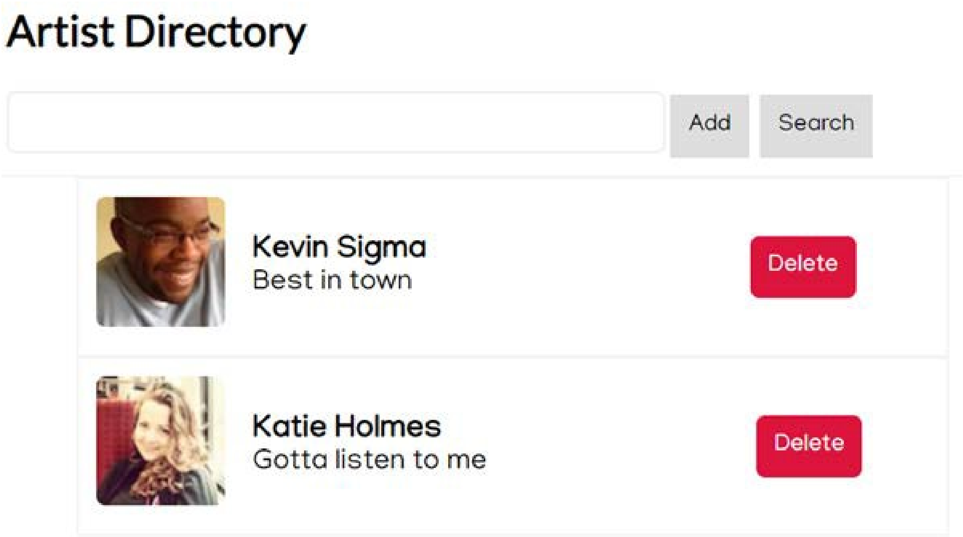
Lab 5

COMP 4711
WINTER 2020

The artist directory was working well with localStorage but we realized that anyone can clear it, so depending on just client side can only take us so far, that was the last nail in the coffin.



Artist Directory

Kevin Sigma
Best in town

Katie Holmes
Gotta listen to me

We need to look to node to help us out with our storage issues.

In the last lecture we explored node and saw that it can serve files.

The idea behind this lab is to serve our front end that we have created so far using node and at the same time use node to somehow store and retrieve the artist data.

One way could be store the data in some sort of file (text/json) and ask node to perform actions on that data, i.e add / delete.

This lab should feel very similar to the previous lab. Note **all** functionality add, delete and search should now be **handled by node** and not be dependent on localStorage/UI.

Marking guideline

Providing a url for lab	1 Marks
Addition of artist	3.5 Marks
Deletion of artist	3.5 Marks
Search	2 Marks

Note (If some of the below are not followed marks will be deducted)

- Cannot use databases (SQL/No-SQL)
- All JS/CSS code should reside in a separate files and use best practices
- fetch may be useful
- readFile / writeFile may be useful
- All node code written should be asynchronous
- Provide a zip of all your code
- Begin from a empty list of artists

Other:

Example for fetch (get request) can be found in Week 3 code samples / notes

Example for readFile / writeFile can be found in Week 4 code samples / notes

Example of POST via JavaScript

```
let user = {
  name: 'John',
  surname: 'Smith'
};


fetch('/somepath', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify(user)
})
.then((response) => response.json())
.then((data) => {
  console.log(data);
})
.catch((err) => console.log(err))
```