Fantasy Baseball

Liam Bowen
bowenl@oxy.edu
Occidental College

1 Problem Context

I've been playing fantasy sports for most of my life, and baseball is the only adaptation of the game that my friends and I struggle to retain focus. When I play fantasy baseball with my friends, we run into the same problem every year. Some people end up drafting a team that performs poorly, and they quit because they are upset with not winning. This happens early in the season, and it leaves the group down a player. By the time we make it halfway through the season, less than half of us are still participating. The MLB season is 162 games, and once we reach the end, there are only two people still active.

A massive issue that exists is that there is a skill curve or more so an "educated" curve to the game as I like to put it. What I mean by this is that winning is solely based on stats and there are certain stats that have a much higher value than others. The metric that impacts scoring the most is wins. It makes sense that the best pitchers generate the most wins, but that is not always the case. How many runs that pitcher's team scores on the day he pitches plays a key role in determining if said pitcher comes away with a win, and five points for the fantasy team he is on. An article from 2017 points out a great example of how the pitcher's team can impact that pitcher's win total. [**Joyce**]. In 2010, Felix Hernandez, who was voted as the best pitcher in baseball that season, and James Shields each recorded the same amount of wins despite Hernandez allowing 54 less earned runs and 56 less hits.

For the most intense fantasy baseball players, it's obvious that Felix Hernandez is not as valuable as his individual numbers show because his team is less likely to do their part to get the win. However, the average player does not notice that. They draft a player like Hernandez, and even though he performs great, his fantasy score is not as high as other names. Wins are just one problematic metric. We see great hitters that do not score great in fantasy because their team is not good, leaving them with very few opportunities to get runs batted in or runs scored.

Fantasy football is year in and year out, way more popular than fantasy baseball. Approximately 60 million people participate in fantasy sports, and football is by the most popular version [**Kravitz**]. About 78 percent of the demographic played fantasy football. Then, there is a massive drop-off to fantasy baseball, which is only 39 percent of the population.

As someone who plays both versions, I find it is way easier to enjoy fantasy football. Fantasy football is way simpler. Roster sizes are much smaller, the season is way shorter, and stats are less influenced by the results of the player's teammates.

My goal was to create a web app that improves the overall fantasy baseball experience. I wanted to show users that there are different ways to visualize and understand the data and construct a team. The app is meant to give users a better idea of how fantasy baseball works, so that they put together a better team when it comes to competing with their friends.

2 Technical Background

For those who are not familiar, fantasy sports is a game where users draft a team of players and compete against one another. Users compete head to head week by week with one another and the team with the best record by the end of the season wins it all. A player's real life performance impacts the scoring, so drafting the best players is important. Each statistical metric has a point value attached to it, and that impact how many points a user gets in a week. Fantasy sports is typically a fun way for users to connect more with friends while discovering new players to watch and root for.

For my project itself, I started with constructing a data set that would be dynamic where users could interact with it and modify it. I exported two csv files from Baseball Reference to create my data set [**Baseball Reference**]. Since batters and pitchers have varying metrics that they get measured on, I needed to make sure that the two data sets are separate. I wrote code to convert the raw csv files into a database in MySQL.

I built my project in python, using a web framework called Flask [**Flask**]. The flask framework is able to interact with the data bases, and displays viable information for users. The flask framework can takes html templates and render them on a browser. Flask was important for writing functions to help users register, log in, and log out. I wrote a function to create another data base that included user information. The flask routes were able to add new users to the data base and search through the existing users to find an account.

Each page of the application has an html file that has all its elements defined. Each html file has a css file that is used to style that page. I wrote the css to style each element. Most pages also include a javascript file. The pages all call for the javascript file with the same name for the functionality of pages. The javascript and css help to shape how the html looks and works. The html files gets called from the flask framework and helps make the project a visible and functional site. For help with formatting and writing html, css, and javascript, I turned to W3 Schools page. All of the properties for styling and proper variables to use in the html can be found [**W3schools**]. This site and its resources can be used to develop an html project in any style.

3 Prior Work

ESPN is the most obvious prior type of work similar to what I worked on [**ESPN**]. Not only is it the most popular site for fantasy sports, but it is the most popular with sports in general. ESPN does a great job with fantasy football, but fantasy baseball is lacking. This is not to the fault of ESPN, it is difficult to promote fantasy baseball when baseball itself has much longer seasons and more complicated fantasy rosters. ESPN however could make changes to generate more users, but it has not done so.

