Student database

We are going to build a simple student database to manage student data. It will consist of the following features:

1. List of all students
2. Form for adding a new student
3. Student page to show data of a single student

Intro

You can use any frontend framework of your choosing (React, Angular, Vue ect.).

Adding styles is completely optional. Just add enough structure so that the features can easily be used.

We recommend starting with React. Facebook has created a simple starter app create-react app (https://github.com/facebook/create-react-app). With this starter app you don't have to worry about setting up the project and you can start coding right away.

Create the app:

`npx create-react-app my-app` OR `npm init react-app my-app`

If you need any help messages me in Whatsapp: +358505750702

Serving the data

All the data of the app is in db.json - file that is included in this folder. Database has two types of data students and courses. You can for example use json-server (https://github.com/typicode/json-server) to serve the data from a simple json backend:

1. Start by placing db.json in the root of the project
2. Install json-server: `npm install -g json-server`
3. Start json-server: `json-server --watch db.json`
4. Now if you go to http://localhost:3000/students/1 you will see the data of a single student:
   ```
   {"id": 1, "name": "Anni Anonen", "birthday": "1992-02-28", "address": "Kivakatu 1", "zipcode": "00500", "city": "Helsinki", "phone": "+358506760702", "email": "anni.anonen@testing.fi", "courses": [1]}
   ```

You now have the following endpoints to use:

```
GET    /students
GET    /students/1
POST   /students
PUT    /students/1
DELETE /students/1
GET    /courses
GET    /courses/1
POST   /courses
PUT    /courses/1
DELETE /courses/1
```

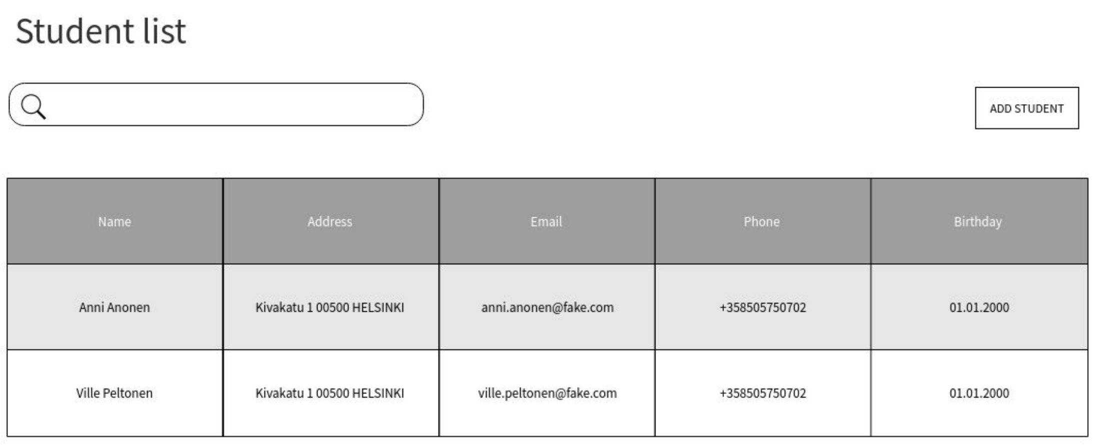
You can use the library of your choosing to make api - requests from react. We recommend axios (https://github.com/axios/axios):

```
const axios = axios.create({ baseURL: 'http://localhost:3000' });

axios.get('/students')
  .then(res =>//DO SOMETHING WITH THE RESPONSE);
```

Student list (1)

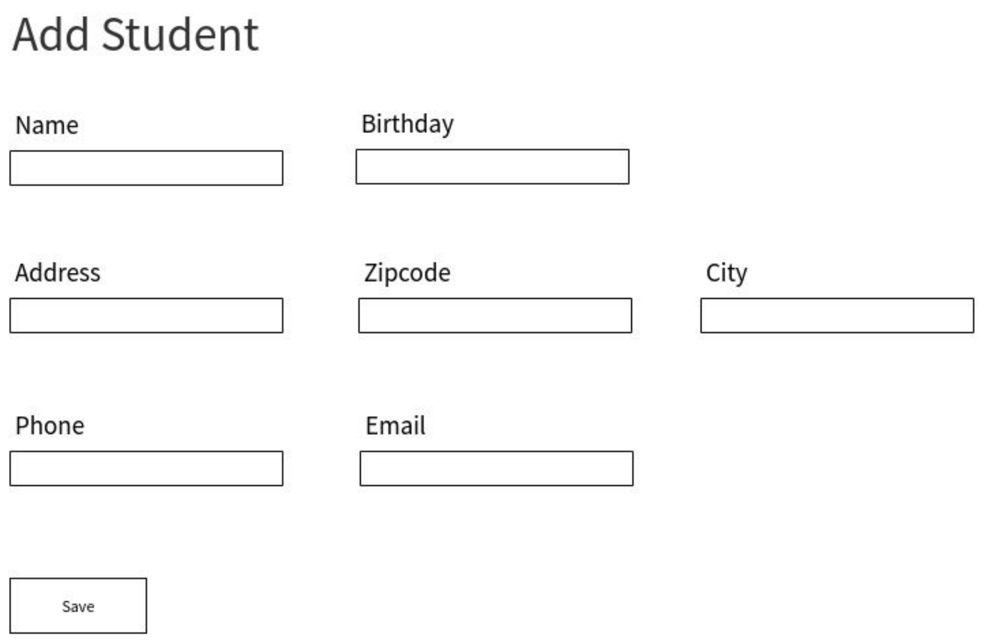
When the app starts we want to fetch all the students and show them in a list. We also want to be able to filter students by name (search field). You can show this component in the root route of your App:



Name	Address	Email	Phone	Birthday
Anni Anonen	Kivakatu 1 00500 HELSINKI	anni.anonen@fake.com	+358505750702	01.01.2000
Ville Peltonen	Kivakatu 1 00500 HELSINKI	ville.peltonen@fake.com	+358505750702	01.01.2000

Adding a student (2)

Add a button “Add student” to the top of the list view. Pressing the button opens a form where you can fill data of the new student (you can show the form in the same view, in a modal, or a separate page):



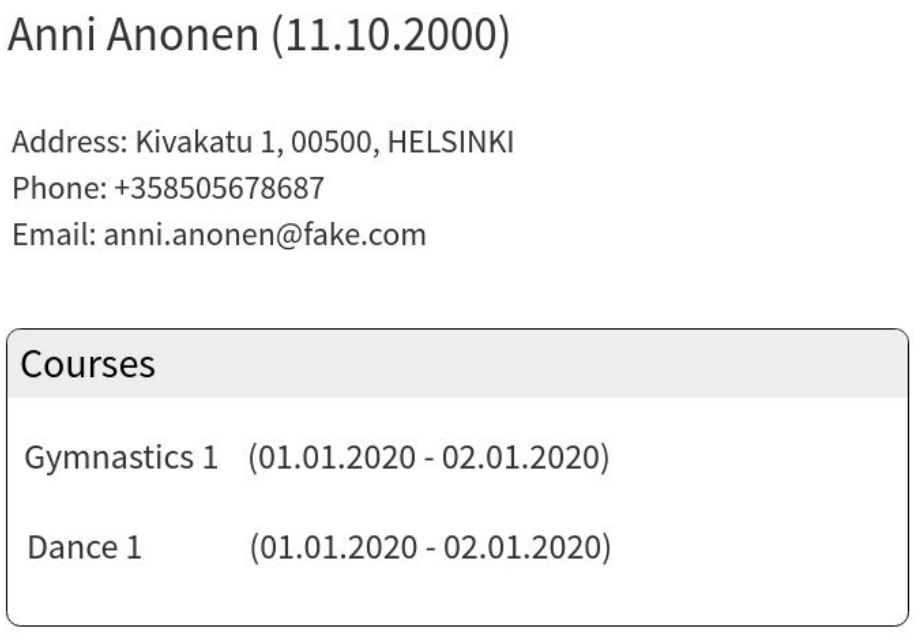
Submitting the form sends a POST request to /students:

```
axios.post('/students', yourStudentObject)
  .then(res => console.log(res.data));
```

Update the state and show the updated student list.

Show student (3)

Add a feature that if you click a student in the student list your app shows information of the corresponding student. Student page should show all the fields of the student in the top and also separate list of all the courses of the student. Course list shows the courses in following format: name (startdate - enddate):



Anni Anonen (11.10.2000)

Address: Kivakatu 1, 00500, HELSINKI
Phone: +358505678687
Email: anni.anonen@fake.com

Courses

Gymnastics 1	(01.01.2020 - 02.01.2020)
Dance 1	(01.01.2020 - 02.01.2020)

Prevent duplicate students (4)

You have received a request from the customer: “We have many users using the system so sometimes two users are adding the same student in the database and we end up having the same student twice in the list. Could you prevent this somehow?”

How would you start to solve this problem? Explain your solution in English (you can also implement it if you have time).

OPTIONAL: Add/delete course of student (5)

Make a feature to add and remove courses on the student show page.

Returning the exercise

After you have done the coding you can publish the repo in github (or another version control platform). Here is the tutorial on how to publish a repo in github:

https://help.github.com/en/github/importing-your-projects-to-github/adding-an-existing-project-to-github-using-the-command-line

After you are done send the link to your repo and your written answer to part 4:
esa.nuurtamo@hobiver.com

Deadline for sending the answer is 6.4.2020.

Hope you enjoy the assignment!