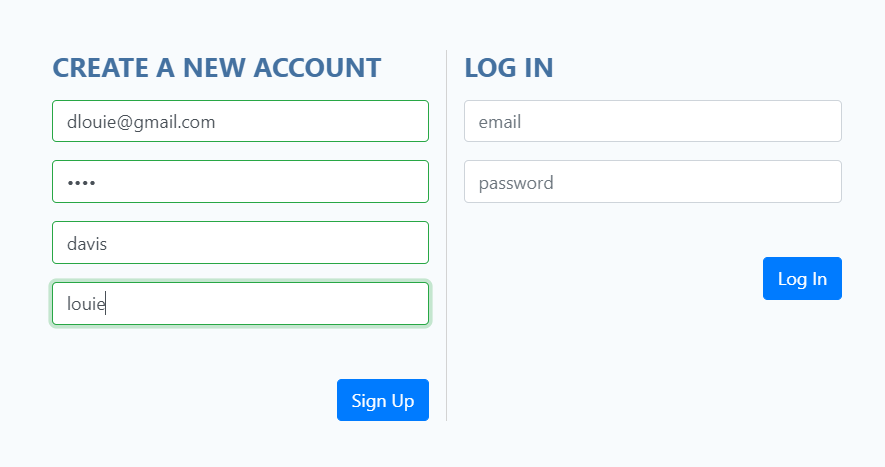
**VTCCCS User Manual**

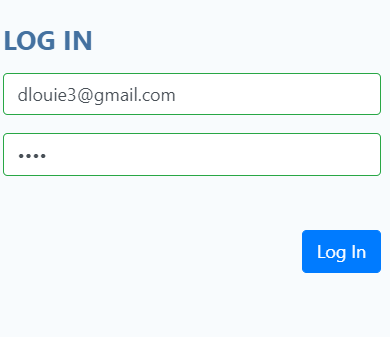
This guide is to aid users in using VTCCCS web application. Jumping into this application without a guide is possible, but several features may be missed at first glance.

**Register/Login**

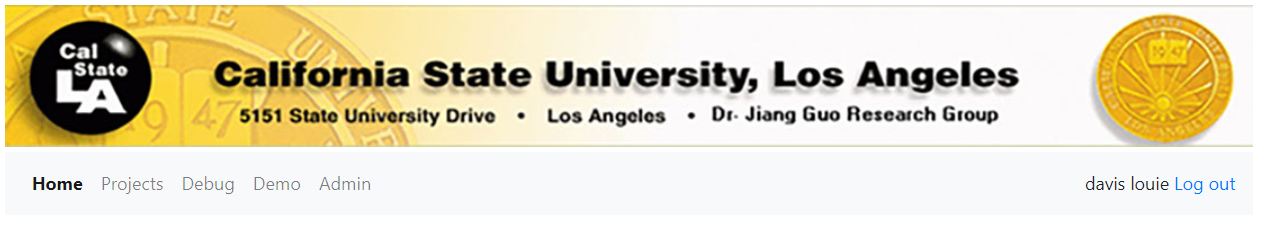
The first step is to login to the web application. Click on the Log in or sign up text in the right-hand side of the navigation bar. This should bring up the login/register modal window. Now, if the users are new to the website, they will use the left side to create a new account. If the users already made an account, they can use the right side of the modal window to log in. After the users log in, they can start creating their projects.



*Figure 1.* Registering a New Account.



