# Project 3– Fostering Hope Inventory: Check Out (Small Ticket Items)

PUT YOUR NAME HERE

|  |
| --- |
| NoName Mchiggens |

PUT GROUP NUMBER HERE

|  |
| --- |
| X99 |

PUT THE NAMES OF YOUR BUDDIES HERE

|  |
| --- |
| Samuel D. Test, Ima Nuther, Buddy Finkerbean |

## Expectations

* Instructor Guidance – **High**
* Independent Effort – **Moderate**
* Originality – **Low**
* Teamwork – **High**

## Description

Fostering Hope is a charity with 10 branches in Pennsylvania. The charity serves families and children who are interacting with the Foster-care system in Pennsylvania. Fostering Hope offers three primary services: “Dignity Duffles” (duffle bags for children entering foster care), “Bags of Hope” (care packages filled with emergency supplies to help foster parents get through the first 48 hours of a placement) and “Foster-share closets”.

This project is focused on the “Foster-share closet” service offered by the Northeast Philadelphia Chapter of Fostering Hope. Your goal is to create a system used to track various “small ticket” items in the closet.

This project (Project 2) is a team project. Students are required to work on this project with instructor-assigned project buddies.

## Terminology

| **Term** | **Description** |
| --- | --- |
| Foster-share closet | A foster-share closet is more than just a closet. It’s more like a thrift-shop. The big difference is that all the items in the foster-share closet are available to foster parents for free. Fostering Hope volunteers will sometimes refer to the foster-share closet as a “resource closet” or simply “the closet”.  |
| Big-Ticket Items | In the foster-share closet, big ticket items are items that need to be tracked individually. Big ticket items are often available in limited quantities due to high demand and/or high cost. Some examples of big-ticket items: strollers, car seats, infant formula, and diapers. A “big-ticket item” is “big” because it is valuable, and needs to be tracked, not necessarily because it is physically large.***We aren’t going to worry about tracking or taking inventory of these “big-ticket” items this semester.***  |
| Small-Ticket Items | Small-ticket items are available in abundance. Clothing items (shirts, shorts, dresses, etc.), hair scrunchies. children’s books … these are all “Small-ticket” items. |
| Check Out | When a parent leaves the closet with one or more small-ticket items, we want the items to quickly and efficiently removed from the closet’s inventory. This is very much like purchasing an item from a store, except of course there is no money being exchanged, as all the foster-share closet items are free. |
| Clerk | A clerk is a Foster-hope volunteer who helps parents “shop” at the closet and also helps keep track of the closet’s inventory.  |
| Super User | A super user can do everything a clerk can do. Super users can also perform things a clerk can’t do (for example, add new big-ticket items to the inventory). |

## Web Service Features

Here are the expected features of your web service:

1. Issue a GET against clients. The response will be all the active clients as a JSON object.
2. Issue a POST against auth and provide the keys username and password. The result will be a JSON object representing the authenticated user.
3. Issue a GET against smallitems. The response will be all smallitems, sorted by name, as a JSON object.
4. Issue a POST against scancheck and provide the keys clientid, userid, quantity, and itemcode of the small item. The result will be a JSON object representing the item name, item description, the date of the transation, and the quantity removed from inventory.
5. Issue a PATCH against reset. The quantity of each small item will be set to 100, and all the small item history will be deleted.
6. Issue a GET against smallitemreport. The result will be a JSON object, showing all the small items, ordered by name, and the date of the last transaction.

## Client side solution – What should it look like when it is done?

* <https://youtu.be/8ZVjwfQFLlE>

## Instructions

1. Did you notice the “PUT YOUR NAME HERE” box at the very beginning of this document? If you haven’t already done so, replace “NoName McHiggens” with your name. ***Every student is expected to turn in their own document.***
2. Specify your group number in the box provided.
3. Put the names of your three project buddies in the second box, replacing the silly names there.
4. Complete your Web Service.
	1. <https://youtu.be/J_L-mKYVVns> (video 1 cleanup + expectations)
	2. <https://youtu.be/D8Rg59MlJ-A> (video 2 the scancheck feature)
5. Test your web service with the Thunder Client.
	1. <https://youtu.be/fmUqcB_OXpQ> (Very important for those server-side buddies)
6. Complete your HTML / CSS / JavaScript mock-up.
	1. <https://youtu.be/Lcco32sgdQ0> (video 1 baseline)
	2. <https://youtu.be/aiUsvbbKt-4> (video 2 links and clean up)
7. Third Party API Integration and FH Web Services Integration.
	1. <https://youtu.be/FjgHDQ8vrjg>
	2. <https://youtu.be/JbNms0yaKF8>
	3. <https://youtu.be/4lXTGxenutQ>
	4. <https://youtu.be/CRzPuFb7I_E>
8. Testing. Make sure that your solution works. Make sure it is free of HTML errors and A11y problems.
	1. <https://youtu.be/8ZVjwfQFLlE>
9. Evaluate your peers. (Your instructor will email you a survey on or near the project due date.)

**CONTINUED…**

## Turn in your work!

When you are done, upload this document to the corresponding “Project 3” assignment out on canvas.

Be sure to fill out the boxes below.

***Every student must submit a project document.***

Box A: The URL to your solution (should be on S3, should be the same for the whole team).

|  |
| --- |
|  |

Box B: The URL to your web service API endpoint (from Lambda, should be the same for the whole team):

|  |
| --- |
|  |

Box C: The code to your web service (from the index.mjs file on AWS Lambda, should be the same for the whole team)

|  |
| --- |
| <<COPY PASTE YOUR AWS LAMBDA CODE HERE >> |

## How will this project be graded?

|  |  |
| --- | --- |
| Item | Points |
| Your name / Step 1Your group number / Step 2Your group members / Step 3 | 5 |
| BOX A – Functionality (point deductions in 5-point increments)* Login
* Logout
* Client list
* Check out
* Inventory list
* Clean Navigation (may be multiple instances)
 | 30 |
| BOX A – A11y check comes back clean. | 5 |
| BOX A – HTML is free from errors. | 5 |
| BOX A – Client-side code is clean. For example: unused div tags (like div-AAA) were removed from HTML, and related JavaScript code was removed as well. | 5 |
| BOX B – Web service documentation is complete and accurate. Don’t forget to edit “created by” and “last modified by” | 5 |
| BOX B – Web service code is clean. No undocumented features. Code related to missing / unused features removed. | 5 |
| BOX B – The web service feature for reset performs all the expected actions. | 5 |
| BOX B – Corrected the security bug in the auth feature. | 5 |
| BOX B – The web service feature for smallitemreport is functional. It does not duplicate small items records. It returns itemcode, name, description, estimated value, quantity, category, and last transaction date in the expected sort order. | 10 |
| Your contributions to the project ( 0, 5, 10, 15, 20) | 20  |
| **TOTAL** | **100 points** |

***Please see the syllabus for the late policy.***