D424 – Software Engineering

Task 4

Capstone Proposal Project Name: Music Catalog Web Application (MCWA)

Student Name: Randy Elias Garcia

Table of Contents

Deployment Documentation

Justification for Selecting Vercel as Cloud Service Provider

When it came to selecting a cloud service provider for deploying the Music Catalog Web Application, Vercel seemed like the most suitable choice for several reasons:

1. **Cost-Effectiveness:** Vercel offers a generous free tier and competitive pricing for additional features. For an independent musician's platform like this application, this cost-effectiveness allows for managing expenses while leveraging robust deployment infrastructure.
2. **Serverless Functions:** Vercel supports serverless functions to handle backend logic, such as payment processing with Stripe and content fetching from AWS, without managing dedicated server infrastructure. This capability aligns perfectly with the architecture of this application.
3. **Ease of Integration with Frontend Frameworks:** Vercel is optimized for frontend frameworks, particularly those built with React.js, Next.js, and similar technologies. Given that this application is a React-based platform, and Vercel offers seamless and quick deployments, it is an ideal choice.
4. **Scalability:** Vercel's infrastructure is designed to handle high traffic loads, automatically scaling the application as needed. This scalability is crucial for the application to accommodate potential growth in user base and traffic without compromising performance.

Explanation of Container Usage

For this project, I chose not to implement container images for the following reasons:

1. **Application Architecture:** The Music Catalog Web Application is primarily a frontend application with serverless backend functions. Vercel is designed to support this type of architecture directly, without the need for containerization. Serverless functions provide the backend capabilities, making containers unnecessary for this particular use case.
2. **Deployment Simplicity:** Using Vercel's built-in deployment features simplifies the deployment process. Containers would add an additional layer of complexity, potentially requiring management of containers, image creation, and maintenance.
3. **Frontend Optimization:** Vercel is strong in optimizing frontend deployments. The Music Catalog Web App benefits from Vercel's ability to optimize and serve static assets efficiently. Containers are more suited for applications that require complex backend services, which is not the case for this application.

By not using containers, I was able to streamline the deployment process and leverage Vercel's strengths to deliver a well performing and scalable application tailored to the needs of the client.