Capstone Report: Playlist

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Faisal Refai

Spring 2022

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Advisor

Nada Basit, Department of Computer Science

Introduction

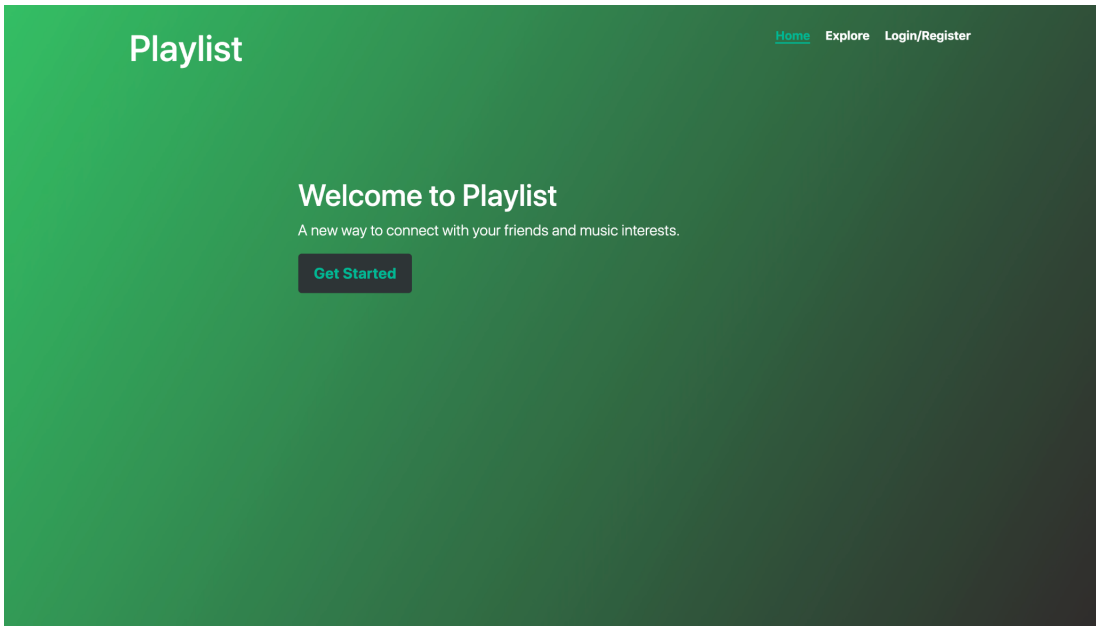
Playlist is a new way to stream music and connect with friends. Like traditional social media applications, Playlist allows users to connect with friends, family, celebrities, and other members of their community. Playlist is novel by only allowing users to have one song associated with their account whether that be self published or by another artist. The motivation for Playlist was the lack of community building within modern music streaming applications. Spotify, Deezer, SoundCloud focus on their ability to scale and deliver media. However, Playlist is focused on emphasizing media which is available on these applications. For example, recently we had the Black Lives Matter Movement. Users would be able to reach out to their following community by changing their account song to a BLM podcast or a song which advances the interest of the movement. Through Playlist, one can connect users with a proper audience to better spread awareness.

Design Decisions

The basic requirements of the project are the app will be built using HTML/CSS, the app will have users, the app will create a playlist based on the users followed, and the app will stream music or provide a plan for streaming.

There was not a lot of time for this project as the whole website needed to be planned, designed, and coded within a span of 8 weeks. As such, I wanted to keep the UI simple. A simple UI would be easier to code to reach the desired deadline and wouldn't detract from the functionality of the website. Having several links and buttons on page can overwhelm the user on how to approach and navigate the website. Instead I wanted a single function on each page with

only the navigation linking the pages.



You will notice that all the pages on the website follow a common theme. A page with minimal text and a gradient/plain background. The gradients are themed after common music streaming applications, the above picture being Spotify. Since this project is just starting out it is important that the users understand what the application is and how it works. That logic fell into the design decision to make the UI easy and approachable.

The next decisions to be made were backend coding and project management. A plethora of languages were available that I have learned from prior classes. While my most comfortable language was Java, I decided to connect the HTML/CSS to the backend using PHP. This was because I had experience using PHP to connect to a database from CS 4750 whereas I have never used Java as a website backend before. Given the short amount of time for the project and the vast amount of resources from CS 4750 that could aid me in challenges, it seemed like a clear decision to use PHP. Furthermore, PHP would be easy to connect to the UVA CS servers already

set up on my account for CS 4640. Using PHP would also solve the issue of hosting and utilizing databases to store user information.

