Exercise Tracker Web Application

Cathal Ryan

B.Sc.(Hons) in Software Development

April 26, 2019

Final Year Project

Advised by: John Healy
Department of Computing and Software Development
Galway-Mayo Institute of Technology (GMIT)

GMIT
INSTITIÚID TEICNEOLAÍOCHTA NA GAILLIMHE-MAIGH EO
GALWAY-MAYO INSTITUTE OF TECHNOLOGY

Contents

About this project

Abstract The main objective for this project is to create a web application which helps the user find gym exercises based on the sports they play.

Authors Cathal Ryan

Chapter 1

Introduction

The main objective for this project was to create a project that displayed the skills that we have developed throughout our the four years we have spent here at GMIT. For this project I hope to create a web application that helps the user find basic gym exercises based on the sports they play. For example if the user were to select rugby exercises, the application will display a list of exercises focused on the upper body.

I decided to spend a lot of time thinking about which technologies would be suitable to create this project. During this time, I conducted research on using technologies such as Java and HTML for the front-end. The reasoning behind this would be that I had used these technologies throughout my time in GMIT and that I was very familiar with them. I also thought of using a mySQL database. The reasoning for this is that I had used this in many projects , especially my previous year in college. I chose not to use these technologies as I wanted to test myself and use technologies that I was not familiar with and see how I would get on.

The technologies that I have chosen to develop this project are React for the front-end, MongoDB to store all of the exercises and the Meteor Platform for developing the web application. I will go into greater detail, regarding the technologies used later in the 'Technology Review' section of this dissertation.

Chapter 2

Methodology

2.1 Research and Planning

Before I started developing this application. I did conducted a lot of research on the many technologies out there at the minute which I could have used.

Initially, I thought of using Angular instead of React but my supervisor strongly suggested using React instead. At this time I am glad I went with React as I feel my skills with front end programming have come a long way since the start of the project.

Early on in the research stages I also thought of using an SQL database with the application as I am very familiar with mySQL databases. However I chose against this to challenge myself and decide to use a MongoDB database. At first this was a challenge but once I read up on the MongoDB documentation and tested it myself, I eventually got the general gist of it.

For quite a long duration I couldn't pick which back-end service I wanted to use. Initially I started with a React Application as I thought that would make the most sense as I was writing the application in React. However I came into a a lot of issues in the earlier stages of the project such as not being able to deploy the application to the local server. After quite a lot of research I then chose to use Meteor[1] as my back end service. I found the Meteor documentation much easier to read and understand which helped me massively through out the project.

The most challenging part of the research in my opinion was the choosing of the cloud platform I was going to use to deploy my Application to. I didn't have much experience in deploying applications to the cloud so a lot of this process was very new to me. In the end I chose to use Heroku[2]. The main reasoning behind this was that I had just used Heroku in a module this semester which really gave me a good insight of it.

2.2 Agile Approach

For this project I chose to to have an Agile Approach. I did this by meeting with my project supervisor multiple times to discuss how the project was going. I tried to have an added feature to the project for every meeting that occurred.

I firstly created the Meteor Application and and deployed to the local host on my machine. At the beginning of this project I thought this step would have taken more time but due to Meteor's detailed and helpful documentation, I was able to to this fairly contently.

I then went about to connecting the database to my application which did turn out to be challenging. I firstly went about it by trying to print out data from the database to the server which would not work for me for quite a long time. I then tried to print data from the database to the console of the local host which worked. Once I had got this far it wasn't hard to print out the database data to the server.

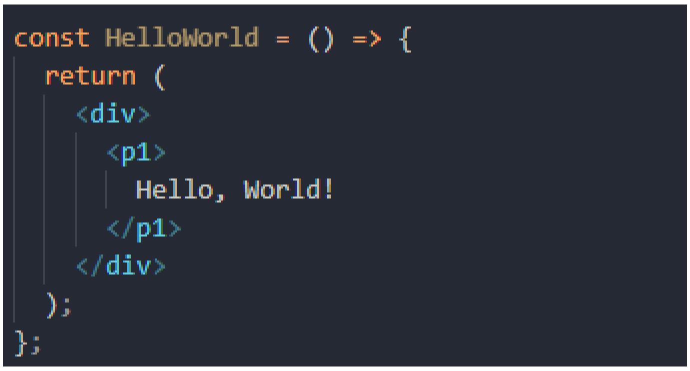
Chapter 3

Technology Review

For this part of the dissertation, I will talking about the technologies in which I incorporated into my project. I thought about the many technologies out there which I could use. For my previous final year project last year I decided to use Java to develop the front-end and used an SQL database to store information. This year for this current final year project I decided to use React for the front end. The reasoning behind this is because React has gotten very popular over the last few years. I really wanted to learn how exactly it compares to other languages used for front end programming. For the back end of the project I decided to use Meteor.

