GNG 2101

Design Project User and Product Manual

Guide to Using the Asset Inventory Manager System

Submitted by:

Scan n' Go – B1.1

<TEAM MEMBER 1, STUDENT NUMBER>

<TEAM MEMBER 2, STUDENT NUMBER>

Matthew Hadjis, 300356139

Nick Mesquta, 300361908

Leonardo Atalla, 300352589

12/03/2024

University of Ottawa

Table of Contents

List of Figures

Insert your list of figures here (right-click to update this field).

List of Tables

List of Acronyms and Glossary

Table 1. Acronyms

Acronym	Definition
AIM	Asset Inventory Manager
UPM	User and Product Manual
BCHC	Bethany Children's Health Center

Table 2. Glossary

Term	Acronym	Definition

1 Introduction

This User and Product Manual (UPM) provides the information necessary for users to effectively use the Asset Inventory Manager (AIM) system and for prototype documentation.

In this document, you will first find an overview of the product, the problem it's solving, and specific requirements and dependencies to begin the downloading process. Then, you will find the steps to download, setup, and begin using the system as well as how to navigate it. Following that is an overview of the functionality of the system ranging from basic features to more advanced shortcuts for experienced users. The next section is an in-depth guide on troubleshooting the system and what steps to take based on specific errors that may occur. Then, an overview of the development process of the product just before the conclusion and plans for the future of the system.

The purpose of this document is to provide an understanding of the AIM system, an easy-to-follow guide for installing the system, and more specific information regarding the AIM. The AIM provides two main services, first being the general inventory management system and second being the scanning functionality allowing employees to check-in and check-out specific assets within the system. The target audience for this document are people who need to manage many assets but has trouble keeping track of locations, specific assets, etc.

When following the steps regarding downloading external software, ensure you are downloading from the correct website to not install anything unwanted onto your machine.

2 Overview

When speaking to a client about the system they were using to handle numerous assets, we discovered 2 major problems with inventory management systems on the market. The first problem is that they typically scale in cost based on number of assets, and the second is that in a system with numerous assets, it's difficult to keep track of who has what and where everything is. In fast-paced work environments, 5 minutes lost in trying to sign out an asset is major. The AIM we designed aims to combat these problems by reducing the time it takes to sign in/out assets through an intuitive, easy-to-use barcode scanning system. As well, as the system is run locally, there is no cost in using the system, even when scaled.

For the user to use the AIM, the user needs a functioning computer that can connect to the internet as the AIM is web-based. If the user wants to use the barcode scanning functionality, then a barcode scanner capable of scanning EAN8 barcodes is needed along with a USB port to connect the scanner to the computer.Figure 2.0.1 - AIM Admin Login Page

