MSAN 695 Group Project

Zainab Danish Andre Duarte Spencer Smith Derek Welborn

***Abstract*—The paper describes how our website, *Nostalgia FM*, works and how a user can navigate it. We use a flask server to dynamically render an HTML webpage that shows the music video of the top song on a user-selected date. This web app is an experiment with responsive design, meaning that it renders nicely on either big and small screens.**

I. Introduction

It is estimated that more than 4 billion mobile devices are currently being used in the world, with this number rapidly increasing [1], [2]. Therefore, it is necessary that modern websites and apps correctly render on both large and small screens. When the original specifications for HTML were determined, and up to several versions later, mobile was never a priority, since no such device existed in the late 1980s/early 1990s. However, using HTML5 and CSS3, it is possible to built such websites without many difficulties.

Additionally, it is very simple nowadays to create and host a dynamic web app using programming languages and common libraries such as Flask [3].

Our goal behind this website is to showcase a simple web application, built by leveraging modern tools, that is also modern and responsive.

II. Material and methods

Nostalgia Station allows the user to select any date from 1959 to 2015 to see what was the top song in the Billboard Top 100 chart that specific day.

The website is a simple combination of only four files:

- *index.html*: a file that contains HTML code for the landing page,
- *main.css*: a file that contains CSS code for the styling of the page,
- *server.py*: a file that runs a flask server in Python, which handles REST requests and renders the result page dynamically based on the user's selection in the landing page,
- *hot100_dict.p*: a file that contains a dictionary of all Billboard Top 100 songs from 1959 to 2015.

These files are available as a *.zip* file upon request to the authors.

III. Results

Figure 1 shows the landing page of the website. The interface is simple, with only some text, a date selection and submit button. This scales well to mobile, and renders nicely on any current smartphone.

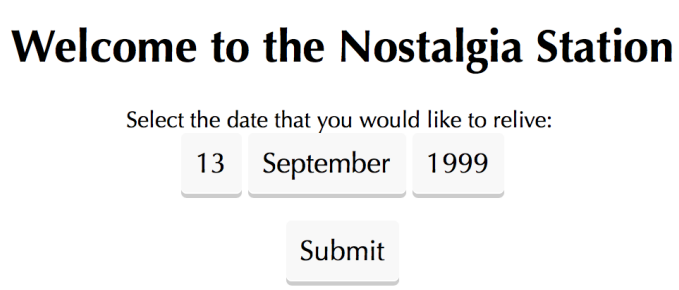

Fig. 1. Landing page

Upon defining a date and clicking on "Submit", we query the dictionary of songs to get the one corresponding to the selected date. With this information, we use BeautifulSoup [4] to fetch the Youtube page relevant to those search terms (song + artist) and return the link to the first video on that page. We then create an iframe tag in HTML [5] to load the corresponding video within the results page and occupy the entire width available for an immersive effect. This is shown in Figure 2.

Fig. 2. Result page

Both the HTML and CSS code pass their respective validators [6], [7].

IV. Google Analytics

Another purpose of this website is to track user activity using Google Analytics. For example, we can see how users interact with our website and maybe see which dates are the most selected (if possible). Google Analytics can help us

keep track of which buttons are being clicked, and how many users come back to use our website.
We can add ads on the sidebars or above/below the video to generate extra revenue.

V. Event tracking

Using Google Analytics, we plan to track events such as button clicks, dropdown selectors, and video clicks. Our objective (goal) is for people to select a date and click on ”Submit” to view the video. Therefore, our objective is reached if users land on the `/process` page. Consequently, our funnel is not very complex: once the user lands on the main `index.html` page, they can optionally select a date, and then click on the ”Submit” button, which is the final goal. Another goal we set up is the idea of going back to try another date after the first view. We can use this as a measure of how engaged the user is with the website. Hence we track whether the user selected a day/month/year combination, whether they clicked on ”Submit”, whether they clicked (hovered) on the video, and whether they clicked on the ”Go back” button to try another date.

VI. Customization

We added custom dimension for date as well as custom metric for date. We wanted to track which dates were most selected. This will only make sense if many users visit our site, which is improbable.

Andre customized the default session timeout to 5 minutes, since this is a reasonable length of time to select a date and watch the corresponding video (see Figure 3). It does not make sense to use the default timeout of 30 minutes, since this is considerably more than landing on the website, selecting a date, and potentially watching the corresponding video.

