Team Members

Kevin Brosam, Nate Lang, John Hattas, Alex Berkhout, Matt Petter

**User Stories:**

Kevin wants to win one billion dollars. Kevin has some picks that he is not sure about and needs something to do a more detailed analysis. He really values three-point percentage over points in the paint. By using this program Kevin can get a more detailed analysis of who will win the game.

Nate goes to the website and wants to generate a bracket to win a lot money. Nate goes to the website and generates a bracket. The website informs the user of all stats of each team and compares them to their opponent. Nate thinks the generated bracket looks fine, but he believes that one team that the website has winning will lose. Nate changes that one team around in his bracket and runs the generation again, except he manually picks the one victory. He then uses the bracket that was generated and wins a billion dollars.

John knows nothing about basketball but is looking for something to help guide his picks. He wants indicators to show which team is more likely to win based on which is picked. It generates a bracket based on the indicators. John is confident about some games and wants to override the picks. The bracket will regenerate the bracket based on the new picks. John wants to win a billion dollars.

**Task Cards:**

* Convert data into usable cvs files
* Create a git library and get the whole team able to pull and commit
* Get whole team using Jupyter and Anaconda3
* Write a program that uses the cvs files correctly
* Randomly pick a bracket, no indicators or bias
* Display the bracket in an easy to understand way
* Modify existing code to have a more precise method of picking winners

**Tests:**

* Have everyone commit to the git
* Have everyone open code using Jupyter
* Cvs file data is accurately read in, team names match ranks
* Bracket has correct teams playing eachother
* Bracket displays in the correct format
* Bracket picks winners based off an indicator rather than randomly

**Sprint Backlog:**

**Product Backlog:**