

**BSc in Science in Computing & Multimedia**  
**Year 3, Semester 1, 2021**

**BACK-END WEB DEVELOPMENT**

**Continuous Assessment II**

<b>Lecturer:</b>	George Blanaru
<b>Student Name:</b>	Alessandra Silva dos Reis
<b>Student ID:</b>	21565
<b>Module Code:</b>	BSC30921
<b>Assessment Type:</b>	Individual Project

## FULL STACK WEB APPLICATION

**GITHUB LINK:** <https://github.com/21565-Alessandra/21565-BEWD-CA2>

For this assessment it was created a new API to be used as a back-end server. The previous API was obtaining *bands* information and the current one retains *singers*' details. A new mongoDB was built and this API counts with its own database server.

### CREATE REACT APP

The react app was installed in order to build users interfaces for the pages of this applications. To install it, the following commands were used:

```
npx create-react-app <app-name>  
cd <app-name>  
npm start
```

Once it was installed, it is needed to type on terminal:

```
npm i (to install/update the packages)  
npm start (to run the application)
```

### REACT ROUTER

React Router is a standard library for routing in React. It enables the navigation among views of various components in a React Application. To install the React Router it was used the following commands on terminal:

```
npm install react-router-dom@6
```

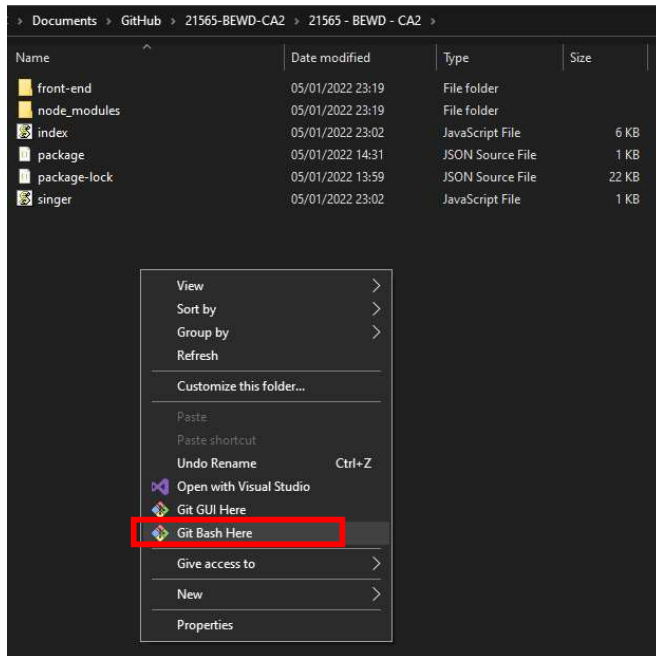
### AXIOS

Axios is a Javascript library used to make HTTP requests from node. To install Axios it was typed on the terminal:

```
npm i axios
```

To run the application, it is necessary to also run the server individually. To make it possible the port of the back-end server can't be the same as the front-end app.

An individual terminal can be opened directly at the folder by right clicking on the folder where the files are in and then clicking at Git Bash Here.



At this terminal it was used the following commands:

```
MINGW64:/c:/Users/reisa/Documents/GitHub/21565-BEWD-CA2/21565 - BEWD - CA2
reisa@DESKTOP-25RL5P5 MINGW64 ~/Documents/GitHub/21565-BEWD-CA2/21565 - BEWD - CA2 (master)
$ npm i
npm WARN web-application@1.0.0 No repository field.

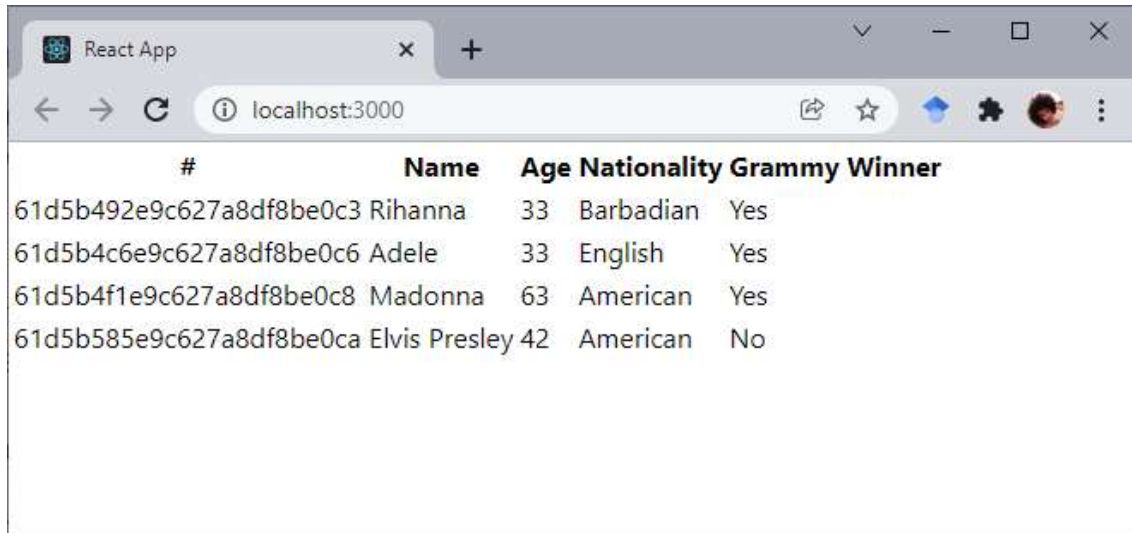
audited 77 packages in 0.825s

4 packages are looking for funding
  run 'npm fund' for details

found 0 vulnerabilities

reisa@DESKTOP-25RL5P5 MINGW64 ~/Documents/GitHub/21565-BEWD-CA2/21565 - BEWD - CA2 (master)
$ nodemon
[nodemon] 2.0.15
[nodemon] to restart at any time, enter 'rs'
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting 'node index.js'
Example app listening at http://localhost:3001
```

