**Knowledge is Power Program Los Angeles (KIPPLA): KASTLE 2.0**

Faculty Advisor:

Russ Abbott

Jungsoo (Sue) Lim

Company Liaison:

Kimberly Eng

Team:

Jose Gonzalez

Safa Al Mahbub

Juan Perez

**Table of Contents**

1. **Introduction 1**
2. **Related works and technologies 2**
3. **System Architecture 2**
4. **Results and Conclusions 3**
5. **References** 5
6. **Introduction**

Knowledge is Power Program Los Angeles (KIPPLA) is the Los Angeles Division of the nation wide charter school system KIPP. KIPPLA’s analytics team gathers and inputs data into a Tableau Server. Tableau is a data analytics program that takes data, stores it, and configures the data into easy to understand visual representations. This visual representation is called a “dashboard”, and that is how we will describe this information for the duration of this report. This data has an embedded link that can be used in html iframes to display the information. In order to make this information available to user, the KIPPLA data analytics teams created the website “KASTLE” with Google sites.

KIPPLA has four different user levels. These user levels are listed below.

* 1. Admin access – admin to edit the website
  2. SST - can view all the dashboard pages
  3. School Leaders - can view all the dashboards except those for SST only
  4. All Users - can only view certain dashboards

Currently KASTLE cannot support multiple users and so has to host three separate websites KASTLE Courtier, KASTLE Knight, and KASTLE Steward. Essentially if a page is available for all users the admin has to create the same webpage with the same embedded Tableau Dashboard across all websites. A second caveat of KASTLE is that it cannot implement a single sign on with tableau. All the information KIPPLA stores on Tableau is private and confidential information. As a result, only those with a user log in information and permission can access the data on tableau. This security is also mimicked on KASTLE. The embedded iframe with the link to the tableau page understands that the information is private and won’t allow a user to see the information until they have signed in to Tableau with the proper credentials. A few other problems with KASLTE is that the website does not have an optimal navigation bar, home page, or search function.

For these reasons, KIPPLA came to California State University Los Angeles (CSULA) to create a better functioning website, KASTLE 2.0. The main goal of the project is to host one website that addresses the following requirements:

1. Have a website that can handle multiple priority levels of users.
2. Have a dynamically created navigation bar that changes with different levels of access
3. Have a search function that can filter pages based on certain taxonomies and categories.
4. Create a home pages with a favorites, recently visited, and newly created pages area
5. Implement Single sign on to tableau and Website.
6. Have daily auto importing of users
7. Import Data from tableau to automatically generate webpages on the website.

As a team we decided to use a Content Management System (CMS) to build the website. The decision to use a CMS instead of building from scratch was largely based on two factors, ease of use and time. This project was given to CSULA after more than half of fall semester was finished, and so only left a few months to work on. A CMS is easy to understand and use once learned and would allow KIPPLA to make changes to the website further along the line if they needed to add more features. We decided to use Wordpress because of all the CMS’ available, it had the lowest learning curve and seemed to have a wide and supportive community to offer help.

The Website will be hosted on a SiteGround server. SiteGround holds many feature that makes host a site easy, such as phpMyAdmin, CMS website transfers, and more. This allowed the website to be easily set up and quick to use.