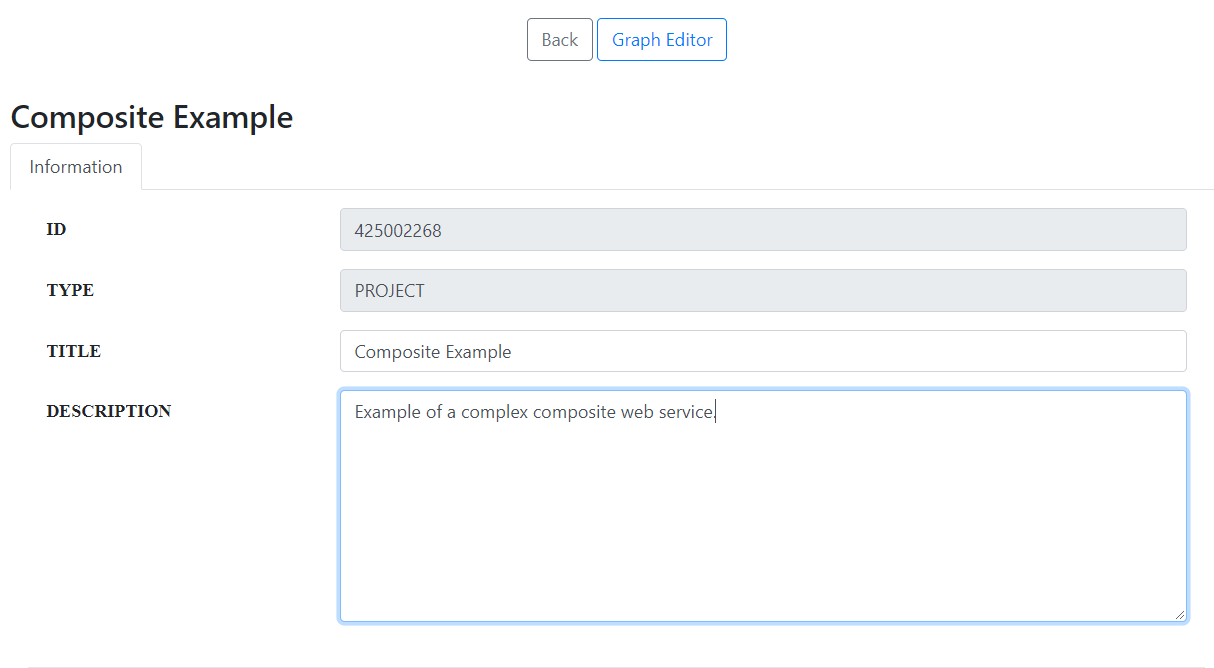
*Figure 2.* Logging into VTCCCS.



*Figure 3.* Successful Login.

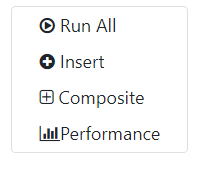
**Project Creation**

Now the user will click on the Projects tab in the navigation bar or click on the start here button to go to the projects page. The user may press the edit button to edit details of the project like the title or description.



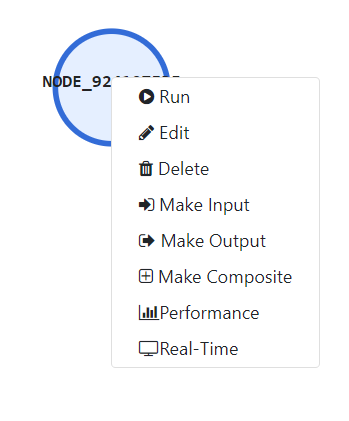
*Figure 4.* Project Details Editor.

Now the user can move on to the graph editor by pressing the graph button on the far right in the graph column of the table. This will take the user to the graph editor of the project. The user can create a node by right-clicking and selecting insert. This will create an empty node. There are other menu options such as Run All, Composite, and Performance. Run All would run all the web services. Composite would allow the user to choose a different project and make that node that composite web service. Performance would display metrics gathered such performance and availability, but more will be covered later.

