UNIVERSITY of INFORMATION
TECHNOLOGY and MANAGEMENT
in Rzeszow, POLAND

FACULTY OF APPLIED INFORMATION TECHNOLOGY

Field of study: INFORMATION TECHNOLOGY Specialty: Programming

Nyengeterayi Mawire,Johnson Mudzingwa
No. of student’s record book w66964,w66984

Workouts To-Do

Supervisor: MSc. Marcin Jagieła

Programming Languages Project

Rzeszów 2024

Contents

Introduction

Working out and fitness is an essential part in human life as human health is key. In our project we decided to implement a workout website. In this website it allows for input of a workout routine to be inputted in the website allowing for a person to see what workouts are on their agenda.

Chapter 1

Project Overview

1.1 Project Background

Objectives

- Input workout type, reps and sets
- Be able to delete a workout
- Be able to create a workout
- Manage all workouts in the database

Chapter 2

Software Requirements and Specifications

2.1 React JS

React.js is JavaScript framework that is fast, secure, and scalable. It provides a fantastic user and developer experience. Its use of components makes programming easier and unlike regular html frameworks its rendering capabilities saves time and processing as only key information and parts of the page are re-rendered instead of requesting for a whole new page.

2.1.1 Libraries

- React-router-dom : used to handle routes for different pages

2.1.2 Functional Requirements

1. Workout inputation:
    - Workouts should be inputted into fields and error messages returned when requirements are not met
    - Output the workout data in the frontend as soon as workout is created
    - Delete the workout from front end as soon as the user clicks button to see updated workouts

2.2 Node JS Back End

2.2.1 Node JS

Node.js is an open source, cross-platform runtime environment and library that allows for the running of web applications outside the client's browser.

2.2.2 Express Framework

Express JS is the most common framework used by Node JS. It is easy to use and is designed for creating web applications and APIs

2.2.3 Express JS API request:

In Express as we are using it as a communication between frontend and backend we are going to need a few requests. Here is a list of the api requests:

- / (Get): Get all workouts

- /{id} (POST): POST a workout

2.2.4 Mongo DB(Database)

MongoDB is an open-source document-oriented database that is designed to store a large scale of data and also allows you to work with that data very efficiently. It is categorized under the NoSQL (Not only SQL) database because the storage and retrieval of data in the MongoDB are not in the form of tables.

MongoDB ATLAS:

MongoDB atlas is a service provided by MongoDB which allows for the deployment of your database across platforms allowing for interaction whilst providing the same functionalities.

Chapter 3

Results and Outputs

3.1.1 Home Page

Fig. 4.1 shows the home page

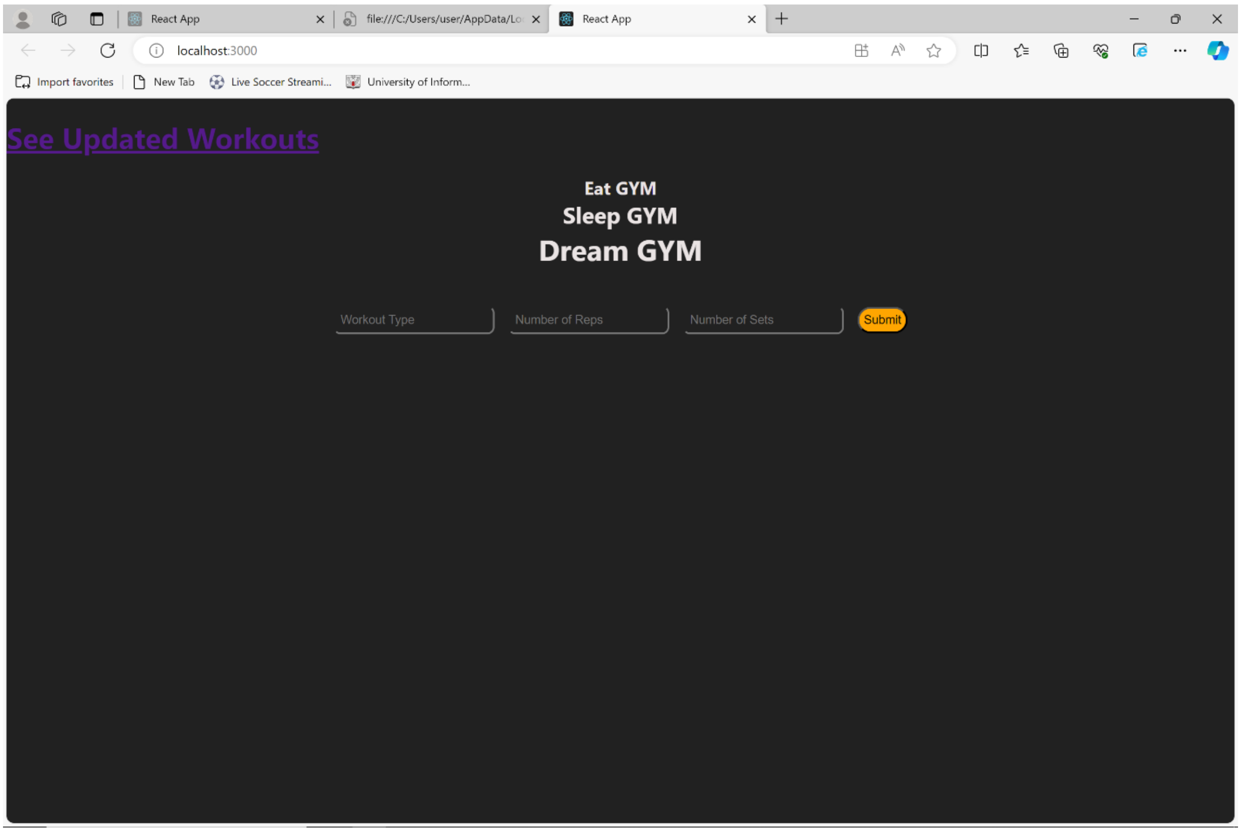


Fig. 4.1: Home Page
Source: Own Study (20.12.2023)