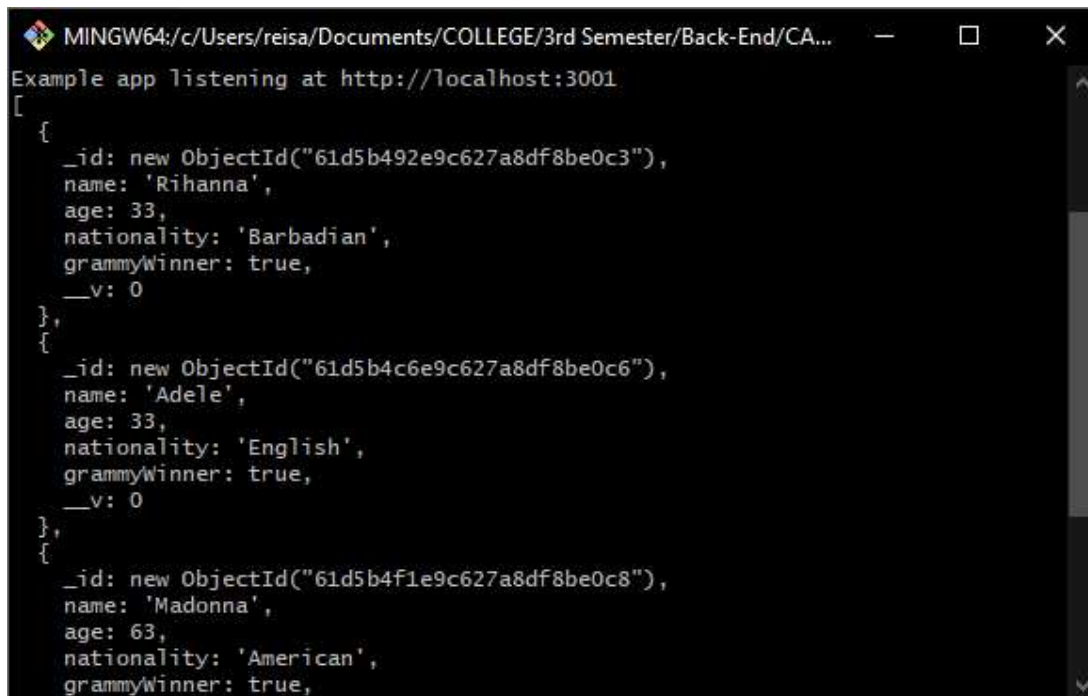
Then, at Visual Studio, in the folder where are the front-end files it was typed *npm start* and the following result was opened at the browser:



A screenshot of a web browser window titled 'React App' showing a table of artists. The browser address bar displays 'localhost:3000'. The table has five columns: '#', 'Name', 'Age', 'Nationality', and 'Grammy Winner'. It contains four rows of data for Rihanna, Adele, Madonna, and Elvis Presley.

#	Name	Age	Nationality	Grammy Winner
61d5b492e9c627a8df8be0c3	Rihanna	33	Barbadian	Yes
61d5b4c6e9c627a8df8be0c6	Adele	33	English	Yes
61d5b4f1e9c627a8df8be0c8	Madonna	63	American	Yes
61d5b585e9c627a8df8be0ca	Elvis Presley	42	American	No

The same information was printed at the back-end terminal:



A screenshot of a terminal window titled 'MINGW64:/c/Users/reisa/Documents/COLLEGE/3rd Semester/Back-End/CA...' showing the output of a server application. The terminal displays a message 'Example app listening at http://localhost:3001' followed by a JSON array of three artist objects. Each object contains fields for \_id, name, age, nationality, grammyWinner, and \_\_v.

```
Example app listening at http://localhost:3001
[
  {
    _id: new ObjectId("61d5b492e9c627a8df8be0c3"),
    name: 'Rihanna',
    age: 33,
    nationality: 'Barbadian',
    grammyWinner: true,
    __v: 0
  },
  {
    _id: new ObjectId("61d5b4c6e9c627a8df8be0c6"),
    name: 'Adele',
    age: 33,
    nationality: 'English',
    grammyWinner: true,
    __v: 0
  },
  {
    _id: new ObjectId("61d5b4f1e9c627a8df8be0c8"),
    name: 'Madonna',
    age: 63,
    nationality: 'American',
    grammyWinner: true,
    __v: 0
  }
]
```