COLOR CODING KEY

* + Code will be in blue text
  + Info will be in orange text
  + Specific to you/can be changed will be in green text
  + Import information will be in red text

At the bottom of the page I have detailed some issues I’ve had and how I fixed them

**INSTALLS/////////////////////////////////////////////////////////**

ONLY INSTALL PACKAGES WHILE IN A VIRTUAL ENVIRONMENT. BE SURE TO UPDATE THE requirements.txt file with the ‘>’ command below.

Packages are necessary to do what we want. However, our project only needs certain packages to work and having all the packages installed in pip can cause issues. I was unable to deploy to Heroku until I deleted the packages mentioned in the error from the requirements.txt folder. This is why we work in virtual environments, so we only use the packages we need for the project we are working on, while not altering the packages stored on our computers.

**Getting Started with Virtual Environments:**

* **Install the virtual environment:** pip3 install virtualenv
  + (one time thing globally)
* **Create a virtualenv:** python3 -m venv venv
  + (Do this inside of project folder, one time generally)

**Venv Activation Commands:**

* **Activate Venv while developing:** source venv/bin/activate
  + (Do before you install any packages)
* **Deactivate Venv session with:** deactivate
  + (Do this when you are done working inside project)
* **Installing Package In the Venv:** pip install PackageYouWantToInstall

**Venv Maintenace Commands:**

* **Update Venv Packages:** venv/bin/pip install -r requirements.txt
  + (This takes all packages specified in requirements.txt and installs them in your venv if it’s activated. You will most likely be doing this if you pulled someone else changes from github. Therefore it’s important to update the requirements.txt file.)
* **Update Requirements.txt file:** pip3 freeze > requirements.txt
  + (Do this after you install a new package in your Venv)

**Python//////////////////////////////////////////////////////////**

**Getting Started with Python:**

* **Download Homebrew:** https://brew.sh/
* **Install Python3:** ‘brew install python3’
* **Update the pip command:** pip3 install –upgrade –force pip
  + This is so the pip command works on the new (2020) python 3 instead of the old 2.7. Check with pip –V
* **Download pip to install:** curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
* **Install pip:** python3 get-pip.py

**Running Python Files:**

* **Run the web app locally**: python3 fiddl.py
* **Lookup python version:** python --version

**Flask//////////////////////////////////////////////////////////**

**Getting Started with Flask:** Flask is a (micro (pretty light)) web framework (way to structure a website). No database or form validation (API/HTTP requests) so you need to Import them as you need them. Essentially more maneuverability (Pinterest and LinkedIn are built on Flask).

* **Install Flask:** pip install flask
* **Check flask version:** flask –version

**GITHUB//////////////////////////////////////////////////////////**

**Getting Started with GitHub (One time thing):**

* **Clone the repo to Local:** git clone <https://github.com/GU-CPSC-2020-21-Seniors/cpsc-491-2-senior-project-facial-recognition-security-lock.git>
* **Update repo:** git pull

**Github Branches:**

* **See what branch you are in:** git branch
* **Point to your branch:** git checkout dev-cmooring
* **View all branches:** git branch –r
* **Make a new branch:** git branch dev-cmooring
* **Delete a branch:** git branch dev-cmooring –d

**Pushing to YOUR BRANCH:**

* **Point to your branch:** git checkout dev-cmooring
* **Check if you have files to add:** git status
  + Check what files you have changed locally that need to be added to github
* **Add new changes:** git add –A
* **Commit changes:** git commit –m “Commit message – be detailed”
* **Push to your branch:** git push

**Update your code with the MASTER BRANCH code:**

You want to update your local code with the master branch code that has stuff others have worked on.