Fig. 4.2 shows how the user inputs data into the required field

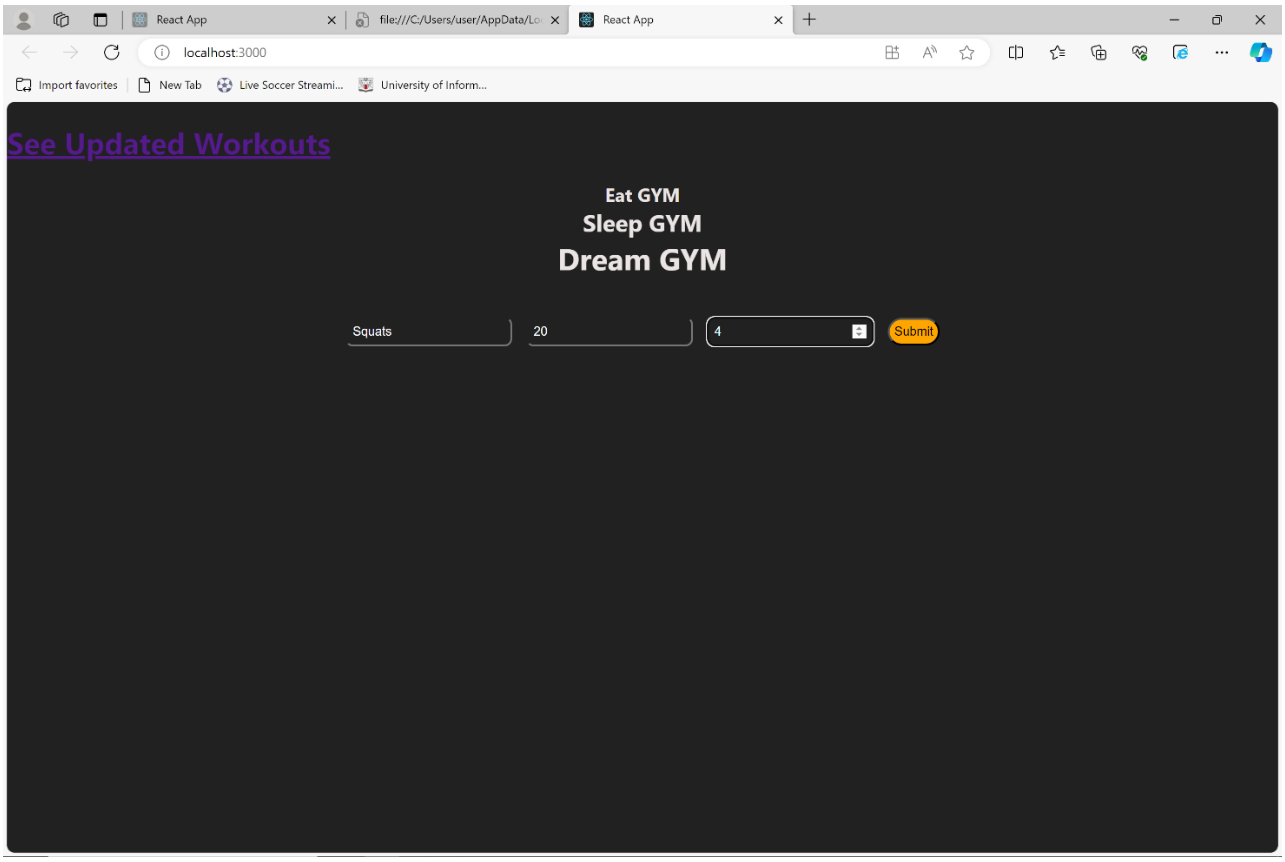
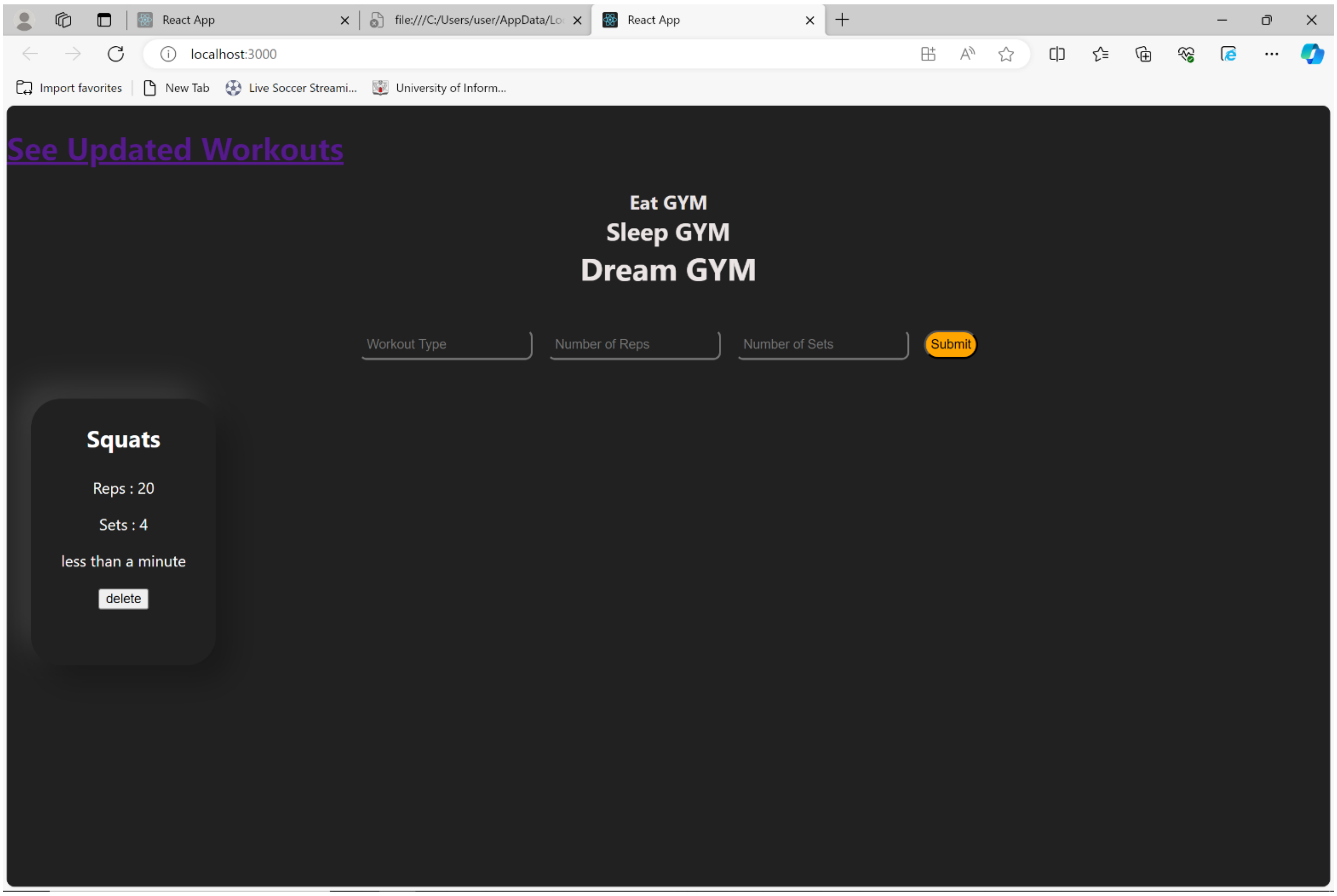


