

# CALIFORNIA POLYTECHNIC UNIVERSITY SAN LUIS OBISPO

## CPE SENIOR PROJECT FORMAL REPORT

### Free2Play

#### **Abstract**

Free2Play is a website based upon the wildly popular PC video game League of Legends created by Riot Gaming. The website contains helpful information for new and old players of the game, such as articles containing the latest League of Legends news, a rune calculator, and recommended item builds for characters. The website uses HTML5, jquery, node.js, MySQL, and PHP to pull data from Riot's web API and Champion.gg's website.

Author:

Corey Wilson

Supervisor:

John Oliver

December 8<sup>th</sup>, 2015

# Contents

Introduction	3
Related Work	4
Champion.gg	4
Surrender@20.net	5
Mobafire.com	5
Formal Product Definition	6
Features	7
Recommended Item Builds	7
Rune Calculator	8
Articles	9
Personas	10
Use Cases	10
Technology and Design	11
Node.js	12
PHP	12
Jquery	12
mySQL	13
Testing	13
Future Work	14
Conclusion	14
References	14

## **Introduction**

I started this project, because I love League of Legends and I wanted to learn cool, new web technologies. League of Legends is a competitive multiplayer game where you control a champion of Runeterra and team up with 4 other players to face off against another team of 5 players. The objective of the game is to invade the enemy's base and destroy a structure known as the nexus. Champions are aided by items that strengthen their stats and occasionally perform an action. The game can get very complex, very fast. With 128 champions, each with unique abilities and stats, and over 100 items the game can become overwhelming for newcomers. Even experienced players can have trouble keeping up with the constant balance updates.

Thus, Free2Play was born. Free2Play is a website that consolidates key information for players in a simple and easy to use format. To help players out I created a tool that recommends the highest win rate item build for each champion that is currently free. I host articles from around the internet pertaining to the latest changes and news for League of Legends. I also created a rune calculator – runes are equipment that can be used to add to a champion's statistics. Using my website will help players improve their game knowledge and it will give them a competitive edge in the game.

To achieve all of this, I had to learn new web technologies so that I could create these tools. A large portion of my time with this project was spent learning how to use technologies like node.js and Riot's web API (application programming interface). Undertaking this project has made me a better programmer and given me knowledge that will be helpful in the workplace.

## **Deliverables**

A .zip containing .html, .css, .php, and javascript files, images, and readme.txt file containing instructions on how to set up the server.

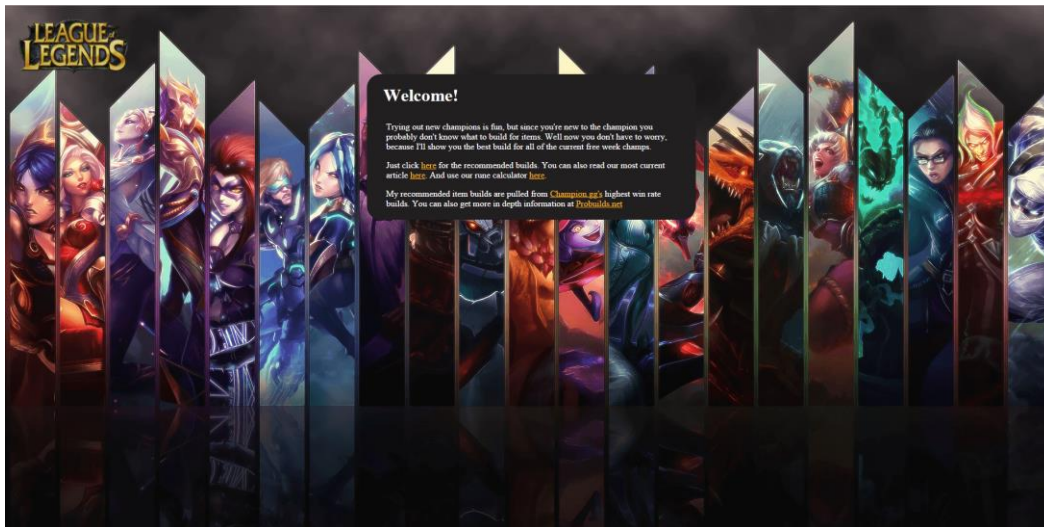


Figure 1: Home page

## Related Work

### Champion.gg

Champion.gg is a website that uses Riot's web API to gather and analyze data on League of Legends to create useful statistics for players. Using the website, you can check win rates for champions and items. I use champion.gg's data on highest win rate item builds for my recommended item builds page. It is an extremely useful website, but all it offers is statistical data, while my website offers news and a rune calculator tool.

