

The server is rather easy to run.

```
cd webserver
python2 runWebserver.py
```

if you wish to add a config file

- 1) look at webserver/webserver.conf.example
 - edit that as you want or make a new file following the same format, I have not tested having empty lines in the config file so make sure all lines are formatted with <option>:<setting> or #comment
- 2) then edit webserver/runWebServer.conf
 - change my_server = WebServer("127.0.0.1", 8008, config_file='<config file>')

The web apps require python3.5, php, php mongodb library (get through pecl install mongodb requires php-devel and php-pear), and mongodb

for the web apps, webapp2 does not actually run with cgi though it is possible to do so with flask since it was no longer a requirement I only worked on functionality. To test it out switch into it's directory and run

```
service/systemctl start mongod (and of course the correct formatting for whichever
command is used)
python3 -m venv flask
flask/bin/pip3 install -r requirements.txt
./makedatabase.py
./run.py
```

you can add users through
./adduserscript.py

These users will only work for webapp2 due to a difference in hashing libraries the default admin/bartender is

koalatea : temporary2017koalatea

for webapp1 you will need one more step cd webapp1
php-cgi signup.php (this is only a script to set up the two test accounts)

then deploy the webapp using deploy.sh (this will put the app into /var/www/cgi-bin and /var/www/html the php goes in the bin and the statics go into the html directory)

Now to use the app you have two accounts (or also koalatea2, same pass as 1 only difference is it has admin but there are no admin features so it means nothing)

user : user

bartender : bartender

the basic User Access Controls are

Unauthed gets access to

/menu.php

no buttons though since they cannot order drinks

/login.php

login duh

/index.php

just tells you to go to the menu

User gets some changes and more access

/menu.php

gains buttons for ordering

/review_order.php

the user can add notes for the bartender

/recent_orders.php

here the user can see the state of their drinks, order again, or cancel a drink, canceling will give a refund

/logout.php

destroys a users session

Bartender gets access to the same features except it also gets access to

/bartender.php and all the utilities that go with it

here the bartender can see all orders and click the button to follow the flow of making a drink, they will see any notes from a user here as well

webapp2 has more features that I chose not to reflect into webapp1 due to being beyond requirements and time. a better place to get webapp2 is the git link found in webapp2/extrareadme.md since that is my active repo