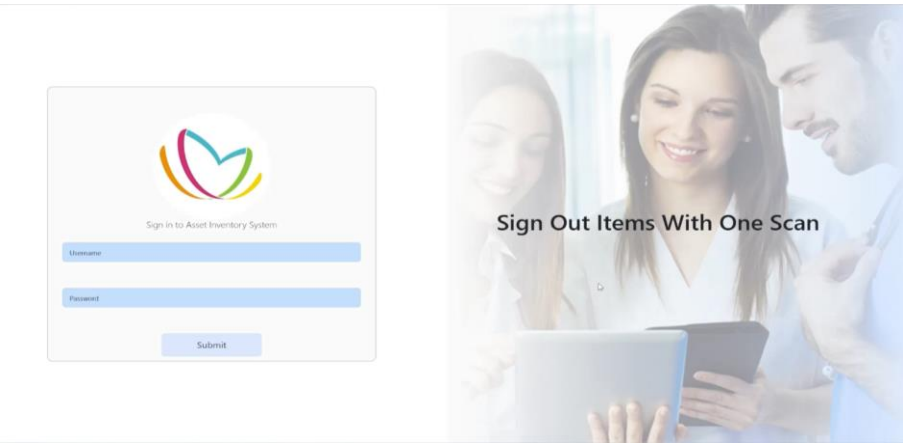


Figure 2.0.2 - AIM Admin Dashboard Page

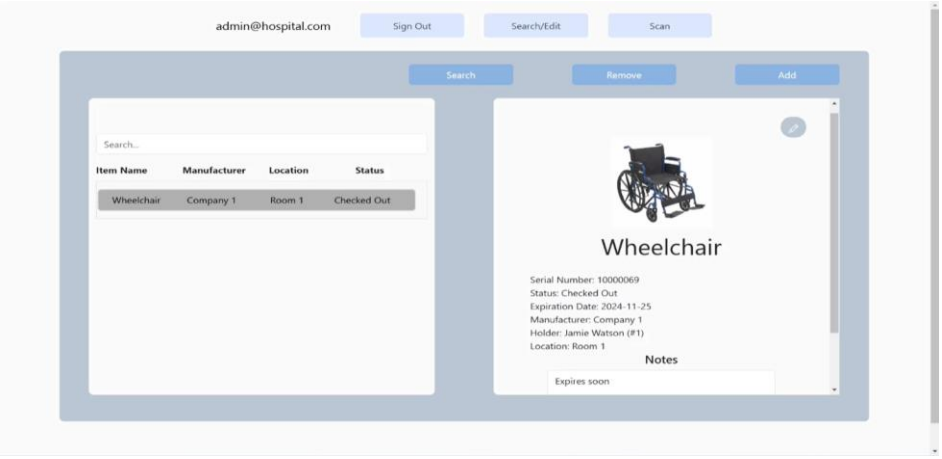


Figure 2.0.3 - AIM Scan Page for Signing Out an Asset

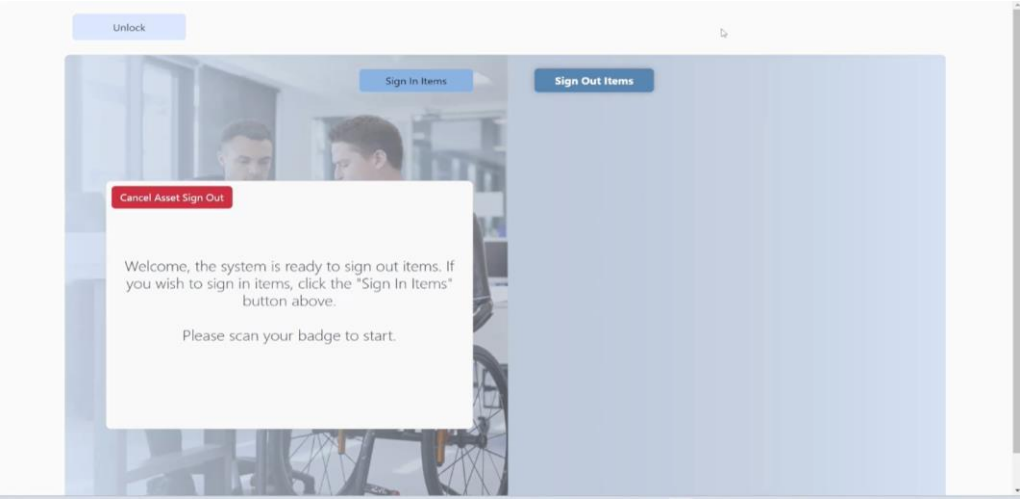


Figure 2.0.1 shows the login page of the system. Admin accounts are used to login as generally not everybody who has access to the check-in/check-out process should have access to the adding/editing/removing of assets. Once logged in, the user is sent to the admin dashboard page (Figure 2.0.2) where an admin can add, remove, or edit assets. There is a search bar used to search for specific assets and a scan button which will send you to the scan page (Figure 2.0.3). Once on this page, the user is locked there until the unlock button is pressed, which will send the user back to the login page if confirmed. On the scan page, users can check-in or out specific assets. This is done by first scanning the employee barcode, then scanning the asset they want to take/put back, then scanning the employee barcode again as confirmation.

2.1 Conventions

Due to the system being run locally, a good amount of setup needs to take place to get the AIM running, meaning there will be lots of commands to run to download dependencies, start servers, etc. It will generally be very clear what command needs to be run. If using Windows (the client has stated they are on Windows), commands are generally done through the command prompt.

A written command may look like: cd <directory>. The part that isn't in < > is what literally needs to be written into command prompt, and the part in the < > is where the command is dependent on your situtation. For example, cd means change directory and whatever you put as <directory> will make command prompt start looking at that file location. For example, if "cd Desktop" is entered into command prompt, it will change its directory to the desktop (if the path exists).