3.1 React

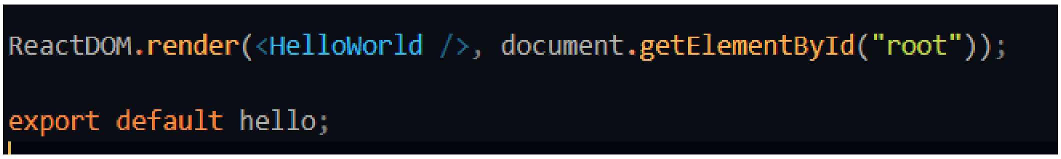
The main reasoning for picking React as the main language I used for this project is because I had never used it prior to this project and that it has become so popular over the last few years. React is a JavaScript library for building user interfaces. It is maintained by Facebook and a community of individual developers and companies-[3]. One of the primary features of React is components. Components allow you to split the UI into independent, reusable pieces and think about each piece in isolation-[4]. Components are the building blocks of a React application. Components are mostly JavaScript classes or functions that accept inputs optionally. A code example of a React component which displays Hello World would look like the following :



```
const HelloWorld = () => {
  return (
    <div>
      <p1>
        Hello, World!
      </p1>
    </div>
  );
};
```

In this example you can see that I created a constant called 'HelloWorld', that returns a function in which it prints out 'Hello, World!"

If I then wanted to use this function I would have to put it in a render function which would look like the following:



```
ReactDOM.render(<HelloWorld />, document.getElementById("root"));

export default hello;
```

3.2 MongoDB

MongoDB is a cross-platform document oriented database program. Classified as a NoSQL datbase program, MongoDB uses JSON like documents with schemata[5]. The main reasoning for using a Mongo database for this project was because I had never used it within a project itself before. They only previous experience that I had with Mongo was a module which I did in my second year of my studies here at GMIT. However throughout this module, we only really covered the basic concepts of the databases such as creating a collection, inserting a document into a collection and deleting them. While doing my research I debated on using an SQL database instead of Mongo. The main reasoning behind my choice to use mongo, was that I had already used SQL databases many times in the past and that I wanted to challenge myself and try something new.

The main differences between a Mongo database and a mySQL database would be:

- In a Mongo database, data is represented in a collection of JSON documents, while in a SQL database, data is represented in tables and rows.

- Another difference would be that mySQL databases supports atomic transactions. This means you can have multiple operations within a transaction and you can roll back if you have a single operation. MongoDB does not support atomic transactions.

- Finally, when using MongoDB, you are not responsible for defining the schema. All you have to do is simply drop in the documents that you wish to store. In a SQL database, you have t define the tables and columns before you try to store anything.

3.3 Meteor

Meteor or MeteorJS is a free open-source isomorphic JavaScipt web framework writte using Node.js. Meteor allows for rapid prototyping and produces cross-platform code [1]. The main features of Meteor would include:

- Both the back-end and the front-end are wrote in JavaScript.
- It automatically applies Node.js in the back end.
- It automatically applies MongoDB for the users database.
- It provides a library of key packages that most applications will require.
-

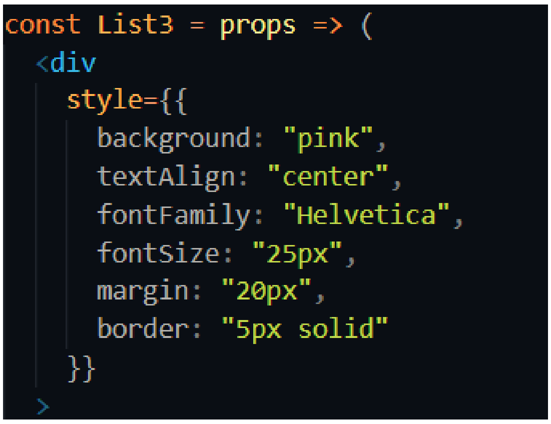
When deciding on what type of back-end I waned to use for this project, I originally had the idea of using React's back-end as I was writing the majority of the code in React. However I came into issues with deploying my app to server using React's methods, which then lead me on to see what other back-end pieces of technology I could use. I then came into contact with Meteor. The main fact as to why I chose this is because I didn't see much back-end technologies that supported react in the research stages of the project and Meteor seemed to have everything I needed to complete the project.

3.4 CSS

Cascading Style Sheets also known as CSS is a styling sheet language of a document written in a markup language such as HTML. CSS is designed to enable separation of presentation and content including layouts, fonts and colours[6].

Using CSS, in my opinion is probably the most well known method of styling code. I was first introduced to it in my first year of studies here at GMIT, where we learned the basics such as changing the background of a simple HTML page and styling lists etc. I decided to use it in this project as I thought it could make the application more appealing which in my opinion, it did just that.

The following is an example of how I used CSS with my project. The image below shows a constant that prints out a list of exercises. Within the style tag you can see that I set background colour, text alignment, font type, font size, margin and the type of border:



```
const List3 = props => (
  <div
    style={{
      background: "pink",
      textAlign: "center",
      fontFamily: "Helvetica",
      fontSize: "25px",
      margin: "20px",
      border: "5px solid"
    }}
  >
```

3.5 Bootstrap

Bootstrap is an open source toolkit for developing with HTML, CSS and JavaScript[7]. It is a free front-end framework that includes design template for typography, forms, buttons, tables, navigation and images.

