

Beta Release

Submitted by: Divya Shrivastava & Kirubel Tadesse

1.Implementation

1.1 Prototype Overview

For the Beta release, we developed a complete working prototype of our visualization tool for Website performance analysis. A preview of the working prototype is shown in Figure 1.

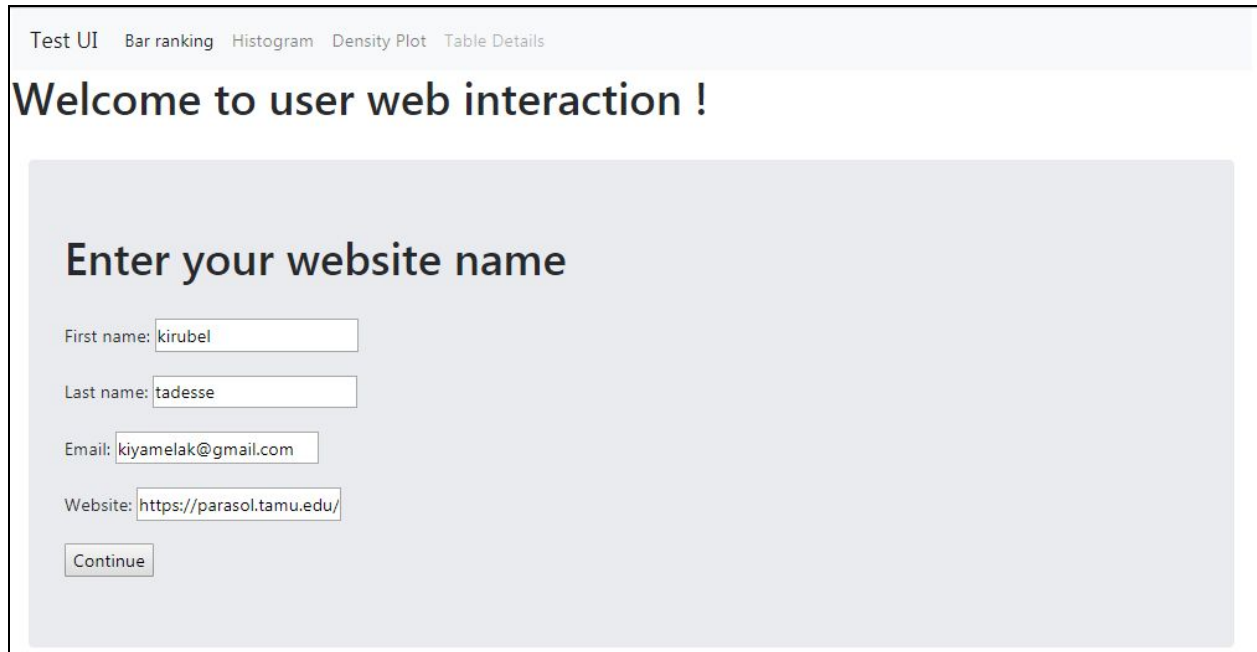
The screenshot shows a web application interface. At the top, there is a navigation bar with five tabs: 'Test UI', 'Bar ranking', 'Histogram', 'Density Plot', and 'Table Details'. Below the navigation bar, a large heading reads 'Welcome to user web interaction !'. Underneath this heading is a light gray rectangular area containing a form titled 'Enter your website name'. The form includes four input fields: 'First name:' with the value 'kirubel', 'Last name:' with the value 'tadesse', 'Email:' with the value 'kiyamelak@gmail.com', and 'Website:' with the value 'https://parasol.tamu.edu/'. A 'Continue' button is located at the bottom left of the form area.

Figure 1: Preview of the Welcome page

Using Webpagetest - a web tool which helps to retrieve the critical performance parameters of a website - we collected the data for the top 100 Alexa websites. The development of the prototype was initiated by automating the collection process using a JavaScript program. Moreover, our script processes the result from Webpagetest and saved it into a CSV file for further analysis. After back-end development, our main task was to integrate the back-end with the UI to complete the prototype.