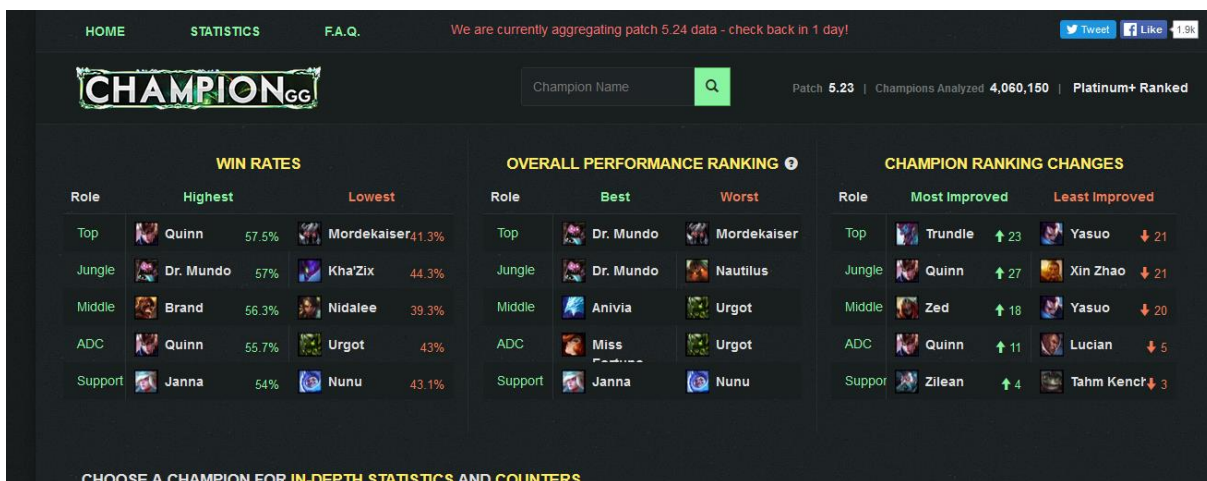


Figure 2: Champion.gg

## Surrender@20.net

Surrender@20 is a website that posts articles about League of Legends. The website uses Riot's web API to data mine for changes made to the games code and posts the latest game changes and other articles. I host a few of their articles on my website. Like champion.gg though, surrender@20 only post articles so other websites are need for different types of League of Legends information.

