Fighting Aggies Analytics Platform
Project Report

Members

Andrew Imwalle, Nick McDonough, Elias Tracy, Tuong Tran

Presentation and Demo Video

https://youtu.be/chHc0VKYA_Y

Important Links

Pivotal Tracker: https://www.pivotaltracker.com/n/projects/2598148
Github: https://github.com/AndrewImwalle/fighting-aggies-platform
Heroku**:** http://fighting-aggies-platform.herokuapp.com/

Summary

The main client is Kratik Malholtra who is Texas A&M University's Director of Sports Science and Analytics (DSSA). Currently, Kratic needs to be able to upload, analyze, visualize, and predict player statistics from practices using the Texas A&M University's new computer vision system. Furthermore, this platform should be efficient and easy to use so that the Sports Science and Analytics department can work at full capacity. This platform should give our client the ability to upload videos which are then sent to the computer vision system (on the cloud), have the videos persist for further analysis, and receive the corresponding analytics and statistics from the computer vision system. Most importantly, this platform should provide Kratic and Texas A&M University's football coaches with the ability to produce report cards for specific players, routes, and plays.

Our platform meets these needs by providing a role specific platform that allows data managers to upload videos, coaches to analyze and develop report cards from data generated from the computer vision system, and administrators to manage all users. Furthermore, this platform is fast, easy to use, accurate (once implemented with the computer vision system), and visually appealing. The primary stakeholders of our platform are the sports analysts, football coaches, and data (video) managers at Texas A&M University. Our platform has been developed for their specific use cases and with their technical adeptness in mind.

User Stories

Iteration 1

- Feature: Administrator Dashboard (6 Points)
    - As a administrator
    - So that I may manage those who have access to the website
    - I want to have an administrator page where I can manage roles and access
- Feature: Google OAuth (3 Points)
    - As a user
    - So that I may access the page securely without passwords
    - I want to be able to sign in through Google

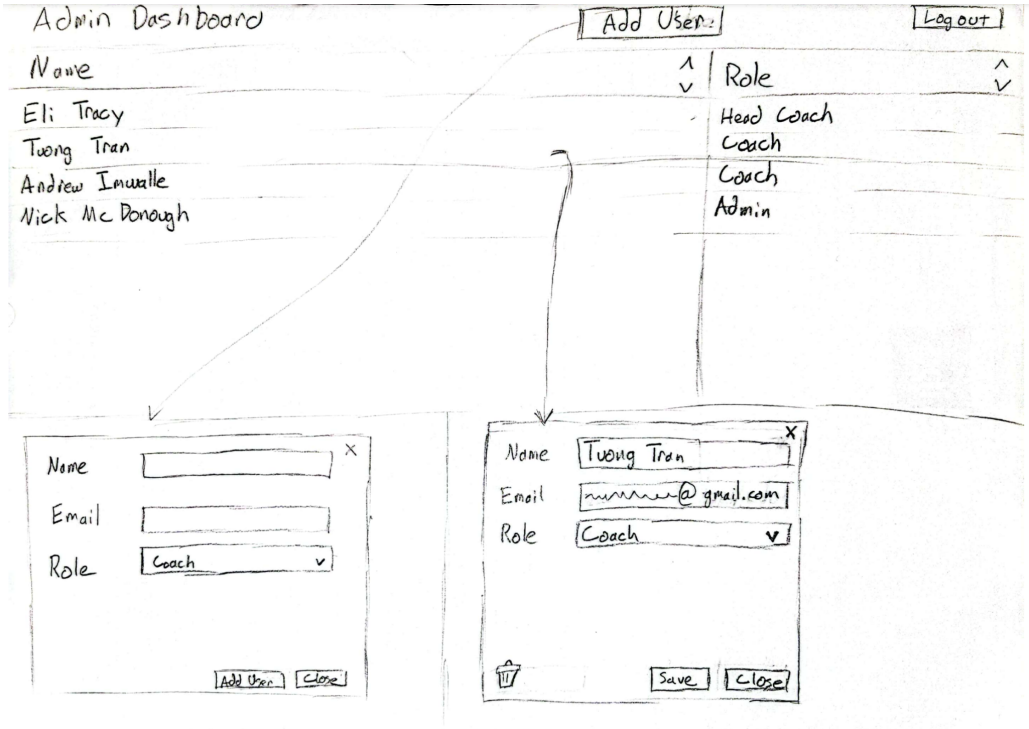
These stories have been successfully implemented.

Google OAuth was given the point value of 3 points because even though this was a straightforward task, the complexity of dealing with Google and the Authenticator gem was high.

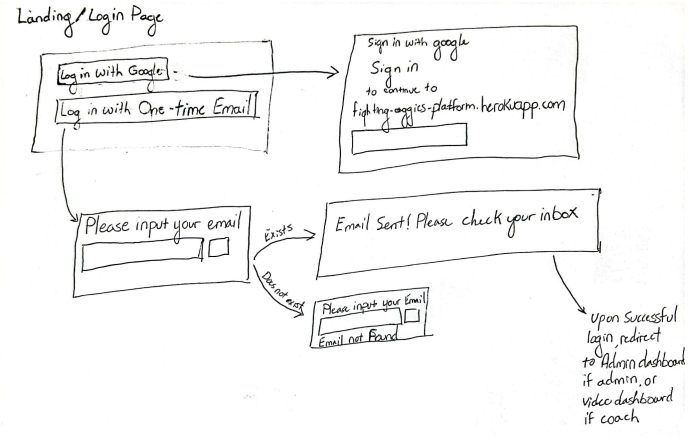
The Administrator dashboard was given the point value of 6 points because there were multiple steps to ensure that this story could be completed to its full potential. This dashboard adapted a few times as the client requested updates to the UI.