Different ideas for implementation evolved while developing the prototype. Initially, we created interactive views from our processed data solely through the Bokeh server. While the server is running, the user gets directed to a dashboard, which has various tabs for the purpose of visualization. The Welcome tab, Histogram view tab, Density distribution tab, and Bar ranking tab are the tabs currently available to the user. The user is prompted to enter a website name to test its performance against other Alexa websites and to analyze the statistical relationships

between the website and their performance parameters. Once the user enters a website name, the server calls the back-end script to collect performance parameters of that website from Webpagetest. Next, the data collected from the server is saved in a CSV/JSON format file for further processing to create or update the visualization views. After the views for that website are generated, the user can switch to different views to analyze the website performance.

1.2 Challenges faced

Our aim was to collect data from the server and save it in a CSV/JSON format file to process it for our visualization tasks. Although Bokeh server was very helpful in collecting the data for the website entered by the user for producing the visualization views, saving the data collected to a JSON/ CSV file was a bit of a challenge. For this purpose, we used Django - a high-level Python web framework which is used for producing web applications. Through Django, a form was created on the dashboard to enable the user to enter their credentials and the website name which they wanted to test. The user credentials are not a required field; however, the website name is a required field in this form. The Django model, which uses the power of relational databases to store data served helpful to retrieve the data of the websites from the database table. After entering the required website name from the user, the model helps in retrieving the website parameters from the database, if it is not present already. If the website is present, the user is prompted with a message that the website data is already loaded in the database. After retrieving the data from the model, currently, the user can only view static visualizations. This is because Bokeh allows only static visualizations while running a web application through Django. We can produce interactive visualization solely through Bokeh server, but when we integrate them to run the server on Django, it creates an issue. We are currently finding a way to produce interactive visualizations while running our visualization tool through Django and Bokeh together.

1.3 Progress & Future Improvements

1.3.1 UI Progress

In the Alpha release, we worked on producing visualization views - Histogram view and Density plot in our Website Performance Visualization tool through Bokeh Server as shown in Figure 2.

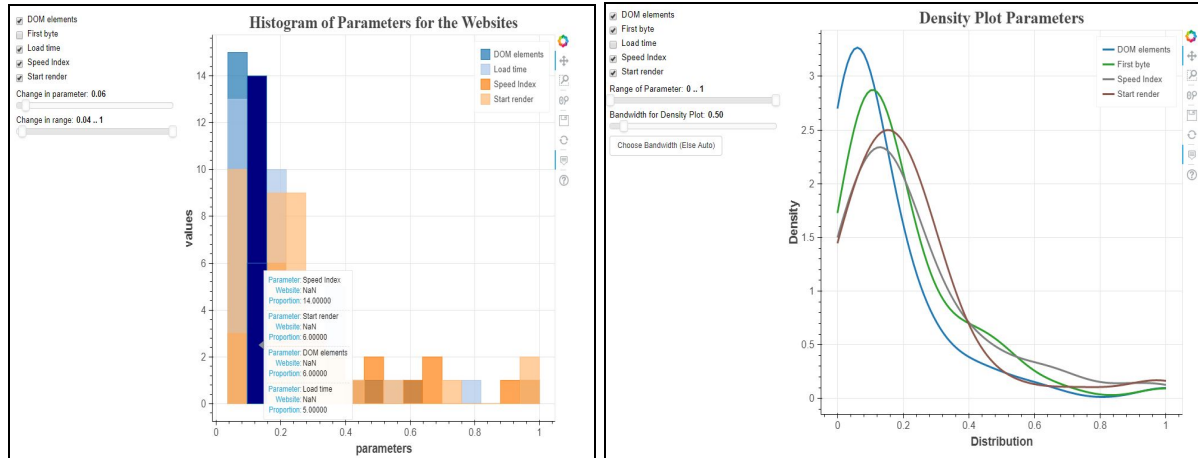


Figure 2: Histogram View and Density view

For Beta release, significant progress has been made to integrate the back-end with the User Interface. In producing additional interactive visualization views, we added a widget to change the values and the parameters for respective views with slide bars. We also added a checkbox to select and de-select the parameters of interest. A widget to adjust the range and width of the plots is also provided to the user. Few of these interactive features were working for Density plot in Alpha release. Currently, all these interactive features are working for Histogram and Density distribution view. A Hover lighting feature is also added to the Histogram view to keep track of the Histogram bar of user's interest. Interactive feature such as Mute glyph and checkboxes were added to the Bar Ranking view. Mute glyph enables the user to click on the legend to mute the visuals of that color and show other views having different colors.

