# Project 3 – Fostering Hope Inventory: Admin

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

In this project we will use the same database that we developed in Project2. The goal of Project 3 is to create an interface that can be used by users with the admin role to add new big ticket items to the database, print a page of QR code labels, view an individual invenrory item, and (if necessary) delete an item from the inventory.

Screen shots of various pages are provided at the end of this document.

## Terminology

Same as Project 2.

## Instructions

1. Delete your old GROUP AWS access KEY and Password. (Did this in class.)
2. Generate a new GROUP AWS access KEY and Password. (Did this in class.)
3. 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.
4. Specify your group number in the box provided.
5. Put the names of your three project buddies in the second box, replacing the silly names there.
6. Identify what third party APIs your project will require. Project 3 will require us to use the third-party API found here: <https://goqr.me/> Test it out. Look at the API documentation page.
7. Meet with your buddies, decide who will do what. (Did this in class)
8. Get yourself set up for group work. (Just like Project 2)
9. Set up your database using the sql script provided on the class site. (Did this in class)
10. Create your Web Service using the start file provided on the class site. (Did this in class)
11. Test your web service with the Thunder Client. (Ongoing)
12. Concurrently:
    1. Go get the HTML start file and edit it, creating div tags for every element of the solution and navigation. (We started this in class, but there’s more work to do!)
    2. Create the functionality described in the API feature list. If you add to or revise a feature, you should document it in the API feature list and communicate with your team.
    3. Integration.
13. HOT TIP! My prediction is that, on the Lambda side, the challenge will be to create what I called the “makeItem” supporting function. The tricky part is generating a good item code. This will require you to think in steps.
    1. Create the new item record. Don’t forget that your new record must have an availability of “Y”.
    2. Generate an item code that is the concatenation of a random 5 digit sequence of letters and numbers, concatenated with the newly created item code.
       1. I used MySQL’s md5 function to create my random sequence, but other approaches are ok.
    3. Update the record you just created with the newly created itemcode
14. ANOTHER HOT TIP: My prediction is that, on the client side, the challenges are to:
    1. Validate all the input necessary to create a new item. Don’t call the API if the data is obviously bad.
    2. Do not allow the user to advance to the confirmation div if the API call fails. Show a user-friendly error message instead.
    3. If you are trying to figure out how to place a piece of text on a specific part of the page, try using a <SPAN></SPAN> tag. Use jQuery to assign the inner html of a span tag.
    4. Rendering the printable labels will be tricky too. That’s why I will spend extra time on this in class.
    5. Regarding the DELETE Item feature, you must send both the userid to the API
15. Testing. Make sure that your solution works. Make sure it is free of HTML errors and A11y problems.
16. Evaluate your peers. (Your instructor will email you a survey on or near the project due date.)

## 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 >> |

CONTINUED …

## How will this project be graded?

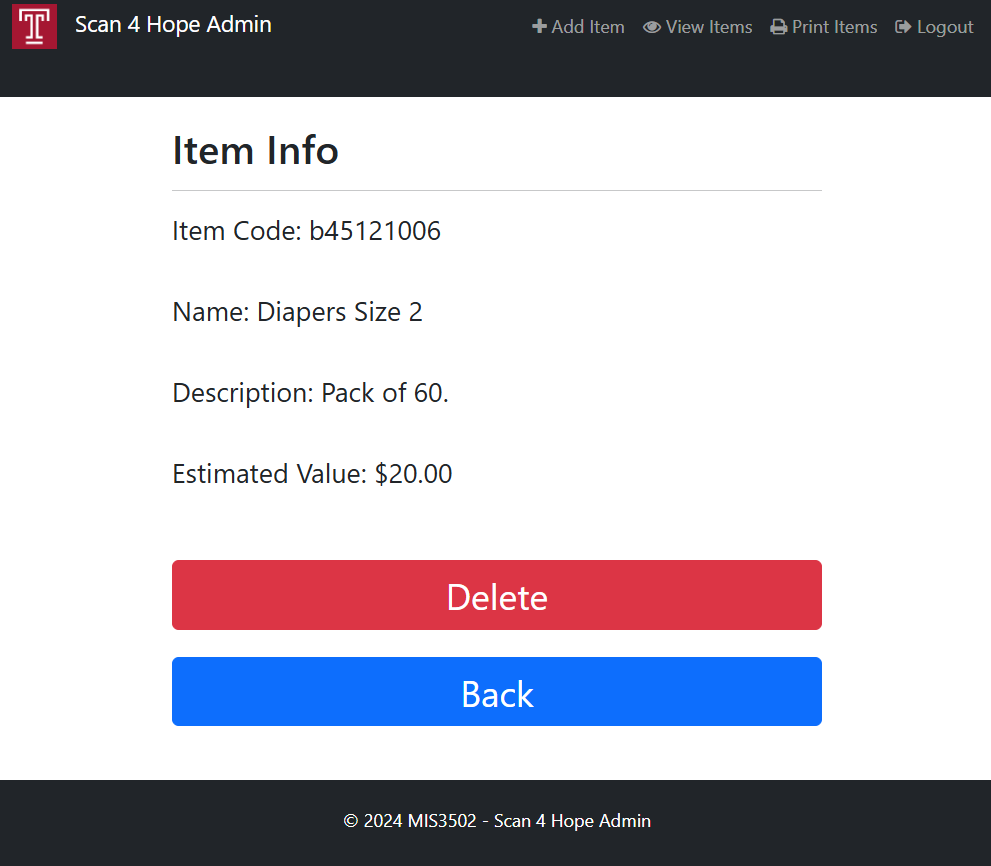
**Shafer’s Advisory**: If I determine that one or more members of a project team were not participatory, then I reserve the right to assign those students **a project grade of zero**. In those rare cases, I am not obliged to use the rubric below. Similarly, I am not obliged to give every student on the team the same grade.

