Sprint 5

Team on a Cob

What happened

- No new features implemented in this sprint

- Running out of work that can realistically be finished this year

- Fixed some security issues

- Began cleaning up code to prepare for handing it off

Attempted new feature

- One new feature that was attempted, but not implemented was a forgot password functionality for resetting passwords
- Ran into technical issues with this
    - Libraries that allowed emailing functionality required subscriptions
- This would potentially be a good goal for another developer next year to work on, but if the project doesn't continue, we should just remove the half-implemented feature.

Sprint 5 Goals

Robert Whitmer
- Make changes to data export according to client
- Make changes to charts according to client
- Set up "forgot password" email service

Brent Gruber
- Implement database side of "forgot password" feature

Andrew Leach
- Error checking on routes

Peter Essman
- Commenting/Cleaning up code

Lucas Nation
- Clean up coach page, add training log page to coaches page

Individual Contributions

Robert:
- Fixed some (hopefully) final bugs with the charts.
- Worked on finishing Update button w/ Brent as well as correcting an issue w/ data export
- Fixed security issue with DB login info being exposed to public

Brent:
- Fixed SQL Injection issue by parameterizing queries
- Fixed bugs with retrieving team association
- Worked on the update button on training log, so that all fields are updated

Peter:
- Removed excessive console.log statements
- Commented code for better documentation

Andrew:
- Fixed NavBar bug where bar would preload overwritten format
- Routing improvements
- Added new toast feedback

Lucas:
- Fixed Coaches page bug where coaches could see all users

Code Cleanup

- Our code had a lot of console.log statements which we had used for debugging purposes
  - These were sending too much information to the console and potentially leaking important information
  - Went through and removed many of these statements
- Starting to add more comments to the code
  - Did not do a good job of this during the initial implementation

Security issues

- Our site was prone to sql injection prior to this sprint
  - Fixed by parameterizing queries

```
//add the conditions that were passed in
if(input.username){
    conditions.push(`username="${input.username}"`);
}
if(input.password){
    conditions.push(`password="${input.password}"`);
}
```

```
//add the conditions that were passed in
if(input.username){
    conditions.push(`username=?`);
    values.push(input.username);
}

if(input.password){
    conditions.push(`password=?`);
    values.push(input.password);
}
```

Security Issues

For the past two semesters, we have been having the server log into the database just by hardcoded values.

This isn't technically bad if you know with certainty that the file will not be accessible to outsiders, but that is not always feasible.

In our case, the project is stored on a public Github repo, so anyone had access to this login information for the past year (there are bots that snoop through public repos looking for this info)

```
var mysql = require('mysql');

var connection = mysql.createConnection({
    host     : 'ouwxc.cevwncmpvhy1.us-west-2.rds.amazonaws.com',
    user     :
    password :
    database : 'PPA'
});
```

Bad

Security Issues

To overcome this issue, AWS allows users to preload in environmental variables in the server's shell.

We can then access those environmental variables by grabbing **process.env.[env_var]**

Environment Properties

The following properties are passed into the application as environment variables. Learn more.

Property Name
password
username

```
1   var mysql = require('mysql');
2   var connection = mysql.createConnection({
3       host      : 'ouwxc.cevwncmpvhy1.us-west-2.rds.amazonaws.com',
4       user      : process.env.username,
5       password  : process.env.password,
6       database  : 'PPA'
7   });
```

Plans to wrap it up

- We believe our application is ready for use as far as data collection and retrieval.
- The indications from our client are that this website is usable for basic functions.
  - Will reduce his throughput time for athlete workouts from about a week to almost instant
- Plenty of room for improvement if project is pursued further
- Need to prepare help documents to allow client (or future devs) to know how to setup AWS server and other small details with our project (env vars, etc).