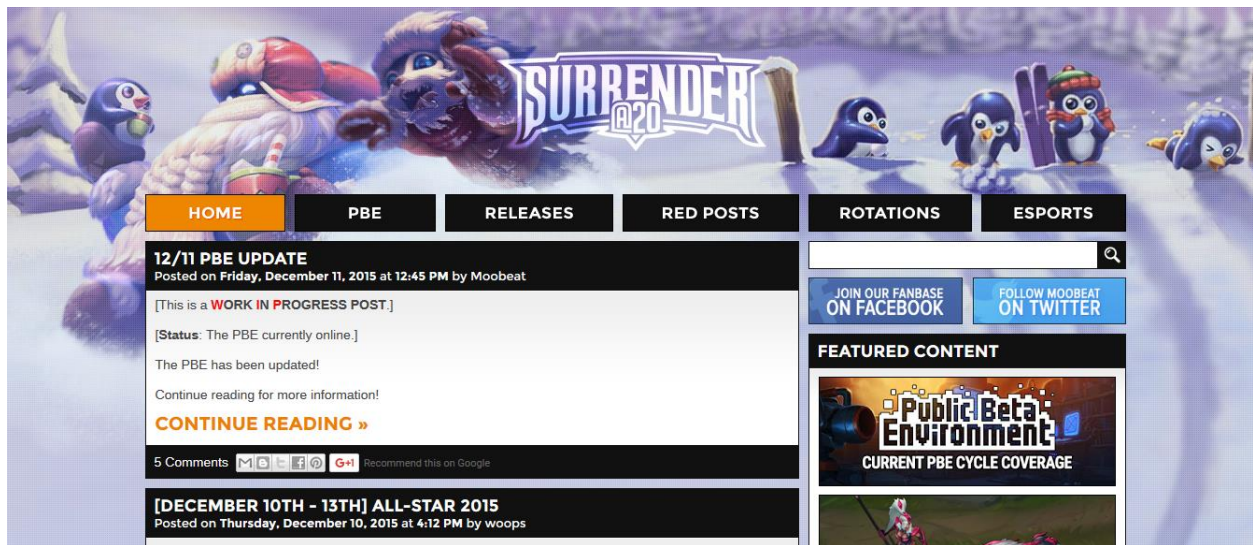


Figure 3: Surrender@20

## Mobafire.com

Mobafire.com hosts character guides that users upload and they have a rune planner tool that is similar to mine. They lack the other features of my website, so I will compete by offering more features. Users will be attracted to my site, because they don't have to use multiple websites to get all the information they want.

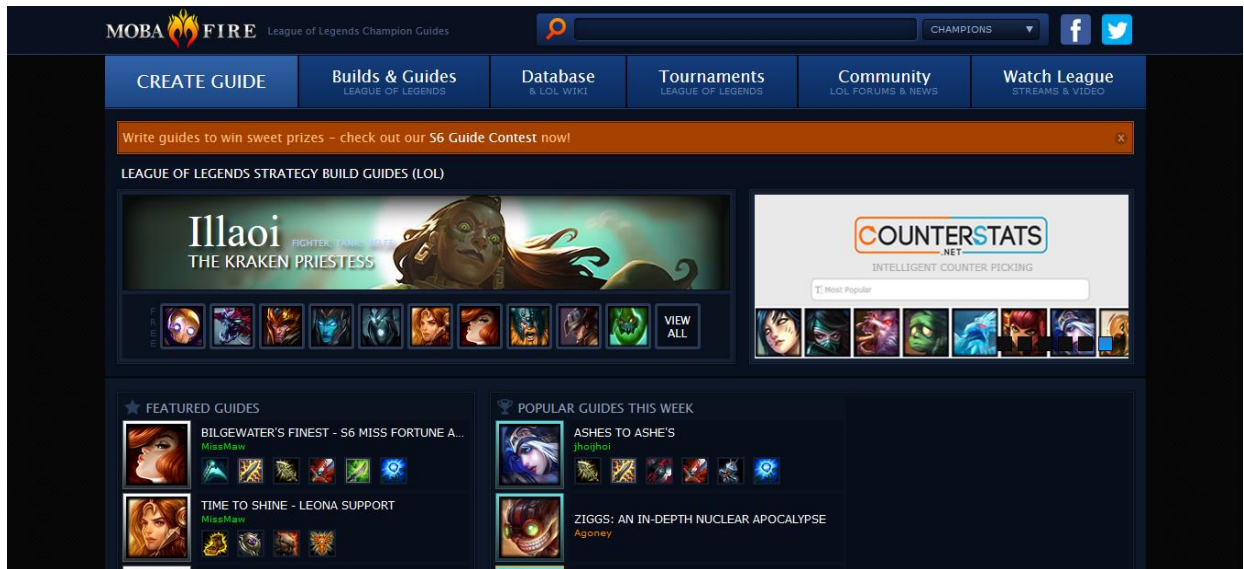


Figure 4: Mobafire

## Formal Product Definition

### Need Statement

There is a need for a singular website that contains helpful information for League of Legends players

### Objectives

To learn how to use web technologies like node.js, to consolidate league of legends data into a single website, and to entice users to visit the website

### Marketing Requirements

- Generate users that will support the site through ad revenue
- Create a simple to use UI
- Provide useful and relevant information about League of Legends
- Accurately gather data
- Support multiple browser resolutions
- Support multiple browsers

## **Engineering Requirements**

- Limit web page connections to 6 per load
- Keep image sizes under 500 KB
- Load web page in less than 2 seconds
- Process data server-side in less than 80 ms

## **Features**

### **Recommended Item Builds**

The first tool I created is the recommended item builds page. Every 2 weeks Riot makes a set of 10 characters free for players to use and try out. If players like the characters they can choose to spend currency earned in-game or use real money to unlock the character permanently. The problem is that 2 weeks is often not long enough to learn how to play a character, much less multiple characters. The players who try out these champions are typically new to these characters, so I created this tool to ease the players into these new champions.