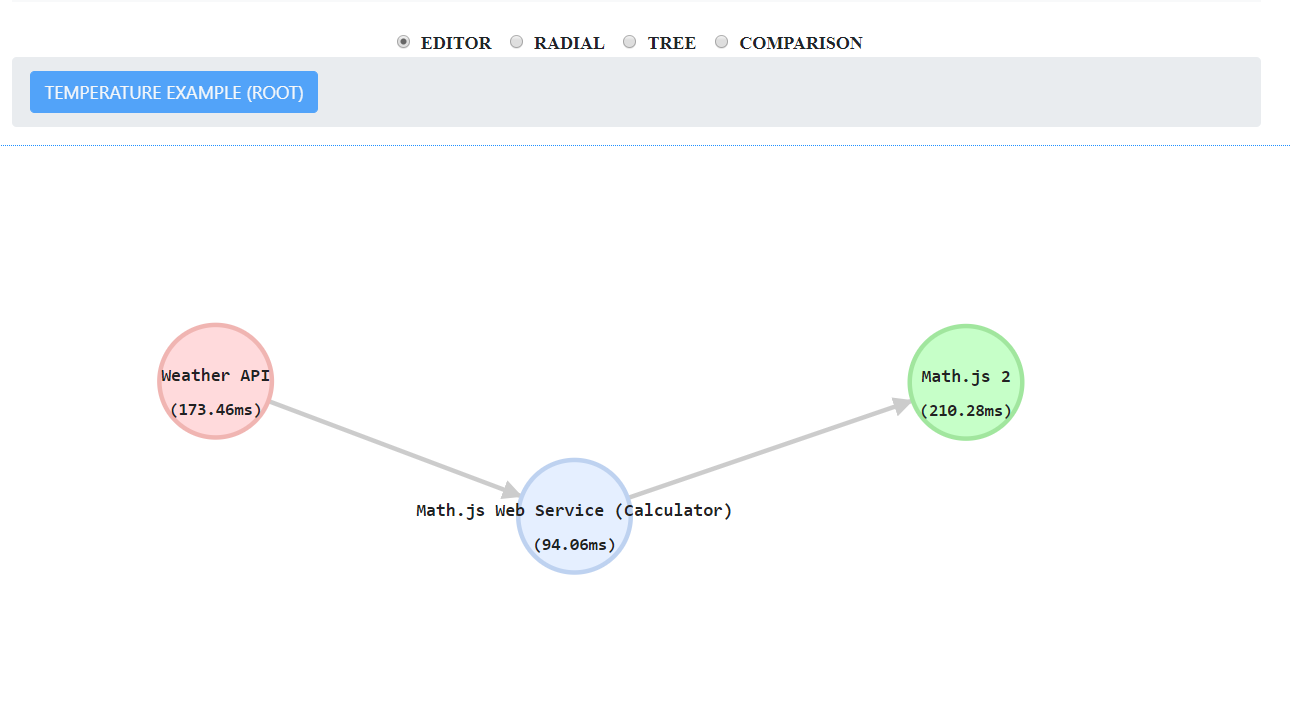


*Figure 5.* Menu Options.

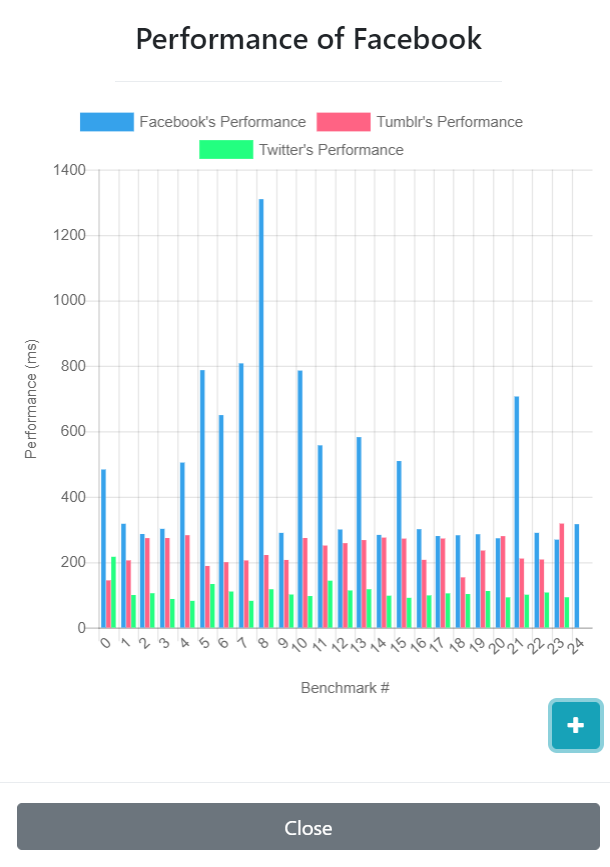
This node can be configured by selecting the node. The border will be thicker to show that you have selected the node. Now if you right-click when the node is selected, there will be more options. Run just runs the individual web service and records the run-time performance. Edit will allow the user to edit the node’s properties and insert a web service. Delete removes the node from the graph editor. Make Input will make the node an Input Node which will pass the input parameters on to the next web service. This is mainly for complex compositions since we need input nodes. Make Output is the result of the current project and it is what this project is sending as output to other web services as inputs. Performance displays the most recent performance metrics. Finally, Real-Time brings up the real-time monitoring window. Also, if the user wants the nodes to be connected, they must click on the node, hold CTRL, drag the mouse to the node the user wants to connect.