Lo-Fi UIs:

Administrator Dashboard:

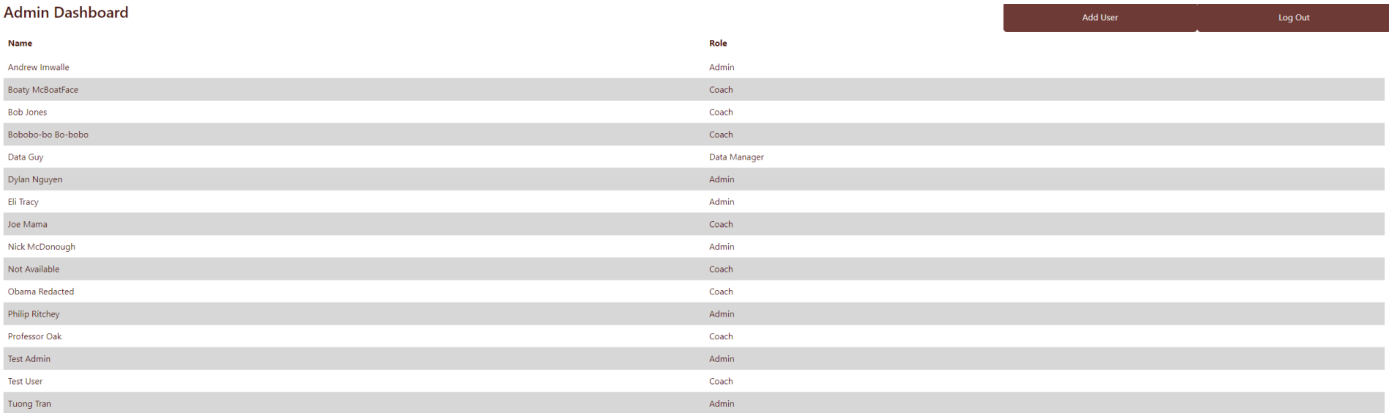


Google OAuth:

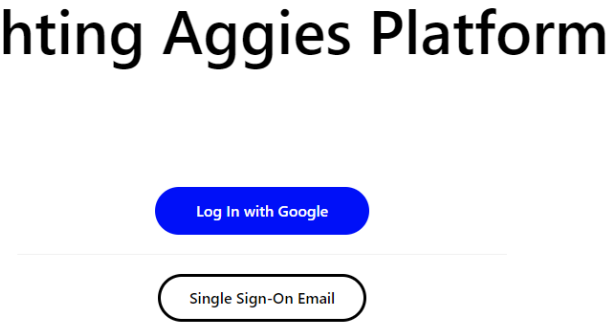


Current Version:

Administrator Dashboard:



Google OAuth:



Iteration 2

- Feature: One-time email link (3 Points)
    - As a user
    - So that I may access the page without a Google login
    - I want to be able to log in through a one time password sent to my email
- Feature: Restriction of access (2 Points)
    - As an administrator
    - So that users have access to the pages they are allowed to see
    - I want to restrict access to the pages, and redirect the user back to their homepage

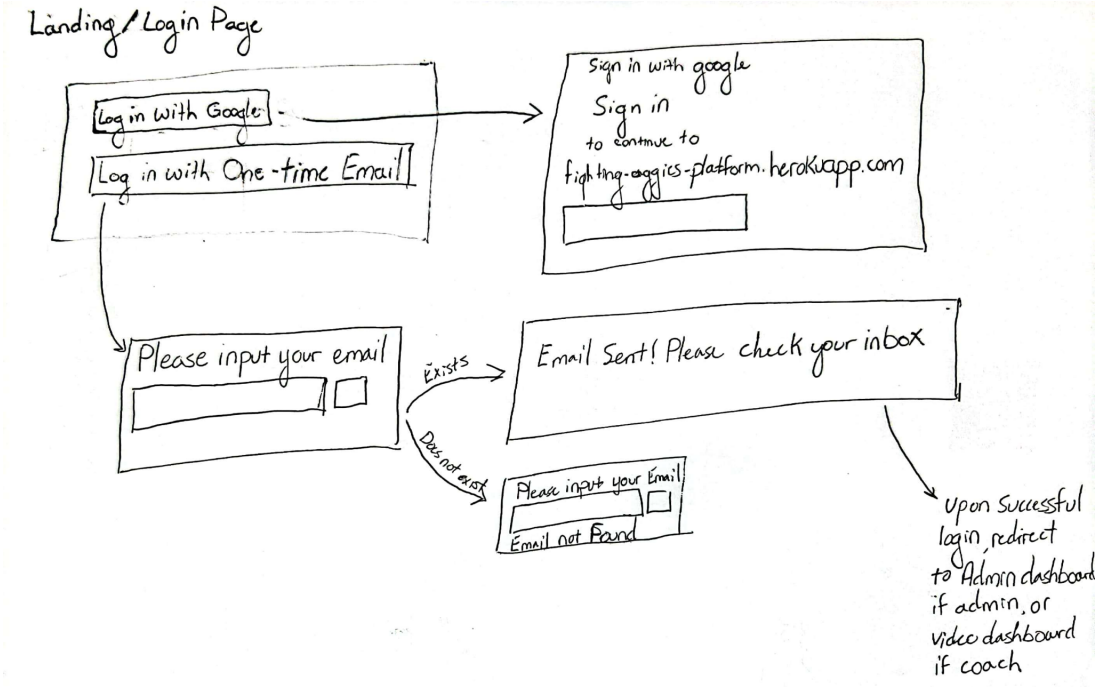
These stories have been successfully implemented.

The One-time email link was given the point value of 3 points because even though this was a straightforward task, the complexity of dealing with an email service and the passwordless gem was high.

The restriction of access was given the point value of 2 because the team had to refactor code to ensure that the access to each page was restrictive. Additionally, due to the frustration that the login process can cause, the team had to ensure that the user would be redirected back to their original page if they try to access a page they do not have access to.