Another fantasy sports site tries to add in a lot of their own algorithms to analyze trades and drafts, but this leads to the user listening to the computer to make their decisions [**FantasyPros**]. If the computer gives the go ahead on a player, the user will follow along almost blindly. A site like this is great, but how much are users really learning. There is no improvement in the skill of a user even if their team gets better. When the computer tells them to do something and that turns out to be a bad decision, users will get frustrated and quit.

These sites either try and do too little or too much to change fantasy baseball. In both ways this does not succeed because they both do not promote new ways for users to enjoy the game. ESPN makes the user do everything themselves and does not have systems in place to enhance how a user analyzes players. People using ESPN are left with fending for themselves and there is not enough information given to them beforehand to draft a successful team. Fantasy Pros goes the other way and gives the user too much to use. The users are not able to do anything without computed analysis being put in front of them. All their decisions are going to be based on what the algorithm says and there is no development of skill in playing. The user is not running their own team at that point because every decision is made by the system.

4 Methods

My first idea that I started running with was a full fantasy baseball website that had its own unique tricks that made it stand out from other versions of fantasy sports. However, this was not reasonable to pursue for a couple reasons. First off, I was going to make my site overly complicated, and in the process confuse the users. I wanted to take inspiration from some fantasy football sites because fantasy football consistently is more popular. I found several fantasy football sites not named ESPN that do a great job at keeping the roots of fantasy sports intact while involving their own ideas. Dynasty Football League is one such site [**Dynasty League**]. They promote the idea that there is not a redraft every season. Players get to keep their team the same season to season. Users can make changes as time passes, but the rosters do not get a full reset at the end of each season. The promotion of their version is based a lot on the idea that there is a stronger sense of team ownership and a better connection to the team.

I wanted to take inspiration from fantasy football sites like the Dynasty Football and incorporate them to baseball. I thought of a midseason redraft where users could protect a few of their players and then conduct a redrafting of the rest of the player base. I liked the idea of teams getting to retain players, but since a baseball season is so much longer, putting a redraft at about the halfway point would be able to add a new level of strategy. Users have to prioritize which players are the most important to their success and make sure that they were aware of who was going to be made available to improve their team the most. Another idea was to incorporate a steal a player feature where if a team with a worse record won against the team with the better record, they could pick one of the players from the other person's team to steal and add to their roster. In return they would send a player back of the same position, so that the stealing was not so lopsided. I chose not to pursue these because I found that this was not a reasonable product that people would play. This would create too much chaos and make the game almost unplayable. Also, the process of writing the code to create these aspects would be overly complicated and take away from my core goals. My goal was not about making a new version of fantasy baseball, but rather creating a site for users to enjoy and understand fantasy baseball more.

I considered making it a basic fantasy baseball app that was very much so similar to that of ESPN fantasy, but I would include a couple different statistics in my database to help add a new level to the basic level of the game. Users would draft their teams and go head to head like in normal fantasy baseball, but games were going to be one day at a time as opposed to a week long. This would help to reach my final goal of an app that helped users have a better experience in fantasy baseball because a team could climb the standings faster and stay involved. I chose to not go down this route because this did not differ enough from what already exists. I felt like this was going to create something that was already easily accessible and known for people.

So, that turned my attention towards creating an app that users can use to see not only see the data, but also select players to construct a team, and then compare that team with other users constructed teams. I wanted to allow the user to then have the ability to make edits to their constructed team by adding and dropping players from the roster. When I was researching other fantasy sites, I felt like creating a connection to one's team was a very important idea because it would get users more involved with their teams and promote more engagement. The dynasty style of football league was able to do that by making the user run a franchise instead of a year by year team. Finding a way to do that without using the idea verbatim was the challenge.

A valid concern is that if I was just displaying the data set, I would be presenting something that is not much different from a spreadsheet. Anyone can just create a spreadsheet and use it however they want. So I needed to make this more than what a spreadsheet can do. One way I set out to do that was to create registration features. A user can register and create their own team name and mess around with data. Another user can come along and is able to see who the first user has on their team and can compare the two teams side by side. Each page has its own function, so it does not feel like everything is cluttered. If I display too much at once, it can overwhelm people and make my project unusable.