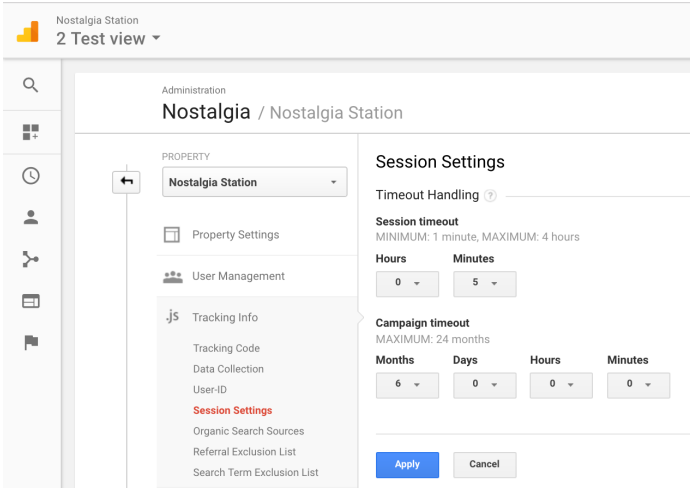

Fig. 3. Andre’s customization: session timeout from 30min to 5min

Zainab added a filter to exclude any hits from Safari in order to determine whether most of the traffic comes from Safari.

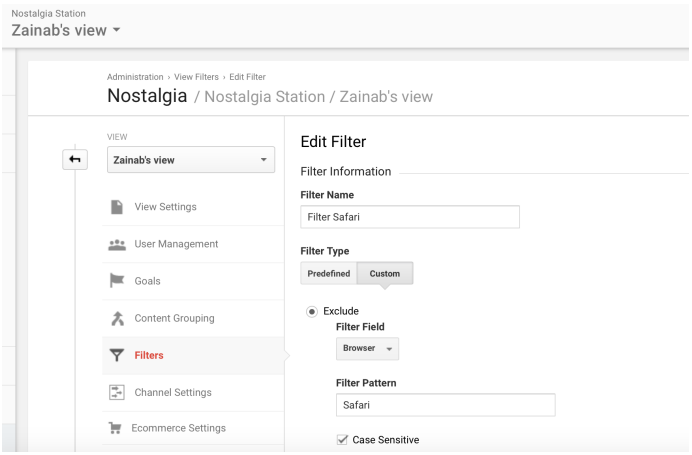

Fig. 4. Zainab’s customization: filter out users browsing through Safari

Derek added a filter to exclude any hits from Kenya in order to see whether it impacted our traffic in any way.

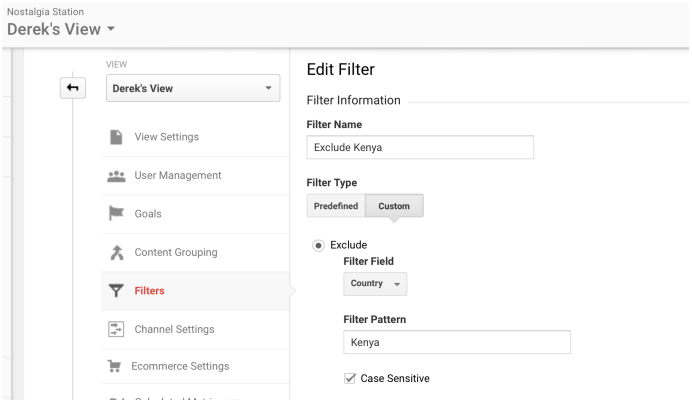

Fig. 5. Derek’s customization: filter out users browsing from Kenya

Spencer added a filter to include only hits from the United States.

VII. Conclusion

Nostalgia Station is a simple proof-of-concept web app that implements modern technologies to create a responsive and dynamic interface that is accessible on big and small screens alike. It is accessible on EC2: http://ec2-52-43-67-18.us-west-2.compute.amazonaws.com.

References

[1] https://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/
[2] https://www.emarketer.com/Article/Mobile-Phone-Smartphone-Usage-Varies-Globally/1014738
[3] http://flask.pocoo.org/
[4] https://www.crummy.com/software/BeautifulSoup/
[5] https://www.w3schools.com/tags/tag_iframe.asp
[6] https://validator.w3.org/
[7] https://jigsaw.w3.org/css-validator/

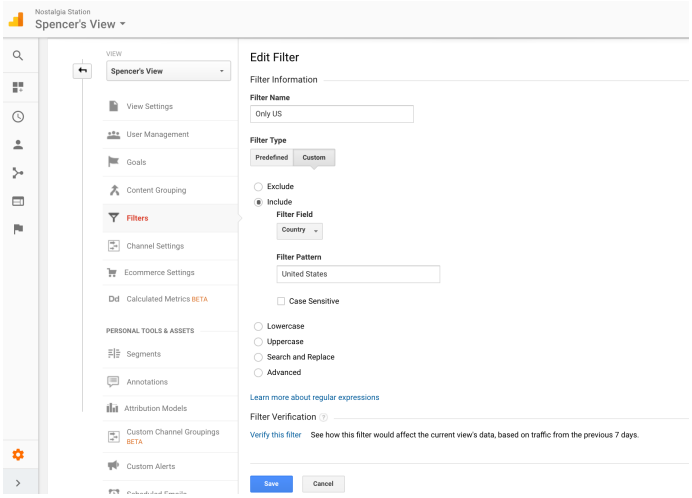

Fig. 6. Spencer's customization: filter out to users browsing from the Unites States