*Figure 6.* Selected Node Options.

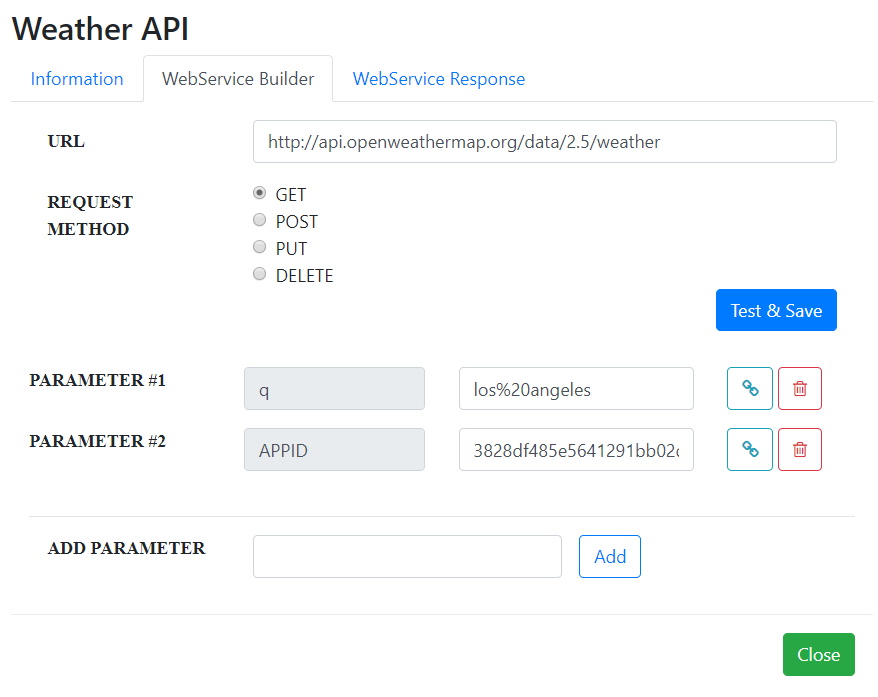


*Figure 7.* Example of Input, Regular, and Output Node Interconnected.

