Due on December 28, 2022

# Project 6

In this assignment you are asked to implement the student management system of Project 5 using React. You may use any React UI Component Library in your project. I personally used the React-Bootstrap UI Library.

## Student Management System: The data and the backend

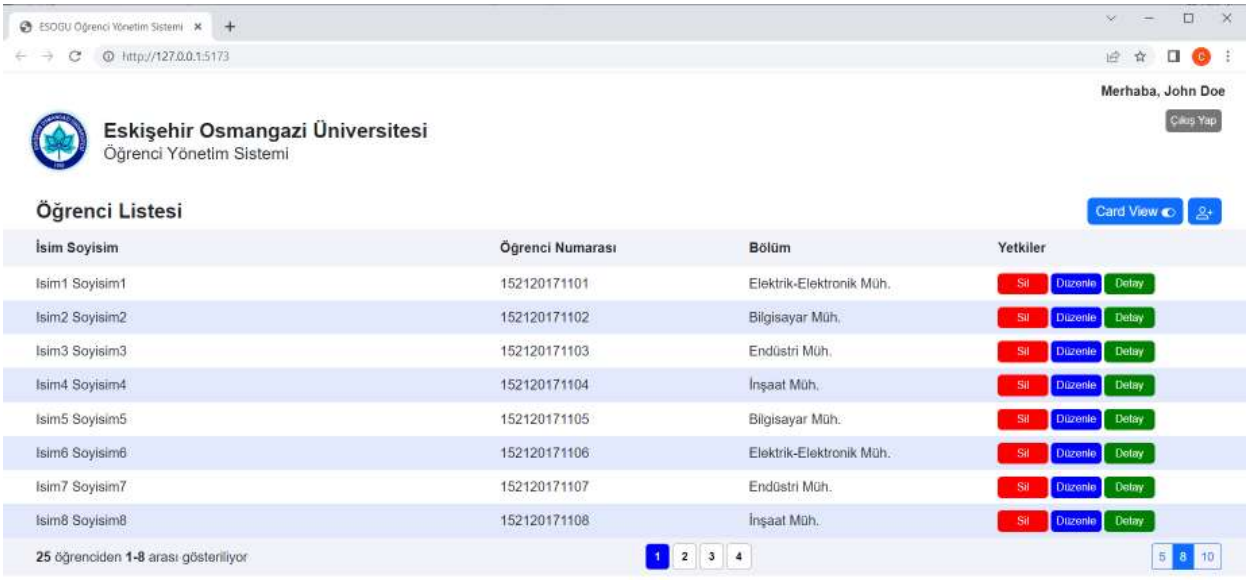
As in Project 5, we will be using the same students.json file as our data. To emulate our Rest API endpoints, run the json-server. However, you are required to run the json-server at port 8000 as follows:

```
% json-server --watch db.json --port 8000
```

