Spring Self-Assessment Essay

Seamus Collins

Senior Design 2

Part A:

My primary contribution to our Spotify genre-based music recommendation system was developing the AWS integration and implementing much of the front-end design. I built the DynamoDBHandler class that manages all interactions with AWS DynamoDB, configuring multiple tables with appropriate access patterns and implementing efficient CRUD operations. This work was built directly upon the cloud computing skills I am currently learning but required me to deepen my understanding of AWS services and NoSQL database design. On the front-end, I designed the user interface and implemented key features including playlist generation and comparison functionality. This stretched my web development abilities as I worked to create intuitive visualizations of complex music taste data. The most significant technical challenge I faced was adapting to Spotify's API changes in November 2024, which required substantial refactoring of our authentication flow and data retrieval methods.

Throughout this project, I developed stronger competencies in full-stack development, particularly in connecting cloud infrastructure with responsive front-end interfaces. I learned valuable lessons about API integration, specifically how to design systems that can adapt to third-party service changes while maintaining functionality. My successes included implementing a robust error handling system that gracefully manages failures in AWS services and Spotify API calls, and creating an efficient caching strategy for artist genres that significantly reduced our API call volume. My primary obstacle was working with a Python back-end and HTML front-end that required embedding JavaScript code directly in the HTML files. This architectural decision created maintenance challenges and complicated the separation of concerns in our codebase. In retrospect, selecting a different language or framework combination could have avoided the need for embedding code in HTML files and would have created cleaner, more maintainable architecture.

Part B:

Our team successfully built a comprehensive web application that analyzes Spotify listening habits, creates personalized genre distributions, and generates custom playlists using machine learning clustering techniques. The system effectively groups users with

similar music preferences and leverages these patterns to recommend new music. We integrated with Spotify's OAuth and API for authentication and data retrieval, implemented real-time data analysis and visualization, and created a cloud-based infrastructure capable of handling user data securely and efficiently. The final product delivers a seamless user experience from login through playlist generation, with intuitive visualizations that help users understand their own music preferences.

Working in a team environment taught me valuable lessons about communication, task distribution, and accountability in software development. Our successful aspects of teamwork included establishing clear roles based on individual strengths, maintaining consistent communication through weekly meetings and daily check-ins, and implementing a structured git workflow that prevented conflicts. As a team, we faced significant challenges when Spotify changed their API capabilities in November 2024, requiring substantial refactoring of our code. Additionally, our choice of Render as a web hosting service proved suboptimal for our demo, causing performance issues and deployment complications. These obstacles taught us valuable lessons about selecting appropriate technologies and planning for third-party service changes. I found my contribution level to be consistent across our team, with each member bringing valuable skills to the project. Noah deserves recognition for his work as our primary machine learning developer who built the core clustering algorithm. Justin made significant contributions by cleaning and preparing data and assisting in machine learning development. Chris handled the remaining development tasks that complemented my AWS and front-end work. The diverse expertise of our team members allowed us to create a comprehensive solution that none of us could have built individually. This experience reinforced that effective software teams require both technical skill and strong collaboration practices, with regular feedback loops being essential to maintaining project momentum.