Fig. 4.2: Home page
Source: Own Study (20.12.2023)

Fig 4.3 shows the result after the user clicks on submit after inputting data into the fields



See Updated Workouts

Eat GYM

Sleep GYM

Dream GYM

Squats

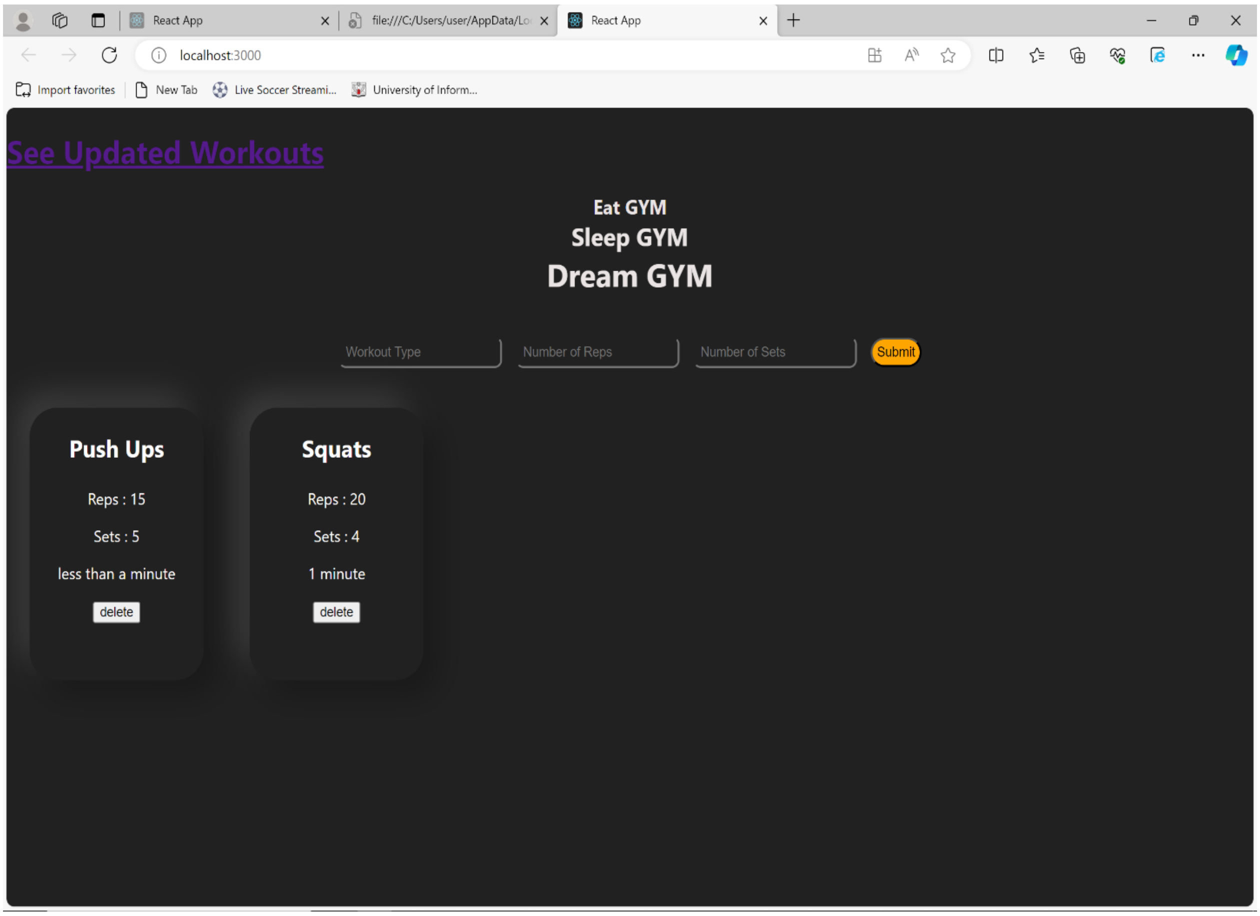
Reps : 20

Sets : 4

less than a minute