As I described earlier, character in League of Legends can buy items with gold that bolster their stats and abilities. Due to the constant balance updates, items frequently change and it can be hard to keep up with the best items for each character. Using Riot's web API, I pull data from their servers about all the characters and find out which champions are currently free. Using PHP, I pull the highest win rate item builds from a website called [champion.gg](http://champion.gg) that collects and displays statistics about League of Legends and I store this information in a MySQL database. When the user loads the webpage I pull the needed data from my database, process the data server-side, and inject the data into the page as an XML document.

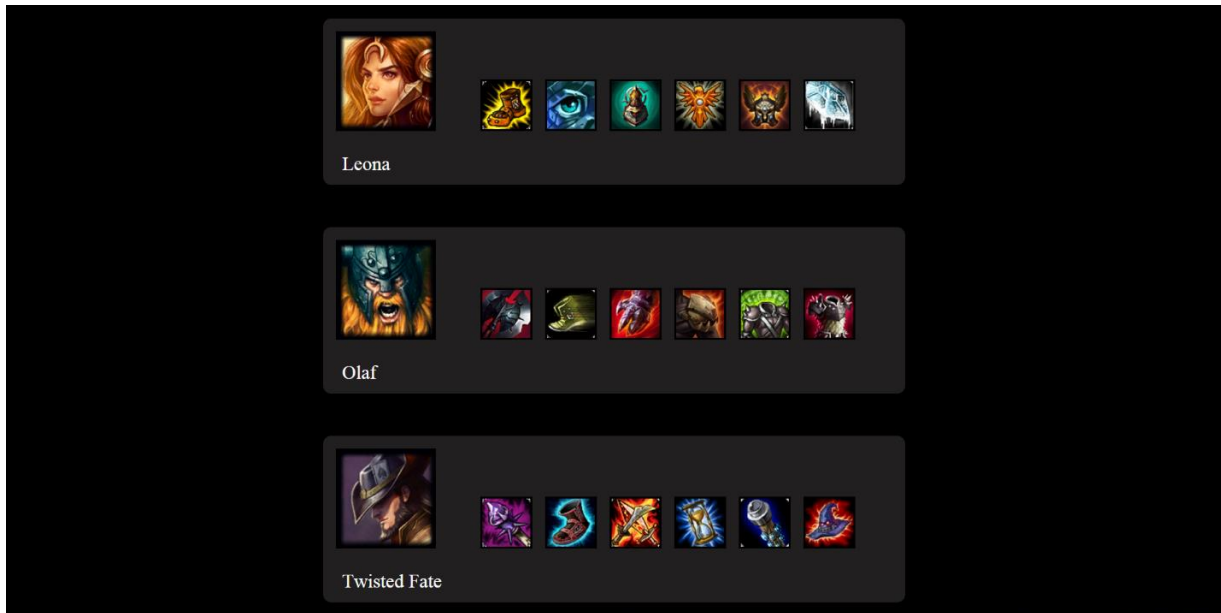


Figure 5: Recommended item builds for 3 free week characters

All of this data is automatically updated using node.js and PHP, so when the page loads it will always display the item builds for the current 10 free characters (unless a champion is currently disabled and not usable, because it has a gameplay bug).

## Rune Calculator

The second tool I created is a rune calculator. Runes are equipment that players can buy and equip to improve a character's stats before a match starts. They allow the player to customize the character for any play style and situation. Unfortunately, there are many types of runes and this adds complexity to the game.

Each player is allowed to equip a total of 30 runes: 3 Quintessences, 9 Marks, 9 Seals, and 9 Glyphs. Each rune improves a stat and each type of rune adds a different amount to that stat. For some this is pretty confusing. My rune calculator allows the user to select the stat they want to improve for each rune you can equip and displays the sum of the stat changes at the bottom. Using this tool, players can plan out what runes they want to use and optimize the stat bonuses they get from runes. The calculator uses jquery to pull the input, calculate the values, and then display the changes.