Lo-Fi UIs:

One Time Email Link:



Current Version:

One Time Email Link:

Iteration 3

- Feature: Video Dashboard (5 Points)
    - As a Data Manager
    - So that I may manage videos
    - I want to have a video dashboard
- Feature: Analytics Dashboard (3 Points)
    - As a coach
    - So that I may view my players performance on practices
    - I want a dashboard that displays all of the data on my practices

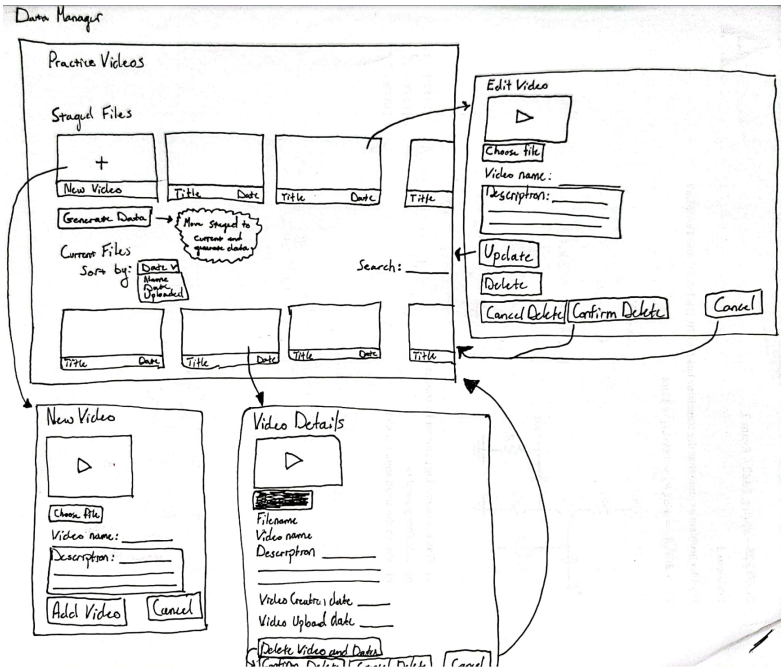
These stories have been successfully implemented.

The video dashboard was given the point value of 5 points because there were multiple steps to ensure that this story could be completed to its full potential. This dashboard adapted a few times as the client requested updates to the UI.

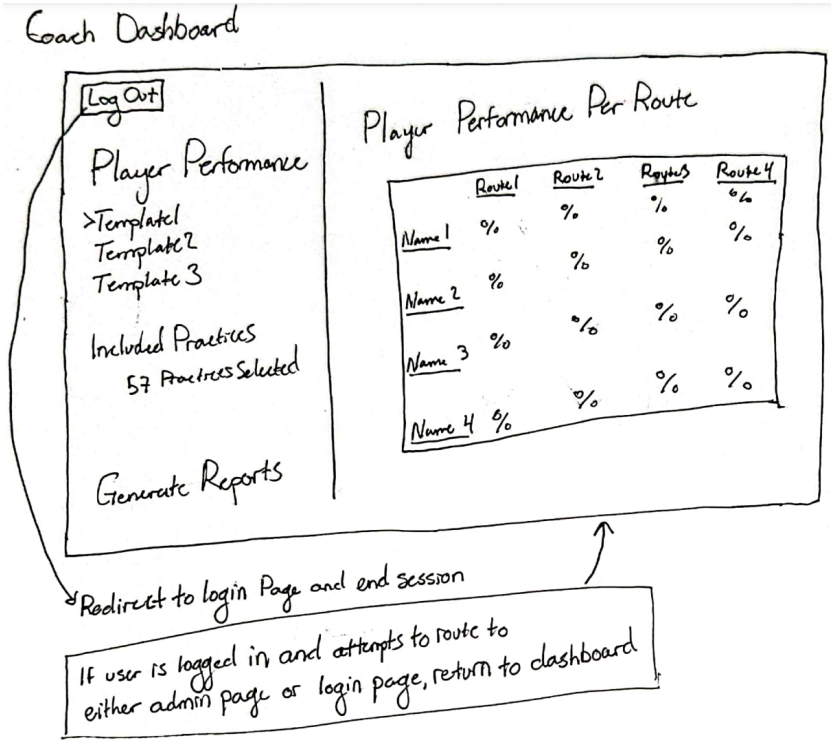
The analytics dashboard was given the point value of 3 points because there was an existing dashboard that had to be refactored. This dashboard adapted a few times as the client requested updates to the UI. This dashboard would be the main focus of the project, but during iteration 3, it was only worth 3 points.

Lo-Fi UIs:

Video Dashboard:

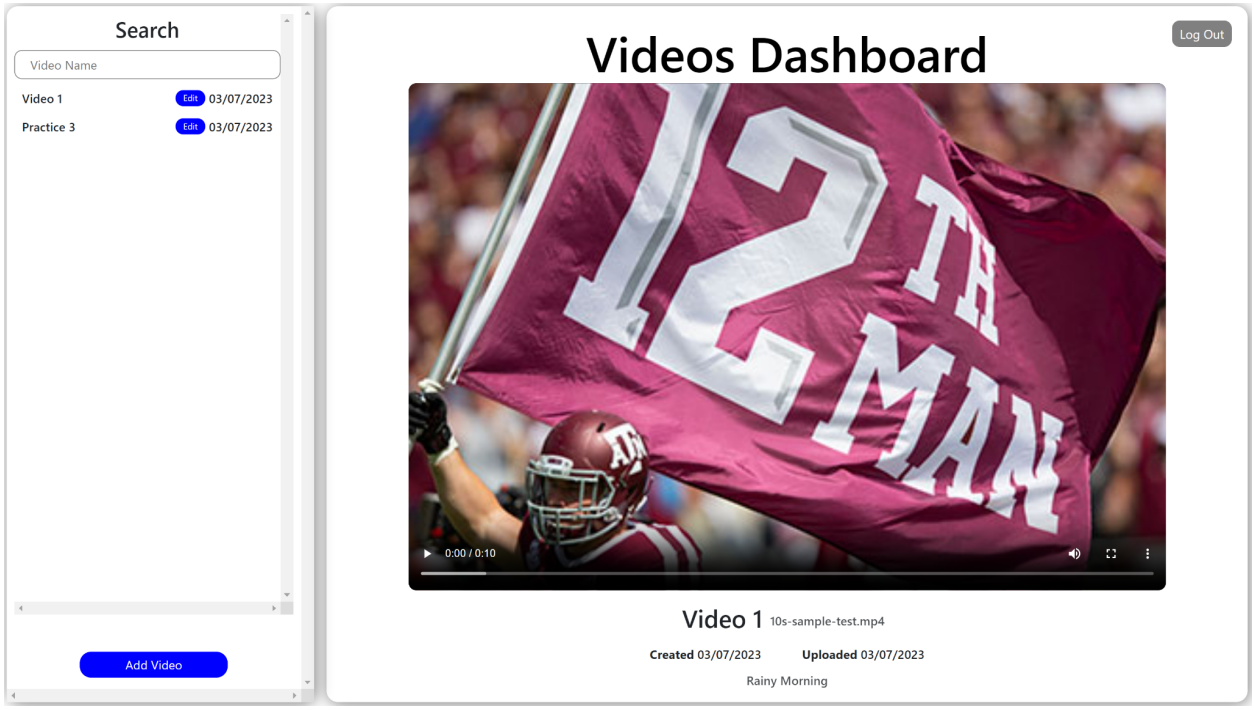


Analytics Dashboard:

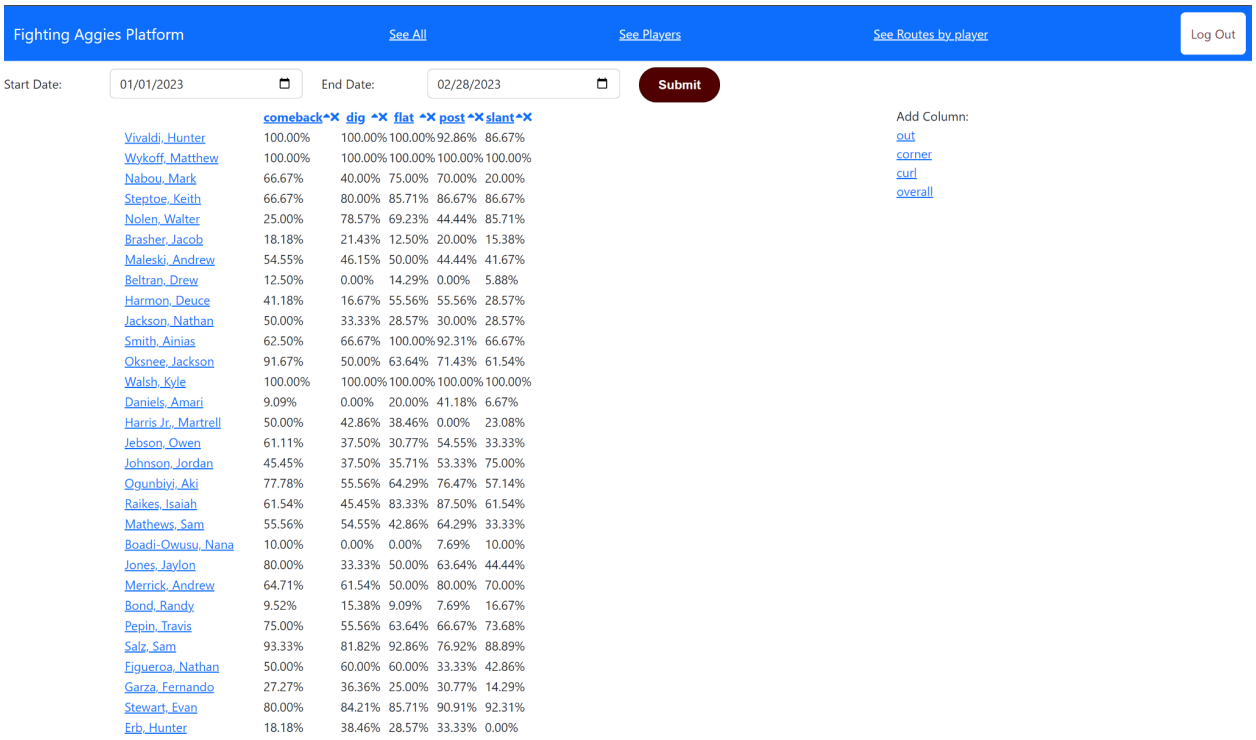


Current Version:

Video Dashboard:



Analytics Dashboard:



	comeback	dig	flat	post	slant
Vivaldi, Hunter	100.00%	100.00%	100.00%	92.86%	86.67%
Wykoff, Matthew	100.00%	100.00%	100.00%	100.00%	100.00%
Nabou, Mark	66.67%	40.00%	75.00%	70.00%	20.00%
Steptoe, Keith	66.67%	80.00%	85.71%	86.67%	86.67%
Nolen, Walter	25.00%	78.57%	69.23%	44.44%	85.71%
Brasher, Jacob	18.18%	21.43%	12.50%	20.00%	15.38%
Maleski, Andrew	54.55%	46.15%	50.00%	44.44%	41.67%
Beltran, Drew	12.50%	0.00%	14.29%	0.00%	5.88%
Harmon, Deuce	41.18%	16.67%	55.56%	55.56%	28.57%
Jackson, Nathan	50.00%	33.33%	28.57%	30.00%	28.57%
Smith, Ainias	62.50%	66.67%	100.00%	92.31%	66.67%
Oksnee, Jackson	91.67%	50.00%	63.64%	71.43%	61.54%
Walsh, Kyle	100.00%	100.00%	100.00%	100.00%	100.00%
Daniels, Amari	9.09%	0.00%	20.00%	41.18%	6.67%
Harris Jr., Martrell	50.00%	42.86%	38.46%	0.00%	23.08%
Jebson, Owen	61.11%	37.50%	30.77%	54.55%	33.33%
Johnson, Jordan	45.45%	37.50%	35.71%	53.33%	75.00%
Ogunbiyi, Aki	77.78%	55.56%	64.29%	76.47%	57.14%
Raikes, Isaiah	61.54%	45.45%	83.33%	87.50%	61.54%
Mathews, Sam	55.56%	54.55%	42.86%	64.29%	33.33%
Boadi-Owusu, Nana	10.00%	0.00%	0.00%	7.69%	10.00%
Jones, Jaylon	80.00%	33.33%	50.00%	63.64%	44.44%
Merrick, Andrew	64.71%	61.54%	50.00%	80.00%	70.00%
Bond, Randy	9.52%	15.38%	9.09%	7.69%	16.67%
Pepin, Travis	75.00%	55.56%	63.64%	66.67%	73.68%
Salz, Sam	93.33%	81.82%	92.86%	76.92%	88.89%
Figueroa, Nathan	50.00%	60.00%	60.00%	33.33%	42.86%
Garza, Fernando	27.27%	36.36%	25.00%	30.77%	14.29%
Stewart, Evan	80.00%	84.21%	85.71%	90.91%	92.31%
Erb, Hunter	18.18%	38.46%	28.57%	33.33%	0.00%

Iteration 4

- Feature: Practice Data Creation (2 Points)
    - As a Coach
    - So that I have data to review
    - I want uploading a video to create practice data
- Feature: Player Report Card (2 Points)
    - As a Coach
    - So that I can see my player's progress through practices easily
    - I want to manipulate the data and display a player report card
- Feature: Sorting Tables (3 Points)
    - As a user
    - So that the data is organized
    - I want to be able to sort data by categories

These stories have been successfully implemented.