3 Getting started

To use the AIM, there are two main processes that need to be run, the backend server and the frontend server and will both be run on separate command prompts. A general run through of the startup process opening two command prompts, traversing to the folder/directory in which the AIM is stored for both, and then running the respective frontend and backend servers. Once the frontend server begins, a website will automatically popup where you can then start using the system. Closing the system is as simple as stopping the local servers which can be done by closing both command prompts or using specific keyboard shortcuts which will be explored later.

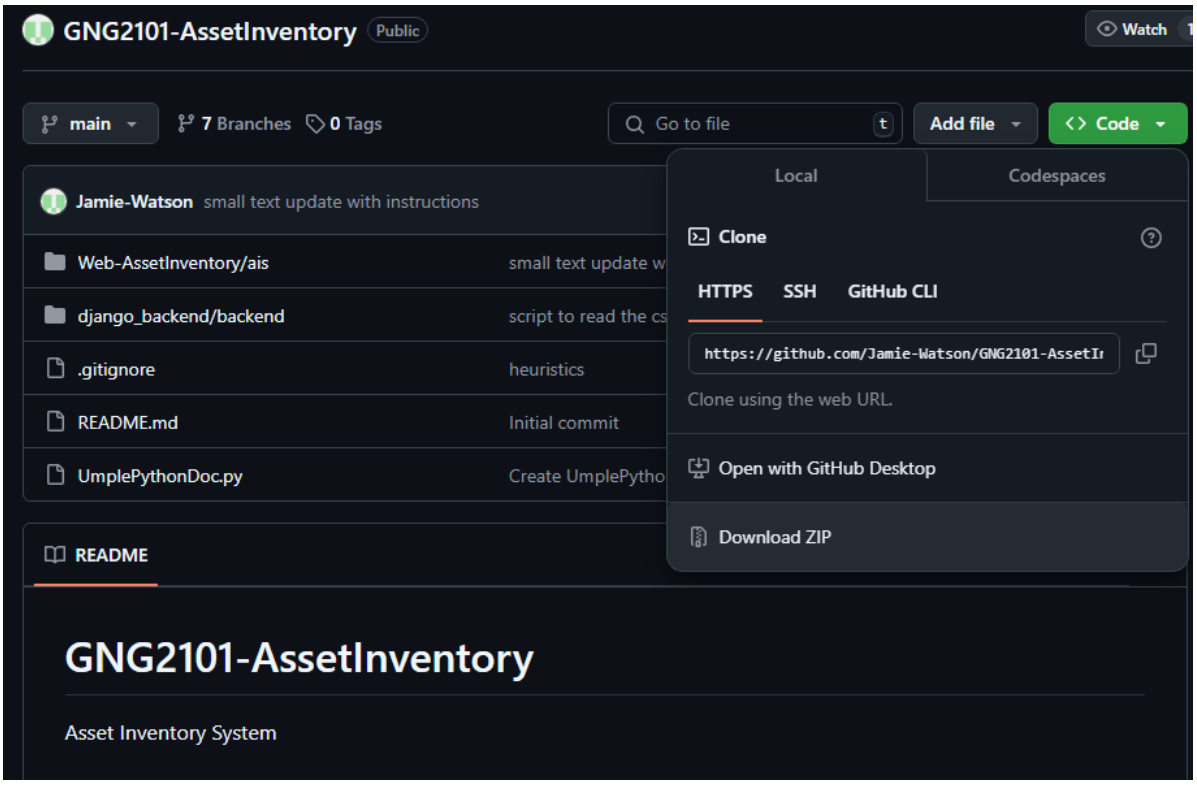
For both the backend and frontend, there are many dependencies and programs that you will need to download to get the AIM to function, namely Python and its pip library for the backend, and Node.js for the frontend. This will be explored now.

3.0.1 Downloading the AIM from Github Repository

We first need to download the AIM files from the Github repository and put them into a folder that will be used to hold everything pertaining to the system.

1. Create a new folder named “AIM” on your desktop which will hold everything pertaining to the system.
2. Go to this repository link, click on the green “code” button, and then "Download ZIP". This will download a folder containing the AIM.