If there is any question or debate regarding who did what work, my decisions will always favor those students who were seen ( by me) in class working on the project and participating constructively

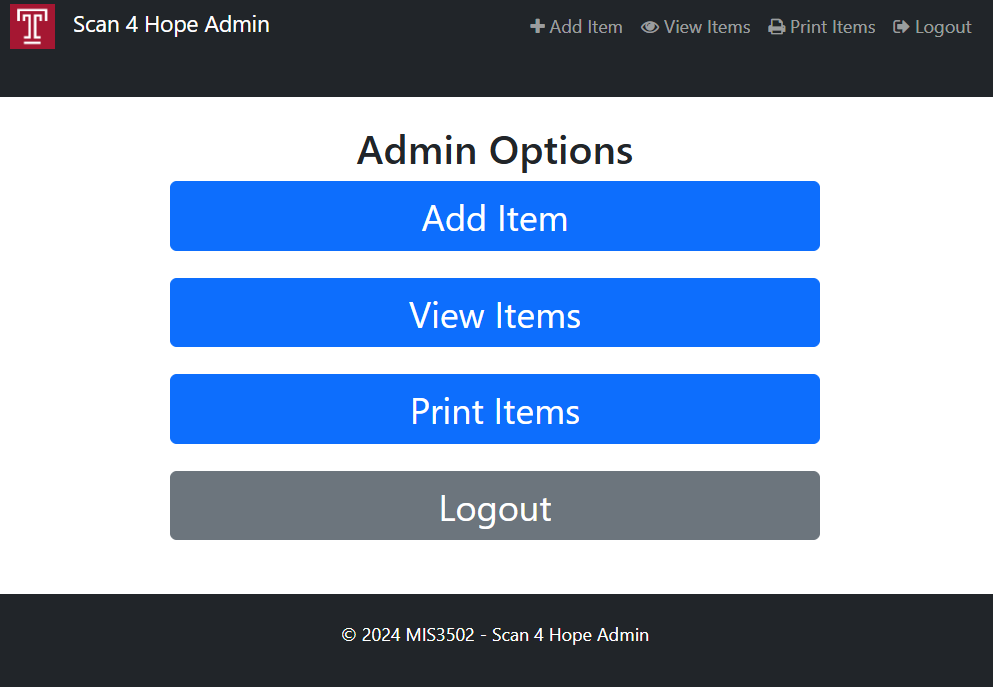
|  |  |
| --- | --- |
| Item | Points |
| Your name / Step 1 Your group number / Step 2 Your group members / Step 3 | 5 |
| BOX A Functionality (point deductions in 5-point increments)   * Login * Logout * Inventory List * Add an item * View an item * Delete an item * Clean Navigation (may be multiple instances) * Good error trapping (may be multiple instances). This includes both client-side and server-side error trapping. * Error trapping implemented on the Web Service features. (e.g. bad data does not make your web service fail with Internal Server Error and/or misleading error messages. | 50 |
| BOX A A11y check comes back clean (0,5) | 5 |
| BOX A HTML is free from errors (0,5,10) | 10 |
| BOX B Web service documentation is complete and accurate. (0,5,10,15,20) | 10 |
| Your contributions to the project ( 0, 5, 10, 15, 20) | 20 |
| **TOTAL** | **100 points** |

