Trysten Giorgione
Assignment 6: Self Assessment
Senior Design
4/14/2025

Individual Assessment

Part A – Personal Contributions and Learning

For this project, I worked independently and served as the sole contributor to the design, development, and deployment of the Spotify Jam web application. The system aims to improve Spotify's "Jam" feature by introducing a fair queueing system that rotates between users, rather than relying on a chronological song order. I implemented full Spotify OAuth authentication, allowing hosts to log in and manage sessions securely. I also developed the functionality for hosts to control playback, enforce fallback playlists when queues are empty, and save session songs as Spotify playlists. The application includes both host and guest views, with guests able to see rotation order, search for songs, and queue them seamlessly—even without a Spotify token. These features were all integrated into a working full-stack web application.

I also built upon skills identified in my initial assessment from last semester. At the time, I had a basic understanding of web development but limited experience in building full-stack systems. Throughout this project, I pushed myself to bridge that gap. I independently handled the architecture, starting from backend session handling and queue logic in Node.js, all the way to developing a responsive frontend using React. I built secure API routes to support the Spotify Web API and developed a consistent pattern for frontend-backend communication using Axios. Each implementation, from authentication handling to session management, required me to extend my skills and think critically about state synchronization, error handling, and user experience.

Implementation and Reflections

One of my biggest technical takeaways from this project was mastering full-stack development in a real-world scenario. I became confident in working with Node.js for building RESTful APIs, and React for building dynamic client-side views. I also gained experience handling OAuth tokens, persisting session data, and integrating third-party APIs—especially in terms of token management and asynchronous communication across components. Implementing features like queue rotation and fallback playlists required me to plan backend logic carefully, while updating the UI in real-time required understanding of React state and component lifecycles. These experiences developed my competencies in modern web frameworks, user session handling, and complex integration testing.

Some notable successes include implementing the fair queueing system, which prevents any one user from dominating the session, building Spotify login and playback control flows, and successfully saving session tracks as a Spotify playlist. These features were fully functional by the end of development. On the flip side, I encountered significant challenges in handling authentication, especially finding ways to allow guests to interact with the queue without violating Spotify's token-based access limitations. Handling these edge cases—while ensuring user privacy and data integrity—was difficult and required several iterations. Debugging cross-device interactions and testing on local networks also posed hurdles. Still, overcoming these obstacles gave me a deeper appreciation for designing scalable and user-friendly systems.

Part B – Group Dynamics and Self-Reflection

Although this project was originally designed to be a group effort, I completed it independently. That means that every technical and design decision—from backend queue logic to frontend layout—was handled by me. I effectively had to manage not only the workload but also the project planning, testing, and iteration cycles typically distributed across a team. Despite the demands, the project turned out to be a strong proof-of-concept for a real-time collaborative Spotify session system, and I'm proud of the polished result I was able to deliver on my own.

Working solo helped me appreciate both the value and the challenges of group work. I learned that while group work can significantly improve output by distributing tasks and leveraging individual strengths, working alone allows for clear direction and consistent vision. There's no need to compromise on ideas or sync schedules when you're the only one responsible for implementation. However, it also made me realize how much more quickly features could have been developed with even one or two additional developers. In the end, I feel a strong sense of ownership over this project—especially because I had limited experience with the tools going in, and yet I was able to design and build a complete application that integrated complex Spotify features in a clean and user-friendly way.