Kristianstad University
SE-291 88 Kristianstad
Sweden
+46 44 250 30 00
www.hkr.se

Lab 1&2

DA219A VT23 Full Stack Development

Stephan Lourentz Hek

Introduction

This report gives a summary of the work done for Lab1&2 of the course Full Stack Development. As Lab 2 is a continuation of Lab 1 they are both combined in this report. An outline is given of the requirements followed by the solutions implented. The report is concluded with a small section on the difficulties encountered.

Task

The goal of the first lab was to create a simple website that utilised *MongoDB* as a database system. The website needed to show information about music albums. The data needed to be visualised as an *index.html* file with the correct layout and framework so that the user could change the album collection with CRUD commands.

In lab 2 this simple website needed to be uploaded using *Render*.

Technical solution

Step 1 – MongoDB setup

In the first step a mongoDB account was created and a database was created for the website. Some test records were created using the MongoDB website features. Finally the IP connection was setup so that this database could be accessed. Given the database was a test project it was decided to make it open for all IP addresses.

Step 2 – Create Model

In this step a model file was created describing the format of an album; *Album.js*. This file specifies what kind of data each album should have and in what format the data is stored.

Step 3– Make MongoDB connection and setup server

In this step the basic code was made to setup the server using *express*; *index.js*. An env file was created that contained the port number and also the url to the *mongoDB*

database created in step 1. This last information was used to setup a connection with the database. This was done with *mongoose*.

Step 4– Add Crud operations

Next the crud operations were added to the index.js file. This contained *get* (view collections) *post* (add and edit collections) and *delete* (remove collection from database). Given it was a simple application it was decided to code everything in the express app endpoints.

Step 5– Connect CRUD operations to html page

In this step the CRUD operations were implemented in the html page. A basic html page was created containing input fields and buttons; index.html. Additionaly a javascript file, *client.js*, was made that contained functions to connect the CRUD operations with the html page. For each CRUD operation a specific function was created in the *client.js* file.

Step 6– Test the application and upload to Github

In this step tall features were tested and adjustmetns were made where neccesary. After this the project was uploaded on github.

git@github.com:Stephan0027/lab1_fullstack.git

Step 7– Upload the file using Render

In this step the application was uploaded to *Render* so that it can be tested by other people. All the steps were followed from the Lab in order to publish the website via *Render*. First an account was created on the Render webpage and this was connected to my Girhub account. Then the relevant github project was selected and the environment variables were added via the webpage functionality.

https://lab2-fullstack-link-test.onrender.com

git@github.com:Stephan0027/Lab2_fullstack.git

Difficulties

This was the first time using MongoDB and Render. After some confusion I managed to code alle the functions using all the course information. The initial confusion was the many seperate programs to setup MongoDB; the atlas website, MongoDB compass and setting up the IP permissions. Given everything was already online at the start the preperation was more cumbersome and delicate compared to working with SQL.

I also could not manage to install the *Render CLI* on my ubuntu pc. I got an error stating the download path was invalid. After several attempts I decided to disregard the CLI feature.