* **Switch to master branch:** git checkout master
  + This will load the most recent master that you pulled from github
* **update local branch:** git pull
  + This will update your local master branch so that it is now up to date with the github master branch
* **Switch to your branch:** git checkout dev-cmooring
  + Now that your local master branch is updated, you can merge it with your personal local branch
* **Merge your branch with master:** git merge master
  + This will add (merge) your branches code with the master branch you just updated. Meaning now your local branch now includes any code from the github master branch you did not have before.
  + This will probably throw a terminal error that is saying you need to take care of merge conflicts. Go to the files mentioned in the error and settle the merge conflicts by selecting the options it has like “keep all”.
  + IT IS IMPORTANT TO NOT GET RID OF ANY IMPORTANT CODE HERE!
* **Add new changes:** git add –A
  + Now your local branch is updated, push to your online github branch
* **Commit changes: git commit –m “Stuff”**
* **Push to your branch (make sure you are in your branch):** git push
  + **Or specify branch:** git push origin dev-cmooring

Now your github branch code has your new stuff and any new stuff from the master

**Heroku//////////////////////////////////////////////////////////**

**Getting Started with Heroku (One time thing)**

First, you want to go to heroku.com and make an account! Be sure to put your account email in the excel file “DeveloperKey” in our group document area. This is so we can add you as contributors.

* **Install gunicorn as server (Be in Venv):** pip install gunicorn
* **Install Heroku:** brew tap heroku/brew && brew install heroku
* **Update heroku: heroku update**
* **Install build packs:** heroku buildpacks:add heroku/python
* **Look at Heroku Build Errors: heroku logs --tail**

To Do this, you must be in a virtual environment first. Read the Install// section above for details. Before you deploy make sure to Update Dependencies: pip freeze > requirements.txt

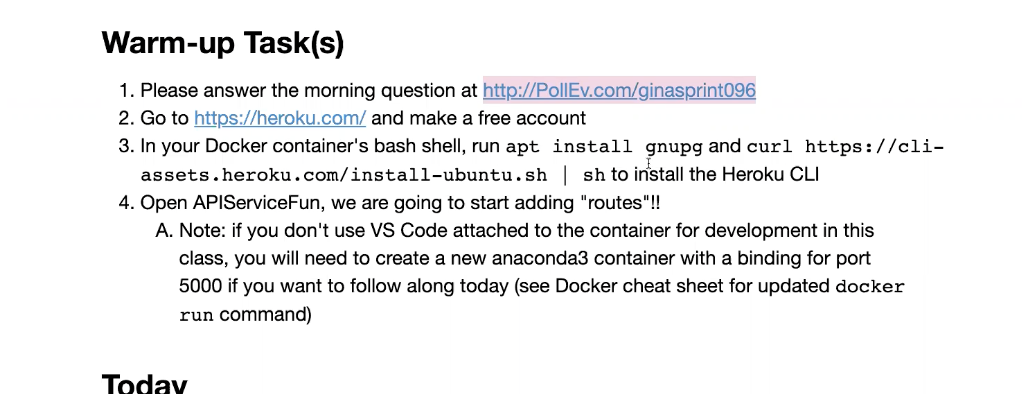
**Now, your system should be all set for Heroku! Now for how to deploy to it!**

