A screenshot of a computer

Description automatically generated with medium confidence**SUBMISSION FOR B4**  
  
Name: Gordon Foo  
Matriculation Number: A0199554L  
Github Link:  
  
1. Using React frontend to interact with backend API  
  
Created a Roster class that queries from a backend server and produces the data on a webpage:  
Graphical user interface, text, application

Description automatically generated  
Code lies in Roster.js   
If results were not queried properly, then the webpage would display the name as “Hardcoded”.  
Evidently, it isn’t hardcoded because the users array in Roster.js is updated by accessing the get api in index.js in the backend.

A screenshot of a computer

Description automatically generated with medium confidenceHere is the index.js that has all the backend apis.

The get request of /users is called in Roster.js and that is how the webpage is able to produce data from the backend.

Roster class is imported into React App:  
Text

Description automatically generated  
  
Delete button in webpage works with the backend:  
  
BEFORE PRESSING DELETE:

Graphical user interface, text, application

Description automatically generated  
Graphical user interface, text, email, website

Description automatically generatedGET request from postman shows that there are 2 entities, inline with what we see in the webpage. Pressing delete on the webpage should result in the next GET request in postman showing only the name that is not deleted. (To prove that deletion happens in the backend and our front end managed to call an API that deals with the data)

AFTER PRESSING DELETE UNDER JEREMY’S NAME:  
Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, email, website

Description automatically generatedAs you can see, after pressing the delete button, GET request only returns 1 entry.   
This means that the delete button on the webpage successfully called an API to update the “database” that an entry is deleted.

2.Used boostrap  
A screenshot of a computer

Description automatically generated imported boostrap in App.js  
Text

Description automatically generated added boostrap into index  
Text

Description automatically generated used bootstrap styling in Roster.js