The following exercises are related to the use of Mongo Database.

1. Create a git repository for your answers to this problem sheet. Push the repository to GitHub. Make a commit and push it to GitHub after each exercise.

2. In our lab we got our React application sending and receiving data from out Node/Express Server. The final solution to the lab can be found at `https://github.com/Data-Rep-MERN-Applicati` `lab_six`. Clone this application if you did not finish the application last week. To clone an the application
   git clone `https://github.com/Data-Rep-MERN-Application/lab_six`

   if the application does not run you must install the project dependencies

   ```
   npm install
   ```

3. (a) Create a free account and free cluster on `https://www.mongodb.com/`. Follow the tutorial provided.
   (b) Allow all IP addresses to connect to your database.
   (c) Add a simple user and password to connect to your database e.g admin admin

4. Get your node/express server to connect to the mongodb Mongo database. We will use mongoose.js module to connection to our Mongo database hosted in the cloud. `https://mongoosejs.com/docs/index.html`

5. Create a data model using Schema Interface to represent the book object we have been creating and listing in the labs to-date. An introduction to mongoose schema can be found at: `https://mongoosejs.com/docs/guide.html`. Use the Schema to construct a database model: `https://mongoosejs.com/docs/models.html`. A compete tutorial on using mogoose.js can be found at: `https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/mongoose`.

6. Make sure your database is empty of all files and collections. Write a method that will write data to your data base use the tutorial available at `https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/mongoose`

7. Write a method that reads all data from the database and gets it to display on the react app.

8. Write a method that reads a document/book by id from your database in your node/express server.