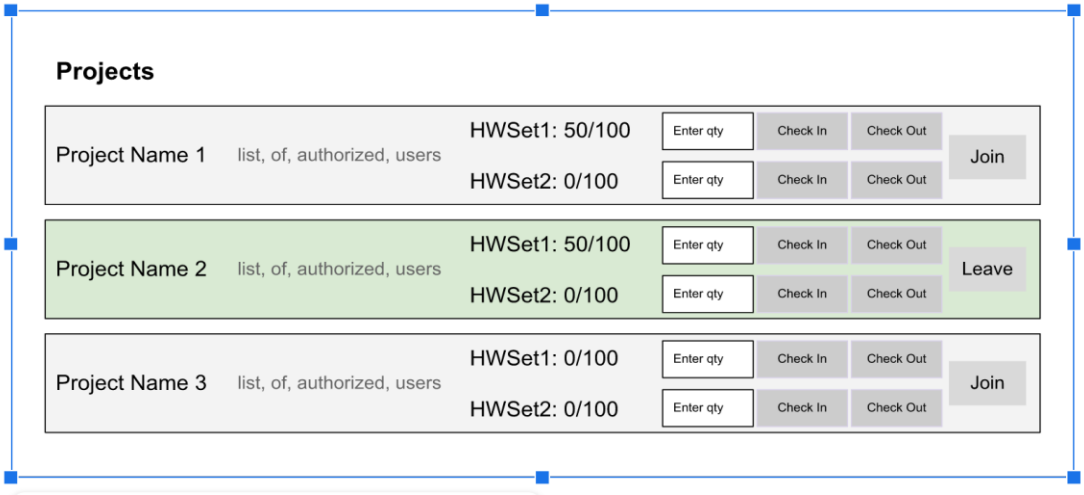
EE461L Homework 6

Bringing React, Flask, Heroku together

In Homework 5, you created a front-end application where users can see a list of projects they are authorized to join, view the hardware checked out from each hardware set, and join and leave projects. Below is a sketch of the page that you created



.

<u>Task 1 (3 pts):</u>

Create the server-side code (backend) for the front-end code that you created. As part of this task, you are required to create the following functions:

```
def checkIn_hardware(projectId, qty)
```

This function queries the projectId and quantity from the URL and returns the project id and quantity to the front end. The front end displays a pop-up message which says “<qty> hardware checked in”

```
def checkOut_hardware(projectid, qty)
```

This function queries the projectId and quantity from the URL and returns the project id and quantity to the front end. The front end displays a pop-up message which says “<qty> hardware checked out”

```
def joinProject(projectid)
```

This function queries the projectId from the URL and returns the project id to the front end. The front end displays a pop-up message which says “Joined <projectId>”

```
def leaveProject(projectid)
```

This function queries the projectId from the URL and returns the project id to the front end. The front end displays a pop-up message which says “Left <projectId>”

Note 1: Each function definition should have the appropriate route decorator for the URL from the front-end application.

Note 2: In the team project, you will connect to Mongodb in the above functions and perform additional steps such as

a) Check if the hardware is available for checkout
b) Decrement the available hardware quantity by the amount requested
c) Increment the available hardware quantity by the amount to be checked in
d) Check if user is authorized to join a project
e) Check if the user has joined the project they are asking to leave
f) Others

To keep the homework simple, we are not asking you to connect to Mongodb. You will define the API, parse the URL, and return information to the front-end code

Rubric

Requirement	Points
`def checkIn_hardware(projectId, qty)`	0.75
`def checkOut_hardware(projectid, qty)`	0.75
`def joinProject(projectid)`	0.75
`def leaveProject(projectid)`	0.75

Task 2 (3 points):

Learn how the toolchain (React, Heroku, Flask) is integrated into your workflow. To do this, follow this tutorial: https://medium.com/swlh/how-to-deploy-a-react-python-flask-project-on-heroku-edb99309311

It is a simple step by step tutorial that ensures that a React Getting Started example, Python flask example, and Heroku setup work together. The instructions for Flask and Heroku are like what we did in the class last week.

Deploy your application from Task 1 to Heroku. For this, create a Python Flask App following the tutorial above and/or the steps we followed in the class last week. Deploy this application to Heroku.

What to submit:

1. A link to your Heroku hosted app that TAs can run.
2. A zip file (yourEID_HW6.zip) containing the source code that was pushed to Heroku master (all the source code files that you created as part of this homework).