Since I wanted to make users better at fantasy baseball or at least make more educated decisions when playing, I had to create a way for users to practice fantasy baseball. That brought up the question, how do people become better at something? The obvious answer is practice. Whether it be for school, sports, video games or overall anything, the old saying is "practice makes perfect". That meant I needed to make this into something that users could practice with. Tying back to my original goal, if I can make users practice drafting teams, they will become better at fantasy baseball and enjoy playing more. Less people will quit and friends will be having fun for the whole season.

I got inspiration from fantasy football mock draft sites for how to create something users could use to practice drafts. A mock draft is a way for users to practice going through a draft before the actual draft to get a better sense of their strategy in the draft [**DraftWizard**]. Users can do a variation of a mock draft without the pressure of fighting for players. They are on their own and can add a set of players to their roster to further analyze. I wanted to include this idea in my project because it gives users a great way to practice improving at drafting. Users are not actively drafting against other people, but rather constructing a roster on their own time, and another user can do the same at a different time. This promotes a less stressful environment for users to practice and learn about fantasy baseball and the players. This builds up a foundation that they can use and understand when they reach draft day.

I looked at trade analyzation sites as a way to involve a way for users to see if their draft was successful or not. A trade analyzer is a site where users can plug in players that they want to trade and who they want to trade for, and the site will return information on how good the trade will be for one side [**Rotorade**]. While having trades on my site was not important, the idea of comparing two sides was useful. This is where I came up with the idea to compare a users team to someone else's. Being able to take the team you built and look at it next to someone else's roster is a great way to see how good you constructed your own roster. This creates a good learning experience that can then be applied to making better decisions when drafting a team.

I wanted to include some numbers in the data set that would be unique to my project and could provide a deeper level of understanding of a player's ability to users. I looked towards Statcast, which is a tracking technology that has been collecting advanced data on players since 2015 [**Statcast**]. Some examples of data that are measured using Statcast includes exit velocity, pitch velocity, barrel percentage, and spin rate. I wanted to use some of these as a way to show users that a player has more value than home runs or strike outs might show. Ultimately, I chose to avoid including these metrics in my data base because these become too complicated to the average user without them conducting extra research. A stat like barrel percentage takes the amount of times that a hitter produced a perfect combination of exit velocity and launch angle. For someone who studies baseball and is a statistical nerd like myself, studying the barrel percentage of players is really exciting. For the average users that my project is intended for, barrel percentage would make no sense. They would either ignore the stat completely, or have to do extra research to understand the meaning. This goes against my initial goal because users that I want to help give a better understanding of how fantasy baseball works will not get any benefit out of these complicated advanced metrics. I would be asking to much out of the average user on my site.

Another reason I wanted to avoid evolving my data set was that these metrics do not have a fantasy point system in place. All the other statistical categories have a point value behind them. For example, a single is one point, a double is two points, a run batted in is a point, strikeouts are minus one point, etc. Barrel percentage does not have a point value assigned to it. I could convert the percentage back to total number of barrels and say that each barrel is a point. However, this causes a problem because each barrel does not necessarily mean that they got a hit. If the hitter got out, but the out was defined as a barrel, do they still warrant a point? That does not make sense, so I thought that maybe I would use only the barrels that also were hits, and award the player an extra point for the hit, but I get back to making things too complicated. This would create a larger skill gap because the best players are going to be the ones to take advantage of those extra points.

Something else that was important was dealing with how many roster spots each person could have. Normal fantasy sites allow for somewhere between five and eight bench spots besides the usual starters. They typically include several utility spots, meaning that any player can be set to that roster slot no matter the position they play. I decided that it was important to not include the utility positions and as many bench positions. I wanted to create a roster crunch that meant users had to really decide who the most important players were. It also means that there is no stashing players on the bench just for the purpose of preventing other users from having that player. It was important to still include some bench spots, so users are able to draft up to ten position players and six pitchers before being restricted. This is important for teaching users better roster control, so when they play a normal season and have a larger roster size, they already know who the most important players to target are. When the roster size is larger, there is less caution with selecting players. The thought process for someone with more roster spots available is, ”it's fine I can still fill the last couple spots with other players.” When I constrain the roster size, it causes users to learn more about who they should prioritize the most. This helps to teach more about how to effectively draft a team.