I decided to use Bootstrap in my project as I think it adds massively to the front-end side of any application. I used it for making simple buttons look more appealing to the user. Bootstrap is very easy to use. The method in which I used to integrate bootstrap with my application was that I firstly installed it onto my machine by using the following command on the command prompt:

```
npm install --save bootstrap
```

I then proceed to import the downloaded libraries to my project. This was done by including the following line of code on the file that I wished to use bootstrap on:

```
import 'bootstrap/dist/css/bootstrap.css';
```

I used Bootstrap throughout the project. An example of where I used it woud be the home page where I used it to style the buttons. Below is an image of the buttons described above:

3.6 Visual Studio Code

Visual Studio Code is a source code editor developed by Microsoft for Windows, Linux and macOS[8] It is a very popular code editor due to its many appealing features. These features include:

- It includes support for debugging.
- Git can be used within the software.
- It has Syntax highlighting.
- It has code re factoring.
- It has customizable themes.

For this project, I used Visual Studio code to write all of the React and CSS code. The reasoning of this is because I have been using this program over the last few years and in my opinion is the best text editor out there at this point in time due to its many features such as the ones listed above.

3.7 Git

Git is a distributed version control system for tracking changes in source code during software development[9]. Git is used to track the progress of projects.

It does this by creating a local repository of your project on your machine. To start a local repository with git, all you have to do is navigate to your project folder and simply type:

```
git init
```

This then creates a local repository on your machine. If you wanted to commit a change to the project you can by simply adding your changes to the project by using the following command:

```
git add .
```

This command adds all of the changes that it can find in the project folder and add it to the local repository. If you then wanted to push these changes to GitHub, you must first add a remote origin to your local repository. This is done by using the following command:

```
git remote add origin https://github.com/user/Project-name.git
```

The command above would link the local repository to a repository on GitHub which then you coud push all of the changes to by using the command below:

```
git push -u origin master
```

3.8 GitHub

GitHub is a web-based hosting service for version control using Git. It offers all of the distributed version control and source code management functionality of Git also and adds its own features[10].

I used both Git and GitHub throughout this project as it was a requirement. Everytime that I had made a major change in the project I committed it and pushed the changes to my repository on GitHub which can be found here:

```
https://github.com/CathalRyan96/Project-Year-4
```

3.9 Cmder

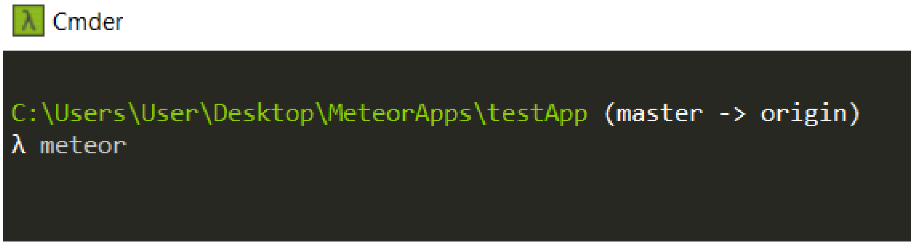
Cmder is a pre-configured software package that provides you with an appealing terminal emulator[11]. Cmder is very similar to the Command Prompt Terminal window that is pre-installed on most machines however it also has many additional features as well.

These features include:

- Git and all of its commands are already installed onto it, which saves you from downloading Git onto your machine.

- You can use Unix commands whilst using it. Commands such as ls, rm and chmod.
- It is very portable. What I mean by saying this is that you can upload it on to any cloud file storing websites such as Dropbox or Google Drive.

I used Cmder throughout the project, mainly for my use of Git. I also used it when deploying my application to the local host. I did this by navigating to my project folder and entering the command ”Meteor”, which can be seen in the image below:



```
C:\Users\User\Desktop\MeteorApps\testApp (master -> origin)
λ meteor
```

3.10 LaTeX

LaTeX is a document preparation system. It provides features designed for the production of technical and scientific documentation[12].

I had only used LaTeX once before last semester in the ”Research Methods” module in which we had to write a Literature Review in LaTeX. It was also recommended by our lecture to use it as they had already created a template from where we could start from.

I enjoyed using LaTeX as in my opinion styles documents in a much more appealing way rather than using just plain text in programs such as Microsoft Word.

3.11 Heroku

Heroku is a cloud platform service that supports many programming languages. It is one of the first cloud platforms as it has been in development since 2007. Back then it only supported the programming language called Ruby. However as of now it supports many languages such as Java, Node.js, Scala, Python and PHP[2].