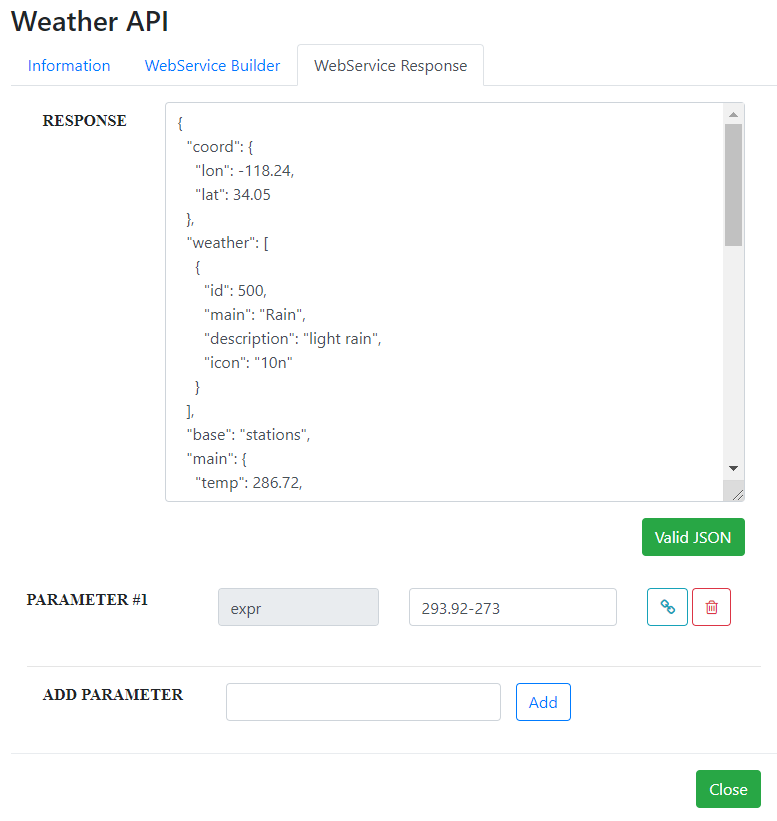


*Figure 8.* Performance Chart of a Web Service.

Now in order to add a web service, the user must set the URL to one of the endpoints of a web API of their choice. In the figure below, this node is using Weather API. Next, the user sets the request method. Now if the web API needs parameters, the key and value must be set properly. For the Weather API, the key is q which stands for query and the value is Los Angeles. If additional parameters are needed, the add parameters textbox can be used to add more parameters. Once the user is done configuring the web service, press the test & save button. This will test to see if the web service is valid. If it is indeed valid, the WebService Response tab will appear, and no error dialogue will pop up. There are also link buttons and delete buttons to the right of each parameter. The link button will allow you to link that parameter to an input that current node has. The only way for a node to see options to link is when another node is giving their output to this node.



*Figure 9.* Example of Linking a Remote API to the Node.