Fig. 4.3: Home page
Source: Own Study

Fig.4.3 shows the outputted workout after the user enters data in the fields and clicks submit



See Updated Workouts

Eat GYM
Sleep GYM
Dream GYM

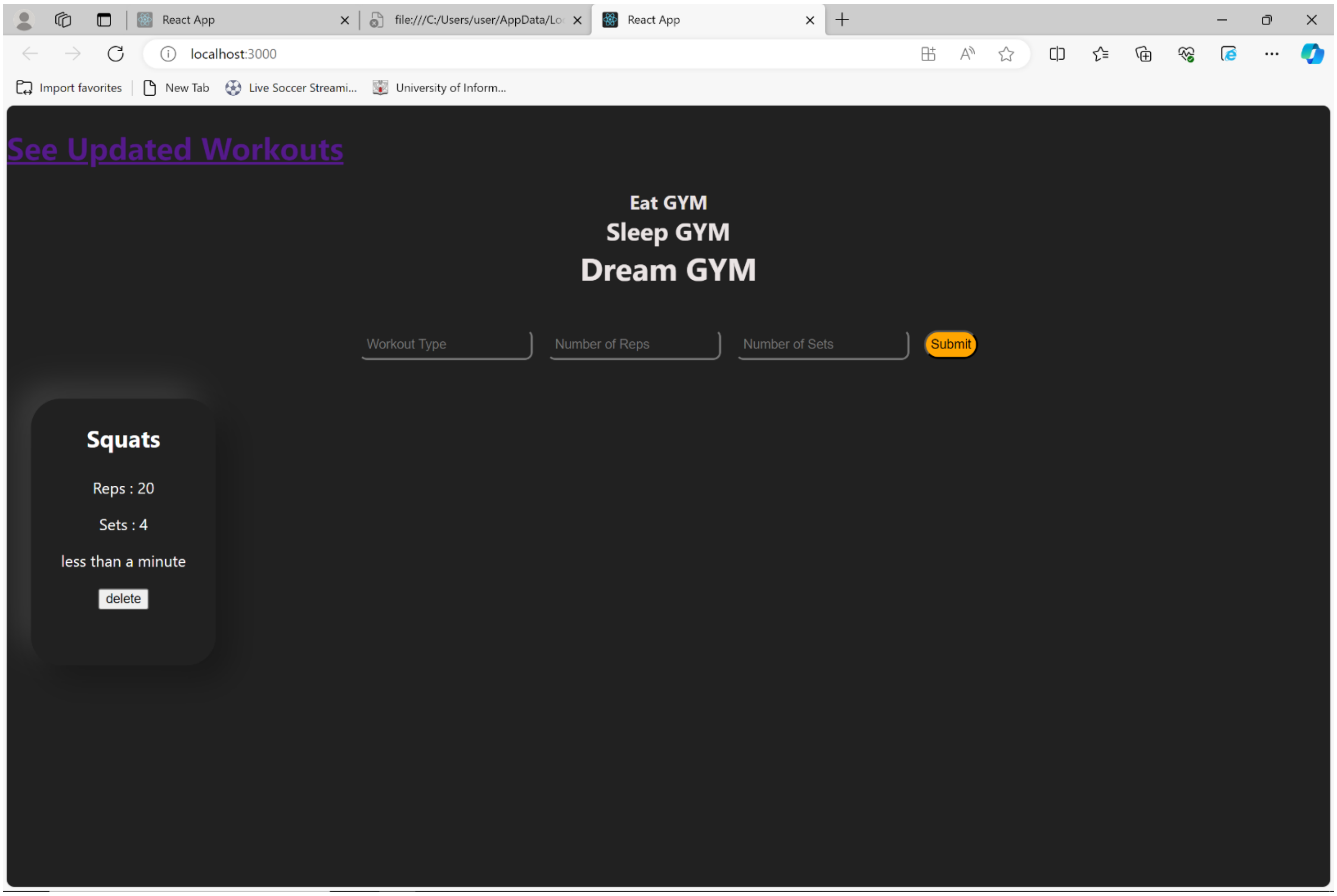
Workout Type | Number of Reps | Number of Sets | Submit