Figure 3.0.1.1 - Download ZIP Option



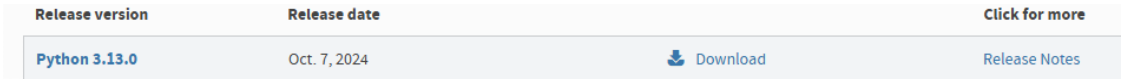
3. Open the ZIP folder and then open the next folder, you should find five files. Copy the files and paste them into the AIM folder you created earlier.

3.0.2 Backend Dependencies and Setup

In this section, we will download Python, pip, and create a virtual environment to hold and download the AIM backend dependencies.

1. If Python is not already installed, go to https://www.python.org/downloads/ and click download on the latest version (the AIM was developed in Python 3.11.7). Then press "Windows installer", choose 64x or 32x depending on your system.

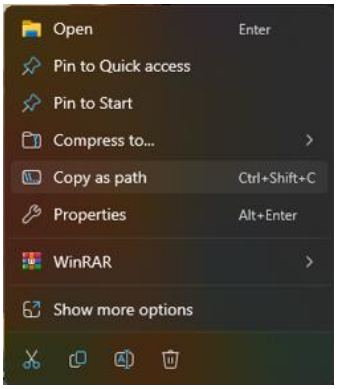
Figure 3.0.2.1 - Python Installation



2. Run the installer. Once the process is complete, you can verify the installation in a new command prompt by running the command "python --version".
3. The pip library for Python should have been installed along with Python itself, test this by running "py -m pip --version". If pip isn't found, then run the command "py -m ensurepip --upgrade" which will install pip if not already installed.
4. Run the command "py -m pip install virtualenv". This will download the Python library used to create a virtual environment.

5. Right click the “AIM” folder on the desktop and click “copy as path”. This will copy the folder’s address. Run the command “cd <paste what you copied here>”, which will make your command prompt point to the new folder.

Figure 3.0.2.2 - Copy as Path Option



6. Run the command “py -m venv vir”. This will create a virtual environment with the name vir. You should see a new folder with the name vir in your AIM folder.
7. Test that the virtual environment works by running “vir\Scripts\activate”. This will activate the virtual env. You can tell that its working because (vir) should show up in the command prompt line.

Figure 3.0.2.3 - Virtual Environment Activated

```
C:\Users\ricar\OneDrive\Desktop\AIM>vir\Scripts\activate

(vir) C:\Users\ricar\OneDrive\Desktop\AIM>
```

8. Once the virtual environment is activated, run the command “cd django_backend/backend”.
9. Run the command “py -m pip install –r requirements.txt”. This will download every needed backend dependency onto the virtual environment.
10. Run the command “py manage.py makemigrations”, then “py manage.py migrate”.
11. Finally, running the command “py manage.py runserver” should make the backend start running. If your command prompt looks the same as the image below, you are done.