For this project I deployed my application to Heroku. This was probably one of the most challenging parts of the project. I say this because, I had a lot of issues regarding connecting my Mongo datbase to my application on the cloud. After a lot of testing I finally came to the conclusion of hosting my database on a cloud service known as mLab. Here I created the collections that were present in my local database and connected it to my Heroku Application. As of now most of my application is being deployed on heroku, however some parts such as the about page are not appearing correctly and I am currently trying to figure out why I am getting this error. The application deployed on heroku can be found here:

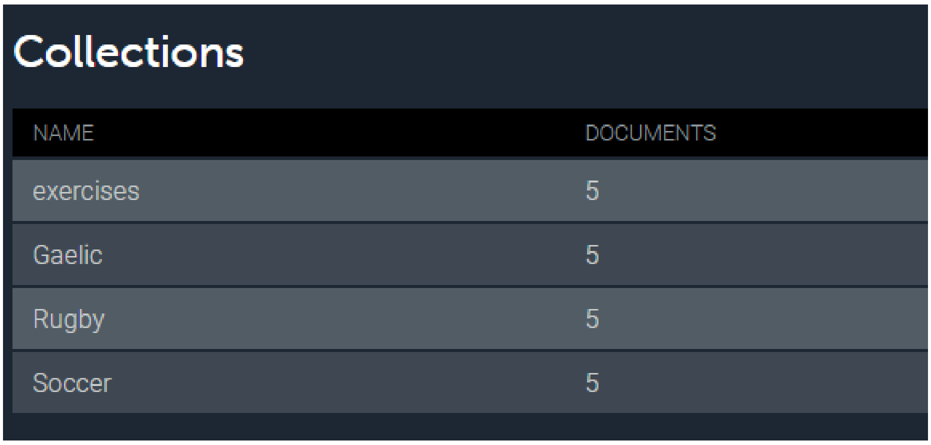
https://cryanfyp.herokuapp.com/

3.12 mLab

mLab is a fully managed cloud database service that hosts MongoDB databases. mLab runs on cloud providers such as Amazon, Google and Microsoft Azure.

I used mLab to host my database in order for it to connect to my Heroku Application. I did this by creating the collections and documents which were present in my local database and used the Mongo URI to connect it to my Heroku application.

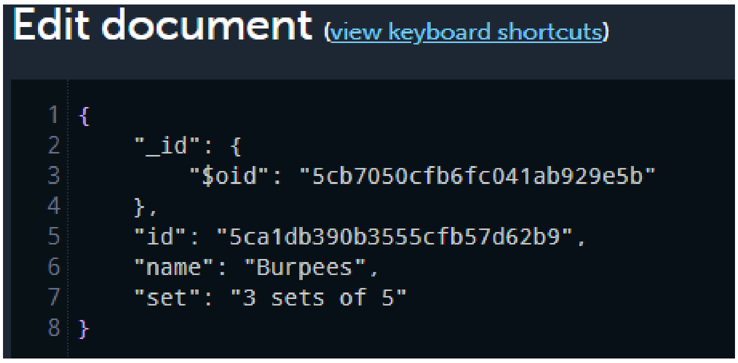
Images of the collections in the database on mLab can be seen below:



Collections

NAME	DOCUMENTS
exercises	5
Gaelic	5
Rugby	5
Soccer	5

Image of an example of a document in the database on mLab can be seen below:



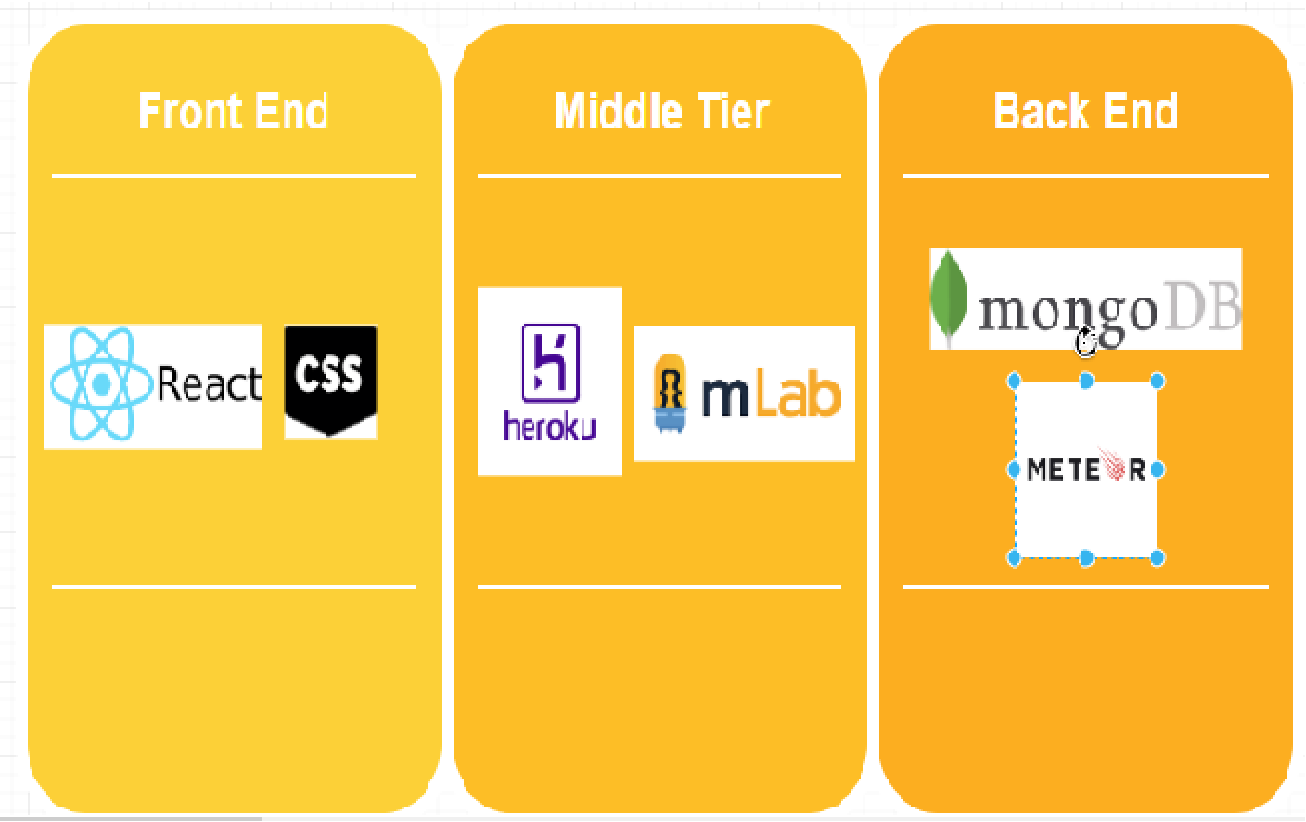
Edit document (view keyboard shortcuts)

```
{
    "_id": {
        "$oid": "5cb7050cfb6fc041ab929e5b"
    },
    "id": "5ca1db390b3555cfb57d62b9",
    "name": "Burpees",
    "set": "3 sets of 5"
}
```

Chapter 4

System Design

4.1 Architecture



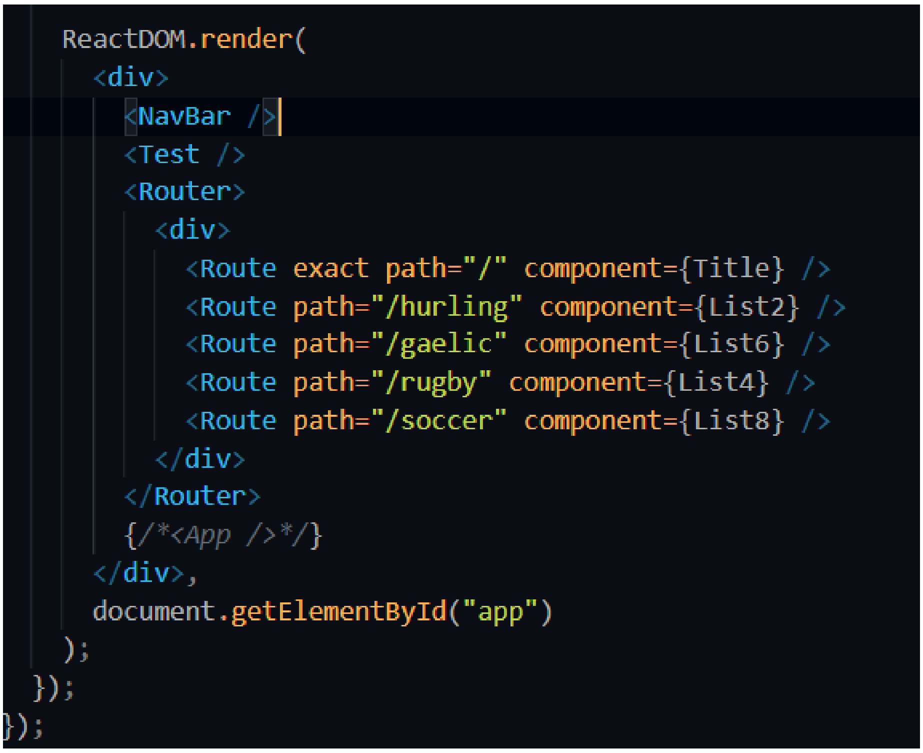
As you can see from the diagram above my application has three tiers, Front end, Middle Tier and Back End. Throughout the next part of the dissertation I will be discussing the these three tiers and how I went about creating them.

4.1.1 Front End

As you can see from the diagram above the front end of my application contains two technologies. These being React and CSS.

The majority of the application is written in React. While using React for the first time I learned quickly that React is built up using components and then using a render function to run these components. In my code I put most of the react code into a file called main.js. In this file I ran the majority of the components I created throughout the project.

