IMY 220
Assignment 6

Advanced Docker

General Instructions

- This assignment must be completed and submitted by the due date which is available on ClickUP.
- This assignment is a take-home assignment and may take one or two days to complete.
- This assignment should be completed on your own, but you may come to the tutorial sessions or email the assistant lecturer if you need further assistance.
- No late / email submissions will be accepted.

This assignment focuses on **Docker and Docker Compose**. You will be required to create your Docker Compose Stack, which is composed of a simple React project served via Express and a MongoDB instance.

Part 1 - React & ExpressJS Dockerfile

Create a simple **React app** using webpack, babel and ExpressJS as taught in the lecture 19 videos (NodeJs & React).

You are more than welcome to copy and paste in a different project that you have already set up from another practical, assignment or your semester project, just make sure that you have a working project, and that you only have what is required from you for this assignment (i.e., please remove any extra files when copying over).

Make use of a frontend/ and backend/ folder like in your project, as you will be fetching data from your MongoDB instance.

Create a **Dockerfile** for your React application. The Dockerfile should be used to build and run the React app.

Part 2 - Setting up Docker Compose

Create a **Docker Compose** configuration that should do the following:

- Run and handle the running of your React app. **All environmental variables and configuration options should be inside the Docker Compose file**.
- Create a MongoDB instance that uses the *mongo* image from Docker Hub.

    - **All configuration for the MongoDB instance should be handled inside of the Docker Compose file.**
    - You do **not** need to provide any authentication for your MongoDB instance, meaning you do **not** need to set a username and password.
    - You will need to set up **volumes** and **networks** as required.
    - *There are lots of examples on how to create a MongoDB instance in Docker Compose on the internet.*
- Your React app depends on your MongoDB instance, **set the configuration as required**.

Part 3 - Creating API Routes

Download the ***posts.json*** file and upload it to your MongoDB instance. You will manipulate this data in your MongoDB instance.

Create **API routes** on the server / backend section of your React app that does the following:

- **Fetch** all posts from the MongoDB instance.
- **Delete** a post from the MongoDB instance.
- **Update** a post in the MongoDB instance.

How you achieve this is up to you. You are required to use the ***mongodb*** package from NPM and use your MongoDB instance that you created in your Docker Compose file.

Part 4 - Using the API Routes

Once you have created the API routes, you should **use them in your React app**. You should make use of components to achieve this assignment. The React app should do the following:

- **Display all posts**. *You should create a component to display an individual post and go through the list.*
- **A way to delete a specific post**. *This should reflect in the MongoDB and the UI as well.*
- **A way to update any aspect of a post (except for the post ID)**. *These updates should reflect in the MongoDB instance as well as the UI.*

All these actions should make use of the MongoDB instance that you are hosting in your Docker Compose and your API routes you created.

Bonus: TailwindCSS Styling

As a bonus, import TailwindCSS into your project and use it to style your React app. The design should still be readable / usable. You should style the text (font-weight, font-colour, etc.), padding / margin, background colour and anything else that seems appropriate. You may make use of the default tailwind fonts, colours, etc. No default HTML/CSS styling may be visible to get the marks.

Submission Instructions

Place these in a folder named `A6_u12345678` where `12345678` is your student number.

- Submit *all* the files required for this assignment to work (including the config files for babel and webpack) **except** for your node_modules folder and its contents.
- Include any commands that you used to run your Docker Compose stack.
- Don’t forget to upload all files to run it with Docker.
- **Do not submit your node_modules folder. This will result in -10% from your assignment mark.**

Zip the **folder** and upload this to ClickUP in the relevant submission slot before the deadline.