Cloud Technologies Oblig 1 Report

Adrian Nysted Riise, adriannr@stud.ntnu.no, 538508

Kristian Wobbes, kristwob@stud.ntnu.no, 539682

URL to app: https://course-registration-clouds.herokuapp.com/

How our system works:

- Created with MongoDB, Express, Node and React.
- Project is separated in two folders, ‘Client’ and ‘Server’
- Client and Server are hosted on the same Heroku app, but as two separate instances. (“start”: “node server/index.js”, “build”: “cd client && npm install && npm run build”)
- Client runs React and fetches data from the server using Axios.
- Server talks to MongoDB to fetch and retrieve data.
- Mongoose schema is defined in ../server/api/studentModel

The application can perform basic CRUD operations such as GET, POST, PUT and DELETE to access and update the information stored on the database.

Timestamps for latency testing

Desktop/WiFi	
t0	1647958149849
t1	1647958150124
t2	1647958150222
t3	1647958150175

Mobile/WiFi	
t0	1647959399799
t1	1647959400106
t2	1647959400202
t3	1647959400210

Mobile/4G	
t0	1647959539774
t1	1647959540261
t2	1647959540353
t3	1647959540387

Results	
Desktop/WiFi	228 ms
Mobile/WiFi	315 ms
Mobile/4G	521 ms

Conclusion of latency results

To conclude the results of the latency testing we noticed that it takes a longer time to reach the MongoDB database because it is located on another server, the communication between the client and server is quicker because they are both hosted on the same location. The connection between the server and the frontend is rather quick because it sends a file instead of a rendered page. We also noticed that the 4G testing is slightly slower than over WiFi, this is due to a slower internet connection or slower hardware.