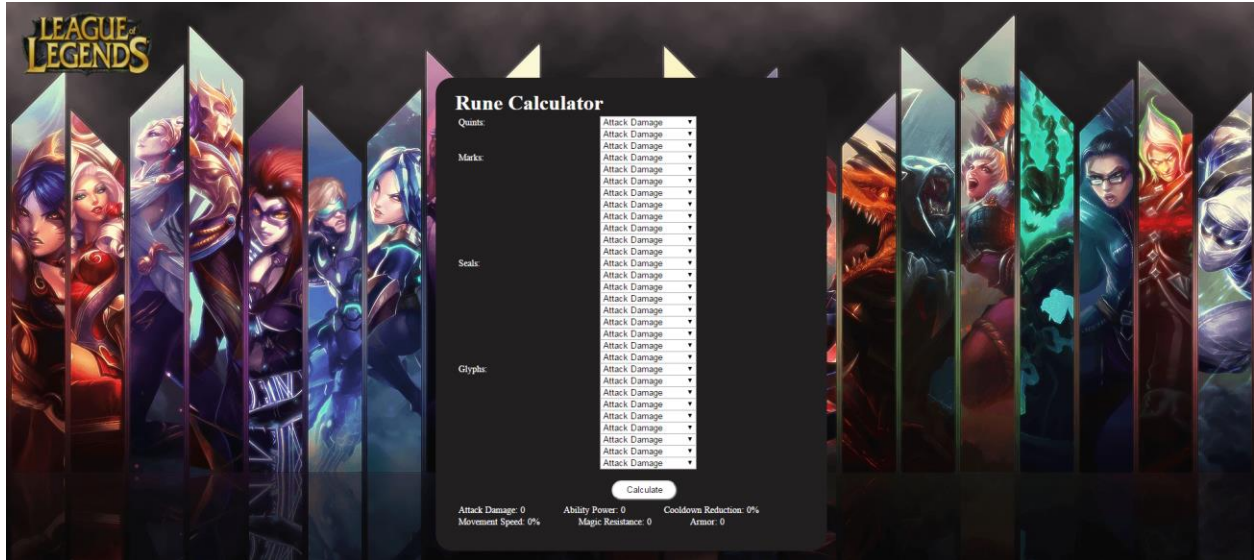


Figure 6: Rune Calculator

## Articles

League of Legends is constantly changing, new characters and skins are released, items are changed, and events are occurring. It's hard to stay on top of the latest news, so I host articles from around the web on my website to consolidate League of Legends news in one easy to read and find place.