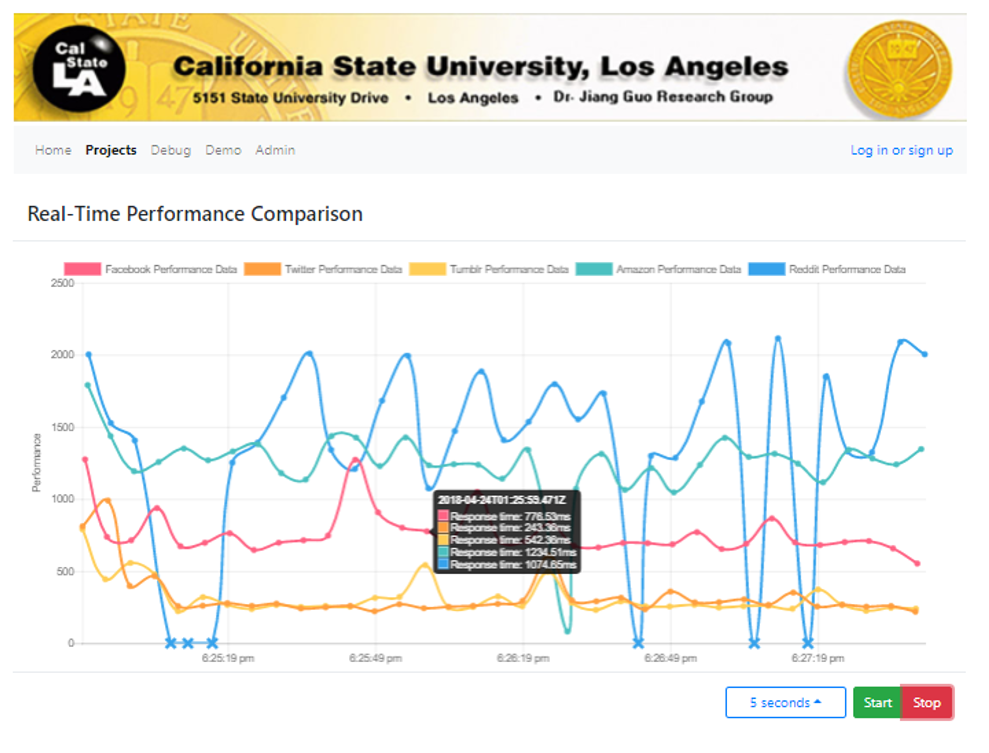


*Figure 10.* Example of a Successful Response from a Web API.

In the WebService Response tab for the Weather API, it returned a JSON file. Now if users want outputs in this node, they must add them manually. It is a similar process to adding parameters for the web API, but the value must be typed. The user must decide what they want to be as outputs.

**Real-Time Performance**

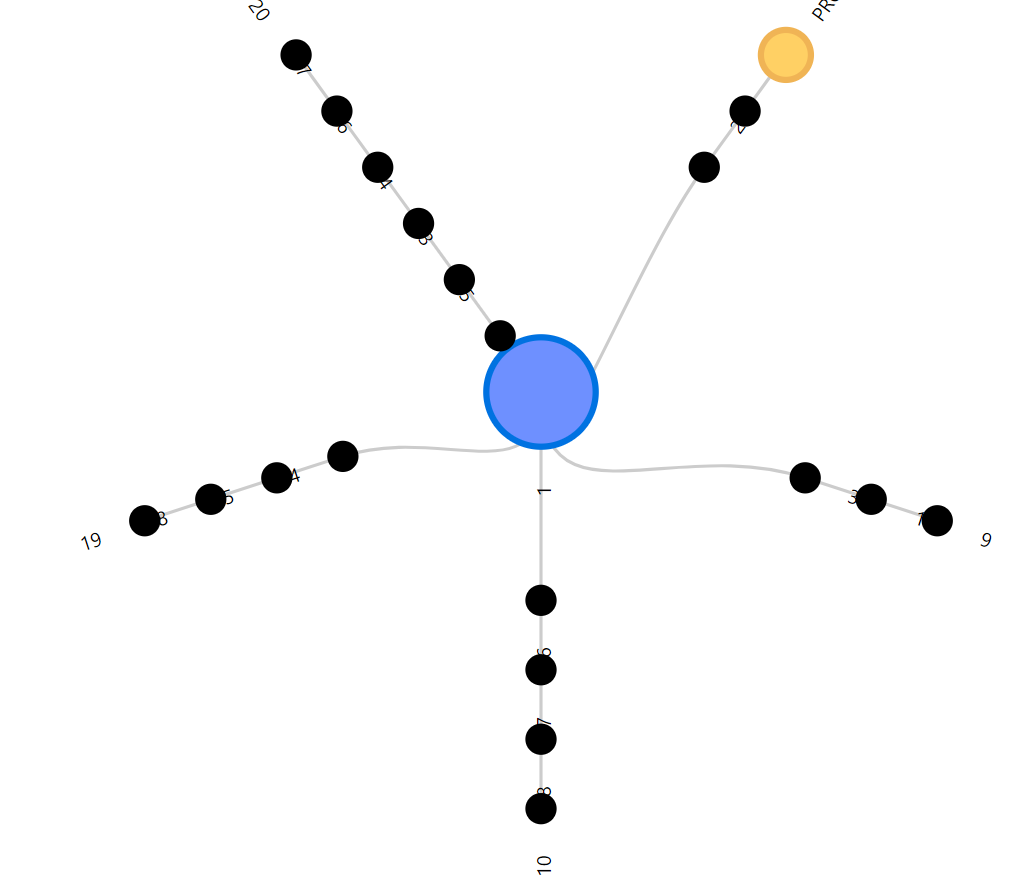
The real-time performance is how this web application collects performance data for web APIs. Without this feature, the user would be manually clicking on run for data. This feature will make the API call every x seconds where x is the chosen interval for this task. This feature also is not restricted to just web APIs. As seen below, the figure shows that this feature can measure response times of websites even though this project’s intended target is web services. There are two different ways of using this feature. The first way is individually testing one web service by right-clicking a node and selecting real-time. The other way is a group comparison by clicking the comparison radio button at the top of the graph editor. In that menu, the user can select which nodes in the current project they want to be monitored together.

