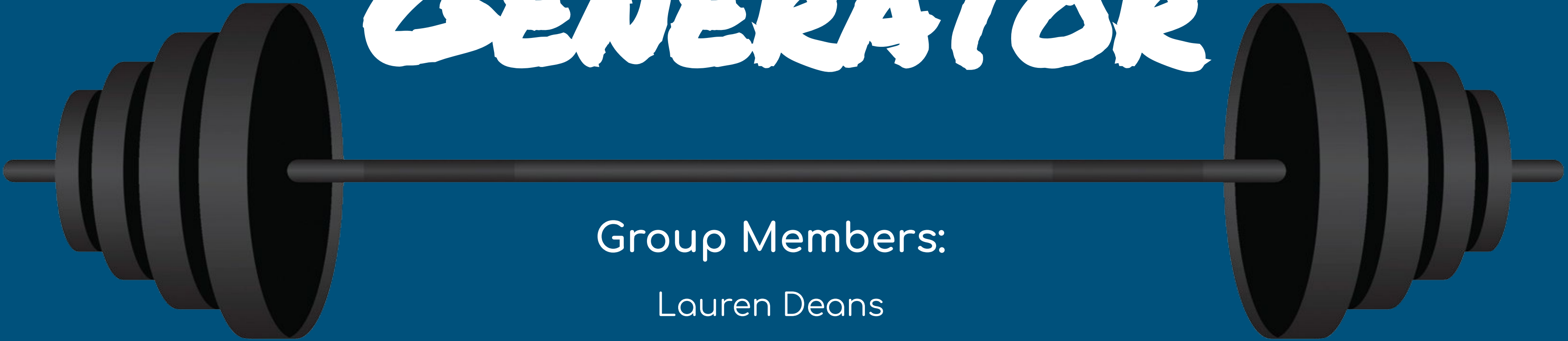
Workout Generator



Group Members:

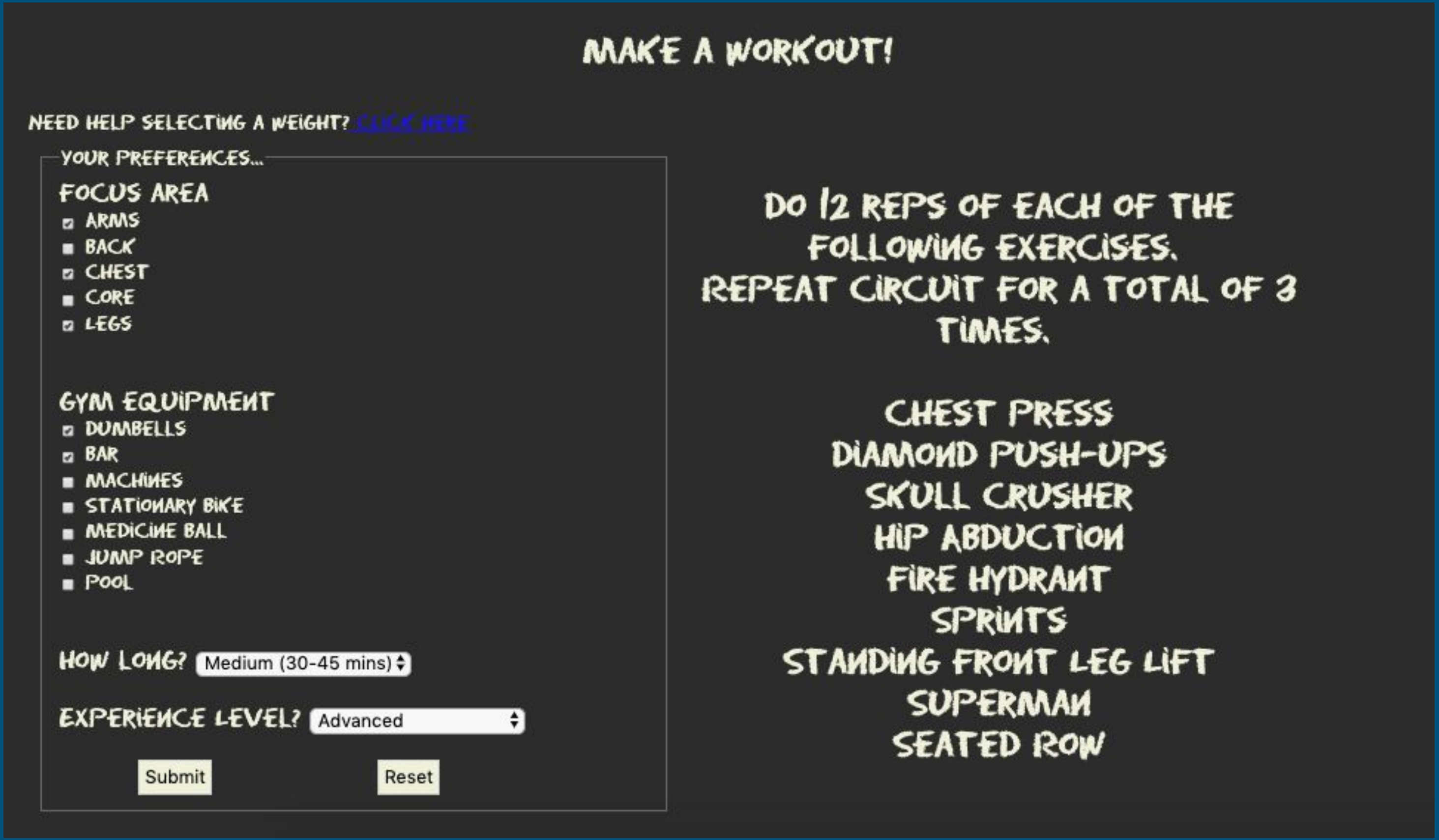
Lauren Deans

Christina Holt

Bret Murray

Derek Sessions

Workout Generator



MAKE A WORKOUT!

NEED HELP SELECTING A WEIGHT? CLICK HERE

YOUR PREFERENCES...

FOCUS AREA

- ☑ ARMS
- ☐ BACK
- ☑ CHEST
- ☐ CORE
- ☑ LEGS

GYM EQUIPMENT

- ☑ DUMBELLS
- ☑ BAR
- ☐ MACHINES
- ☐ STATIONARY BIKE
- ☐ MEDICINE BALL
- ☐ JUMP ROPE
- ☐ POOL

HOW LONG? Medium (30-45 mins)

EXPERIENCE LEVEL? Advanced

Do 12 REPS OF EACH OF THE FOLLOWING EXERCISES.
REPEAT CIRCUIT FOR A TOTAL OF 3 TIMES.

CHEST PRESS
DIAMOND PUSH-UPS
SKULL CRUSHER
HIP ABDUCTION
FIRE HYDRANT
SPRINTS
STANDING FRONT LEG LIFT
SUPERMAN
SEATED ROW

Utilized Development Tools

slack

zoom

Vim

Flask

HTML

CSS

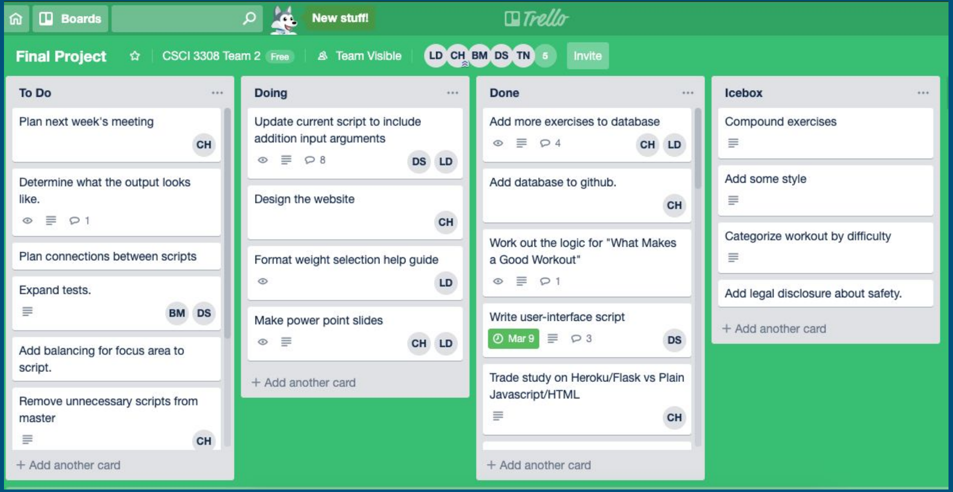
python™

Trello

GitHub

HEROKU

Google Sheets



To Do
- Plan next week's meeting — CH
- Determine what the output looks like. — 1
- Plan connections between scripts
- Expand tests. — BM DS
- Add balancing for focus area to script.
- Remove unnecessary scripts from master — CH
- + Add another card