When a user comes to my site, they get prompted to create an account with a unique team name. From there, the user goes to the draft page where they can pick to look at pitchers or batters. They can sort based on any of the metrics displayed in both ascending and descending order. Users can mix and match with the metrics they want to in order to put together the team that they would like to. Once they've drafted a full roster, users can head their own teams page and look at how the full roster is constructed with the numbers. There is an option to remove players if they are not happy with how the roster is configured. The user can click on the comparison page and compare their team with teams that other users put together. This allows the user to look at the two sets of data side by side. At any time, they can go back to the draft room and add new players to their roster in order to create a team they are happy with.

With this layout, I was trying to create something dynamic that would allow a user to make better judgements in drafting for fantasy baseball. My goal was to create a better fantasy baseball experience. Creating a way for a user to compare a team they put together to a team that another user previously built is important to give them a way to visualize how their team would match up if the two teams were to go head to head. I did not want anything to be set in stone, so the ability to remove players and refill those roster spots was important. If users get stuck with the team they draft, they are not able to really learn anything. Giving them the opportunity to go back and construct a team again is important to help people create the best team they possibly can come draft day.

I wanted to make the draft room simple because that would help make the important data more visible to users. Normal drafts have a timer that puts users on a clock, but that is a stressful aspect that is not necessary for my goal. I wanted to give the users as much time as they would please, so they can really play with the data set and find players that they would never have seen if they were given 30 seconds. If I am forcing them to rush, I am not promoting an environment where users can properly visualize a good team. If they can not do that, then they will not be able to learn the right approach to take. Having no timer creates a space where the data can really be broken into and some hidden gems can be found.

Something important that I did that varied from most fantasy sites, was that I chose to hide the total points fantasy points a player would earn. While other sites include the points on the data set, most people will look only at that as the defining metric. I chose to not include the total points because it forces users to think more and understand how each metric impacts the results. If I made points visible, it could create a blindness to what metrics helped to produce those points. Users may want only the top points scorers on their teams, and would not take the time to properly analyze the stats on each player's resumé. This means the user would not be learning and understanding the data enough to make decisions. The point values will trick them into thinking they put together a great team, when in reality, they selected players that hit a lot of singles and did not hit a lot of home runs. A player of this caliber is fluky because they rely too much on one metric. It's important to find players that are well rounded and succeed in several categories because these are the players that are going to give the most consistent results.

5 Evaluation Metrics

My early evaluations of my project started with drawings on paper. I made colorful drawings that I would show to people to get a sense of if it looked like something they would use. I had several pages drawn up and handed them to people to look at. I wanted to see if it felt like something they could use without getting lost or confused. Each page had boxes that represented buttons to switch between pages or to pick players to their team. I asked questions during each test because I wanted to see how my project would flow and if the user enjoyed using what I put in front of them. I did this to work as a preview of what I was creating before creating it because if people hated it, I would not have wasted as much time writing the code to illustrate that. I have included some of the raw drawings in my repository to show what I started with.

I conducted a lot of user interviews to pick the brains of fantasy baseball players. I was originally focused on creating something that I wanted in my mind, but I needed to focus more on the needs of my target audience. In my interviews, I presented my ideas and asked for opinions. I wanted to see if people would use my site and enjoy it if I included ideas like the redraft or stealing a player. I asked for what kind of data set would be the most appealing to them and what I could put forth to make my project enjoyable for all fantasy baseball players.

I wanted to make sure my data set was an effective data set to make sure that users could use it to make strong analytical decisions. So a part of my early evaluations involved asking about the categories of the data set. I needed to make sure that. I wanted to use complicated metrics, but it was important to ask users what metrics they felt would be important for them to properly study and improve their fantasy baseball skills. I wanted to use metrics related to Statcast as I discussed earlier, and I used some interviews to discuss if those would be necessary in a data set. The only way I was going to achieve my goal was if I put forth a data set that all users wanted and used. I could not justify including advanced metrics if my interviews found that users did not want to look at those and could not benefit from studying those metrics.

I wanted to create something with live stats in season for the user, but since the major league season starts in April and ends in September, I could not test with users using live statistics. I considered running a simulation of the stats from the last season, but ultimately did not do this because some people could cheat the system because the season already happened and they would be able to pick the best players without using any intuition or skill. I decided to it was best to give users the end of season numbers because it would give them the best sense of who each player could be. Going through a week by week simulation of data would not give users a good sense of who would the best players would be because a player could have one really good week at the start of the year and then they struggle for the rest of the season. A simulation would highlight those misleading results to a user and create traps, so I did not want to run any tests using simulation data. My whole goal was to make the data as clear and easy for users to learn from and creating a simulation to run people through would trick them.