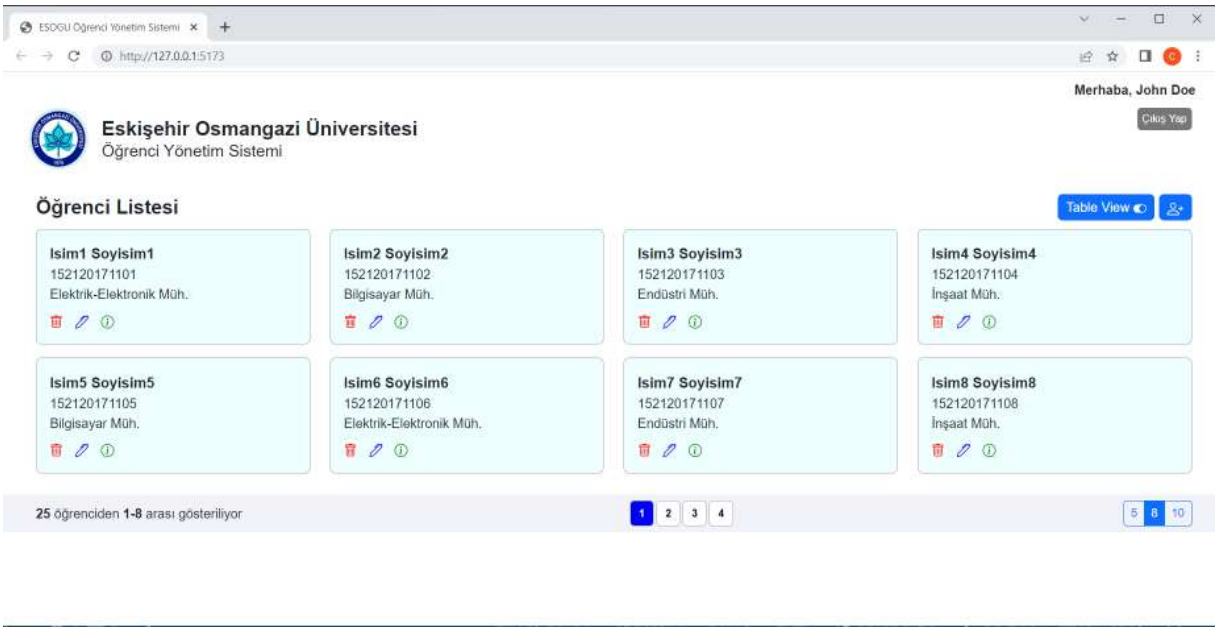
The reason for using port 8000 is the following. When you create a React app using *create-react-app* and start the live server using *npm start*, you app is served from TCP port 3000, which is the same port that the json-server uses by default. In order to avoid port collision, we move the json-server's TCP port to 8000.

## Student Management System: The front-end

The initial front-end of your app will be the same as in Project 5 with one additional button as shown below:




As you can see, we now have an additional button named "Card View" that will let us toggle between the 'table' view (the front-end above), and the 'card' view, which is shown below:




In the card view, we show each student's information inside a small Card component. Here we have 8 Cards because we display 8 students per page. The Card container will also be responsive: For screens < 768, we will see just 1 Card per row. For medium screens (768 <= screen size < 992), we will see 2 Cards per row. For large screens (992 <= screen size < 1200), we will see 3 Cards per row. And for extra large screens (screen size >= 1200), we will see 4 Cards per row. In the figure above, we have 4 Cards per row because this is an extra-large screen.

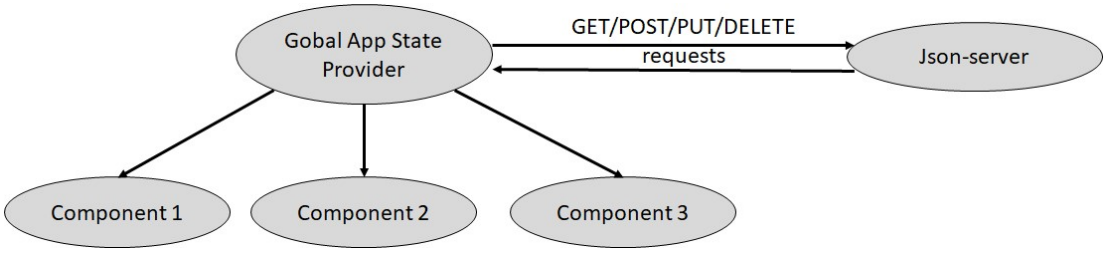
### Bootstrap Icons

In the card component, the controls are shown as icons. I used the following bootstrap icons in this project:

1. Sil: trash (https://icons.getbootstrap.com/icons/trash/)
2. Düzenle: pen (https://icons.getbootstrap.com/icons/pen/)
3. Detay: info-circle (https://icons.getbootstrap.com/icons/info-circle/)

### Implementation

You can either store the app state in the App component and distribute the data to child components using props, or use the Context API, which is what I used. I simply put all global app state in a React Context and accessed this context in the child components using the **useContext** hook. I also used the **useEffect** hook in the global app state provider component to fetch all students from the json-server when the app starts. The data is then stored in a JS object using the **useState** hook and distributed to other components through the provider component's value property. This is illustrated in the following figure.



**Submission**

Zip all project files and submit them through UZEM. Do not forget to include the group member names.