Doing
- Update current script to include addition input arguments — 8 — DS LD
- Design the website — CH
- Format weight selection help guide — LD
- Make power point slides — CH LD
- + Add another card

Done
- Add more exercises to database — 4 — CH LD
- Add database to github. — CH
- Work out the logic for "What Makes a Good Workout" — 1
- Write user-interface script — Mar 9 — 3 — DS
- Trade study on Heroku/Flask vs Plain Javascript/HTML — CH
- + Add another card

Icebox
- Compound exercises
- Add some style
- Categorize workout by difficulty
- Add legal disclosure about safety.
- + Add another card

Trello is a collaboration tool used for Project Tracking.

We utilized Trello by visually placing tasks into 4 categories:

- "To-Do"
- "Doing"
- "Done"
- "Icebox"

This is similar to Kanban approach.

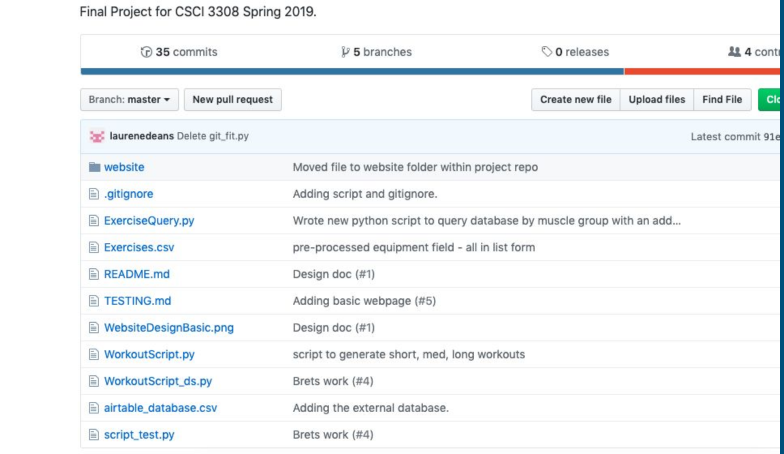
Each task:

- Could be assigned to a particular team member(s)
- Given a due date
- Enabled comments, attachments, and links to other tasks.

Evaluation of Trello:

- Effective for accountability
- Helped keep us organized during weekly scrum meetings
- Very useful tool for any work- individual or team- to track progress

GitHub



Final Project for CSCI 3308 Spring 2019.

laurenedeans Delete git_fit.py

Latest commit 91e

website	Moved file to website folder within project repo
.gitignore	Adding script and gitignore.
ExerciseQuery.py	Wrote new python script to query database by muscle group with an add…
Exercises.csv	pre-processed equipment field - all in list form
README.md	Design doc (#1)
TESTING.md	Adding basic webpage (#5)
WebsiteDesignBasic.png	Design doc (#1)
WorkoutScript.py	script to generate short, med, long workouts
WorkoutScript_ds.py	Brets work (#4)
airtable_database.csv	Adding the external database.
script_test.py	Brets work (#4)

GitHub is a web-based tool used for Version Control using git.

We utilized a git repository hosted on GitHub. We interacted with our repository by using git from the command line and also from GitHub directly.