Push Ups
Reps : 15
Sets : 5
less than a minute
delete

Squats
Reps : 20
Sets : 4
1 minute
delete

Fig. 4.4: Home page
Source: Own Study

Fig 4.5 shows the result of whats outputted when a user deletes the push ups workout and then clicks the “see updated workouts button”



See Updated Workouts

Eat GYM

Sleep GYM

Dream GYM

Squats

Reps : 20

Sets : 4

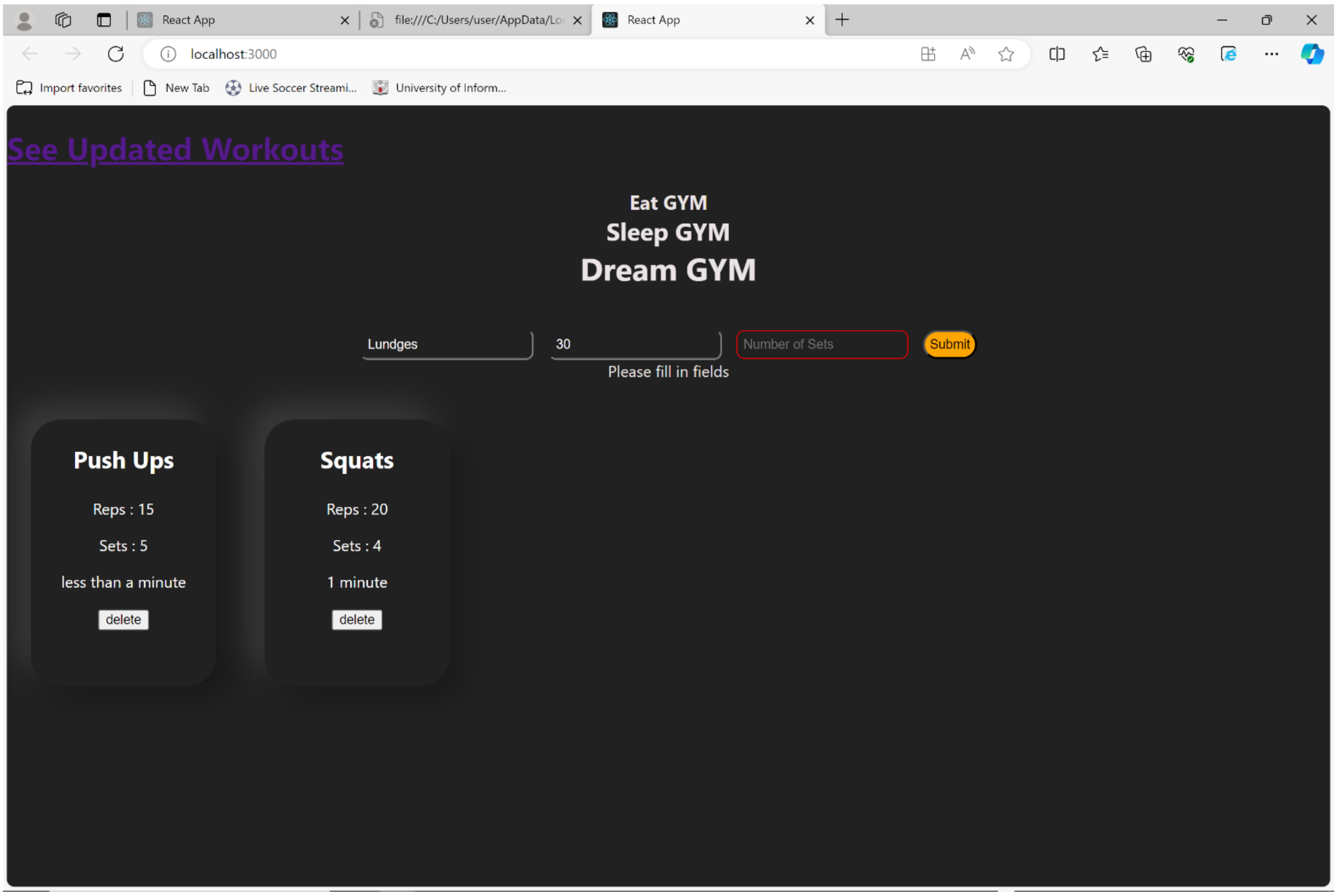
less than a minute

Fig. 4.5: Home Page

Source: Own Study

4.1.2 Errors

Fig 4.6 shows an error when a user attempts to add data with a sets missing field



