Lab1 Report

To start, I set up the server with the necessary packages and dependencies, including dotenv for environment variables, path for file paths, and mongoose for database operations. I also defined the port for the server to run on.

Next, I defined the routes for the API, including endpoints for showing all albums, retrieving a specific album by ID, creating a new album, updating an album, and deleting an album. I also added a route for retrieving a specific album by title.

To interact with the database, I created a model for albums using mongoose, which defines the schema for the data and provides methods for database operations.

In the client-side code, I used HTML, CSS, and JavaScript to create the user interface. I created a table to display the albums and added buttons for each album to show details, update, or delete the album. I also added a form for adding a new album.

One problem I faced was with updating an album. I had to make sure that the updated data was sent to the server in the correct format and that the correct album was updated in the database. I solved this by using the album ID to find the correct album in the database and updating it with the new data.

Overall, I was successful in creating a fully functional web application for managing albums, with a user-friendly interface and efficient database operations.