Högskolan Kristianstad
291 88 Kristianstad
044 250 30 00
www.hkr.se

Fullstack Project DA219A
Spring Term 2023

Lab 1

Theme: Integrate Mongodb and Nodejs

Author: Wibecka Oliver

Högskolan Kristianstad
291 88 Kristianstad
044 250 30 00
www.hkr.se

Task:

Implement CRUD operations to a REST api for music albums. You should be able to add, update, delete and see details for each album. All albums should also be visible when logging into the home page. Albums should exist in a Mongodb database and accessed through a server where data is then fetched for access on the frontend.

Implementation:

For better organization I divided my code into several folders, with all frontend code within a separate part of the folder structure. I also utilized the Model-view-controller or MVC architectural structure within my backend environment.

Backend

The index.js file acts as the main server, which then connects to the database, starts the server on the correct port via the .env file and links to the routes directed within the albumRoutes.js file within the router folder. Within the models folder I have the schema for the album, which is used when adding and updating information into the database via mongoose. Then, within the controllers folder, the album controller takes care of the request data and alters the database accordingly.

Frontend

The frontend folder is divided into the main index.html file, a corresponding css stylesheet, and then two separate folders for the images I wanted to use and the js files needed. The first js file, fetchData.js does the work of connecting to the routes set up by the server and getting the desired information to the client side of the program. The main.js file is linked to the html file, and displays the data fetched in the appropriate way. It also collects and uses relevant user input to be sent when accessing and fetching information for each specific CRUD operation performed.

Struggles:

While much of the lab was straightforward, I did struggle a bit trying to configure a modal to pop up when confirming if a user wanted to delete an entry or not. It was tricky to figure out how to send information through the html components to track the exact item the user wanted to delete. I think if I would refactor the code, I may attempt to do it in a more straightforward way such as through a form rather than the getAttribute method.