Figure 3.0.2.4 - Backend Server Started

```
(vir) C:\Users\ricar\OneDrive\Desktop\AIM\django_backend\backend>py manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
November 30, 2024 - 15:52:48
Django version 5.1.2, using settings 'backend.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

3.0.3 Frontend Dependencies and Setup

Assuming the project has already downloaded node.js, the frontend can be run quickly.

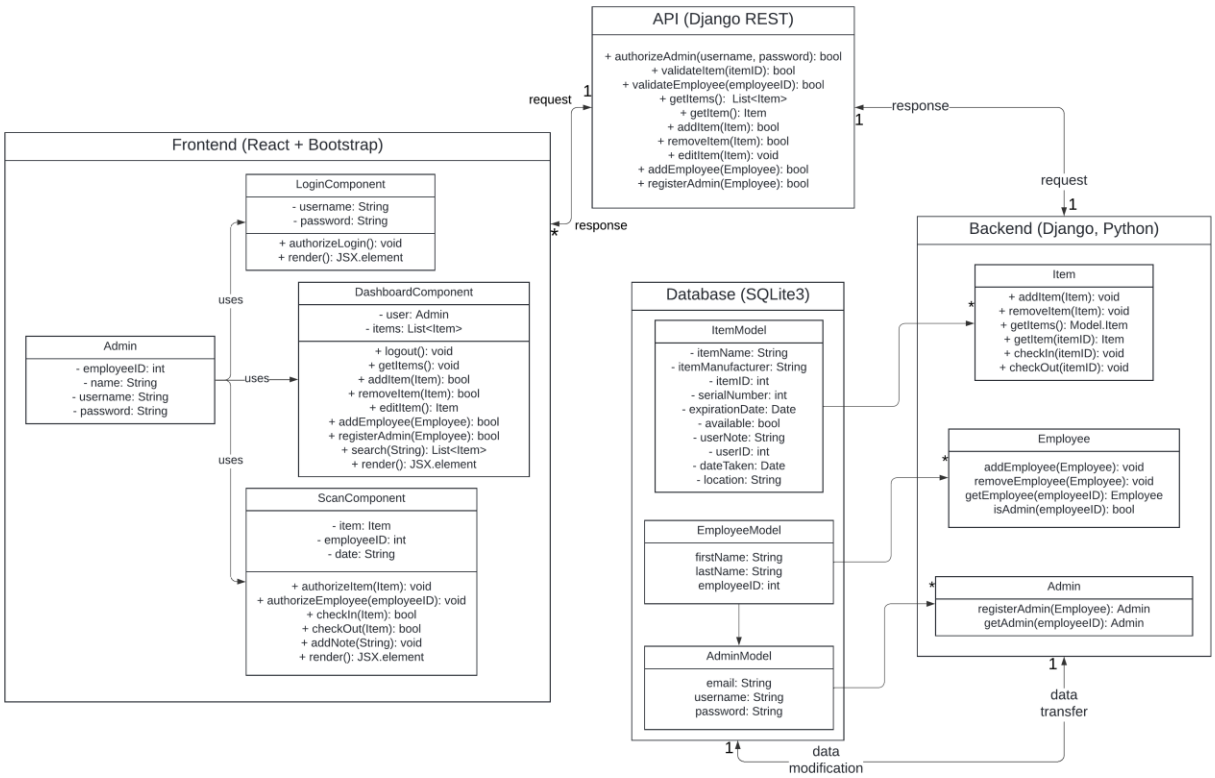
1. Navigate to the directory titled “AIS”.
2. Run command “npm install”.
3. Run command “npm run start” which will start the app and open it in the default browser.

3.1 Configuration Considerations

This system makes use of a frontend, API, backend, and database for all its functionality. The frontend is what the user sees and interacts with. It is what takes the users input. Then, the backend is what processes the user’s input and performs the necessary output. Both components run on and communicate through a local host, However, since the frontend and backend are separate, they need to be connected somehow.

The API is what sends information between the frontend and the backend. It can be seen as a translator of information between both components. A typical flow for the API is it first takes a request from the frontend, formats and sends it to the backend where it will then be processed, then an appropriate response is sent back to the frontend from the backend based on the success or failure of the request. Then, the database is what stores all data on assets, employees, and admin accounts. It can only be accessed from the backend. The diagram below shows how the components of the system are linked to each other.

Figure 3.1.1 - UML Class Diagram of the AIM



3.2 User Access Considerations

Generally, there will be 2 types of users of this system. The first is admins who should have access to all aspects of the system and should also be the only people with login credentials. The other people using the system should be whoever needs to sign in or out assets and should only have access to the scan page. The admin will set the system to the scan page and leave it there for an employee to scan whatever they need.

3.3 Accessing/setting up the System

To start the system once the entire setup process has happened, you need to run both the frontend and backend servers simultaneously. This is done using 2 separate command prompts.