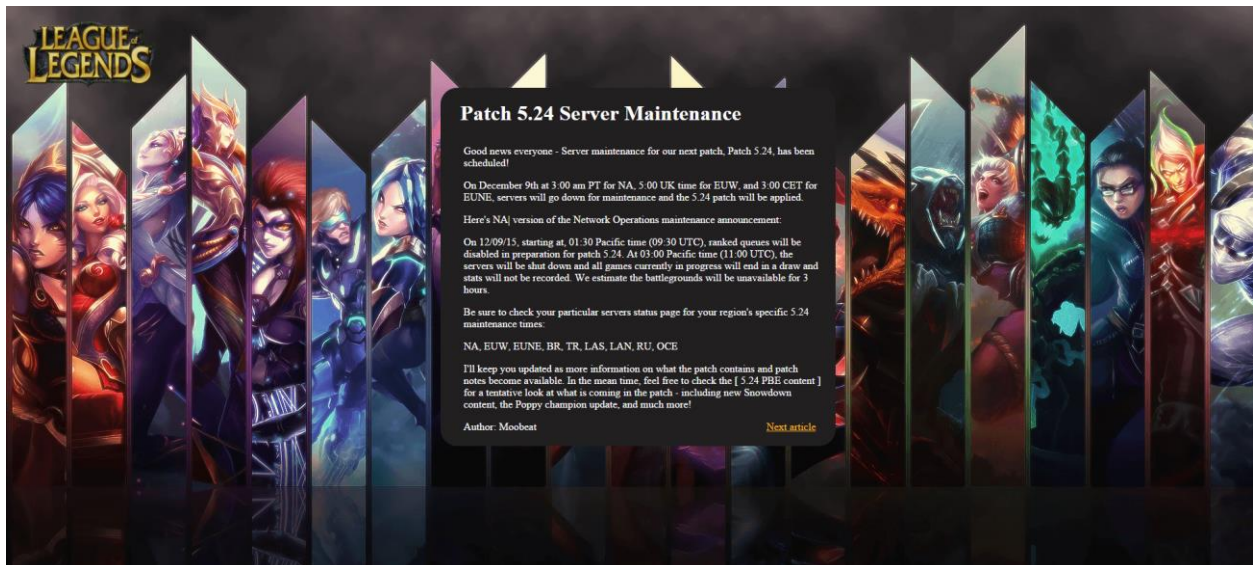


Figure 7: An article about server maintenance

## Personas

Experienced Player – The experienced player wants the latest League of Legends news. He wants to be competitive, so he wants find out how to win more often. The experienced player knows what he needs and may use other League of Legends websites.

Novice Player – The novice player wants to learn how to play the game. He doesn't own many characters, so he uses the free week characters frequently. This player doesn't know many League of Legends websites and would likely visit the site again. The primary demographic.

## Use Cases

Case: Reading about the newest patch changes

Primary Actor: League of Legends player

Scope: A website

Steps:

- User navigates to the website
- User clicks the link to the articles
- User may click the next article link to see other articles
- User reads about the patch changes

Case: Make a plan for a rune page

Primary Actor: League of Legends player

Scope: A calculator

Steps:

- User navigates to the website
- User clicks the link to the rune calculator
- User selects various combinations of runes and clicks the calculate button
- The web page displays the stat boosts for the currently selected runes

Case: Find out the highest win rate item build for a character

Primary Actor: League of Legends player

Scope: A website

Steps:

- User navigates to the website
- User clicks the link to the recommended items page
- Web page displays the highest win rate item builds for the active free week champions

## Technology and Design

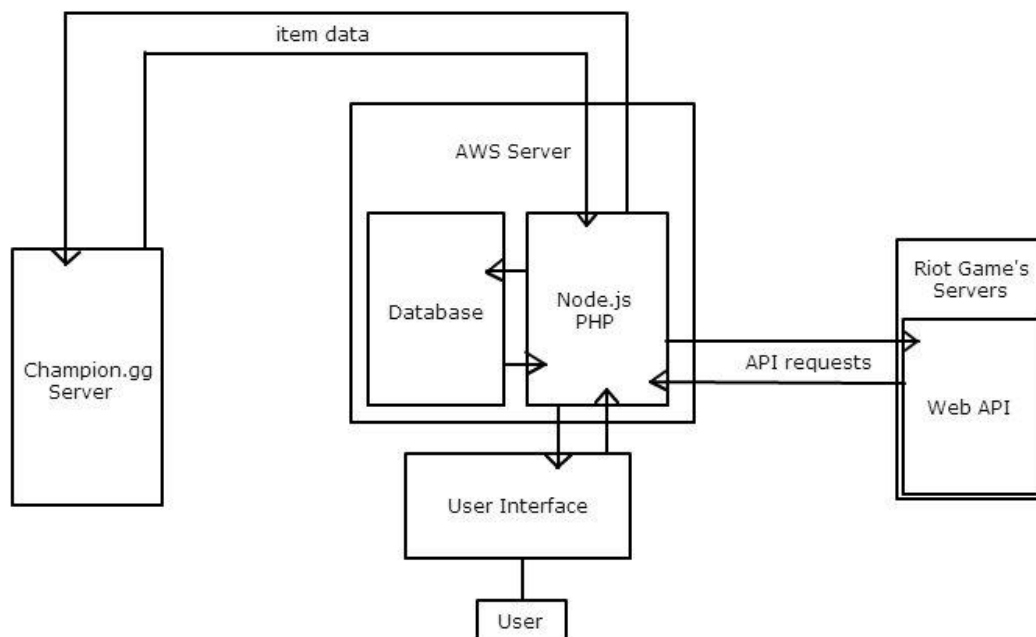


Figure 8: Dataflow diagram

The user accesses the UI through a browser. HTML, Javascript, and CSS files are sent from the server to display the web page. Node.js and PHP work together to connect to Riot's server to access a JSON object containing relevant free week champion info and they connect to champion.gg's servers to pull the item data. This data is then stored in the database using PHP. When a user accesses a webpage the relevant data retrieved from the database and sent to the UI as an XML document to be displayed.

