



NORTHERN ARIZONA
UNIVERSITY

College of Engineering, Forestry & Natural Sciences

Collegiate Esports Platform

<https://github.com/deltarod/CS386-Group-20>

Members:

Blake Lawton
Tristan Miller
Ruben Rincon
Jennie Ryckman

Course:

CS 386
Marco Gerosa
Spring 2018

D.5 Release 1

1. Introduction

Collegiate Esport Finder is a website that allows college students to find other video game players and create teams for their college, allowing them to find players of similar games, and build an online community around a college. Consumer segment: College students that like to play video games have a hard time finding users.

The website allows players to find schools based off their name of their university. It also allows you to build your own profile under that schools name. The idea behind this is to enable players to look up their schools and sign into their schools to then relate to other players at that school. We also were able implement a database that stores all the users signing up for the website.

Trello link: <https://trello.com/b/NPTDEIUD/e-sports-project>

Github link: <https://github.com/deltarod/CS386-Group-20>

2. Implemented Requirements

Search functionality(Tristan, Blake): We implemented a basic search feature that can search for pages containing a keyword.

Search for college or player(Tristan, Blake): Adding on to the search functionality, the search bar allows users to search schools/colleges and players if their name matches the keyword.

Profile creation(Tristan, Ruben): A user can currently register and create a profile with basic information such as their name and the college they belong to.

Join college as user(Tristan, Jennie): If unaffiliated with a college, a user can join a selected college and add themselves to the player list. They can only join one college and must leave their current choice before joining another one.

3. Adopted technologies

Django: A high-level Python Web framework. We chose this because it uses Python, a language we're somewhat familiar with. Django also claimed to take some hassle out of web development and encourages rapid/clean design.

SQLite3: We chose SQLite due to it being included in Python by default. It's more lightweight than a dedicated SQL platform and can support mobile devices in the future.

Python: Since Django is a Python framework, the usage of Python is mandatory for this.

HTML: Building any site layout requires HTML, which we need since our first target is to build the website before the apps.

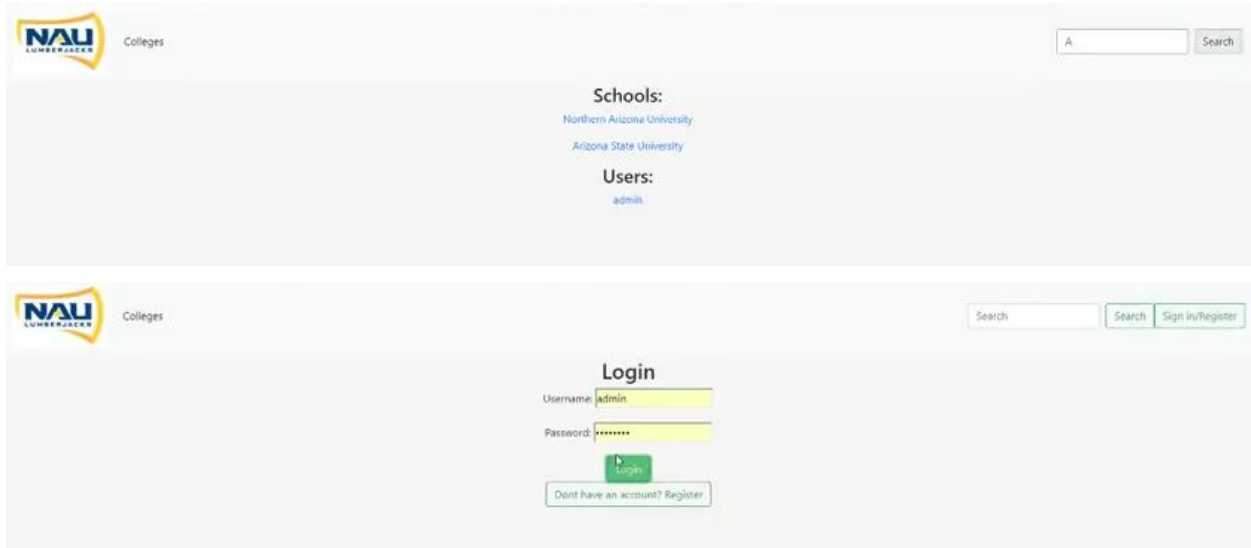
CSS: Since we didn't want our website to look like a set of plaintext files, it's a given for us to use CSS to apply designs to the web pages as needed.

4. Licensing

For licensing, we decided to go with the MIT license, this allows others to use this site for commercial use, and improve upon it. The MIT license also explicitly states that the software is offered as is, so there is no liability or warranty stated

5. Look & feel

Since this is our first release, we were focusing more on functionality than design. Still, we did add basic CSS so everything wouldn't look like plaintext. Our guidelines were to make all of our pages have a basic white color scheme with easily visible buttons. There isn't anything to distract the user from the site functions besides the NAU logo in the upper-left corner. If most of our features are finished in the future we might focus on more colorful site designs.



6. Learning/training

All of us have some experience with HTML, CSS, and Python, but the biggest challenge was learning how to use the Django framework. Although it's Python based we had to look up videos and Django tutorials on the web. Learning SQLite wasn't too bad since it's already built into Python. For HTML and CSS we just looked up refresher tutorials.

7. Lessons Learned

We learned that trying to code modularity into the code was extremely difficult as the language we were using was based off modules. We were able to use a lot of online resources to style out our pages mostly css pages. Another key lesson we learned is that the uml class diagram wasn't as helpful as we thought it was going to be as we used django to code our website.

8. Demo

<https://youtu.be/Bh6FufaclA8>

9. Group Participation

Tristan Miller: Wrote website code, did #4, #8 (25%)
Ruben Rincon: Code for using college, #2, #3 (25%)
Blake Lawton: Code for search function, #1, #7 (25%)
Jennie Ryckman: Code for joining a college, #5, #6 (25%)