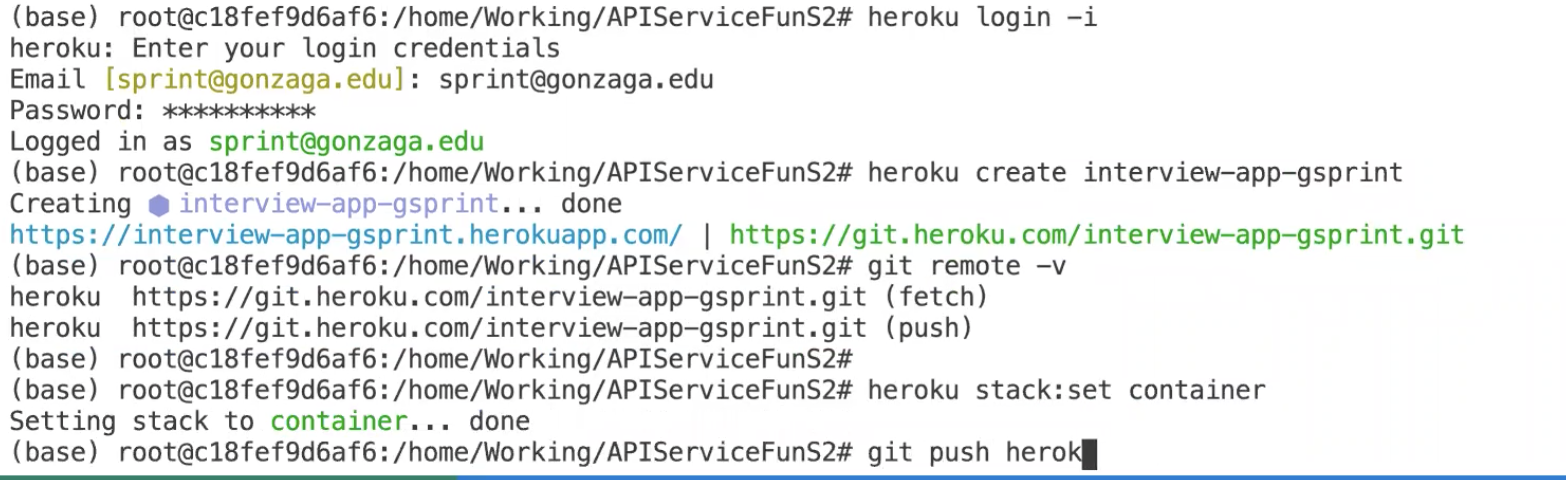
**Push to YOUR test Heroku:**

* **DO the Install/////// updates first**
* **Switch to master branch:** git checkout master
* **Login to heroku:** heroku login
  + Login to heroku from terminal
  + heroku login -i to stay in terminal
* **Add new changes:** git add –A
  + Heroku pushes with github, so make sure github branch is updated
* **Commit changes:** git commit –m “Stuff”
  + **Check Git Branch before pushing!:** git branch
* **Push to your branch:** git push
  + **Check remotes with:** git remote –v
  + **~~Change your remote?:~~** ~~heroku git:remote -a dev-cmooring-tester~~
  + **~~Push your git branch to your heroku app: git push -f heroku dev-cmooring:master~~**
* **Push master github branch:** git push -f heroku master

**Heroku//////////////////////////////////////////////////////////**

**Gina’s Guide:** [**https://github.com/GonzagaCPSC322/U0-Introduction/blob/master/B%20Environment%20Setup.ipynb**](https://github.com/GonzagaCPSC322/U0-Introduction/blob/master/B%20Environment%20Setup.ipynb)





[**https://stackoverflow.com/questions/55313610/importerror-libgl-so-1-cannot-open-shared-object-file-no-such-file-or-directo**](https://stackoverflow.com/questions/55313610/importerror-libgl-so-1-cannot-open-shared-object-file-no-such-file-or-directo)

[**https://medium.com/@tasnuva2606/dockerize-flask-app-4998a378a6aa**](https://medium.com/@tasnuva2606/dockerize-flask-app-4998a378a6aa)

[**https://pythonspeed.com/articles/activate-virtualenv-dockerfile/**](https://pythonspeed.com/articles/activate-virtualenv-dockerfile/)

**Problems////////////////////////////////////////////////////**

# ImportError: libGL.so.1: cannot open shared object file: No such file or directory**:**

<https://stackoverflow.com/a/51004957/14705655>

<https://elements.heroku.com/buildpacks/heroku/heroku-buildpack-apt>

* Heroku: Everything up-to-date, but you still want to:

<https://stackoverflow.com/questions/41345744/redeploy-heroku-app-without-code-changes>

* New Way to check sessions:

<https://pythonise.com/series/learning-flask/flask-session-object#:~:text=We%20use%20session.,doesn't%20exist%2C%20session>.

* **How I fixed my virtual env problem:**

It Seems that each time you work on the project you should be in a venv. However each of us would have different file paths for our venv isntall locations on our computers. Due to this, it would be wiser to just make a new venv every time since it is not that hard. However this is where all your packages are stored to make the code work. To get around this we have to be sure to keep the requirements.txt file up to date with the pip3 freeze > requirements.txt . This is because when you make a new venv you can install all the packages from the requrements.txt file onto the venv using this command: pip3 **install** -r **requirements**.**txt**

All my virtual environments had the same packages, this is becasue I accidentally installed them globally. I had to go into my folders and find where these packages were installed globally and delete them.