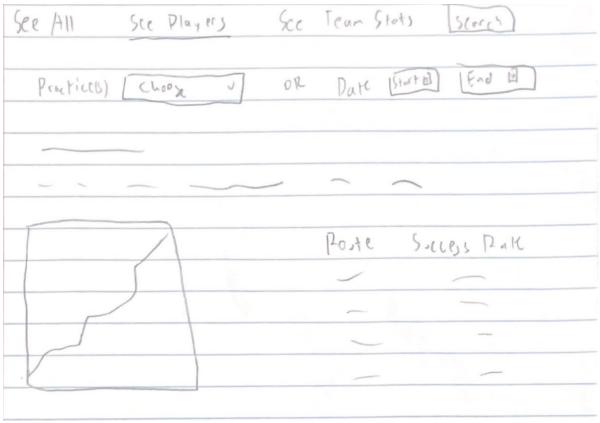
The practice data creation story was given the value of 2 points because the team had to create sample data and only access it when there would be a new video upload. Additionally the uploaded data had to be tied back to the video that was uploaded.

The player report card was given the value of 2 points because this was a straightforward task that the initial creation of would be simple. This feature would become more complex in iteration 5, but would only be worth 2 points during iteration 4.

The sorting of tables would be given the value of 3 points. This is because there are many tables throughout the application, and having the ability to sort each of them is beneficial.

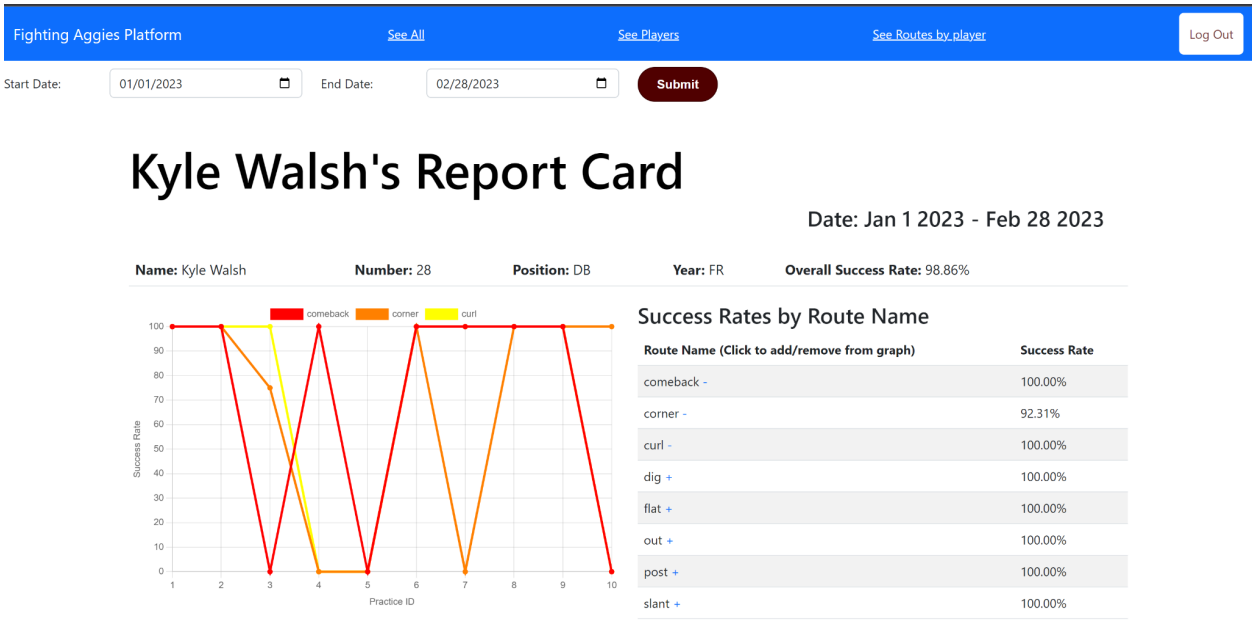
Lo-Fi UIs:

Player Report Cards:



Current Version:

Player Report Cards:



Iteration 5

- Feature: Automatically Populate Current Date (1 Points)
    - As a data manager
    - So that I may quickly input data
    - I want the video create date to automatically populate with the current date
- Feature: Restrict File Duplication (1 Points)
    - As a data manager
    - So that I do not upload duplicate data
    - I want to restrict uploads to require different files
- Feature: Metadata from Uploaded Videos (3 Points)
    - As a data manager
    - To speed up the upload process of new data
    - I want to pull the video metadata and automatically populate fields with the data
- Feature: Filter Practice Data(16)
    - As a coach
    - So that I can view specific data
    - I want to be able to filter practice Data

These stories have been successfully implemented, except for the Metadata from Uploaded Videos.