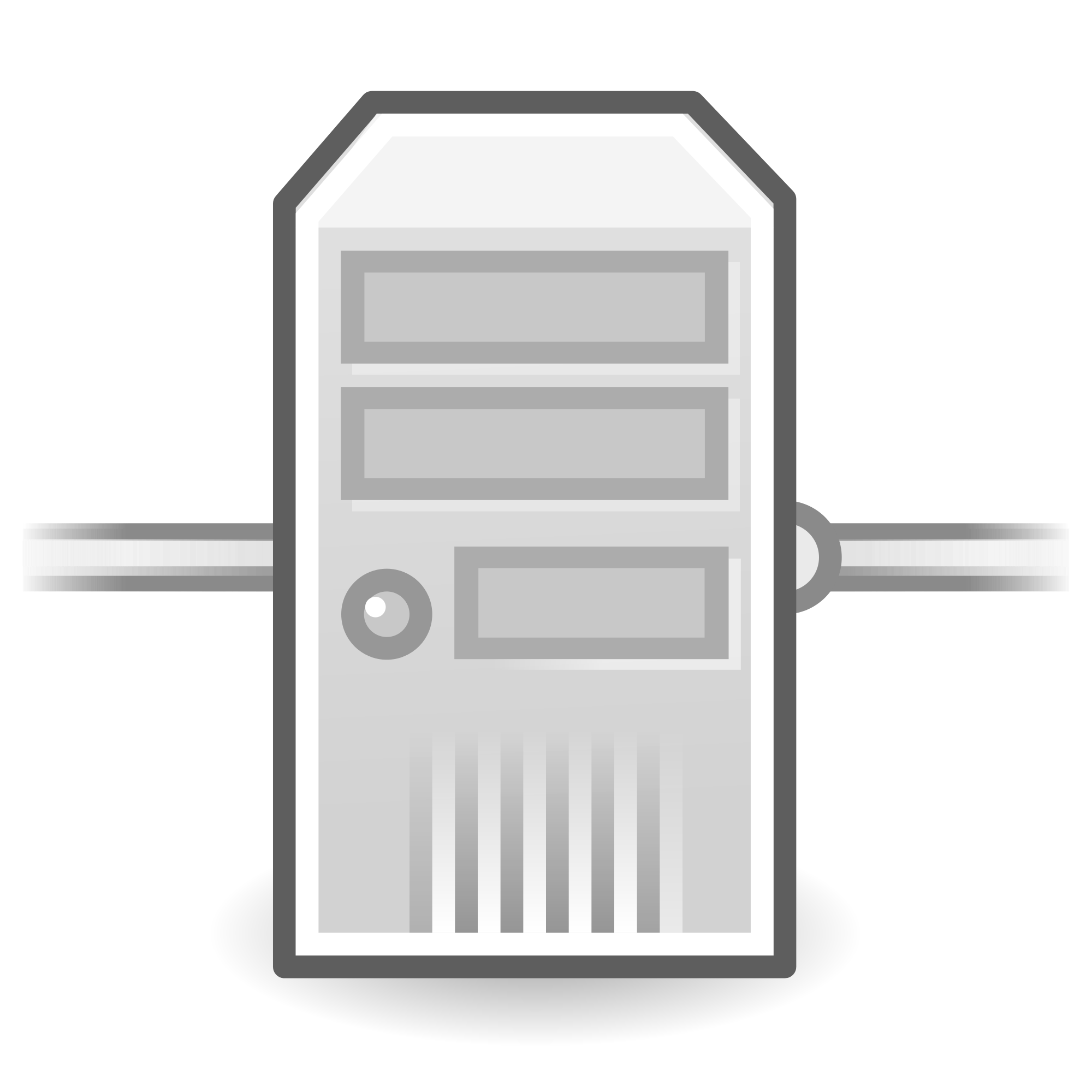
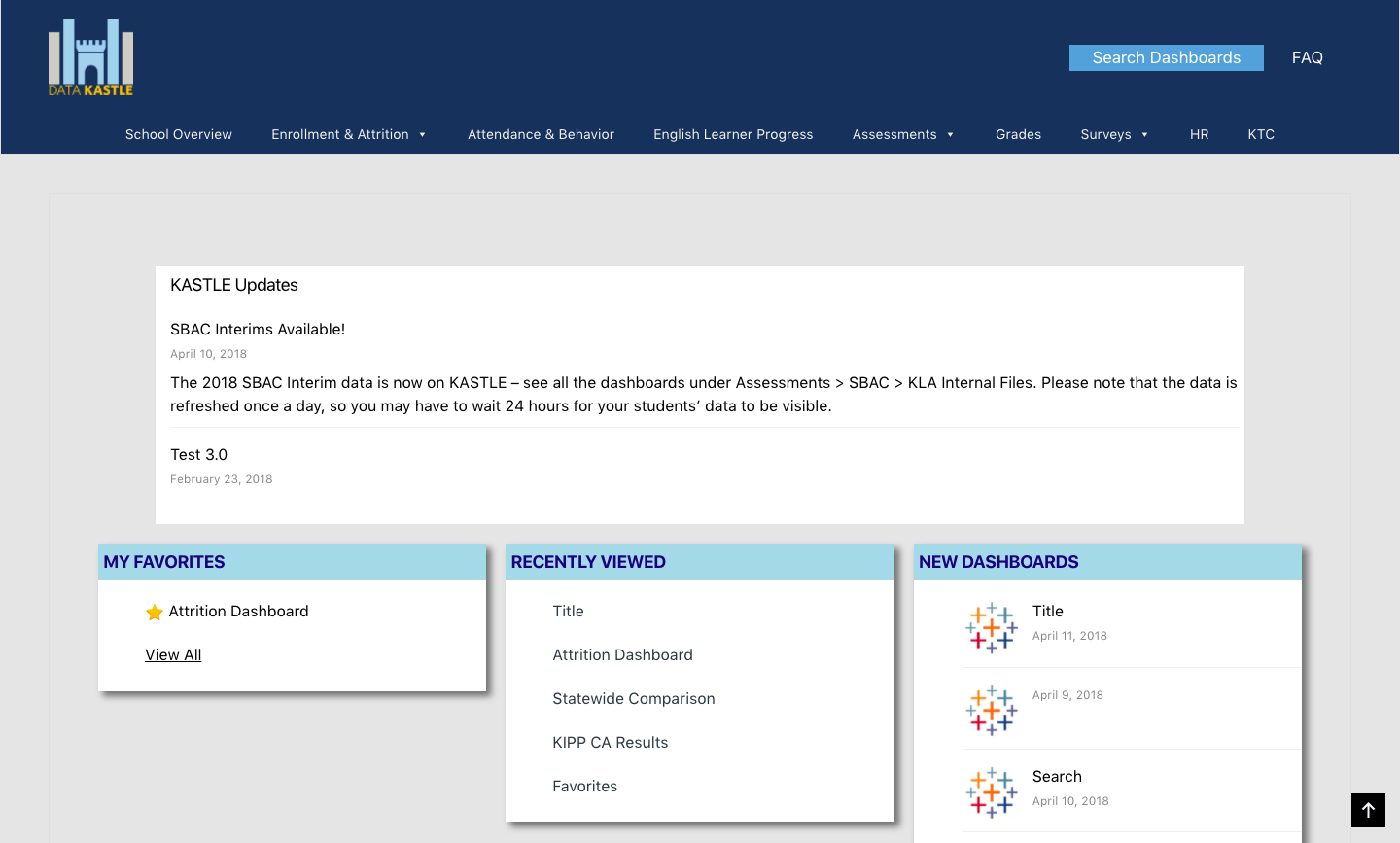