Inside of a virtual machine I put: pip show requests

This returned Location: /Library/Frameworks/Python.framework/Versions/3.8/lib/python3.8/site-packages

This is the location I looked for on my mac and I found the site-packages that were being installed in every venv I had been making. Delete everything in the site-packages folder manually. Doing this though meant I had to reinstall pip:

* + Use the following command to download pip directly,curl <https://bootstrap.pypa.io/get-pip.py> -o get-pip.py
  + Now execute the downloaded file using below commandpython3 get-pip.py

and requests pip install requests and virtualenv pip3 install virtualenv. These should all be kept global there, but all other Python packages should be installed in a venv that you start. This includes flask.

Now, I went to my project directory and made a virtualenv and activated it. Then I did a which pip

. [This stack overflow post](https://stackoverflow.com/a/43294133/14705655) told me what that command was supposed to output. Mine was showing the wrong path so I made a pip alias while in the venv that was the projects file path using: (which in venv)

alias pip3='~/Desktop/ClassesSpring2021/CPSC491\_(Senior\_Design)/FIDDL.nosync/cpsc-491-2-senior-project-facial-recognition-security-lock/venv/bin/pip3'

Now `which pip` will show that path and any installs while the venv is running will install to your projects path! Wooo!

* **How I fixed "how to use a file path in deployed python” or Absolute vs relative paths:**

Heroku was not able to store files how we were locally so I needed to fix my paths to relative ones. I had done this already with the app.config stuff but didn’t realize it

[**https://stackoverflow.com/a/49471006/14705655**](https://stackoverflow.com/a/49471006/14705655)

[**https://stackoverflow.com/questions/9271464/what-does-the-file-variable-mean-do**](https://stackoverflow.com/questions/9271464/what-does-the-file-variable-mean-do)

* **Heroku issues with sessions, made a config variable to keep a consistent secret key, otherwise local host uses the one in \_\_main\_\_**

[**https://devcenter.heroku.com/articles/config-vars**](https://devcenter.heroku.com/articles/config-vars)

* **Admin firebase added so we can delete photos from storage**

[**https://stackoverflow.com/questions/52883534/firebase-storage-upload-file-python**](https://stackoverflow.com/questions/52883534/firebase-storage-upload-file-python)

[**https://stackoverflow.com/a/66668548/14705655**](https://stackoverflow.com/a/66668548/14705655) **my post on it**

* **how to reduce Compiled slug size to heroku so it wont fail deploy**

[**https://thoughtbot.com/blog/how-to-reduce-a-large-heroku-compiled-slug-size**](https://thoughtbot.com/blog/how-to-reduce-a-large-heroku-compiled-slug-size)

* **Printing useful exceptions**

[**https://stackoverflow.com/questions/14519177/python-exception-handling-line-number/20264059**](https://stackoverflow.com/questions/14519177/python-exception-handling-line-number/20264059)

* **DOcker Install Issues**

[**https://stackoverflow.com/questions/60595581/failed-to-install-gcc-on-python-3-7-alpine-docker-container**](https://stackoverflow.com/questions/60595581/failed-to-install-gcc-on-python-3-7-alpine-docker-container)

**Use ‘apt install’ not ‘apk add’**

[**https://stackoverflow.com/questions/49911550/how-to-upgrade-disutils-package-pyyaml**](https://stackoverflow.com/questions/49911550/how-to-upgrade-disutils-package-pyyaml)

**Removed Pyrebase from requirements.txt since we use Pyrebase4 instead**

* **Upgrade python in venv**

<https://stackoverflow.com/questions/53040930/upgrading-python-to-3-7-inside-venv>

* **Skit learn version issues**
  + pip install --upgrade scikit-learn
  + Another issue comes from updating the package but not the embedding files, be sure to do that!