See Updated Workouts

Eat GYM

Sleep GYM

Dream GYM

Lundges

30

Please fill in fields

Push Ups

Reps : 15

Sets : 5

less than a minute

Squats

Reps : 20

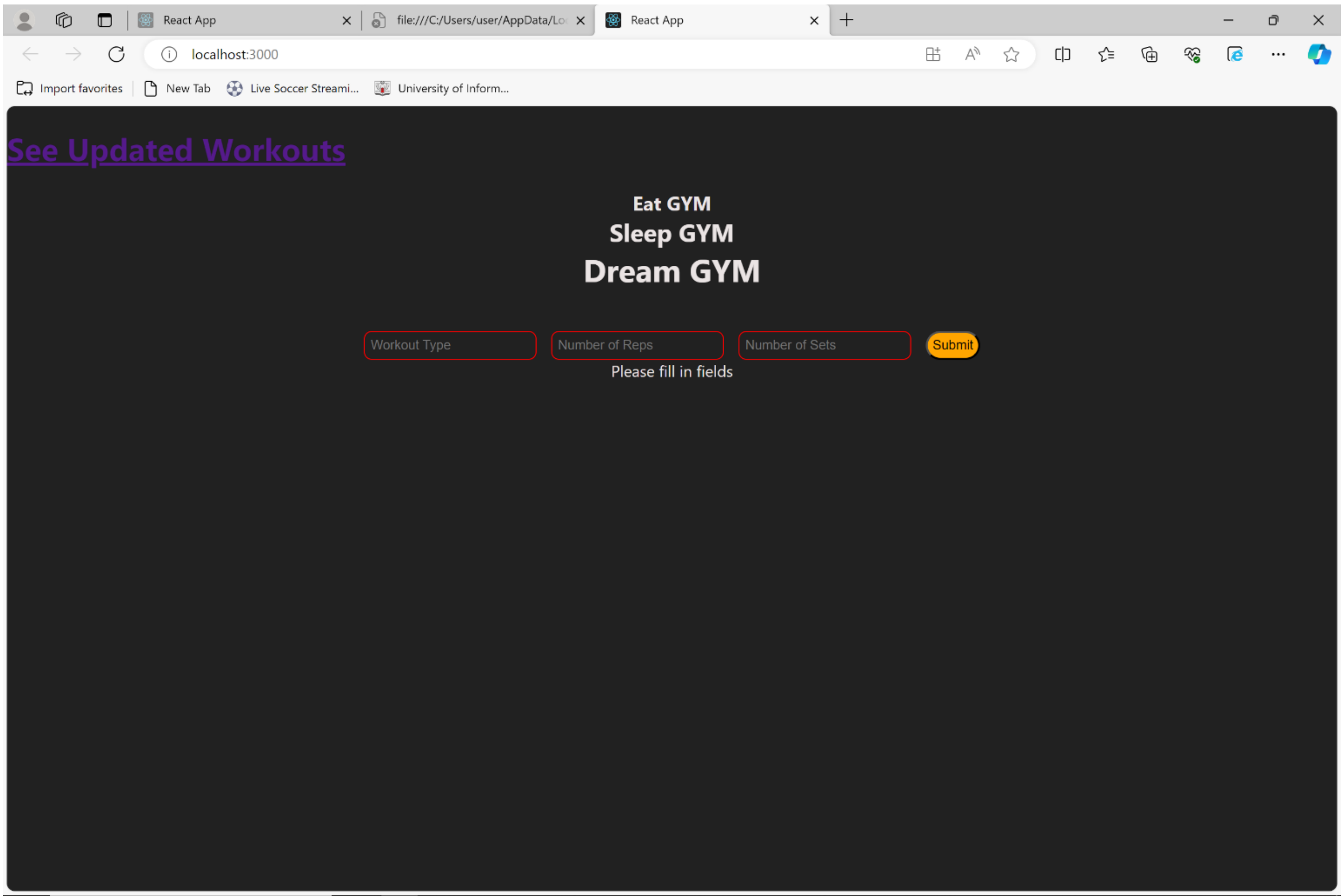
Sets : 4

1 minute

Fig. 4.6: Errors in Home Page

Source: Own Study

Fig. 4.7 Shows the error that occurs when a user attempts to submit missing fields. The fields are then highlighted in red showing which field has an error and an error message is outputted below the fields.



