

Sprint 1 Owner: Jason Ledon (jledon)

- Complete Product Backlog
  - Graphing with AJAX
    - Users will be able to dynamically graph their entered data for a variety of categories such as squat maxes, macronutrients, and weight.
    - Users will be able to filter this data to within a specific time range
  - HTML Framework
    - The website will include a visually appealing and interactive UI that will allow the user to easily navigate through the website.
  - OAuth
    - Rather than creating an account for the website, users will be required to use a google linked account.
    - In their profile screen, they will have the option to provide us with additional information such as zip code or name (if they aren't already received by OAuth)
  - Geolocation
    - User matching based on location (and potentially other aspects such as lift maxes and gym experience)
    - Upon clicking on a gym shown as nearby, a panel should pop up with information relative to that gym received from the google maps API
  - Messaging
    - Users will be able to communicate with their Gym Buddies through the app once they have matched based on certain preferences in order to schedule workout times.
- First Sprint Backlog
  - Graphing with AJAX (Jason)
    - Using AJAX to jump between visualizations of user filtered data
  - HTML framework (Edison, Jason)
    - Have a semi-decent looking UI with interactive features and working redirections.
    - Have a base framework for pages such as Daily Diary, Profile, Gym Info, and Gym Buddies which will later be populated and implemented.
    - Have a working progress page with buttons the user can use in order to update the page's visuals.
  - OAuth (Mukundh)
    - Currently in progress, working on minor fixes in order to implement fully.
- Data Models

```
from codecs import backslashreplace_errors
from email import message
from django.db import models
```

```

from django.contrib.auth.models import User

# Create your models here.
class Person(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE) #
    UNSURE: refer back to Cascade
    zipcode = models.CharField(blank=True, max_length=10)
    def __str__(self):
        return f'Person({self.user.first_name})'

class ExerciseList(models.Model):
    squat_weight = models.IntegerField(default=0)
    squat_reps = models.IntegerField(default=0)
    deadlift_weight = models.IntegerField(default=0)
    deadlift_reps = models.IntegerField(default=0)
    bench_weight = models.IntegerField(default=0)
    bench_reps = models.IntegerField(default=0)

    def __str__(self):
        return f'Excercises(Squat: {self.squat_weight}-{self.squat_reps},
Deadlift: {self.deadlift_weight}-{self.deadlift_reps}, Bench:
{self.bench_weight}-{self.bench_reps})'

class DailyDiary(models.Model):
    owner = models.ForeignKey(Person, on_delete=models.PROTECT)
    creation_time = models.DateTimeField()

    exercises = models.OneToOneField(ExerciseList,
on_delete=models.CASCADE)

    protein = models.IntegerField(default=0)
    carbs = models.IntegerField(default=0)
    fats = models.IntegerField(default=0)

    attended_gym = models.BooleanField(default=False)
    weight = models.IntegerField(default=0)

    def __str__(self):
        return f'DailyDiary({self.owner} - p:{self.protein}, c:{self.carbs},
f:{self.fats})' # id: {self.id},

class Message(models.Model):
    sender = models.ForeignKey(Person, on_delete=models.PROTECT,
related_name="sender")

```

```

    recipient      = models.ForeignKey(Person, on_delete=models.PROTECT,
related_name="recipient")
    creation_time   = models.DateTimeField()
    message         = models.CharField(max_length=500)

```

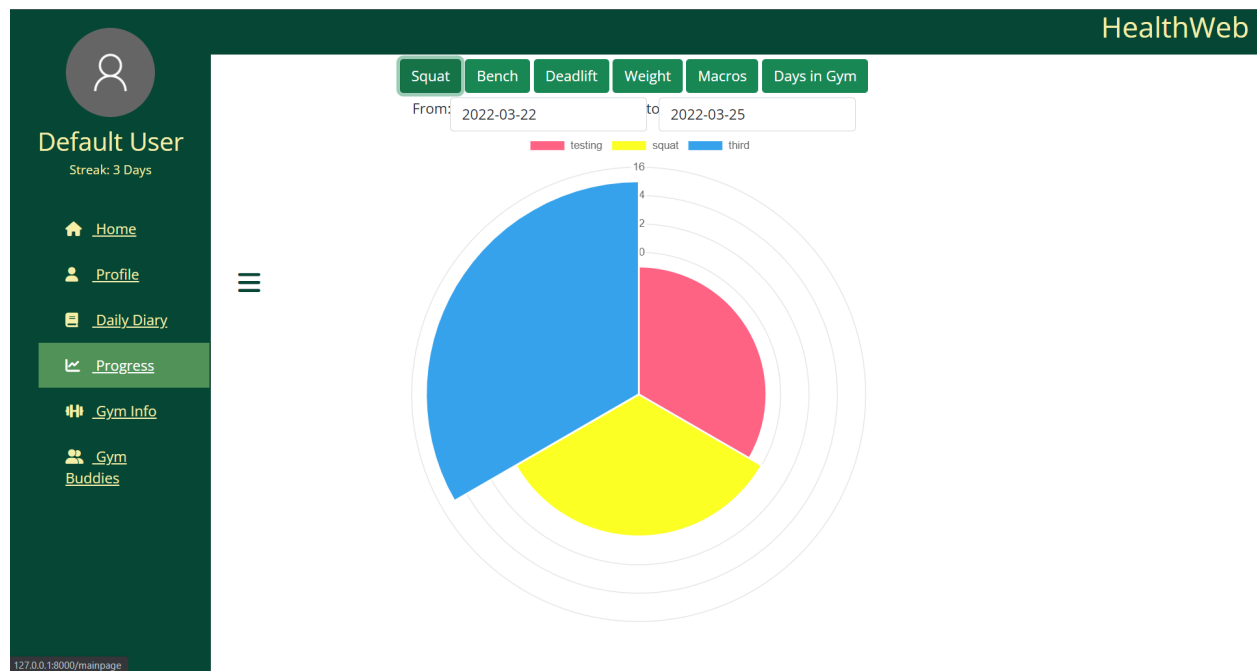
- Drawn Frameworks

HealthWeb

**Default User**  
Streak: 3 Days

- [Home](#)
- [Profile](#)
- [Daily Diary](#)
- [Progress](#)
- [Gym Info](#)
- [Gym Buddies](#)

Proteins(g)* <input type="text" value="0"/>	Squat Weight* <input type="text" value="0"/>	Squat Reps* <input type="text" value="0"/>
Carbs(g)* <input type="text" value="0"/>	Bench Weight* <input type="text" value="0"/>	Bench Reps* <input type="text" value="0"/>
Fats(g)* <input type="text" value="0"/>	Deadlift Weight* <input type="text" value="0"/>	Deadlift Reps* <input type="text" value="0"/>
Current Weight* <input type="text" value="0"/>	<a href="#">Complete Diary</a>	



Page name		Max Squat Max Deadlift Max Bench	Logout
Daily Diary  progress  geolocat			<p>Info about home/selected gym</p> <p>Address Open Hours Phone Number Website Reviews</p> <p>maybe the graph that google shows about buys hours if we can get access to it</p>

Shows your location and the location of the gyms around you

Page name		Max Squat Max Deadlift Max Bench	Logout
Daily Diary  progress  geolocat  friends	list of friends	messages	
		<div></div> <div></div> <div></div>	
		text box	