**Code I don’t want to lose after deprecation/////////////////////**

**Old Code for grabbing the names of user's photos and then getting the URLs for those photos' names:**

data = db.child("users").child(session['localId']).child("photos").get().val() #opens users in db, then finds person by uid in db

print("data", data)

# data has the id of each photo paired with the actual photo file name

# Parse the returned OrderedDict for filenames of user photos

for val in data.values():

print("val: " + str(val)) # val = file names of photos stored (from database aka dictionary)

#storage.child("images/" + userId + "/" + val).download(val, val) # dowloads image to local folder, testing only

imageURL = storage.child("images/" + userId + "/" + val).get\_url(None) # URL for Google Storage Photo location

print("imageURL: " + str(imageURL))

images.append(imageURL) # Stores the URL of each photo for the user

USER["image\_locations"] = images # Stores the users URL list in Global variable. TODO: Make sure to delete this when session ends

**Code that you can use with google firebase:**

print("session['usr']: " + session['usr'])

print("user: " + user)

print("auth.current\_user: " + auth.current\_user)

**Save photo locally:**

#Save photo to local directory, Testing only

image.save(os.path.join(current\_app.config["IMAGE\_UPLOAD"], filename)) #save images to /photosTest for testing

**JSON Response from Nest:**

2021-04-13T05:03:09.247504+00:00 app[web.1]: --------------------2021-04-13T05:03:09.253231+00:00 app[web.1]:

envelope

{'message':

{'data': '',

'messageId': '2169779074539096',

'message\_id': '2169779074539096',

'publishTime': '2021-04-13T05:03:09.047Z',

'publish\_time': '2021-04-13T05:03:09.047Z'},

'subscription': 'projects/fiddl-1604901867274/subscriptions/fiddl-sub'}

2021-04-13T05:03:09.253296+00:00 app[web.1]:

payload b'{\n

"eventId": "5c5ec2b9-f348-4530-bcdc-d6d7c410748f",\n

"timestamp": "2021-04-13T05:03:08.443Z",\n

"resourceUpdate": {\n

"name": "enterprises/428fdda2-61c1-41b1-b271- 1a657994ac/devices/AVPHwEvDLJQsZiLE6- AA7XH4J3OLKg5pqUIh69kYCOlASNxvmDK0IRRa0bj

HoNEWLLoHiOanMP1stHjUXVspoNc0DG1luQ",\n

"events" {\n

“sdm.devices.events.DoorbellChime.Chime": {\n

"eventSessionId": "AVPHwEuY\_P4tqRzdhYKWJ1kbeaKQEWVv5mFE1HiELHXM11EaznHZNC 3NiiDOerD25YirAbi9GuhMwFdfoXlwKeLufUpapQ",\n

"eventId": "CiQA2vuxrwbrzIXrhxXXYjOO843cw4DO6NGXFuoOOCf1RIJT5\_kSjAEAq- xCHsq3AQ83viopB16eQw4RwQdTX\_xpuTwjXOafBWggBdmEHuBU9o X7Ho0s3m7ZyqrgNYXOnqAeK8R6MmfwYL8bZDj1nWh3GL ZMx7UymA4gq3E2W6grLH75ToeHQJp68tf6OundbQzZyWOjxKlZMYQBgnj-T- \_EM0lGAdHHGUNK5FvRmB\_m7pUxhQ"\n }\n }\n },\n

"userId": "AVPHwEvdGjjc5zYAyEP2prjLIt6huo1DNEmgCs\_K5WcL",\n

"resourceGroup": ["enterprises/428fdda2-61c1-41b1-b271- 1e4a657994ac/devices/AVPHwEvDLJQsZiLE6- AA7XH4J3OLKg5pqUIh69kYCOlASNxvmDK0IRRa0 bjHoNEWLLoHiOanMP1stHjUXVspoNc0DG1luQ"]\n}'

2021-04-13T05:03:09.254715+00:00 app[web.1]: 10.95.191.28 - - [13/Apr/2021:05:03:09 +0000] "POST /doorbell HTTP/1.1" 200 2 "-"

"APIs-Google; (+https://developers.google.com/webmasters/APIs-Google.html)"

2021-04-13T05:03:09.260515+00:00 heroku[router]: at=info method=POST path="/doorbell" host=fiddl-dev.herokuapp.com request\_id=5704580f-033c-4f54-966d-426fc9f7549a fwd="66.102.8.82" dyno=web.1 connect=1ms service=9ms status=200 bytes=161 protocol=https