Additionally, an interactive Bar Ranking view is added. This view provides a side-by-side comparison of the website the user would like to test against the average performance of the Alexa websites. This view is very useful to the user to analyze how much improvement is needed to make the performance of the website better than their standard performance for each parameter. We observe interesting results in this view. The visualization shows high values for most of the performance parameters for websites having top rankings in Alexa. We also observed there some critical performance parameters for each website which makes it perform better than the rest. For example - considering facebook.com has the highest ranking among all websites, our tool displays few critical performance parameters due to which this website enjoys the top rank amongst all websites. This information can be very useful to the website owners to understand which the strongest performance parameter of the respective website and which parameters should be improved for better rankings. A screenshot of the Bar Ranking view is shown below in Figure 4.

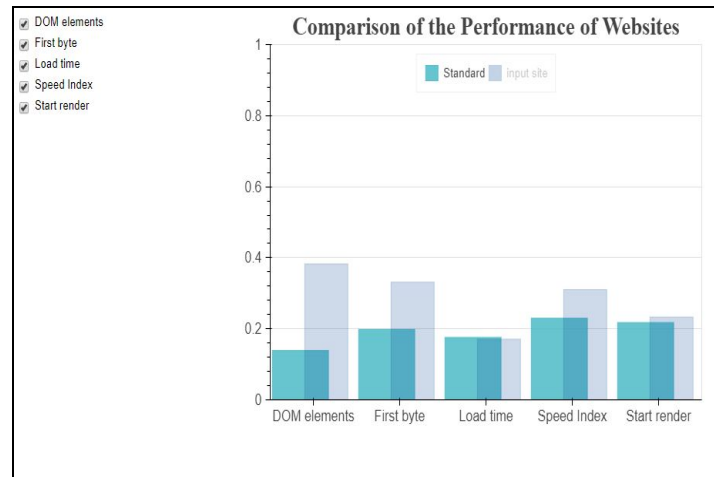


Figure 4: Bar Ranking view: Comparison of Website Performance

There are several interactive features currently working in our visualization tool. The tool allows to hover over the parameters of interests and display their value on the UI. The user can also use the box zoom feature to select a box on the interface and zoom the plot to investigate more in detail. Another useful interaction offered by this tool is the resetting feature which enables the user to reset the view to its original state so that the user can go back to the previous visuals. Further, the visualization tool contains a save functionality where the users can save their plots of interest in their local machine. The save features save the visual shown on the screen to a PNG file in the user's local machine. This feature is useful to the user if the user wants to save interesting plots and results obtained from this tool for their research or study.

1.3.2 Back-end Progress

In the back-end, we have developed the Django-framework to collect input from the user and save the fetched parameter results for the website. Using the Model-View-Control (MVC) framework of Django, the user input directly activate a JavaScript file that collects the parameters from the Webpagetest. This process takes a few seconds until the results are available. Then, the values for each parameter are automatically saved to our model.

Flawlessly, the user is directed to the different views for as soon as the results are saved. Next time, if the user tried to check the performance of the same website he/she will be prompted that the website data are in the database. Based on the result from the Alexa websites, we have created a template to give a performance prediction for each improvement the a each parameter.

1.3.3 Future Work

In the future, we plan to give the users the option to visualize the saved parameter if the website exists in our database or fetch for recent results of the parameters. Moreover, we anticipate completing the integration of Django and Bokeh. By adding additional views such as Table view, users will get a better representation of the collected parameters. Based on our template, we plan to integrate prediction model views to the UI if time permits.