The Automatically Populate Current Date was given 1 point as this was a simple straightforward task requested by the client

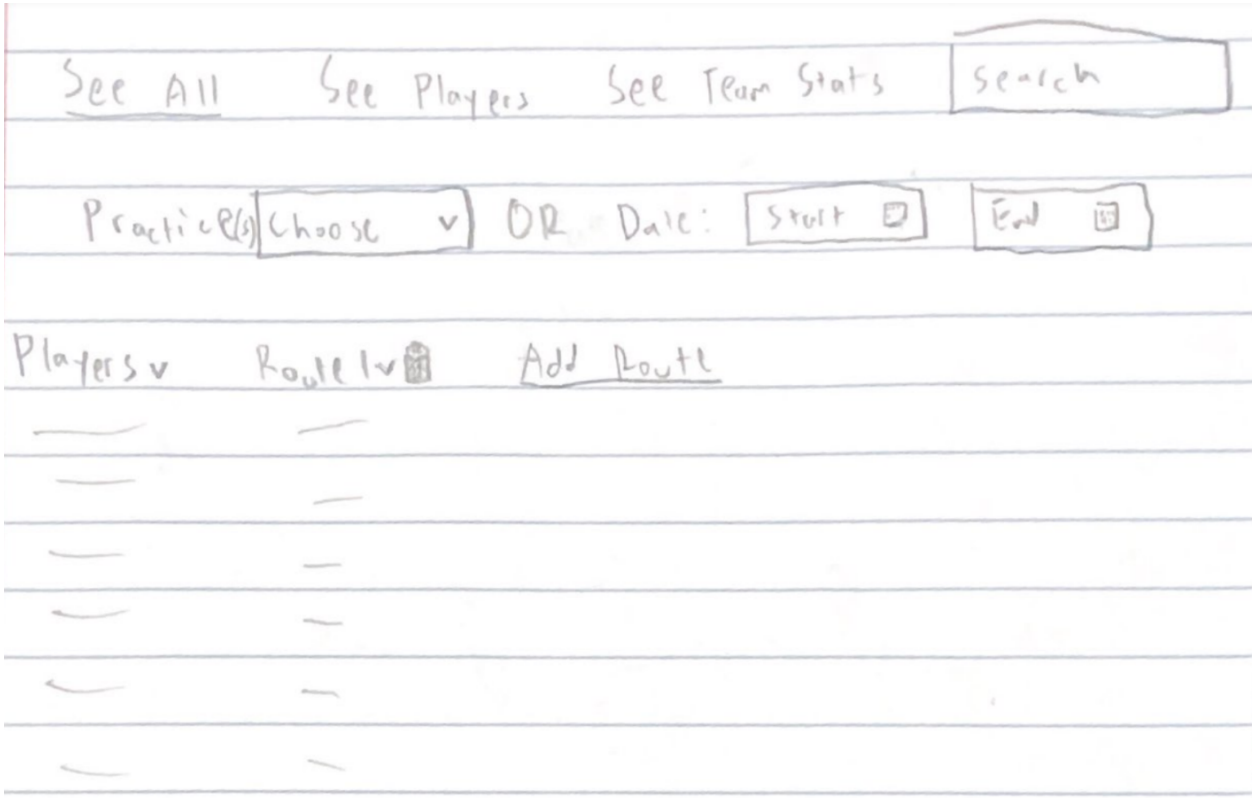
The Restrict File Duplication was also given 1 point as this was a simple task requested by the client.

The Metadata from Uploaded Videos is given 3 points as this task has been attempted multiple times with no success. This is difficult to do due to privacy issues given videos along with different formatting issues. Therefore this story was incomplete.

Filtering Practice Data was given 16 points due to the complexity and the importance of this code. This is the core functionality of the application. The client wants to be able to filter by player, routes, plays, and practices, create a report card, add and remove routes, and edit the report card.

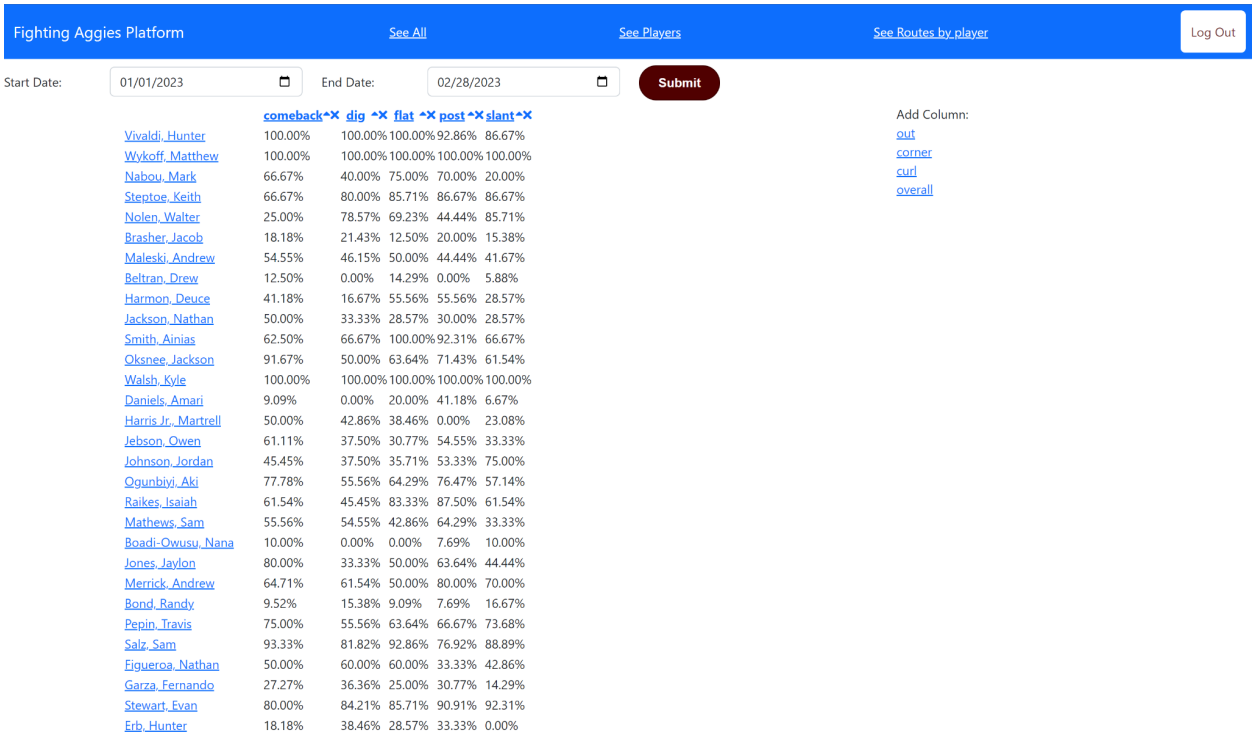
Lo-Fi UIs:

Filter Practice Data:



Current Version:

Filter Practice Data:



	comeback	dig	flat	post	slant
Vivaldi, Hunter	100.00%	100.00%	100.00%	92.86%	86.67%
Wykoff, Matthew	100.00%	100.00%	100.00%	100.00%	100.00%
Nabou, Mark	66.67%	40.00%	75.00%	70.00%	20.00%
Steptoe, Keith	66.67%	80.00%	85.71%	86.67%	86.67%
Nolen, Walter	25.00%	78.57%	69.23%	44.44%	85.71%
Brasher, Jacob	18.18%	21.43%	12.50%	20.00%	15.38%
Maleski, Andrew	54.55%	46.15%	50.00%	44.44%	41.67%
Beltran, Drew	12.50%	0.00%	14.29%	0.00%	5.88%
Harmon, Deuce	41.18%	16.67%	55.56%	55.56%	28.57%
Jackson, Nathan	50.00%	33.33%	28.57%	30.00%	28.57%
Smith, Ainias	62.50%	66.67%	100.00%	92.31%	66.67%
Oksnee, Jackson	91.67%	50.00%	63.64%	71.43%	61.54%
Walsh, Kyle	100.00%	100.00%	100.00%	100.00%	100.00%
Daniels, Amari	9.09%	0.00%	20.00%	41.18%	6.67%
Harris Jr., Martrell	50.00%	42.86%	38.46%	0.00%	23.08%
Jebson, Owen	61.11%	37.50%	30.77%	54.55%	33.33%
Johnson, Jordan	45.45%	37.50%	35.71%	53.33%	75.00%
Ogunbiyi, Aki	77.78%	55.56%	64.29%	76.47%	57.14%
Raikes, Isaiah	61.54%	45.45%	83.33%	87.50%	61.54%
Mathews, Sam	55.56%	54.55%	42.86%	64.29%	33.33%
Boadi-Owusu, Nana	10.00%	0.00%	0.00%	7.69%	10.00%
Jones, Jaylon	80.00%	33.33%	50.00%	63.64%	44.44%
Merrick, Andrew	64.71%	61.54%	50.00%	80.00%	70.00%
Bond, Randy	9.52%	15.38%	9.09%	7.69%	16.67%
Pepin, Travis	75.00%	55.56%	63.64%	66.67%	73.68%
Salz, Sam	93.33%	81.82%	92.86%	76.92%	88.89%
Figueroa, Nathan	50.00%	60.00%	60.00%	33.33%	42.86%
Garza, Fernando	27.27%	36.36%	25.00%	30.77%	14.29%
Stewart, Evan	80.00%	84.21%	85.71%	90.91%	92.31%
Erb, Hunter	18.18%	38.46%	28.57%	33.33%	0.00%

Legacy Code

Upon receiving the project, we spent a couple of days looking over the previous team's code and working out how to install, run, and deploy their project. The code was manipulated using the existing test cases to learn how the code operated. Upon the client's request, we removed the React frontend portion of the project and replaced it with a Ruby on Rails application. In the project we received from the previous team, authentication was conducted using Firebase, which we refactored to use Google Authentication as well as a Backup Single Sign On Email. We also modified the previous team's main site into two separate parts: a Coach Dashboard for creating report cards and looking at different data, and a Video Management system where an authorized user could upload new videos and edit video metadata. Additionally, we have created an administrator dashboard so that an administrator can control who has access.

Project Roles

Iteration	Product Owner	Scrum Master	Developers
Iteration 1	Andrew Imwalle	Nick McDonough	Elias Tracy Tuong Tran
Iteration 2	Elias Tracy	Tuong Tran	Andrew Imwalle Nick McDonough

Iteration 3	Nick McDonough	Andrew Imwalle	Elias Tracy Tuong Tran
Iteration 4	Tuong Tran	Elias Tracy	Andrew Imwalle Nick McDonough
Iteration 5	Elias Tracy	Tuong Tran	Andrew Imwalle Nick McDonough

Iteration Summaries

- Iteration 0 (0 points) - For this iteration the team focused on getting up to speed on the legacy issues of this project, making sure legacy test cases pass, and setting up the development environment for this project. All of these tasks were completed by the end of the iteration.

- Iteration 1 (7 points) - For this iteration the team focused on getting administrator functionality working. During this iteration the team also fixed all legacy testing issues, developed an administrator dashboard to manage users, added CRUD functionality for the users database, and added the necessary test cases. By the end of the iteration all CRUD operations were implemented and legacy video uploads were functional in production.

- Iteration 2 (15 points) - For this iteration the team focused on developing secure user login functionality through Google OAuth, and SSO with email as well as implementing user CRUD operations with the administrator dashboard. By the end of the iteration all login functionality was complete, all user functionality was accessible through the administrator dashboard, and all corresponding test cases were implemented.

- Iteration 3 (13 points) - For this iteration the group focused heavily on the video upload functionality of the platform. We found that the legacy video upload functionality was not sufficient for our client's needs and therefore needed to rebuild this functionality from scratch. During this iteration our team established basic CRUD functionality for videos, persistent storage of videos with Amazon S3 buckets, developed the video dashboard page, and the CI/CD pipeline was developed for easier deployment and integration.

- Iteration 4 (11 points)- For this iteration our team focused on fixing bugs with video uploads, linking video CRUD operations to player data, and began the data exploration portion of our platform. By the end of the iteration users could

successfully add and remove videos that were linked to player data, dig into and sort player data.

- Iteration 5 (19 points) - For this iteration our team focused heavily on cleaning up bugs which were prevalent in existing functionality on our platform as well as finalizing report card functionality and video upload functionality. By the end of this iteration, users could upload and search videos, sort and sift through player data, and produce specialized report cards.

