Gaming Central is a hub built to coalesce the varying aspects of the gaming entertainment ecosystem. Without Gaming Central users would have to go to several different access points to get the gaming information and entertainment they were looking for.

(i.e. if a user found out about a new game they would have to go to one place to get information on the game, another place to watch gameplay to see if they are interested, yet another if they want to actually buy the game. Another use case is for users that are already a part of a game’s community. They would have to visit various places to get recent game news, competitive information, and statistics for those tournaments and matches.)

Utilizes Angular and MaterializeCSS for the Front End Development and Design. Uses a Node and Express server along with MongoDB to save the User and Match information.

Some highlights:

* Security : Through the use of the Passport authentication framework and Google / Steam strategies with OAuth2.0 I am able to give the User the convenience of not having to set up a completely different logins and profiles for Gaming Central. A bonus on the security side due to that is I also do not have to store any user login information for either Google or Steam since the actual authentication is done through their systems.
* Real-time Information : Throughout the Game, Streams, News and Matches sections Users are given real-time information which is pulled either through one of several API’s that my application is using (where available). Or in the case of the game news information and the upcoming matches / player statistics, API’s were not available so I opted to build a couple of scraping scripts using Cheerio to pull the information from relevant websites.
* Ease of Administration : Since I am using so many real-time data sources it is extremely easy to keep the application up to date with new information. However, using methods such as web scraping do pose major issues. (i.e. if a page I am scraping changes design patterns). Therefore to protect against the possibility of getting bad information I have created an Admin panel where I can take the