The image below is my render function in my main file which runs the components which I call:



```
    ReactDOM.render(
      <div>
        <NavBar />
        <Test />
        <Router>
          <div>
            <Route exact path="/" component={Title} />
            <Route path="/hurling" component={List2} />
            <Route path="/gaelic" component={List6} />
            <Route path="/rugby" component={List4} />
            <Route path="/soccer" component={List8} />
          </div>
        </Router>
        {/*<App />*/}
      </div>,
      document.getElementById("app")
    );
  });
});
```

As you can see from the image above I render multiple components in the render function. The NavBar component displays the navigation bar at the top of the application which also has two NavLinks. These links are located at the right hand side of the Navigation bar and are called ”About” and ”Github”. The about link takes the user to the about page of the app that displays images of all the exercises listed in the application. The GitHub link takes the user the github repository of the project.

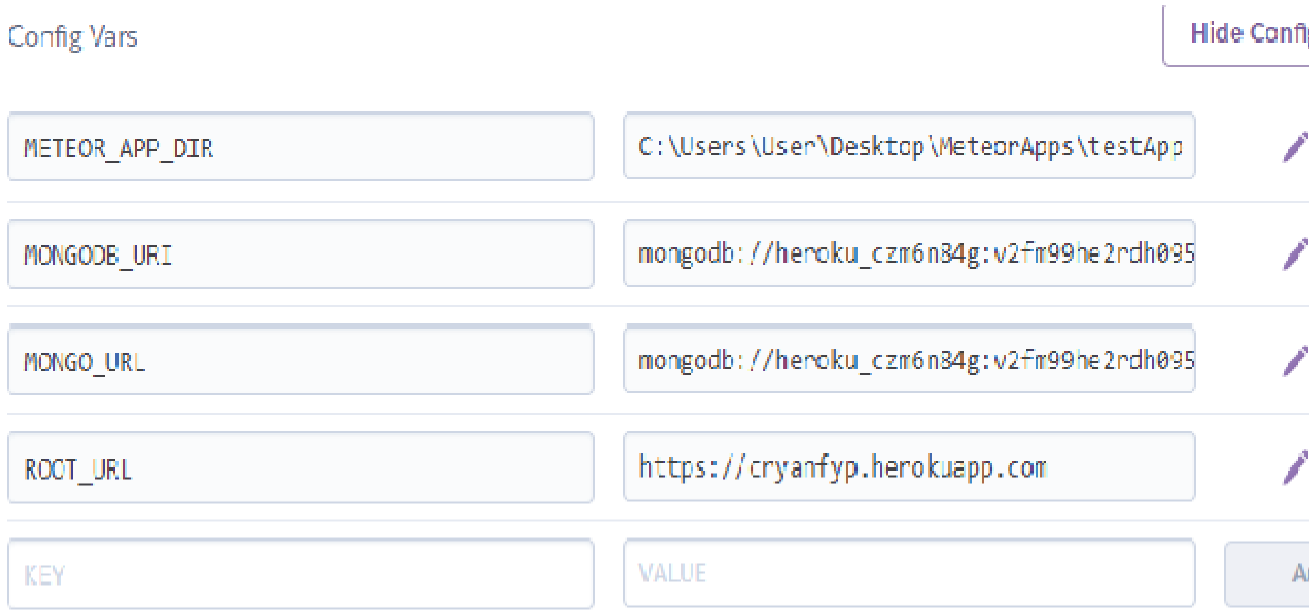
Within the router tag you can see that I have created five routes in which the user is redirected to if they select certain buttons. This was done by using React Router. React Router is used for redirection in React and is very enjoyable to use once you get the full grasp of it.

4.1.2 Middle Tier

As you can see from the diagram of the architecture, you can see that there are two technologies in the middle tier section, these technologies being Heroku and mLab.

Heroku was used to deploy my application the the cloud. I found this the most difficult process of the project as I couldn’t figure out fully how to connect my local database to it. After many days of research I came to the conclusion of hosting my database on mLab. I also had problems of setting the CONGIG VAR variables on Heroku but again after research I found out which ones I needed and how to use them.

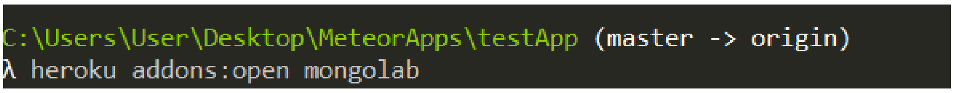
An image of the CONFIG VAR variables can seen on my Heroku account below:



Config Vars

METEOR_APP_DIR	C:\Users\User\Desktop\MeteorApps\testApp
MONGODB_URI	mongodb://heroku_czm6n84g:v2fm99he2rdh095
MONGO_URL	mongodb://heroku_czm6n84g:v2fm99he2rdh095
ROOT_URL	https://cryanfyp.herokuapp.com

I used mLab to host my database in order to use it with Heroku. Once I figured out how exactly to connect the both of them together, it was a massive part of the application complete. I very good technique I found to open the database associated with the Heroku application was the following :



```
C:\Users\User\Desktop\MeteorApps\testApp (master -> origin)
λ heroku addons:open mongolab
```

This command opens the mLab database that is associated with the heroku application.

