ACIT-3695 Group 3
Notes API
Reza Banitaba
Christian Malit
Jasbir Singh

Week 11

For this week we began by compartmentalizing our code into smaller modules to ensure that all code is easily accessible and easier to manage as we add more features. Christian did so by separating the schema, database and cron-job modules to their own independent file, which he then linked to ensure all modules can communicate with each other.

Also, since graphql-yoga was an npm package that was not being updated regularly, we decided to upgrade our GraphQL implementation, and therefore we upgraded our application to use Apollo server, which is an up-to-date library.

Our first feature was one we did in class which was the image upload support via Cloudinary, which allows us to upload our graphql imported images to the cloud. We can then use the uploaded pictures later on for further features that where we can utilize it. Reza was able to set-up the cloudinary database, and modified our schema to utilize the feature.

Second, A feature we wanted was the ability to export our Notes database into a useful file-format, in this case CSV. This functionality allows us to more easily access and view the data within our MongoDB database. Christian utilized FileSystem API (FS), and Json2Csv modules to parse our data and translate them into a csv file. The file export can be triggered by the saveNotesTrigger query, albeit this might not work as expected on the deployed app. It is therefore recommended to try this in a local environment.

Finally we used DigitalOcean App Platform, a Platform-as-a-Service (PaaS) which allows developers to publish code directly to DigitalOcean servers without worrying about the underlying infrastructure. App Platform automatically analyzed our GitHub repository, created the artifacts (in the form of containers), and published our application to the cloud.

The deployed version of the application can be accessed here:

https://acit-3695-notes-api-8st3p.ondigitalocean.app/