## **Node.js**

Node.js is a javascript library that allow for server-side script. Scripts are programs that don't need to be compiled to run. Node.js was by far my favorite technology to learn and use on this project, because of its versatility. Node.js runs asynchronously so it can process anything it gets sent on multiple threads. In my project, I use node.js to run my PHP scripts that pull new data and automatically update my database. My script is running constantly, so the moment any of the data for the recommended item builds changes, it runs the necessary scripts to handle this event. I also use it to process some data server-side. If a user disables javascript, my website won't completely fail.

## **PHP**

PHP is a web programming language used to interact with servers and databases. In my program I use PHP to "crawl" champion.gg and pull the highest win rate item data from the website. I also insert data into my database through PHP functions. For my recommended item builds tool I insert into the table using PHP. PHP is a core web development language that is essential to the operation of most websites.

## **Jquery**

Jquery is the most popular javascript library. It is a compilation of functions that make javascript easier to use and it makes it more versatile. Jquery makes up the backbone of my rune calculator. When I connect through Riot's web API, I use Jquery calls to return a JSON (Javascript Object Notation) object and then use it to parse the object for the data fields I need. Jquery is also an integral part of most modern websites.

```

{"id":103,"active":true,"botEnabled":false,"freeToPlay":false,"botMmEnabled":false,"rankedPlayEnabled":true},
{"id":84,"active":true,"botEnabled":false,"freeToPlay":false,"botMmEnabled":false,"rankedPlayEnabled":true},
{"id":12,"active":true,"botEnabled":true,"freeToPlay":false,"botMmEnabled":true,"rankedPlayEnabled":true},
{"id":32,"active":true,"botEnabled":true,"freeToPlay":false,"botMmEnabled":true,"rankedPlayEnabled":true},
{"id":34,"active":true,"botEnabled":false,"freeToPlay":false,"botMmEnabled":false,"rankedPlayEnabled":true},
{"id":1,"active":true,"botEnabled":true,"freeToPlay":false,"botMmEnabled":true,"rankedPlayEnabled":true},
{"id":22,"active":true,"botEnabled":true,"freeToPlay":false,"botMmEnabled":true,"rankedPlayEnabled":true},
{"id":268,"active":true,"botEnabled":false,"freeToPlay":false,"botMmEnabled":false,"rankedPlayEnabled":true},
{"id":432,"active":true,"botEnabled":false,"freeToPlay":false,"botMmEnabled":false,"rankedPlayEnabled":true},
{"id":53,"active":true,"botEnabled":true,"freeToPlay":true,"botMmEnabled":true,"rankedPlayEnabled":true},
{"id":63,"active":true,"botEnabled":true,"freeToPlay":false,"botMmEnabled":true,"rankedPlayEnabled":true},
{"id":201,"active":true,"botEnabled":false,"freeToPlay":false,"botMmEnabled":false,"rankedPlayEnabled":true},
{"id":51,"active":true,"botEnabled":true,"freeToPlay":false,"botMmEnabled":true,"rankedPlayEnabled":true},
{"id":69,"active":true,"botEnabled":true,"freeToPlay":false,"botMmEnabled":true,"rankedPlayEnabled":true},
{"id":31,"active":true,"botEnabled":true,"freeToPlay":true,"botMmEnabled":true,"rankedPlayEnabled":true},
{"id":42,"active":true,"botEnabled":false,"freeToPlay":false,"botMmEnabled":false,"rankedPlayEnabled":true},
{"id":122,"active":true,"botEnabled":true,"freeToPlay":false,"botMmEnabled":true,"rankedPlayEnabled":true},
{"id":131,"active":true,"botEnabled":false,"freeToPlay":false,"botMmEnabled":false,"rankedPlayEnabled":true},
{"id":119,"active":true,"botEnabled":false,"freeToPlay":false,"botMmEnabled":false,"rankedPlayEnabled":true},
{"id":36,"active":true,"botEnabled":true,"freeToPlay":false,"botMmEnabled":true,"rankedPlayEnabled":true},
{"id":245,"active":true,"botEnabled":false,"freeToPlay":false,"botMmEnabled":false,"rankedPlayEnabled":true},
{"id":60,"active":true,"botEnabled":false,"freeToPlay":false,"botMmEnabled":false,"rankedPlayEnabled":true},
{"id":28,"active":true,"botEnabled":false,"freeToPlay":false,"botMmEnabled":false,"rankedPlayEnabled":true},
{"id":81,"active":true,"botEnabled":true,"freeToPlay":false,"botMmEnabled":true,"rankedPlayEnabled":true},
{"id":9,"active":true,"botEnabled":true,"freeToPlay":false,"botMmEnabled":true,"rankedPlayEnabled":true},
{"id":114,"active":true,"botEnabled":false,"freeToPlay":false,"botMmEnabled":false,"rankedPlayEnabled":true},
{"id":105,"active":true,"botEnabled":true,"freeToPlay":false,"botMmEnabled":false,"rankedPlayEnabled":true},
{"id":3,"active":true,"botEnabled":true,"freeToPlay":false,"botMmEnabled":true,"rankedPlayEnabled":true},
{"id":41,"active":true,"botEnabled":false,"freeToPlay":false,"botMmEnabled":false,"rankedPlayEnabled":true},
{"id":86,"active":true,"botEnabled":true,"freeToPlay":false,"botMmEnabled":true,"rankedPlayEnabled":true},
{"id":150,"active":true,"botEnabled":false,"freeToPlay":false,"botMmEnabled":false,"rankedPlayEnabled":true},
{"id":79,"active":true,"botEnabled":false,"freeToPlay":false,"botMmEnabled":false,"rankedPlayEnabled":true},
{"id":104,"active":true,"botEnabled":true,"freeToPlay":false,"botMmEnabled":true,"rankedPlayEnabled":true},
{"id":120,"active":true,"botEnabled":false,"freeToPlay":false,"botMmEnabled":false,"rankedPlayEnabled":true},
{"id":74,"active":true,"botEnabled":false,"freeToPlay":false,"botMmEnabled":false,"rankedPlayEnabled":true},