Customer Meetings

- 1/10/2023 Iteration 1
    - The team demoed the administrator page as well as the Google Login.
    - The client described their desires for the UI updates for the Administrator Dashboard
- 1/13/2023 Iteration 2
    - The team demoed the updated UI for the Administrator page and the prototype of the Analytics Dashboard
    - The client described the desired functionality of the Analytics Dashboard
    - The client expressed the desire to feel as if the view is all on one page
    - The client expressed the need for a Data Manager role
- 1/30/2023 Iteration 3
    - The team demoed the Analytics Dashboard update
    - The client expressed that the data shown should be the summary data until drilling down into lower levels
    - There should not be a set page that is loaded for the Analytics Dashboard
    - The client discussed the ideal view for the Data Manager dashboard
    - The client expressed that the main functionality is the report cards
- 2/24/2023 Iteration 4
    - The team demoed the Data Manager Dashboard, and the client likes the view with a few critiques regarding the details about uploading a video in order to streamline the process
    - The team demoed the new Data Manager Dashboard and the player report cards
    - The client expressed that the report card needs to be customizable and data needs to be added and removed.
    - The client also proposed the search functionality on the navigation bar
- 3/10/2023 Iteration 5
    - The team demoed the updates to the Data Manager Dashboard and the client approved

- The team demoed the updated Analytics Dashboard and the client had a few notes on some UI changes that they would like to see before the completion of the project

Test Driven Development

For this project we used the test driven development process to guide our group's progress. We found this process a little tedious at first but as we continued to develop our platform found it to be a helpful way to manage progress and functionality. For the TDD process we would start by discussing what functionality we wanted to have done by the end of the iteration, develop the corresponding user stories, and then create test cases which correlated directly to these user stories. This direct correlation let us know which features we still needed to develop, were stuck on, or that might possibly have bugs. The TDD process was especially helpful as we integrated and deployed our code because all of our features were verified by test cases. TDD allowed our group to develop a CI/CD pipeline which would push our code to production given it was properly formatted and that all test cases passed. So while TDD slowed our progress at first it later gave way to a far more efficient and automated production process.

Configuration Management

We utilized Git as our version control system for project configuration management. We recognized the significance of creating separate branches for each functionality added to the project, which allowed us to avoid potential merge conflicts that might have occurred if we had not used Git. We employed several branches for various project components, and once we completed work on each component, we merged and deleted the relevant branch to ensure a well-organized and uncluttered working directory. There were a few spikes that occurred after meeting with the client such that the team could ensure they had a solid idea of how they would like to approach the project. During the project there were at least 17 branches and five releases.

Deployment Issues

During the beginning of the project, several members of the team had issues with Heroku, so there was one website that represented the deployed version of the project. Early in the third iteration, a member of the team set up a CI/CD pipeline that would make several verifications of the main branch on github before automatically deploying to Heroku. This pipeline would check that all tests that are created pass, that the gemfile is up to date, and that the coding standards set by Rubocop were followed before automatically deploying. After this was implemented, there were very few issues with Heroku, as all code was thoroughly tested.

Issues with tools

During our development process, we did not encounter any issues with GitHub, but there were a few issues with AWS Lightsail where the size and implementation of the code would cause the instance to crash and force a reboot of the instance. After replacing the legacy code and significantly reducing the size of the application file, these issues were reduced.

In order to store the videos, we used Amazon's S3 bucket. Implementing this was difficult because there are secrets that must be shared with Heroku in order to successfully utilize this service. The steps in order to set this up for a new website are in the README file.

To facilitate the login process, Google OAuth was used. This setup for this should be easy as the team simply has to register the website for google along with the proper redirection.

Other tools

The Passwordless gem was used to facilitate the one time password link creation. This allows the user to login without a password through an email that they receive.

SimpleCov is a gem that was used to ensure the team was testing as much of the code as possible. This means that any code that exists and passes the tests is less likely to have a bug.

Rubocop is a Ruby code style checker (linter) and formatter based on the community-driven Ruby Style Guide. This ensures that the team has a consistent style when submitting code to GitHub

Rubycritic analyzes the codebase and evaluates various metrics. RubyCritic provides the team with insights into their code quality and helps them identify areas that need improvement. By using RubyCritic, the team can easily spot potential issues, optimize their codebase, and ensure that their code is efficient and maintainable.

Repository Contents

In order to ensure that setup would be as easy as possible for the following teams, a setup script was developed and updated every iteration to ensure that all a new user would have to do is run the setup script to install the dependencies related to the code. In order to do so, the user must run the following command:

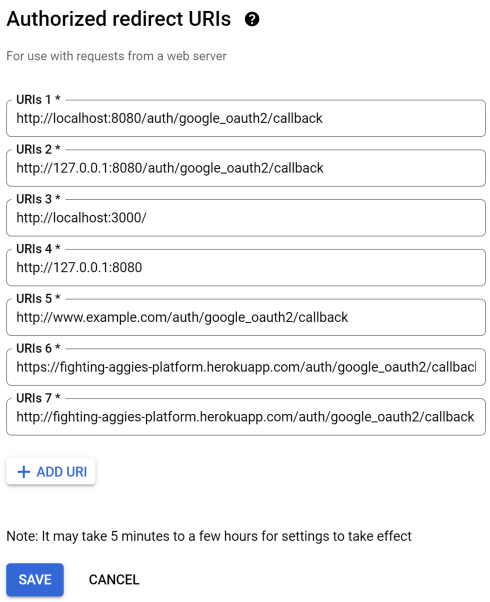
```
./setup.sh
```

This will set up the development environment. In order to run the code on a local machine the user must run the following command:

`bin/dev`

This will create a local host. When using the single sign on email link instead of Google OAuth, input a valid email address and after submitting, return to the console where there will be a link for the developer to click. This allows the user to access all parts of the code

For Heroku Deployment, follow the steps on the README. These steps include migrating and seeding the database and installing ffmpeg on Heroku. In order to successfully access the pages using Google OAuth, visit and login to https://console.cloud.google.com/. Next, type credentials in the search bar and click credentials. Then click create credentials and create OAuth client id credentials. Add the redirect uris listed in the image below. Then modify the .env file to include your Google Client ID and Google Client Secret Id on the credentials page. More details on this process are in the README. Finally, create an Amazon S3 Bucket to store the videos. This is an in depth process that requires multiple steps that can be found on the README.



Finally, if there is an issue with the one time link email, the team may have to create a new email address. If the team does have to edit this, the team must ensure that there is an App Password and to replace the current one. Additionally, ensure that this gmail account has 2-step verification enabled to avoid Google blocking the emails.