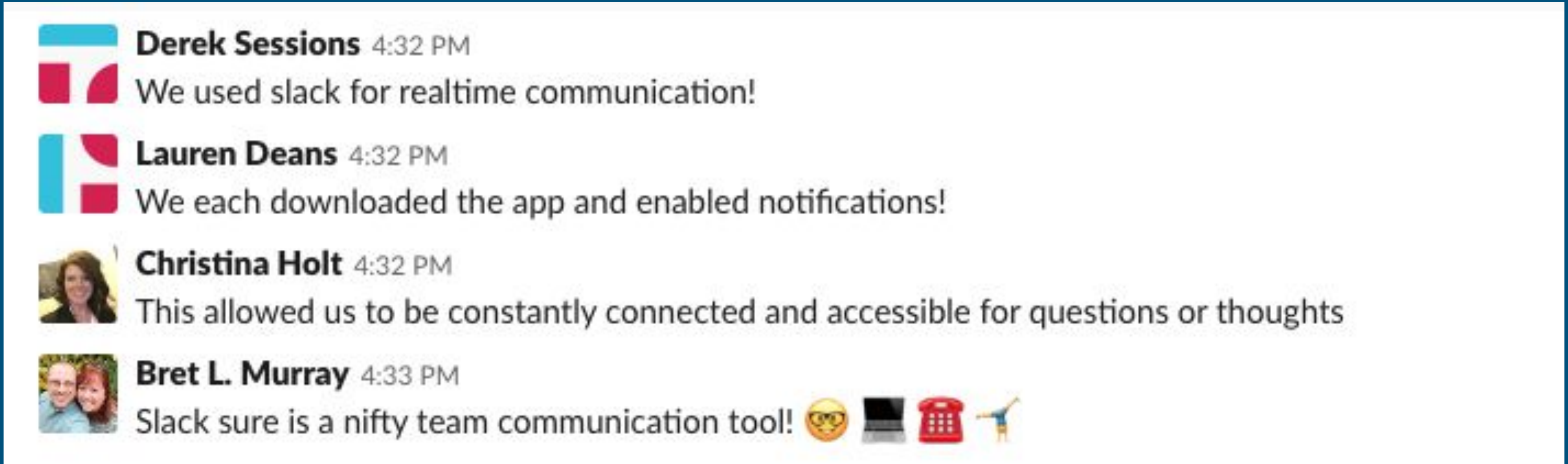
Benefits of using GitHub:

- Able to collaborate in shared environment
- Complete version history accessible through commits
- Able to work out of individual branches
- Easily accessible online repository

Our Evaluation:

- Effective for version control
- Became easier with more experience
- Appreciated the security of version history

slack



Derek Sessions 4:32 PM
We used slack for realtime communication!

Lauren Deans 4:32 PM
We each downloaded the app and enabled notifications!

Christina Holt 4:32 PM
This allowed us to be constantly connected and accessible for questions or thoughts

Bret L. Murray 4:33 PM
Slack sure is a nifty team communication tool! 🤓 💻 ☎ 🤸

zoom

Zoom is a remote video conferencing service that uses cloud computing.

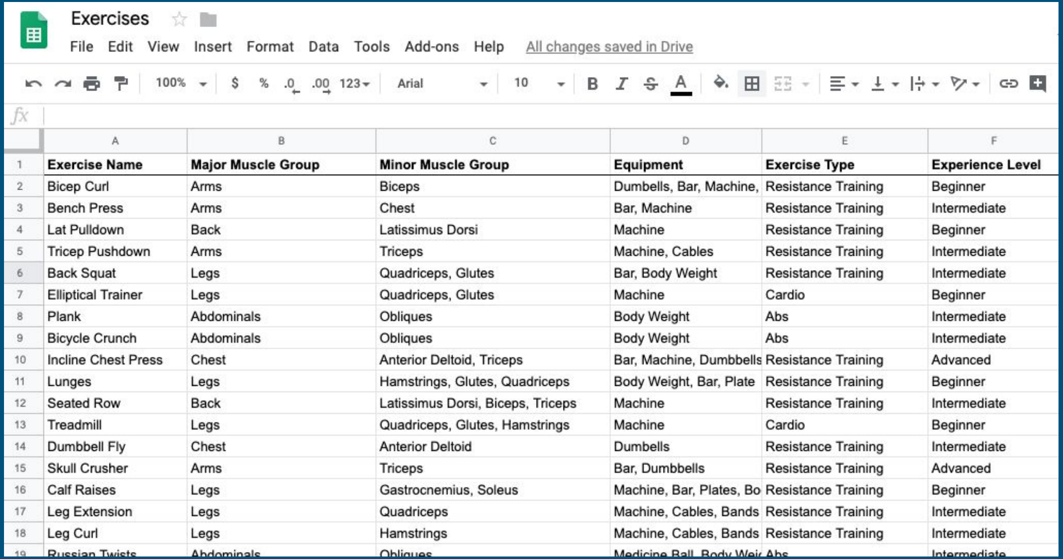
We utilized Zoom for our weekly standup meetings. Benefits of using Zoom:

- Able to all meet in one place
- Easy to set up a Zoom call and provide shareable link to team members
- Options to screen share, text chat, and record a meeting

Our Evaluation:

- Screen sharing was efficient for pair programming/ troubleshooting code
- Connectivity was consistent (very few lags)
- Effective tool for working in remote groups.