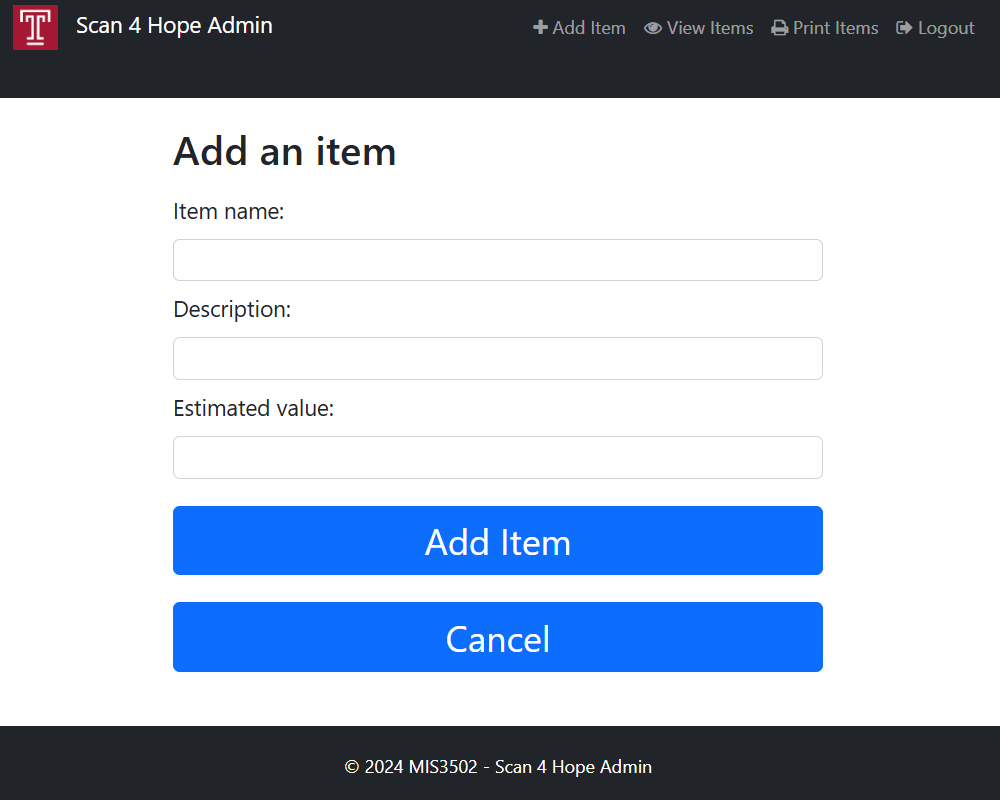
Please see the syllabus for the late policy.

SCREEN SHOTS ( In no particular order)

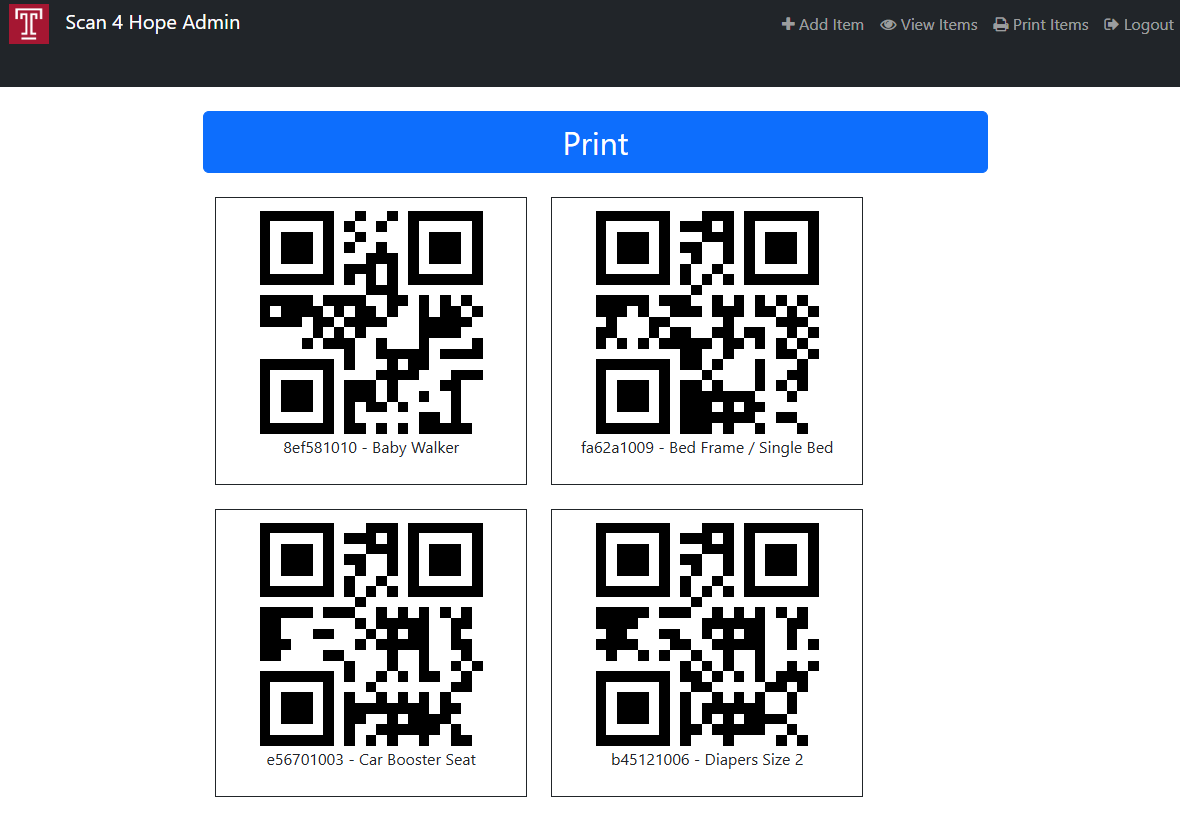


Use <span> tags here.   
  
Note the formatting as currency.





Error trap here. If a problem, then put an alert in a message div.



This page is longer than as shown here…

Our target label dimensions for printing are: Avery Labels Product # 95940