****

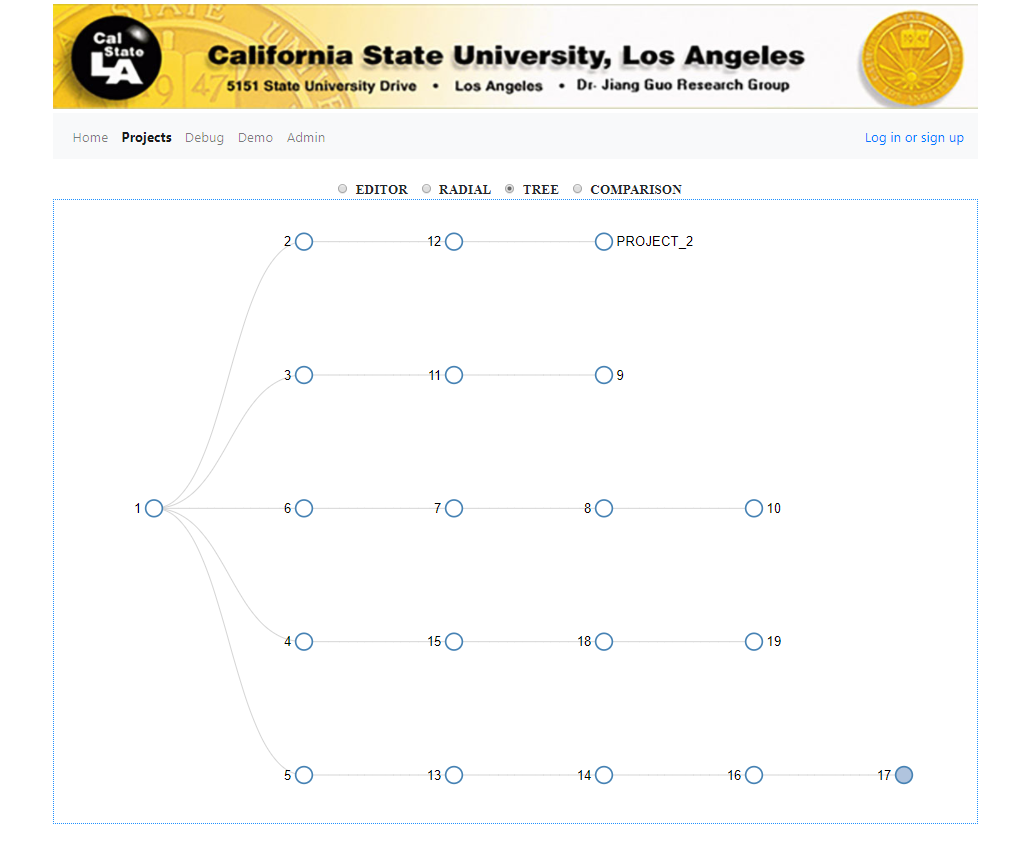
*Figure 11.* The Real-Time Performance Comparison Monitoring 5 Websites.

**Visualization**

Currently, VTCCCS only has two ways to visualize the project. The first one is the Radial View and the second one is the Tree View. Currently, the Radial View and Tree View can display simple compositions, but if the composition gets too complex, the graphs may not give a great visualization. The user can access these views with the radio buttons at the top of the graph editor.



*Figure 12.* Radial View of a Project.



*Figure 13.* Tree View of a Project.