Google Sheets


Exercises
File Edit View Insert Format Data Tools Add-ons Help All changes saved in Drive
100% $ % .0 .00 123 Arial 10 B I S A
fx

	A	B	C	D	E	F
1	Exercise Name	Major Muscle Group	Minor Muscle Group	Equipment	Exercise Type	Experience Level
2	Bicep Curl	Arms	Biceps	Dumbells, Bar, Machine,	Resistance Training	Beginner
3	Bench Press	Arms	Chest	Bar, Machine	Resistance Training	Intermediate
4	Lat Pulldown	Back	Latissimus Dorsi	Machine	Resistance Training	Beginner
5	Tricep Pushdown	Arms	Triceps	Machine, Cables	Resistance Training	Intermediate
6	Back Squat	Legs	Quadriceps, Glutes	Bar, Body Weight	Resistance Training	Intermediate
7	Elliptical Trainer	Legs	Quadriceps, Glutes	Machine	Cardio	Beginner
8	Plank	Abdominals	Obliques	Body Weight	Abs	Intermediate
9	Bicycle Crunch	Abdominals	Obliques	Body Weight	Abs	Intermediate
10	Incline Chest Press	Chest	Anterior Deltoid, Triceps	Bar, Machine, Dumbbells	Resistance Training	Advanced
11	Lunges	Legs	Hamstrings, Glutes, Quadriceps	Body Weight, Bar, Plate	Resistance Training	Beginner
12	Seated Row	Back	Latissimus Dorsi, Biceps, Triceps	Machine	Resistance Training	Intermediate
13	Treadmill	Legs	Quadriceps, Glutes, Hamstrings	Machine	Cardio	Beginner
14	Dumbbell Fly	Chest	Anterior Deltoid	Dumbells	Resistance Training	Intermediate
15	Skull Crusher	Arms	Triceps	Bar, Dumbbells	Resistance Training	Advanced
16	Calf Raises	Legs	Gastrocnemius, Soleus	Machine, Bar, Plates, Bo	Resistance Training	Beginner
17	Leg Extension	Legs	Quadriceps	Machine, Cables, Bands	Resistance Training	Intermediate
18	Leg Curl	Legs	Hamstrings	Machine, Cables, Bands	Resistance Training	Intermediate
19	Russian Twists	Abdominals	Obliques	Medicine Ball, Body Weic	Abs	Intermediate

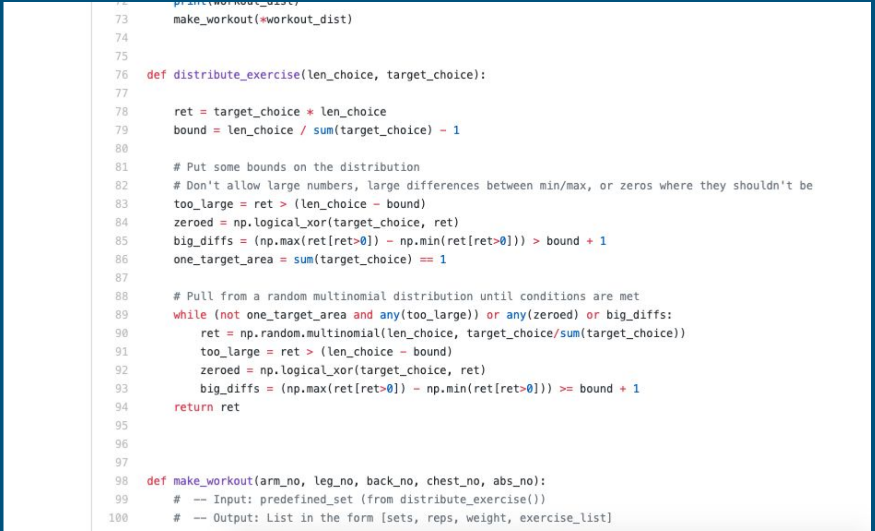
We used Google Sheets for Database Management.