Once I reached a tangible project, I put it in front of people and had them do the same as they did with the drawings. I targeted people that have played fantasy baseball before and wanted to see if my site was able to teach them how to draft a better team. A major point I wanted to learn was if they felt like they would be more successful when they would compete against friends once baseball season came around. They had the ability to modify the data set to look how they wanted it to and were able to pick their roster and compare that with some sample teams I added to the system. Following each test, I asked users if they felt as if this was a good way to practice for other future fantasy drafts.

Since the season is not starting for months, I needed to come up with a way to get results that show that I was able to teach people how to form a better draft. I also needed to come up with a way to evaluate if they would then enjoy fantasy baseball more. I surveyed the people who tested what I did to see if they could learn from what I gave them. I made sure to see how much preparation each person would go through before each season. I also wanted to to know how successful they usually were in a fantasy baseball season. If they were not, I would ask if they would keep competing or if they would quit.

6 Results and Discussion

From my evaluations, I found that users were able to use my project to conduct analysis on the players. I watched as users studied the data. From there I saw that they were willing and able to alter the structure of their roster to try and make their team better. I saw some users repeat this process several times because they wanted to be perfectionists. From this I was able to gather that users were showing a willingness to learn. Most people I ran user testing have all quit a fantasy baseball league early in the season before. Some of them do so every year. They keep going back and signing up again hoping for a different result, but they always lose their first few matchups and find themselves quitting once again.

I saw them playing with the data and switching players in and out of their roster to try and put together the best team they could. For a lot of them I was giving them something to actually help them practice drafting a team. Since there is not going to be a season until April, it is difficult to see if someone is going to have more success once the season arrives, but I was able to create an environment that makes them feel more confident and prepared for the season. Most of the population I tested on had quit in a fantasy baseball season in years prior. Their reasoning was usually that their team sucked or were bored because their team sucked. From my questions, I found that users felt like they would have a better chance at competing.

It seemed as if most people lacked the proper preparation in order to get positive results in fantasy baseball. A general opinion was that it would not be worth it to put in an effort to prepare when fantasy baseball is very luck based. While that is true, there is a level of skill and predictability to how a season will go. The key to getting that skill and predicting the right players to succeed is through the data. I find that the previous year's data is a great indication of how a player is going to look in the future.

Some people I ran tests on really enjoy looking at data more than anything. They got to pick around in the draft room and look at how certain players rank in each statistical category. I watched as they looked at some of the players with the most home runs and then toggled by statistics such as strike outs and runs batted in. They did this to see if home runs might be an outlier in the player's ability because if they had low runs batted in they would not be as worth it to have on a roster because they are not able to generate as many points. If the player was striking out too many times, that would be more negative points that would outweigh the amount of home runs that happened.

The most important part of my overall goal was to create something that would enhance the user experience in fantasy baseball. From my results, I believe that I have successfully done so. Users can pick and choose from a set of the entire Major League player base. From my tests, I was able to learn that people had no idea who a lot of players were and would have glanced over their name and moved on in an actual draft. However, I created a site that allowed them to see those players for the first time. They were able to realize that some of the names they would have never considered actually have a lot of value and perform better than they would have guessed.

I can conclude that users were able to learn from the data I present them with. Now all they need to do is apply the new knowledge in their drafts in April. With a better draft on the horizon, they can finally be excited to play fantasy baseball. The competition level is more even as the worst players in the league can easily learn to compete with some of the best.

7 Ethical Considerations

An ethical issue with my app is the handling of player data. Users are prompted to register an account and then login each time they return. Each user has a password attached to their team, and I have to properly handle that information and the privacy of their information. Uploading my code to a Github repository also attaches the data of players. This is concerning because if people use a password that is similar to what they use on other sites, anyone can access that information on the repository and use that with malicious intent. For all my user testing, I prefaced that they should not use a password similar to any that they use on other apps. I have no way of encrypting the passwords, and that would be something for me to pursue if I am to continue with the project. There would be no way of me knowing that there would be a data breach as someone can just pluck information out of my repository [**Tupaz**].