Possible future adaptations

- More Data analysis for coaches (Zone calculation, etc)
- Native mobile apps
  - Though, our client has mentioned that he doesn't really believe this is necessary anymore as he found the mobile version of our site to be very fluid
- Add in ability to sync with smart wear: Fitbit, Garmin etc…
- Try out a different charting library that allows features similar to Excel (can scroll along horizontal, etc)
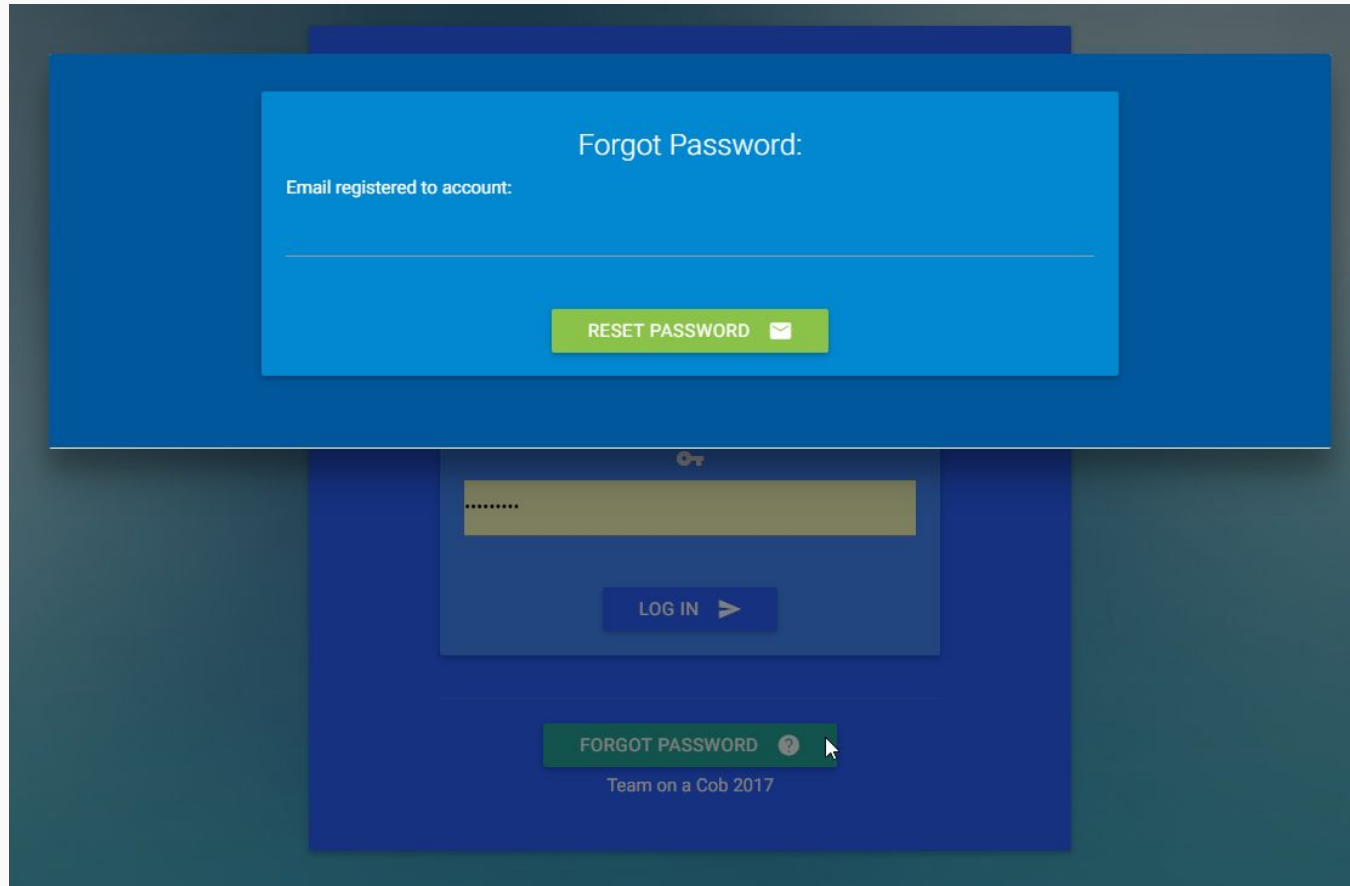
Reflections on AGILE process

- While the Agile Development process works well in the real world, in the academic world (where students all have different schedules with different times where they can work), things inherently get pushed back. Because there will be times where some students simply have no time to develop, features will get pushed back and that slows the development process.
- Having a Tu/Thr class with work days where we can guarantee we'll have at least an hour or so of development time would relieve this issue.
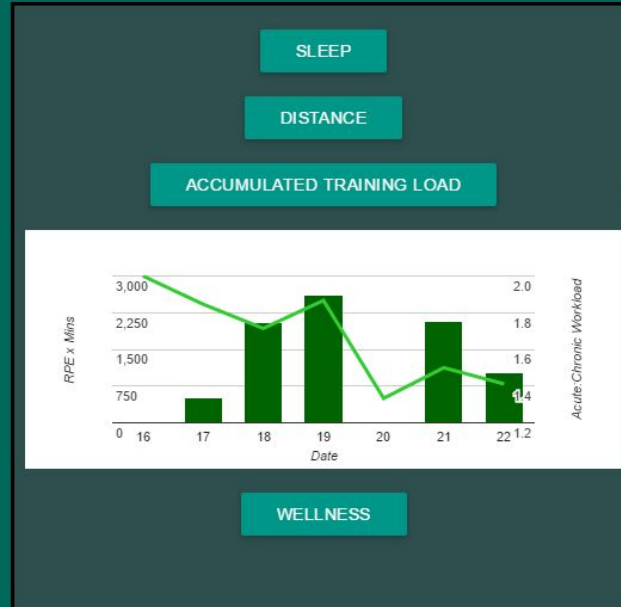
Live Demo - Final version

http://ouwxcpp.ik3pvw7c5h.us-west-2.elasticbeanstalk.com

Coaches View Athletes

VIEW TEAM - ["OUWXC","OUWXC","OUWXC","CS"]

Username	First	Last	
Clev	Mike	Clevidence	VIEW CHARTS
JYoYO	Jen	Yoder	VIEW CHARTS
Andy	Andy	L	VIEW CHARTS
brent	brent	gruber	VIEW CHARTS

Remove a User:

Username:

----   ▼

Andy
Clev
Cleviden
JYoYO
LIL JON SON
brent
jeffdunham
joethefro
johnsmith
roberto

Note:
Please ensure you want to permanently remove this user, this process **cannot** be reversed.

REMOVE USER ➤