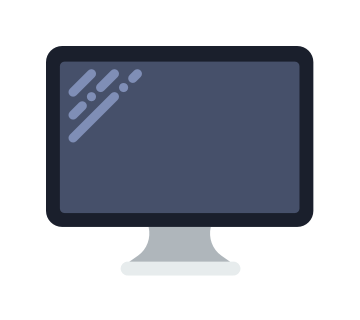
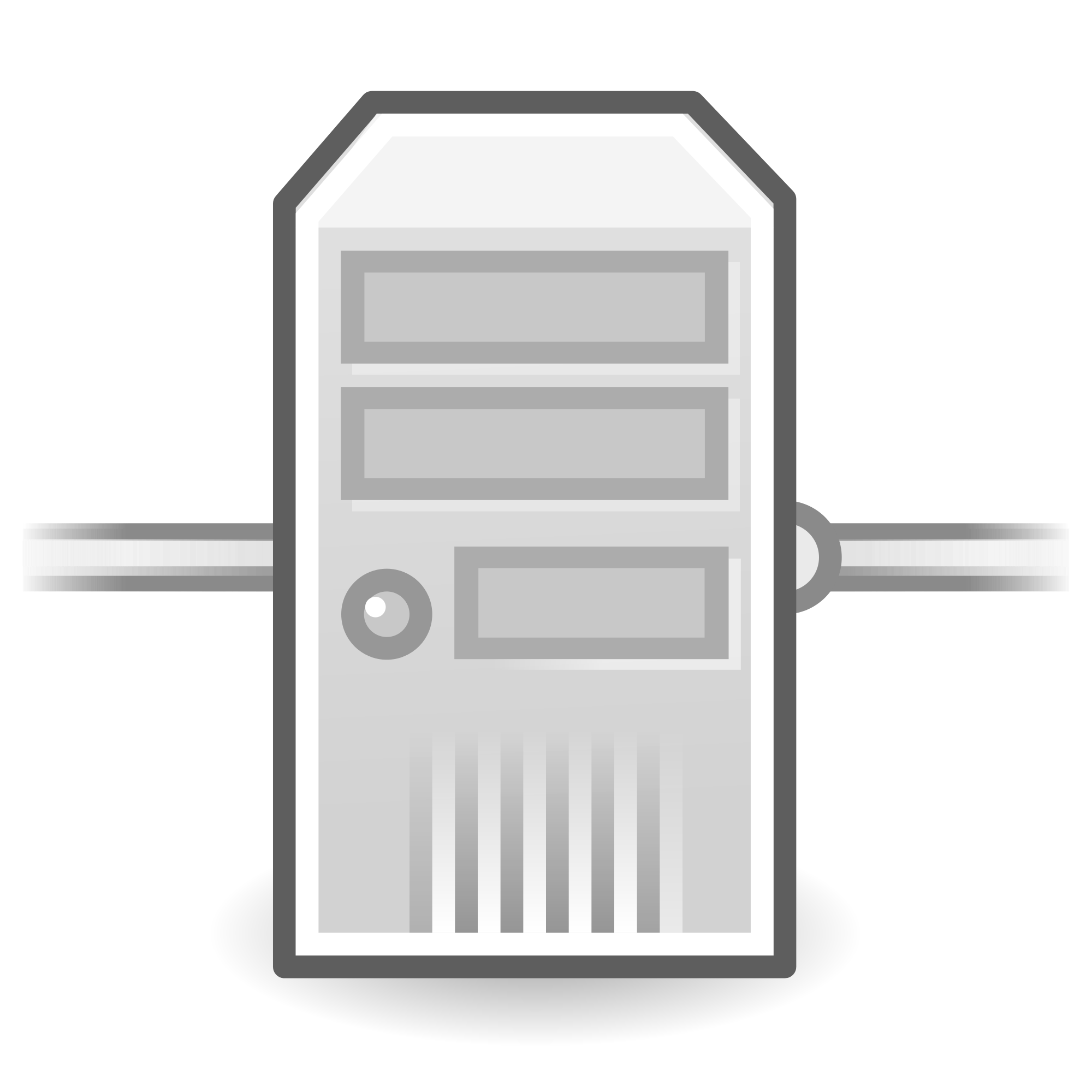
1. **Related works and technologies**