It is worth mentioning that in picking the data set that I wanted to visualize to the users, I can be considered biased towards certain metrics. I wanted to show not only the core metrics used on every site, but a couple extra ones to better highlight some areas of a player's game. I had to pick from a wide array of metrics and narrow it down to just a couple that I felt would most help users. I was biased in my selection of these metrics, and other people may have went with different ones or maybe none at all. My selection process was opinionated, and so I created a site where some people might not be satisfied with the data set.

Another ethical consideration worth mentioning is inappropriate content. Users have the ability to pick their team name, but what can follow is a problem. Someone could take that as an opportunity to write a team name that is vulgar or offensive to others. If a user were to create a team name that offends others, that name gets displayed on the site for them to see. It can be considered my fault for allowing that team name to get put into the system because I do not have any functionality developed to prevent that from happening. Almost all sites and games have a system in place to prevent inappropriate usernames from getting used. One thing to pursue further for me would be creating banned words and phrases that are not able to get passed through when a user registers. It gets extremely difficult because people get really creative with how they create these names. I notice it every time I go online that someone has a username with some creative spelling of a curse word or slur. They use special characters to replace certain letters and sometimes add spaces between letters to help work around the algorithm that blocks that word. I found too many Reddit threads that discuss how funny inappropriate names are. On these threads people love to brainstorm with each other to come up with new funny ones that are gross and offensive. If someone wants to be rude and create an account with a gross name, they do not even need to think of one on their own. They can one to one of these many Reddit threads and pick the one that is their favorite. And since these names are so complicated, a simple algorithm is not suffice to prevent these names from being registered. The major companies have built in a feature for players to report other names when they see one that is inappropriate. When a name or post gets reported, it gets sent to their review team to determine if it is actually inappropriate. If they deem it violates ethical guidelines, they can ban that user from their application.

Since I am one person developing a small project, I am not able to have an ethical team working for me to review each reported name. If I install a report system, it falls on me to analyze the reported names. This can cause problems since it then comes down to only my opinion on the name. There are words and phrases that I have not seen before that could be highly offensive to certain cultures, but since I have never seen the word before, I would think nothing of it. This is ethically a problem because I need to educate myself more to know what terms to be paying attention to more. If a name contains a word that someone reports because it is inappropriate to them, and I do not find that word to be inappropriate, I could upset that person. There are some really obvious words that are inappropriate, but I either may not know what the words means or I could find that the word is really tame and harmless. Why would they want to use an app that allows for users to have words that they do not like included in the username? So I would need to find a balance and think about how each review would impact the population online.

References

[1] Joyce, Tom. “The Problem with Fantasy Baseball.” Splice Today, https://www.splicetoday.com/sports/the-problem-with-fantasy-baseball.

[2] Kravitz, Brandon. “Why Is Fantasy Football so Much More Popular than Fantasy Baseball?” FM 96.9 The Game, 29 Mar. 2021, https://969thegame.iheart.com/content/2021-03-29-why-is-fantasy-football-so-much-more-popular-than-fantasy-baseball/.

[3] “Welcome to Flask¶.” Welcome to Flask - Flask Documentation (2.2.x), https://flask.palletsprojects.com/en/2.2.x/.

[4] “W3Schools Free Online Web Tutorials.” W3Schools Online Web Tutorials, https://www.w3schools.com/default.asp.

[5] “Fantasy Games.” ESPN, ESPN Internet Ventures, https://fantasy.espn.com/.

[6] “FantasyPros.” FantasyPros, https://www.fantasypros.com/.

[7] “What Is a Dynasty Fantasy Football League.” Dynasty League Football, 11 Feb. 2022, https://dynastyleaguefootball.com/dynasty-fantasy-football/.

[8] “Fantasy Football Mock Draft Simulator.” 2022 Fantasy Football Mock Draft Simulator, https://draftwizard.fantasypros.com/football/mock-draft-simulator/.

[9] “Trade Analyzer.” RotoTrade, https://www.rototrade.com/fantasy-football-trade-analyzer.

[10] “Statcast: Glossary.” MLB.com, https://www.mlb.com/glossary/statcast.

[11] Luzzitto Tupaz, https://sites.wp.odu.edu/ltupa001/2022/01/30/four-ethical-issues-in-storing-individuals-information/.

[12] “MLB Stats, Scores, History, and Records.” Baseball, https://www.baseball-reference.com/.