Benefits of using Google Sheets:

- Very simple and easy to use
- Most individuals have prior experience
- No installation required, accessible from Chrome Browser (Software as a Service)

Our Evaluation:

- We needed our database early on in the project (before learning SQL), so we used an already familiar tool
- Served purpose of designing and managing database
- Easy to export as .csv file



```python
    print(workout_dist)
    make_workout(*workout_dist)


def distribute_exercise(len_choice, target_choice):

    ret = target_choice * len_choice
    bound = len_choice / sum(target_choice) - 1

    # Put some bounds on the distribution
    # Don't allow large numbers, large differences between min/max, or zeros where they shouldn't be
    too_large = ret > (len_choice - bound)
    zeroed = np.logical_xor(target_choice, ret)
    big_diffs = (np.max(ret[ret>0]) - np.min(ret[ret>0])) > bound + 1
    one_target_area = sum(target_choice) == 1

    # Pull from a random multinomial distribution until conditions are met
    while (not one_target_area and any(too_large)) or any(zeroed) or big_diffs:
        ret = np.random.multinomial(len_choice, target_choice/sum(target_choice))
        too_large = ret > (len_choice - bound)
        zeroed = np.logical_xor(target_choice, ret)
        big_diffs = (np.max(ret[ret>0]) - np.min(ret[ret>0])) >= bound + 1
    return ret



def make_workout(arm_no, leg_no, back_no, chest_no, abs_no):
    #  -- Input: predefined_set (from distribute_exercise())
    #  -- Output: List in the form [sets, reps, weight, exercise_list]
```

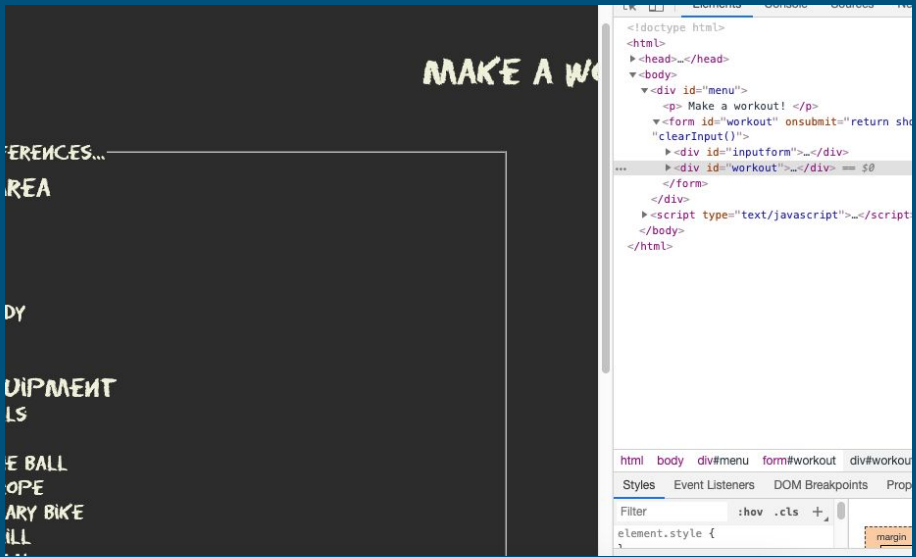
Python is an object-oriented, high-level programming language.

Benefits of using Python:

- As a group we had the most combined experience programming in Python
- User-friendly data structures
- Easy readability
- Offers many open-source packages which are helpful for backend and frontend development

Our Evaluation:

- We used Python for writing our main workout generator script and testing scripts.
- Preferred over other languages such as C++
- Was effective programming language for this project


MAKE A W
FERENCES...
REA
DY
UIPMENT
LS
E BALL
OPE
ARY BIKE
ILL

```
<!doctype html>
<html>
 ▶<head>…</head>
 ▼<body>
   ▼<div id="menu">
      <p> Make a workout! </p>
     ▼<form id="workout" onsubmit="return sho
     "clearInput()">
       ▶<div id="inputform">…</div>
…      ▶<div id="workout">…</div> == $0
      </form>
    </div>
   ▶<script type="text/javascript">…</script
  </body>
</html>
```

html body div#menu form#workout div#workou
Styles Event Listeners DOM Breakpoints Prop
Filter :hov .cls +
margin

```
element.style {
```

We used HTML and CSS for the framework of our web page.