See Updated Workouts

Eat GYM

Sleep GYM

Dream GYM

Please fill in fields

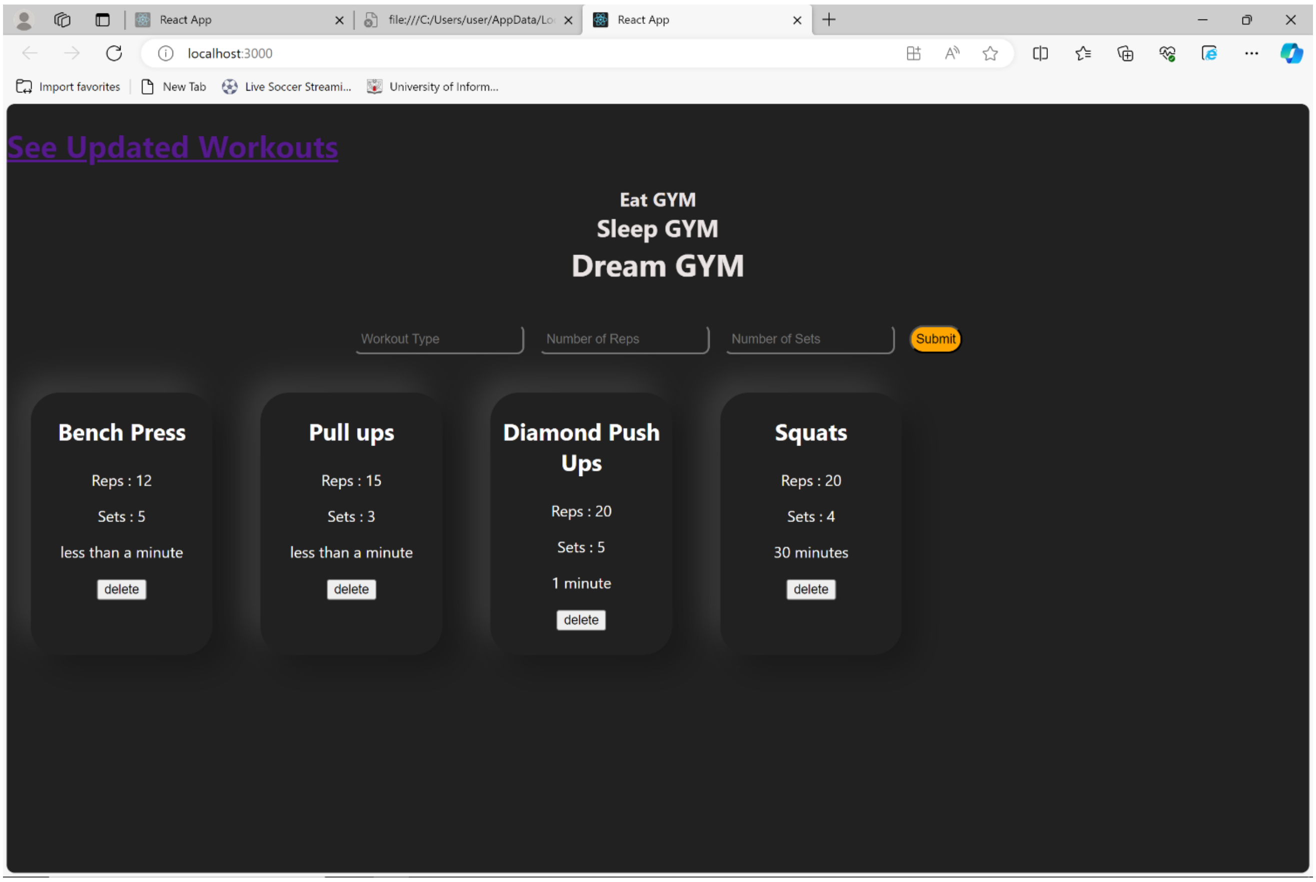
Fig. 4.7: Errors in Home Page

Source: Own Study

3.2Database Table

3.2.1 Stored workouts data

Fig.4.8 shows what workouts are stored in the database



See Updated Workouts

Eat GYM
Sleep GYM
Dream GYM

Bench Press
Reps : 12
Sets : 5
less than a minute
delete

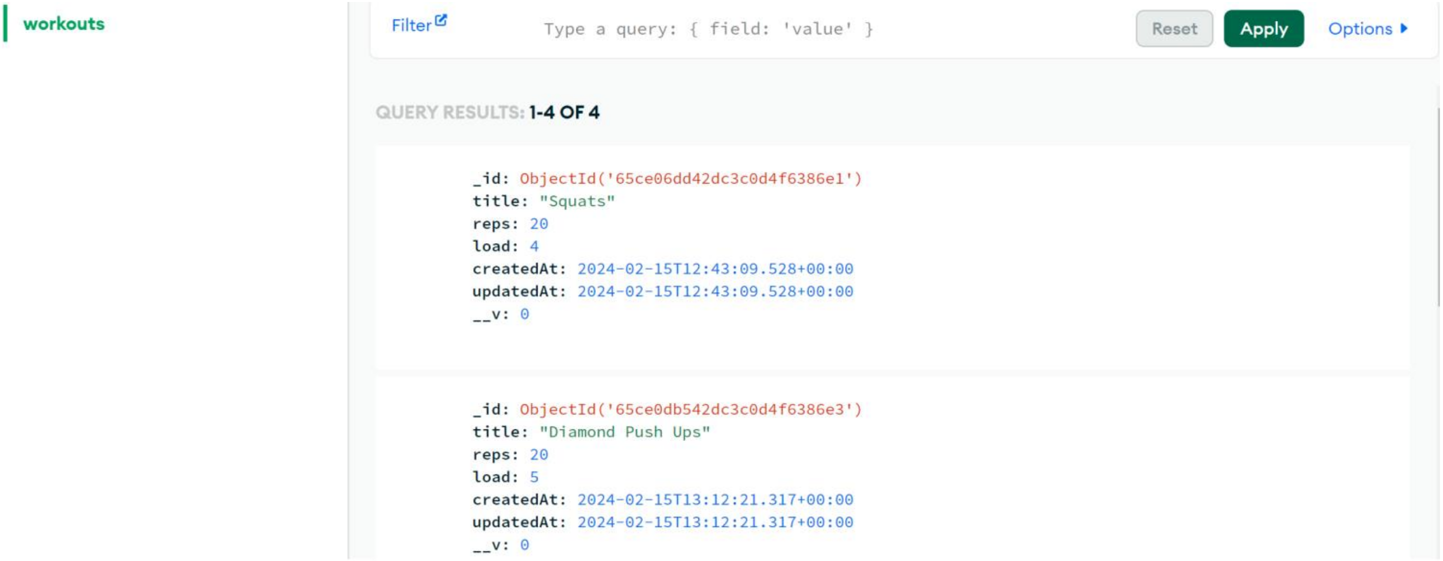
Pull ups
Reps : 15
Sets : 3
less than a minute
delete

Diamond Push Ups
Reps : 20
Sets : 5
1 minute
delete

Squats
Reps : 20
Sets : 4
30 minutes
delete

Fig. 4.8: Database Table: User Table
Source: Own Study (31.12.2023)