Coding

Coding the website followed a fairly standard website once design decisions were made. I thought about and made rough outlines on paper as to what I wanted the website to look like. I started by making the home page with the navigation bar, background, on page text, and interactive css. Once this page had the design laid out I copied and used it as an outline for the other pages. Using this the general layout of the website was coded fairly quickly. After that, I went page by page and thought about the functionality. I walked through what I as a user would expect to be on the page and tried to create it. I put in placeholder elements for items that would be populated using the backend and then noted down any PHP I would need to connect to the page. Once the UI design was coded and complete, I changed all the pages to PHP. This was very easy without any actual backend as all it entailed was adding a php tag at the top and renaming the file.

Following the UI creation was working on the backend. The first thing that needed to be created was the users. I made the login simple with a database which hashed the password and stored it in the database. It also kept track of the user's favorite song, album, and artist while having an entity database for users followed. This was done through the login/register page with some minor input validation for checking existing accounts, password requirements, and all information filled out.

The next step for coding was to create the following and playlist functionality. Following users was difficult as I wanted it to have a live response as to whether a user was followed or not followed to prevent duplicates and confusion among users. I started by creating the animation for

following and unfollowing using javascript. Then I used Ajax to display the users dynamically and ensured that the follow and unfollow buttons still worked. Lastly, I created two methods to follow and unfollow a user. Utilizing the backend for the following functionality was relatively simple as all I had to utilize was the Follows entity database for the logged in user. Once I ensured that all the following functionality worked together I could move onto the playlist functionality.

The playlist functionality was also relatively simple as we could natural join the users and the follows table to get all the accounts the current user follows and then get the song information from that table. Using a simple loop I could populate the playlist with the same design as the placeholder elements.

I did various edge and case testing on the functionality created and found no large issues with the website. I plan to continue working on the website throughout the next semester as part of STS 4600 in which I will have a beta version for the users to test. I figured any minor coding issues would be reported to me through beta feedback.

Challenges

A challenge I ran into was streaming the music on the website. Unfortunately, all the mainstream applications such as Spotify, SoundCloud, etc. do not allow music streaming using their APIs. This makes sense as people would be able to replicate their main functionality using the company's API. It didn't seem possible to incorporate this functionality into the application without building music streaming from the ground up which would require a tremendous amount of work (copyright, storing all the songs, scaling, etc.). To combat this challenge I found an alternative utilizing the SoundsGood API.

The SoundsGood API is an engine which allows music streaming applications to connect to popular music streaming services. There is a function within the API that allows users to export created playlists. This would allow the main functionality of created playlist based on users followed to still be apparent even without music playing from the website.

The music streaming was the only major challenge which was faced during the course of this project, however with research and a plan of action, a solution was able to be created.

Relation to STS 4500

My STS 4500 prospectus was about this capstone project. In that paper I outline the work I would be doing on Playlist and the idea behind the project but the research was focused around a new trending technology, machine learning. In the paper I outline how from a developer standpoint machine learning seems like a great inclusion. However, incorporating machine learning into applications may not always be useful. Research shows that machine learning might fail to understand its users as "algorithms are not only constituted by rational procedures, but also involve institutions, people in a multitude of roles, and various intersecting contexts" (Storms 2019).

Future Work

If given more time and resources I would try and create a prototype custom music streaming application which could connect to my application. By downloading a few songs and testing it privately (to avoid copyright infringement) I could create a proof of concept for how streaming on Playlist might work.

I think building a music streaming application from scratch would be fun and interesting. It would allow me to culminate more knowledge from many classes into a realistic project. I

would approach this by thinking about scalability, incorporating AWS, and also shifting the website to a framework such as Django or Spring.

Conclusion

Overall, this project was really enjoyable to work on and I feel as though it did culminate my learning here as UVA. I was able to incorporate several elements learned in my CS classes such as databases, PHP, website design, data structures, etc. These all played a role in some functionality throughout the website and to code a product which utilized these features was insightful into why the UVA CS curriculum is structured the way it is. Additionally, through STS 4500, I was able to incorporate some of my more interpersonal learnings. I researched human behaviors and interactions with technical products to understand how I could make better design decisions. I was excited to see that the humanities aspect of my UVA education could also be brought into this project to make it better. I hope to continue working on this project throughout next semester.

Link to website: https://cs4640.cs.virginia.edu/ftr3vf/capstone/index.php