There are many different websites that are run and used by Wordpress, however because it is a CMS, there is not free code to base a website off of. There are however, many different themes that are free to use that make it easy to model the website.

1. **System Architecture**

Creating this project through a CMS and many different free open source plugin to Wordpress, we had very little control over the internal and external data flow of the website, server, and Tableau. Many of the features that we implemented, such as favorites and recently viewed are plugins made rom Wordpress or are built by other developers to work within Wordpress. Furthermore, it is hard to distinguish what the data flow of the whole system is as a whole.

There is however a very clear data flow between Wordpress and tableau on the Dashboard pages. When a User goes to a page, the website loads the page information onto the browser. One the page reaches the iframe; Wordpress makes a request to the tableau server to gain access to the information. If the user has the credentials to see the information, Tableau will give back the information. However if the user does not have the right credentials, they will be denied access until it is provided. Once the access is granted, the information will be displayed in the iframe embedded on the Wordpress page. Essentially, the SiteGround Server that hosts KASLTE 2.0 acts as a wrapper for the Tableau Dashboard Pages. Each wrapper page is stored in SiteGround, and the information is retrieved from tableau when each page is accessed.



Store’s Webpage Data, Users, and User’s Permission Level

Store’s Tableau Dashboards and User Permissions

Send and retrieve Website Information

Ask for Information

Send Information

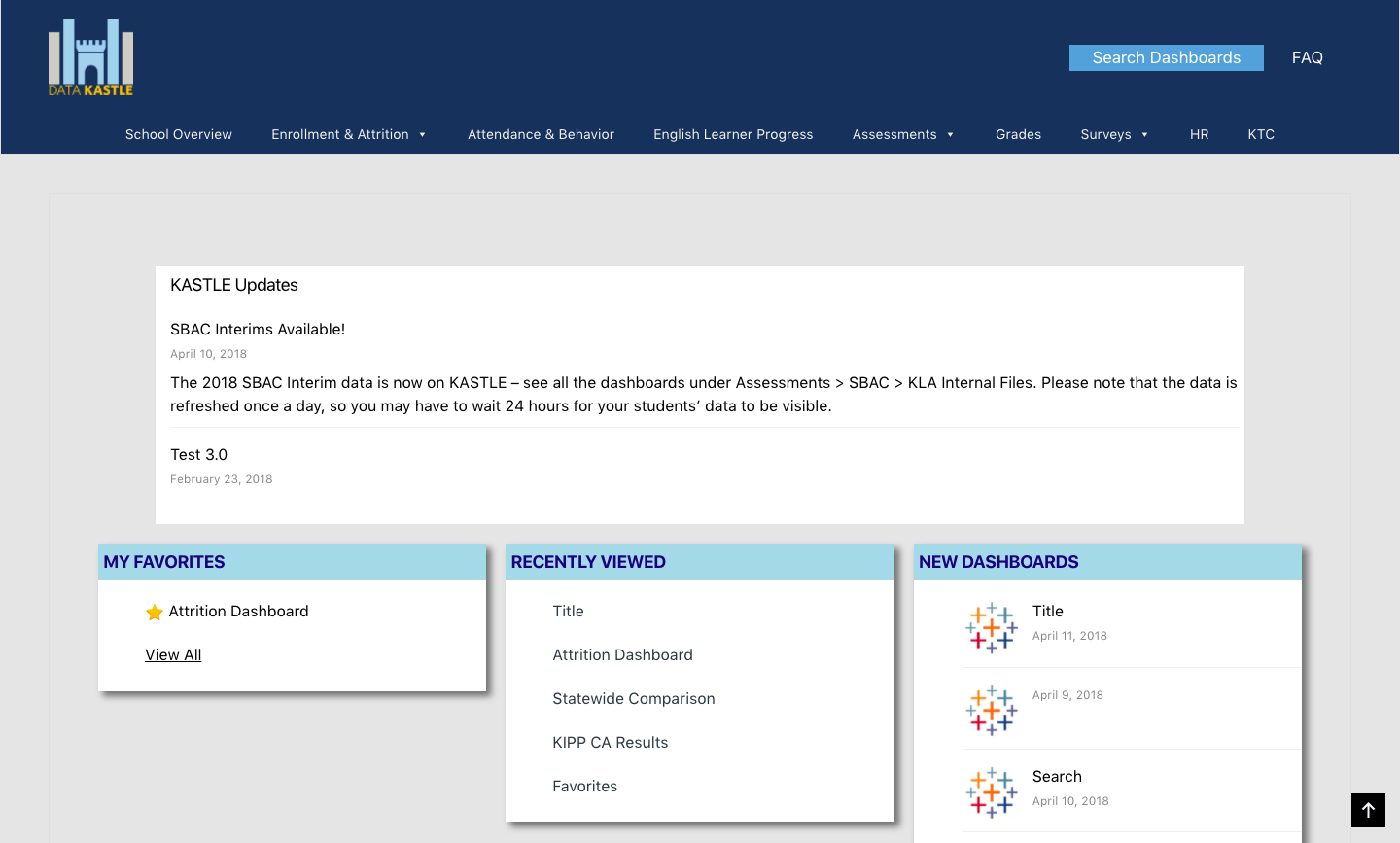
Ask for Tableau Data

Send Tableau Data

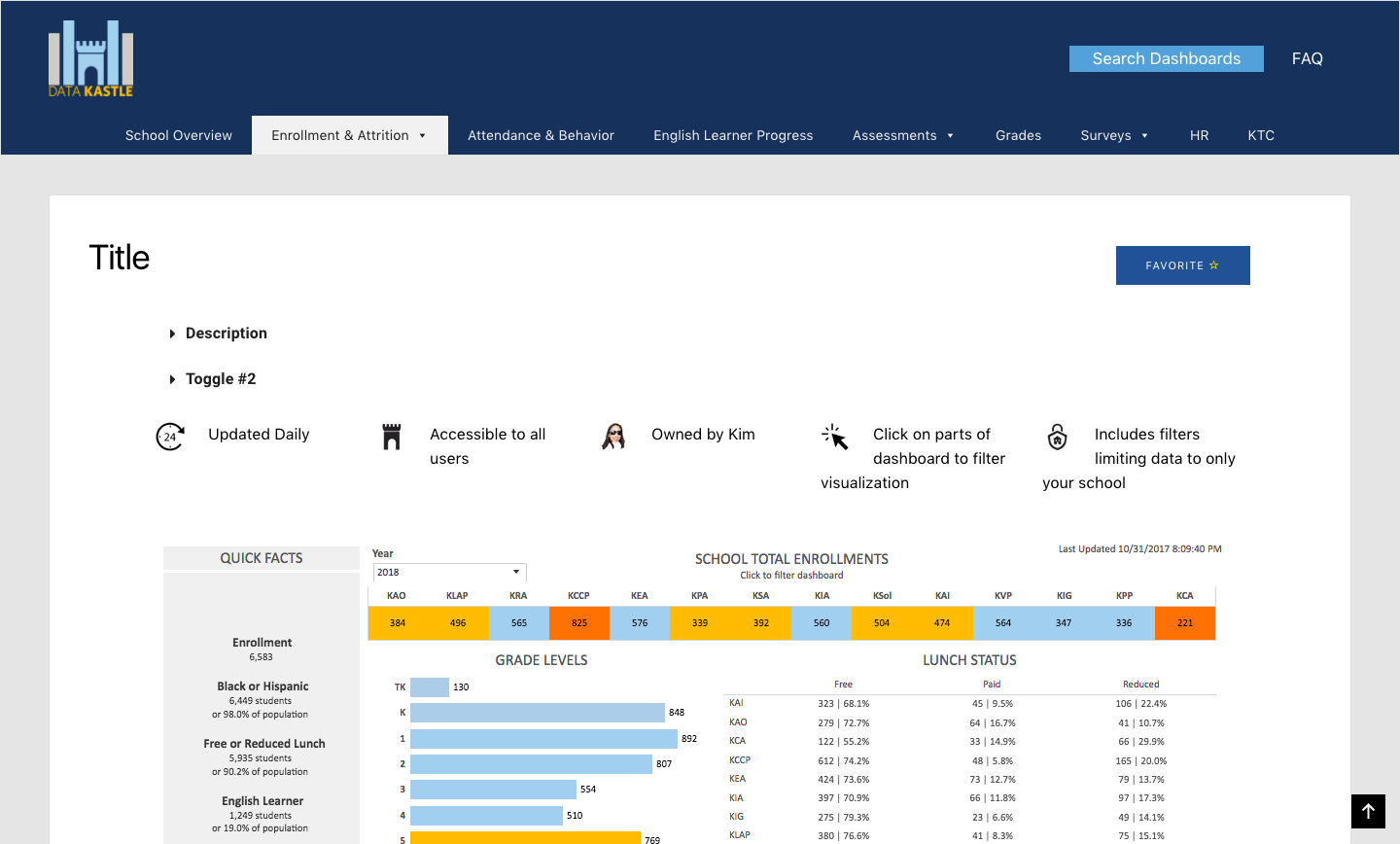
1. **Results and Conclusions**

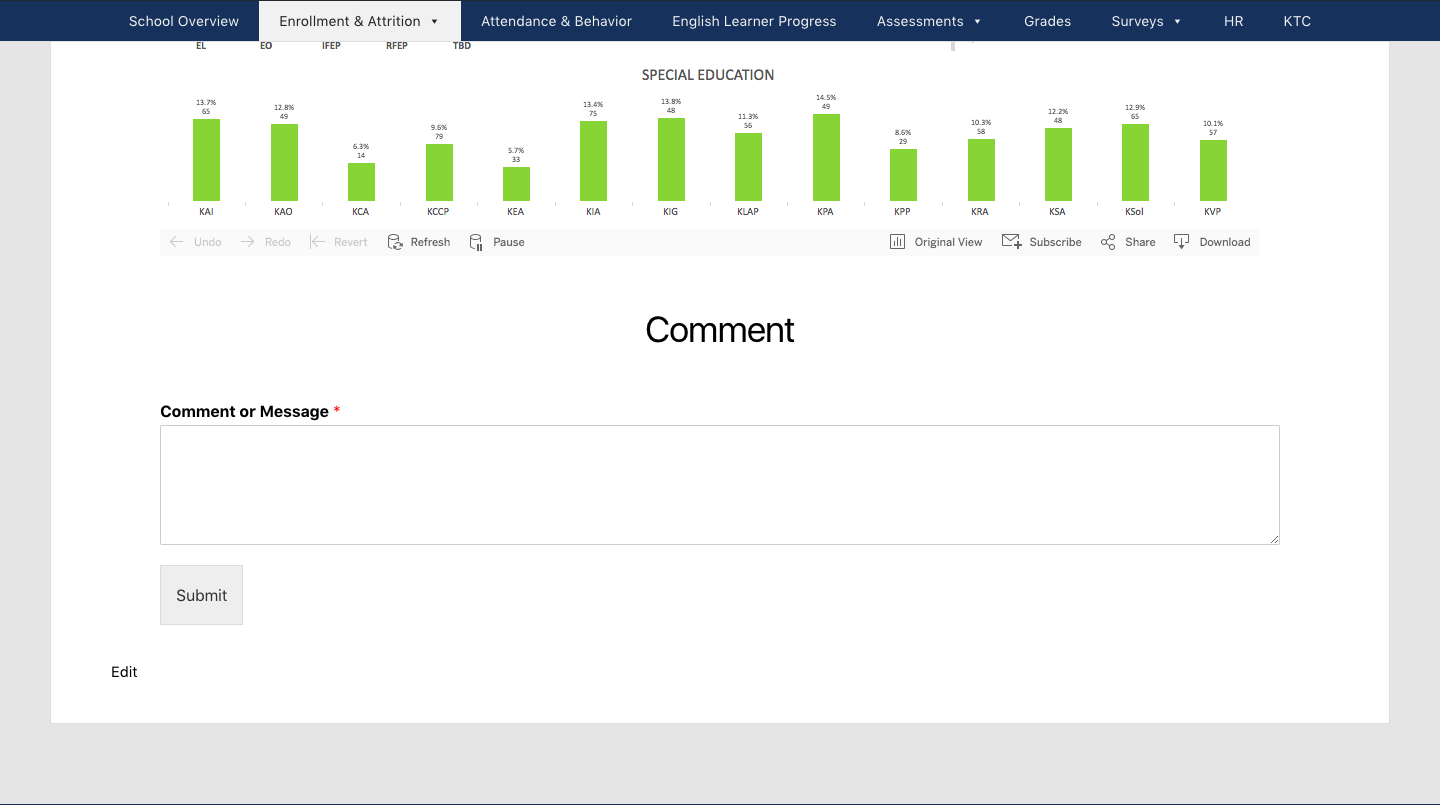
We were able to implement a lot of the functionalities that was required for KASTLE2.0. Hosting only one website, there are four levels of user permissions. These user permissions are determined by the admin, so one a user logs in they are automatically linked to their permission level. Once logged in, the user is redirected to a home page. This home page has four main areas, the updates area in the top of the page, a favorites area, recently viewed, and new pages.

There is a static navigation bar that will change depending on the devices used to access the website. This navigation area has the logo that acts as an anchor to the home page. There is a Frequently Asked Questions (FAQ) button that will redirect the user to a FAQ page with more information about the website and the company. There is a search button that will redirect users to a search page. This Search page will list all the pages on the site on the right, and on the left will have a series of filters to filter the information with. And lastly there is a navigation bar. This navigation bar is a multi level menu that holds all the different categories and sub categories that KIPPLA uses to categorize their pages. It is also dynamically generated based on a users permissions level. So a user cannot see a page that they are not supposed to see. If you hover over one option, it will reveal the next level, until you go to the last level of pages. Here if you click on a page it will redirect you to the page selected.



The actual dashboard page is a simple page. There is the dashboard title listed at the top of the page, as well as a favorites button. Underneath that will be a few toggles that the admin and author can chose to use. Bellow that there are icons that are dynamically displayed to portray more information about the dashboard that the user will be viewing. After that there will be the Dashboard from Tableau that is embedded into the page using an iframe. Underneath that there will be a comment box that will take a comment about the page and send it to the author of the page or the admin.





One major requirement that was not implemented was the creation of a single sign on. This was probably the lengthiest part of the project in terms of research and work, but in the end was unattainable in the time allotted. As stated before, this project was to be done in half a semester, a total of around three months. With an allotted time of a month to add this feature, it was still nowhere close to finished. An executive decision was made by the team that one feature was not worth more then the actual website.

1. **References**