Fig.4.9 shows the data stored in database and how it is stored



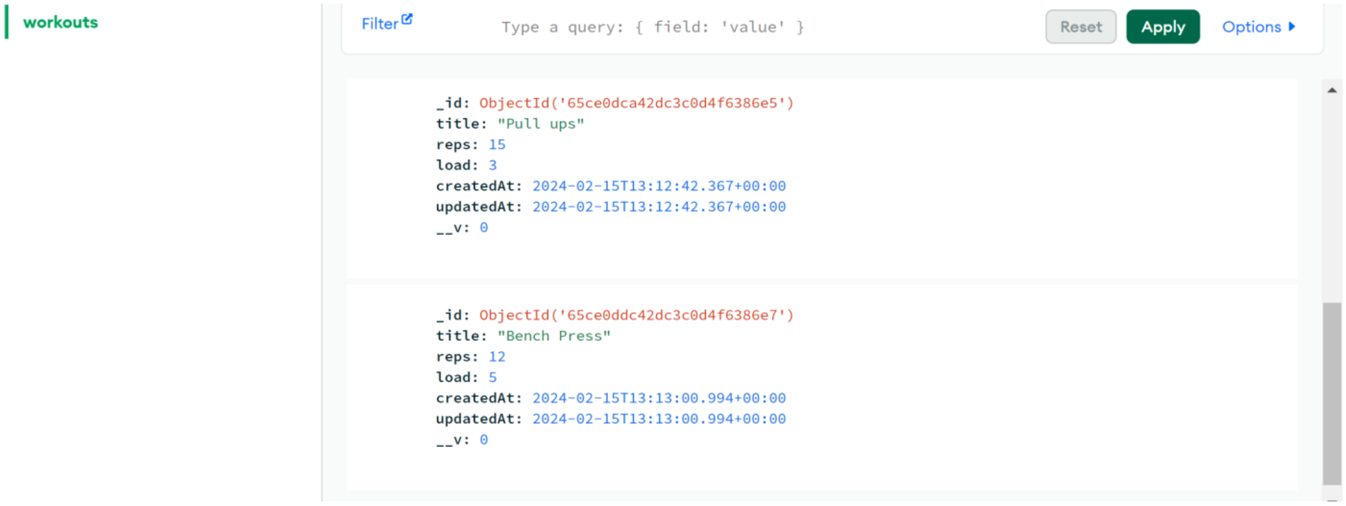
QUERY RESULTS: 1-4 OF 4

```
_id: ObjectId('65ce06dd42dc3c0d4f6386e1')
title: "Squats"
reps: 20
load: 4
createdAt: 2024-02-15T12:43:09.528+00:00
updatedAt: 2024-02-15T12:43:09.528+00:00
__v: 0

_id: ObjectId('65ce0db542dc3c0d4f6386e3')
title: "Diamond Push Ups"
reps: 20
load: 5
createdAt: 2024-02-15T13:12:21.317+00:00
updatedAt: 2024-02-15T13:12:21.317+00:00
__v: 0
```

Fig. 4.9: Database Table: Workout
Source: Own Study (31.12.2023)

Fig.4.10 shows the data stored in database and how it is stored



```
_id: ObjectId('65ce0dca42dc3c0d4f6386e5')
title: "Pull ups"
reps: 15
load: 3
createdAt: 2024-02-15T13:12:42.367+00:00
updatedAt: 2024-02-15T13:12:42.367+00:00
__v: 0
```

```
_id: ObjectId('65ce0ddc42dc3c0d4f6386e7')
title: "Bench Press"
reps: 12
load: 5
createdAt: 2024-02-15T13:13:00.994+00:00
updatedAt: 2024-02-15T13:13:00.994+00:00
__v: 0
```

Fig. 4.10: Database Table: Workouts
Source: Own Study (31.12.2023)

Chapter 4

Conclusion

In conclusion we have managed to implement a fully functional workout to-do website which also meets all of our requirements that we set of to achieve in the beginning. We have managed to implement the create a workout functionality. We were able to implement an update of the created workout by outputting it to our frontend immediately after creation. We were able to implement a functional delete operation that deletes a workout from the database through the frontend operation.

Even though we able to implement all of these functions we however failed to implement an automatic render of the deleted workout. Instead of the data in the database being outputted immediately, we used a different approach by re-rendering the page by clicking the see updated workouts button which in our program is a function that runs a fetch execution to our database.

Even with this issue we still managed to make all of our functionalities work, but just nost in the most efficient manner.

Bibliography

[1] React Documenation: https://react.dev/reference/react

[2] NodeJS Documentation :https://nodejs.org/en/learn/command-line/how-to-read-environment-variables-from-nodejs

[3] React Routing :https://expressjs.com/en/guide/routing.html

[4] MongoDB : https://www.mongodb.com/docs/manual/core/document/

[5] CORS : https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS

[6] React beginners course : https://www.youtube.com/watch?v=f55qeKGgB_M

[7] ExpressJS tutorial: https://www.youtube.com/watch?v=39znK--Yo1o&list=PL_cUvD4qzbkwp6pxx27pqgohrsP8v1Wj2

Summary
Workouts to-do

Author: Nyengeterayi Mawire, Johnson Mudzingwa
Supervisor: MSc. Marcin Jagieła

We have created a workout to-do site that acts as a to-do list for workouts. It allows for a user to create a workout and delete it in order for them to track the workouts they have to do. The website was implemented using React JS in the frontend , Node Js and MongoDB as our backend.