Benefits of using HTML and CSS:

- HTML is widely supported, almost all browsers are compatible
- CSS (Cascading Style Sheets) assist in formatting, keep code clean
- Both HTML and CSS are straightforward and not difficult to use or learn

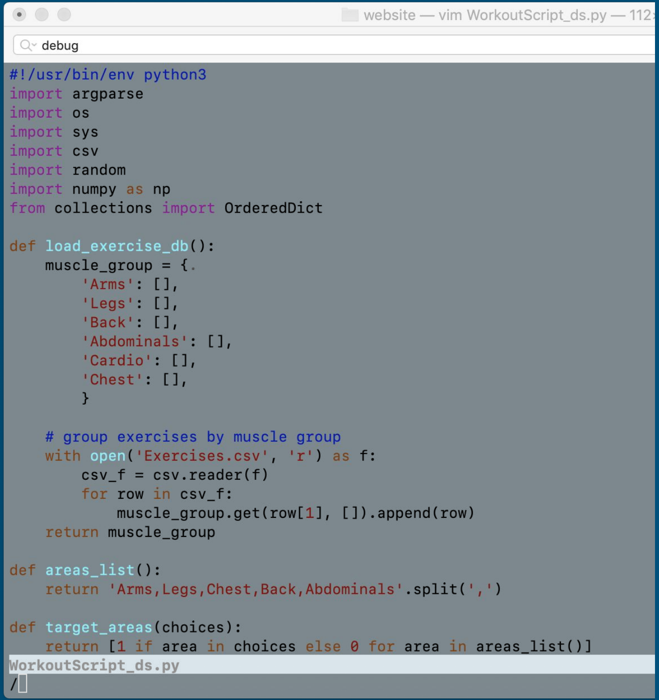
Our Evaluation:

- HTML and CSS were appropriate languages to utilize for the project
- Effective for accomplishing what was intended
- Fun to learn and utilize, not overly complicated

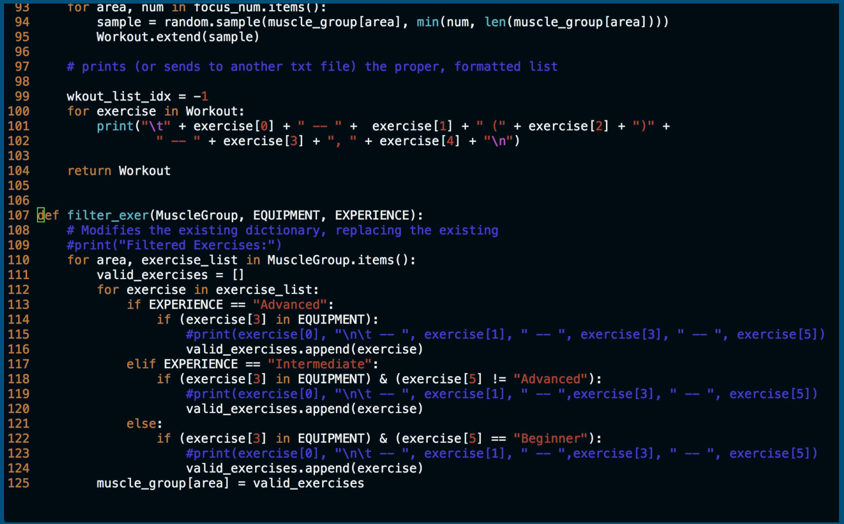
We utilized Terminal to interact with code and Vim as a text editor to edit code.

Our Evaluation:

- Enjoyed the customization features of Vim and Terminal
- No installation required
- Effective tools for code writing/editing.



```
#!/usr/bin/env python3
import argparse
import os
import sys
import csv
import random
import numpy as np
from collections import OrderedDict

def load_exercise_db():
    muscle_group = {
        'Arms': [],
        'Legs': [],
        'Back': [],
        'Abdominals': [],
        'Cardio': [],
        'Chest': [],
        }

    # group exercises by muscle group
    with open('Exercises.csv', 'r') as f:
        csv_f = csv.reader(f)
        for row in csv_f:
            muscle_group.get(row[1], []).append(row)
    return muscle_group

def areas_list():
    return 'Arms,Legs,Chest,Back,Abdominals'.split(',')

def target_areas(choices):
    return [1 if area in choices else 0 for area in areas_list()]
WorkoutScript_ds.py
/
```