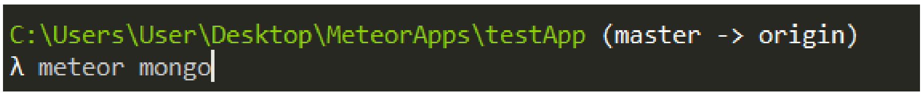
4.1.3 Back End

As you can see from the diagram of the architecture of the project the back end tier contains two technologies.

Meteor was used for the back end of this project which I really enjoyed using as it was my first time using it. It was used to created the initial application. As I stated before Meteor is an open source JavaScript web framework written in Node.js.

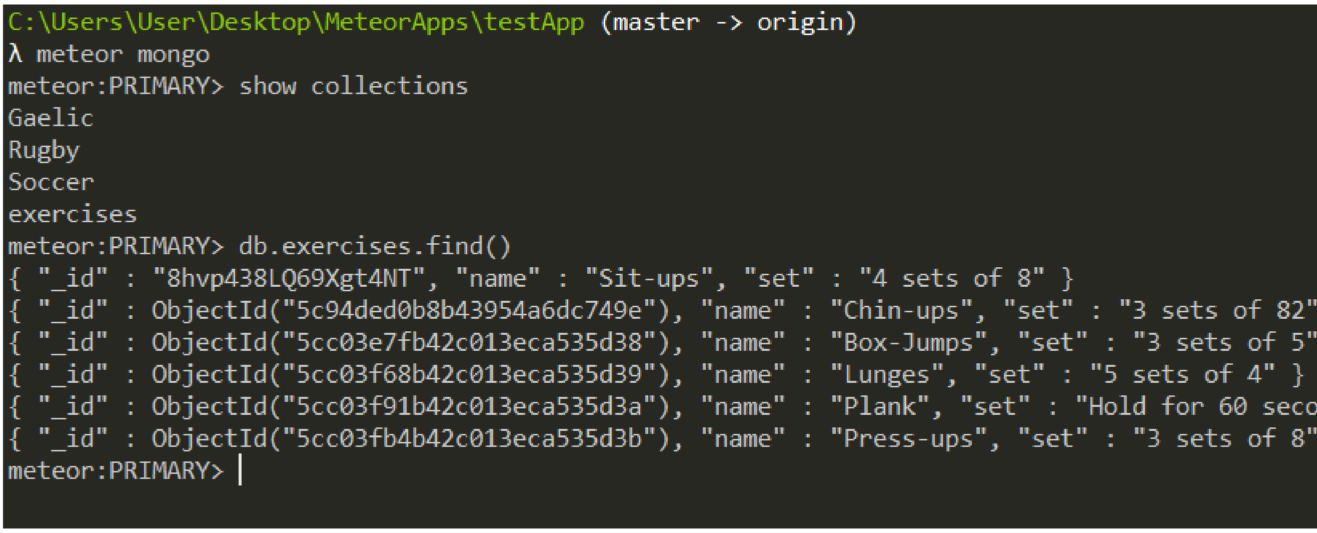
MongoDB was used for storing all of the information regarding the exercises. I did this by using one of meteor features which is Meteor mongo. This is a version of Mongo that can be used within the meteor application.

For example to gain access to this you must navigate to your project folder on the command prompt and enter the following command:



```
C:\Users\User\Desktop\MeteorApps\testApp (master -> origin)
λ meteor mongo
```

When this command is run you are then available to use mongo as normal. Below is an example of all of the collections I created and also examples of documents:



```
C:\Users\User\Desktop\MeteorApps\testApp (master -> origin)
λ meteor mongo
meteor:PRIMARY> show collections
Gaelic
Rugby
Soccer
exercises
meteor:PRIMARY> db.exercises.find()
{ "_id" : "8hvp438LQ69Xgt4NT", "name" : "Sit-ups", "set" : "4 sets of 8" }
{ "_id" : ObjectId("5c94ded0b8b43954a6dc749e"), "name" : "Chin-ups", "set" : "3 sets of 82"
{ "_id" : ObjectId("5cc03e7fb42c013eca535d38"), "name" : "Box-Jumps", "set" : "3 sets of 5"
{ "_id" : ObjectId("5cc03f68b42c013eca535d39"), "name" : "Lunges", "set" : "5 sets of 4" }
{ "_id" : ObjectId("5cc03f91b42c013eca535d3a"), "name" : "Plank", "set" : "Hold for 60 seco
{ "_id" : ObjectId("5cc03fb4b42c013eca535d3b"), "name" : "Press-ups", "set" : "3 sets of 8"
meteor:PRIMARY>
```

Chapter 5

System Evaluation

5.0.1 Screenshots

Below are images are of some of the different screens and areas of the about page of the application:

Image that displays a Sit up:

Sit-ups:

Image that displays a Burpee:

Burpees:

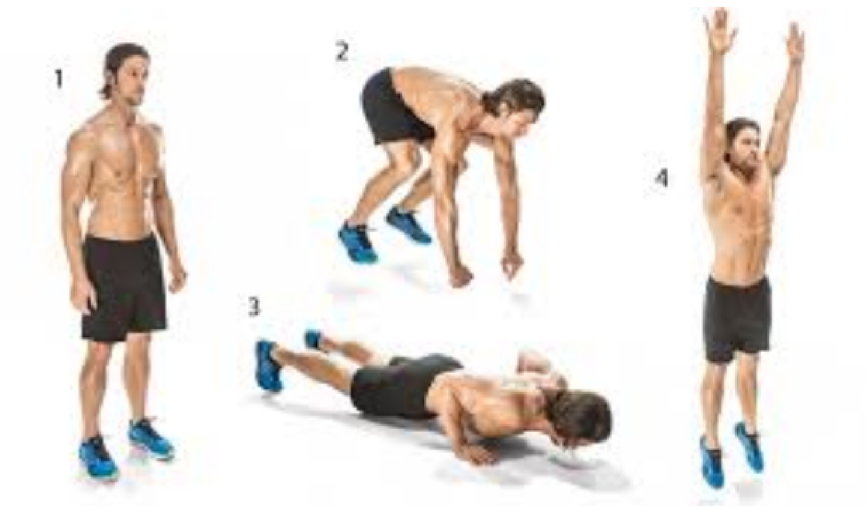
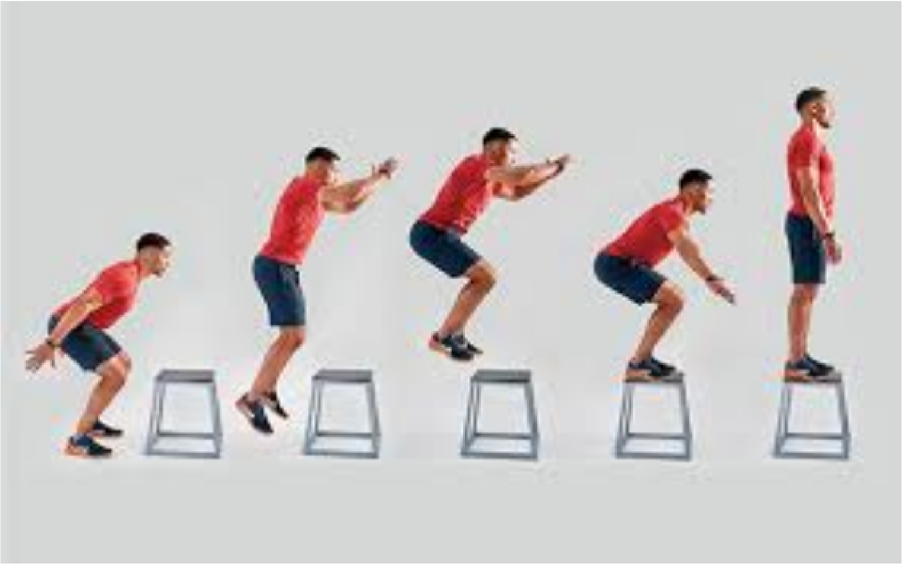


Image that displays a Box Jump:

Box-Jumps:



Chapter 6

Conclusion

At the start of the year I set a target of getting a web application up and running with a technologies that I was not familiar with and I think I have achieved this. I feel a massive sense of accomplishment on getting the project to work with the use of connecting so many technologies together.

Before I conducted this project I had no knowledge of how to use React , Meteor, Heroku and mLab.

My main objective of this project was to get a full three tier application deployed on the cloud which I did in the most part on time for the deadline.

6.0.1 Known Problems

As of now I am aware that alot of the features that I created cannot be accessed on the Heroku based application. For example the project project works fully locally but when deployed on the cloud the about page does not display properly as the images doe not display.

As of know I am trying to fix this error and hope to have it fully working soon.

Chapter 7

Appendices

GitHub: https://github.com/CathalRyan96/Project-Year-4
Heroku: https://cryanfyp.herokuapp.com/

Bibliography

[1] Meteor, “https://en.wikipedia.org/wiki/meteor$_{(}web_{f}ramework).''$

[2] Heroku, “https://en.wikipedia.org/wiki/heroku.”

[3] React, “https://en.wikipedia.org/wiki/react$_{(}javascript_{l}ibrary).''$

[4] R. Components, “https://dev.to/mcarpenter/react—stateless-vs-stateful-components–5c3m.”

[5] MongoDB, “https://en.wikipedia.org/wiki/mongodb.”

[6] CSS, “https://en.wikipedia.org/wiki/cascading$_{s}tyle_{s}heets.''$

[7] Bootstrap, “https://getbootstrap.com/.”

[8] V. S. Code, “https://en.wikipedia.org/wiki/visual$_{s}tudio_{c}ode.''$

[9] Git, “https://en.wikipedia.org/wiki/git.”

[10] GitHub, “https://en.wikipedia.org/wiki/github.”

[11] Cmder, “https://github.com/cmderdev/cmder/wiki.”

[12] LaTeX, “https://www.latex-project.org/.”