The first command prompt will be for the backend. Performing these actions will start the backend server:

- Right click the AIM folder on your desktop and copy the path (Figure 3.0.2.2).
- In your command prompt, run the command “cd <paste what you copied here>.

- Start the virtual environment by running “vir\Scripts\activate”.
- Traverse to the backend by running “cd django_backend/backend”.
- Start the server by running “py manage.py runserver”.

After this, open a new command prompt in which the frontend server will run after performing these actions:

- Right click the AIM folder on your desktop and copy the path (Figure 3.0.2.2).
- In your command prompt, run the command “cd <paste what you copied here>.
- Traverse to the frontend by running “cd web-assetinventory/ais”.

Start the server by running “npm run start”.In terms of login, a base account is made when the system begins for the first time with email “admin@hospital.com” and password “123”. Use this to login for the first time.

3.4 System Organization & Navigation

The first screen that is seen when the system starts is the login page. Logging in with the credentials above will bring the user to an instruction page on all basic functionality of the system. If the sign out button is pressed at any time, then the system will go back to the login screen.

The first thing you can do from the instruction page is navigate to the admin dashboard by pressing the Search/Edit option at the top of the screen. Here, users can see and search for every asset in the system, add assets, remove assets, and edit assets. It is a reliable asset management tool. Every time an asset or employee is added to the system, a barcode is generated for them. The barcodes are to be printed and used using a barcode scanner.

The second functionality is the scanning system which you can get to by pressing the Scan button at the top of the screen. Once the button is pressed, the system will be locked into scanning mode. When you want to exit scanning mode, the system will send you back to the login page. The scanning page will listen for barcode scanning and depending on whether you’ve chosen the sign in or sign out option, will allow people to take out or give back assets. These are the three main windows where all all of the functionality of the system can be accessed.

3.5 Exiting the System