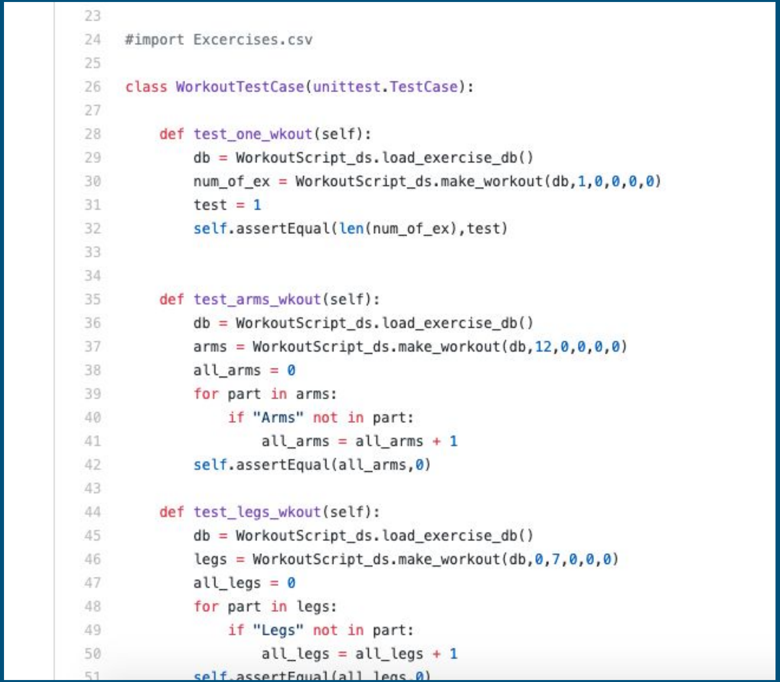


```
    for area, num in focus_num.items():
        sample = random.sample(muscle_group[area], min(num, len(muscle_group[area])))
        Workout.extend(sample)

    # prints (or sends to another txt file) the proper, formatted list

    wkout_list_idx = -1
    for exercise in Workout:
        print("\t" + exercise[0] + " -- " +  exercise[1] + " (" + exercise[2] + ")" +
                " -- " + exercise[3] + ", " + exercise[4] + "\n")

    return Workout


def filter_exer(MuscleGroup, EQUIPMENT, EXPERIENCE):
    # Modifies the existing dictionary, replacing the existing
    #print("Filtered Exercises:")
    for area, exercise_list in MuscleGroup.items():
        valid_exercises = []
        for exercise in exercise_list:
            if EXPERIENCE == "Advanced":
                if (exercise[3] in EQUIPMENT):
                    #print(exercise[0], "\n\t -- ", exercise[1], " -- ", exercise[3], " -- ", exercise[5])
                    valid_exercises.append(exercise)
            elif EXPERIENCE == "Intermediate":
                if (exercise[3] in EQUIPMENT) & (exercise[5] != "Advanced"):
                    #print(exercise[0], "\n\t -- ", exercise[1], " -- ",exercise[3], " -- ", exercise[5])
                    valid_exercises.append(exercise)
            else:
                if (exercise[3] in EQUIPMENT) & (exercise[5] == "Beginner"):
                    #print(exercise[0], "\n\t -- ", exercise[1], " -- ",exercise[3], " -- ", exercise[5])
                    valid_exercises.append(exercise)
        muscle_group[area] = valid_exercises
```

Deployment Environment

We utilized a Python script with Flask framework and deployed our web application using Heroku.

Testing



```python

#import Excercises.csv

class WorkoutTestCase(unittest.TestCase):

    def test_one_wkout(self):
        db = WorkoutScript_ds.load_exercise_db()
        num_of_ex = WorkoutScript_ds.make_workout(db,1,0,0,0,0)
        test = 1
        self.assertEqual(len(num_of_ex),test)


    def test_arms_wkout(self):
        db = WorkoutScript_ds.load_exercise_db()
        arms = WorkoutScript_ds.make_workout(db,12,0,0,0,0)
        all_arms = 0
        for part in arms:
            if "Arms" not in part:
                all_arms = all_arms + 1
        self.assertEqual(all_arms,0)

    def test_legs_wkout(self):
        db = WorkoutScript_ds.load_exercise_db()
        legs = WorkoutScript_ds.make_workout(db,0,7,0,0,0)
        all_legs = 0
        for part in legs:
            if "Legs" not in part:
                all_legs = all_legs + 1
        self.assertEqual(all_legs,0)
```

- Used the unittest framework
- Completed a subset of tests necessary to fully test the script
- Struggles arose from:
    - Disjoint communication/development cycle
    - Learning unittest
    - Learning our own workout scripts

Our Challenges:

- Communication
- Maintaining agile environment in a dispersed team
- Broadening our knowledge/experience while concurrently building our application
- Learning a plethora of new tools
- Managing data structures
- Time/connecting schedules
- Not asking enough questions of each other, the rest of the class, and the instructor
- Knowing which questions to ask