**Briefly describe the artifact. What is it? When was it created?**

The artifact I chose for the database enhancement is a Trivia app. The trivia application is an app built with HTML/CSS/Javascript. This trivia app was designed and built in a coding bootcamp course I completed in 2022. The app is a quiz that gives the user ten random questions regarding music albums, the user then has to select one of four multiple choices to answer which artist the album belongs to. A total score is given to the user at the end of the quiz, and the user can restart it after completion.

**Justify the inclusion of the artifact in your ePortfolio. Why did you select this item? What specific components of the artifact showcase your skills and abilities in software development? How was the artifact improved?**

I chose this artifact for my ePortfolio because it demonstrates my ability to develop a full-stack application with both a front-end interface and a back-end powered by a MySQL database. This Trivia App integrates multiple areas of software development, including database design, server-side programming, and client-side rendering of dynamic content. It reflects my proficiency in developing a structured, data-driven application, from setting up the database schema to handling user interactions in a browser-based interface. The artifact improved from having all the questions stored in a single array of objects to having all the questions stored in a MySQL database.

**Did you meet the course outcomes you planned to meet with this enhancement in Module One?**

The course outcome I planned to meet with this enhancement was: “Design and evaluate computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solution while managing the trade-offs involved in design choices.” I accomplished this outcome by designing the database schema to store quiz questions efficiently. I used structured SQL queries to retrieve data dynamically based on user input. This highlights my understanding of algorithmic efficiency when dealing with relational data, ensuring that database operations were optimized for performance. I made trade-offs between various approaches for data handling. For example, I opted for a relational database over storing data in a JSON file for scalability, despite the added complexity of setting up a MySQL server. This decision allowed for better query performance.

**Do you have any updates to your outcome-coverage plans?**

I have covered every outcome I planned to meet which includes:

Demonstrating an ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals. This project demonstrates how innovative tools like Node.js, MySQL, and Fetch API can be used to accomplish industry-specific goals, such as building responsive, scalable web applications. It also shows how important connecting various components such as the client, server, and database, in an efficient and effective manner, ensuring the solution is both functional and user-friendly.

I also accomplished the course outcome of designing, developing, and delivering professional-quality oral, written, and visual communications that are coherent, technically sound, and appropriately adapted to specific audiences and contexts.

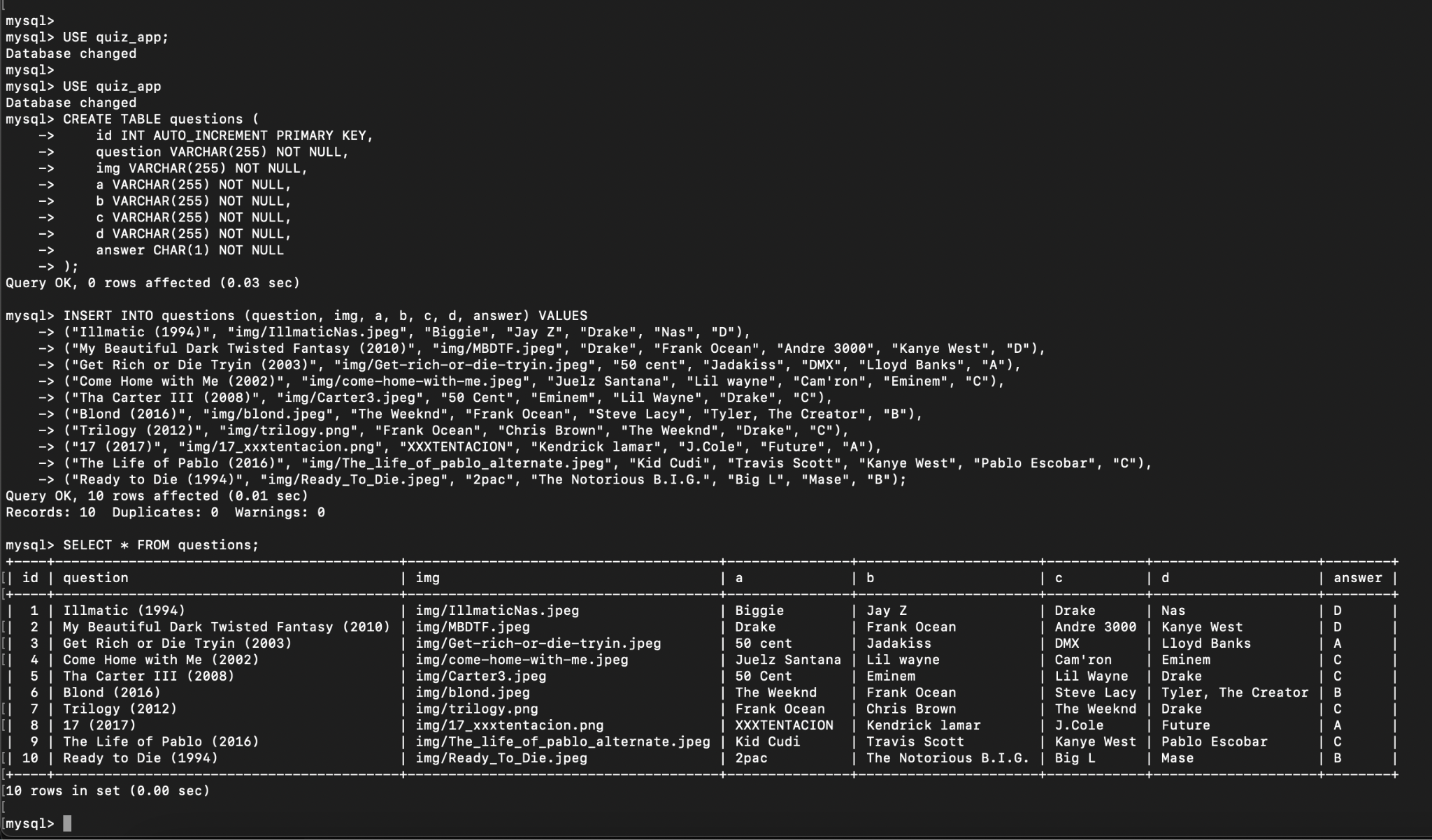
**Reflect on the process of enhancing and modifying the artifact. What did you learn as you were creating it and improving it? What challenges did you face?**

I utilized Express.js for the back-end server and MySQL as the database to store quiz questions, showing my skills in full-stack development. I implemented asynchronous fetch requests to ensure the app could dynamically retrieve questions from the server without reloading the page, which enhances user experience—a key industry goal in web development.

Additionally, I applied CORS techniques to handle security policies, ensuring that data from the server could be accessed by the client-side code, reflecting my understanding of real-world security practices. By combining front-end JavaScript with database-driven content, I showcased how modern computing practices can be used to create a seamless, interactive experience that delivers value by allowing users to engage with dynamically generated quiz content.

While enhancing and modifying the database-backed Trivia App artifact, I learned a lot about integrating front-end functionality with a back-end database and how different components communicate to create a dynamic, interactive user experience.

One of the key challenges was managing data flow between the server, database, and client-side code, particularly with the implementation of CORS and fetching data asynchronously. Setting up a MySQL database and ensuring the right data is retrieved for the quiz questions was also tricky, as it required understanding SQL queries and ensuring the database was structured correctly.



Another challenge was ensuring that the quiz app remained responsive and interactive, even when fetching data from the server, and handling issues such as missing or incorrect data. This required testing and troubleshooting server responses, as well as improving the user interface to handle errors gracefully.



The process taught me valuable lessons in managing both front-end and back-end development, handling real-world constraints like security policies, and refining code to ensure scalability and performance.