```

Figure 9: A visual display of a JSON object

## MySQL

MySQL is a relational database often used in web development. While there are many relational database languages you can use, MySQL is one of the most popular language with the most documentation and it fit my needs for my website. I use a relational table with fields for a character's id, name, picture link, and links for each of the items. I run the database on an Amazon Web Services micro server running Ubuntu and a LAMP (Linux Apache MySQL PHP) stack.

## Testing

Browser Testing – I used Firefox, Edge, Chrome, and Safari to display the website. I visually checked for any bugs

Feature Testing – I clicked each link or button to make sure they functioned correctly

Debug Console – All actions print errors in the debug console. I generated the tests myself.

## **Future Work**

I'd like to continue adding more tools for League of Legends players. In particular I'd like to work more with node.js. I would need to apply to be considered a full-time web developer under Riot's system. Right now I'm limited in the number of API inquiries I can make. I need to register for a domain name, and move to a larger server if I want to expand the website. To support the costs of a domain name and larger server I would need to create an ad revenue model. I'd also like to create a menu bar as I add more features

## **Conclusion**

When I started my project I wanted to help League of Legend players and learn web technologies. This project taught me a lot about node.js, PHP, and mySQL, but there's always more to learn, and there are more tools I can create to help players using these technologies, but for the scope of the project I am satisfied with the outcome. The biggest challenge ahead will come from advertising my site and getting it used by the public. There are websites that perform some of the functions my website does, but none of the competing websites have a broad range of tools like mine. I will sell my website as an all-in-one site and beat the competition by offering easy access to any information a user needs.

## **References**

<http://w3schools.com> – Web development tutorials and references

<http://champion.gg> – League of Legends data website

<https://aws.amazon.com> – Amazon web server hosting website

<http://stackoverflow.com> – Programming question and answer site