %}

</table>

<form action="/" method="POST">

<input type="text" name="content" id="content" />

<input type="submit" value="Add Task" />

</form>

</div>

{% endblock %}

App.py

# import flask

from flask import Flask, render\_template, url\_for, request,redirect

# import databse

from flask\_sqlalchemy import SQLAlchemy

from datetime import datetime

# set up our application

# reference this file

app = Flask(\_\_name\_\_)

# tell our app where our database is located

# sqlite:/// relative path, reside in the project location

# sqlite://// absolute path

app.config['SQLALCHEMY\_DATABASE\_URI'] = 'sqlite:///test.db'

# initial the database

db = SQLAlchemy(app)

#create a model

class Todo(db.Model):

id = db.Column(db.Integer,primary\_key=True)

content = db.Column(db.String(200),nullable=False)

date\_created = db.Column(db.DateTime, default = datetime.utcnow)

#get a function when we create the string everytime, we create a new element

def \_\_repr\_\_(self):

#return task, and the id of the task that just been created

return '<Task %r>' % self.id

# create an index route

# @app.route('/')

@app.route('/',methods=['POST','GET'])

# define function for that route

def index():

if request.method =='POST':

#create a variable call task\_content

# content is the name in index.html-> form

task\_content = request.form['content']

#create a Todo object

new\_task = Todo(content=task\_content)

try:

# add this to the database

db.session.add(new\_task)

db.session.commit()

return redirect('/')

except:

return 'There was an issue adding your task'

else:

# create a variable called tasks,

# query in our database, ordering them by the day create

tasks = Todo.query.order\_by(Todo.date\_created).all()

#pass the tasks to the template

return render\_template('index.html',tasks=tasks)

if \_\_name\_\_ == "\_\_main\_\_":

# if there are any errors they will pop up on the webpage

app.run(debug=True)

Delete(31:27)