Turning off the system is as simple as closing the command prompts, however, if you don’t want to restart the whole setup process for each of the command prompts, you can cleanly stop each server by doing these things.

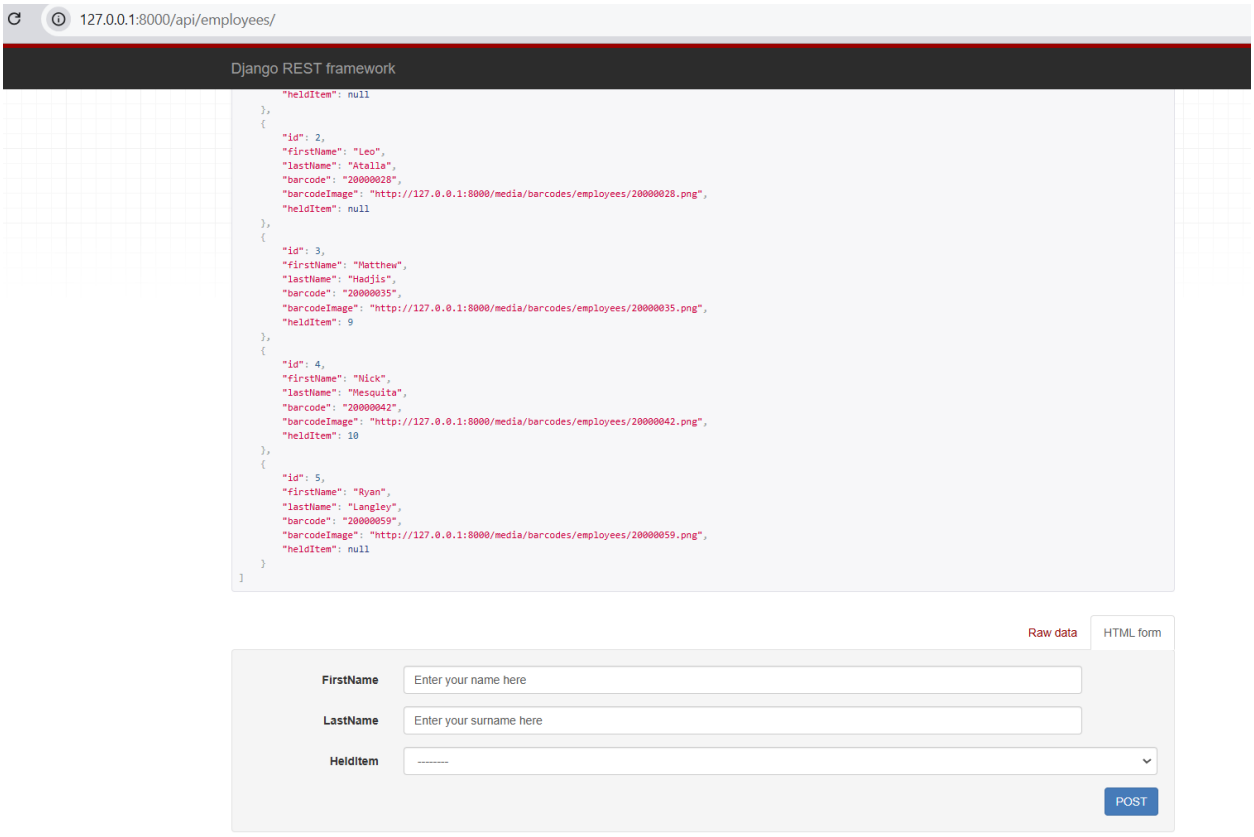
For the backend, simply using the CTRL + C keyboard shortcut on the command prompt running the backend server will end the server. For the frontend, doing CTRL + C will prompt you with a yes or no (Y/N). Typing Y and then Enter will then end the server.

4 Using the System

4.1.0 Admin Login:

To use the AIS, the user must first create an admin account, which will provide access to the system. Creating an admin account requires the user to first create an employee account. To begin this process, navigate to: http://127.0.0.1:8000/api/employees/. This is the location of the server, as previously mentioned in figure 3.0.2.4. Enter your first and last name into the server fields, ensuring to not edit the last field. To confirm the creation of an employee account, click the POST button.

Figure 4.1.0.1 - Visual of Creating an Employee Account



After the creation of an employee account, the upper half of the screen will display the account ‘s information. Note the ID of this account.

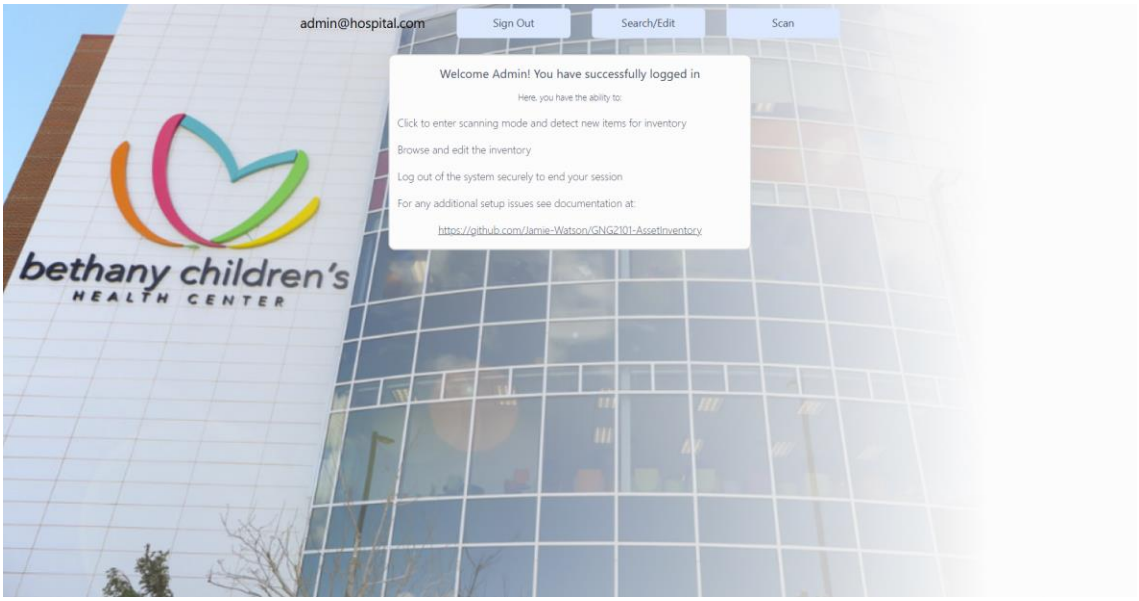
Figure 4.1.0.2 - The Employee Creation Information

```
HTTP 201 Created
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "id": 6,
    "firstName": "sample name",
    "lastName": "sample surname",
    "barcode": "20000066",
    "barcodeImage": "http://127.0.0.1:8000/media/barcodes/employees/20000066.png",
    "heldItem": null
}
```


With the employee account created, you can now proceed to create an admin account. To begin, navigate to: http://127.0.0.1:8000/api/admins/. This process similar to creating an employee account: enter the username, email, and password details. The key difference is selecting the correct employee object, titled “Employee object (<Your ID>)”. After completing the fields, click the POST button to add the admin to the system. To log in as an admin, open the AIM and you will see the login page (as shown in figure 2.0.1). Enter the username and password used during admin creation. Successful login will be indicated by the instruction page, which will show your chosen username in the upper left-hand side of the screen.

Figure 4.1.0.3 - Instruction Page



4.1.1 Admin Dashboard Page

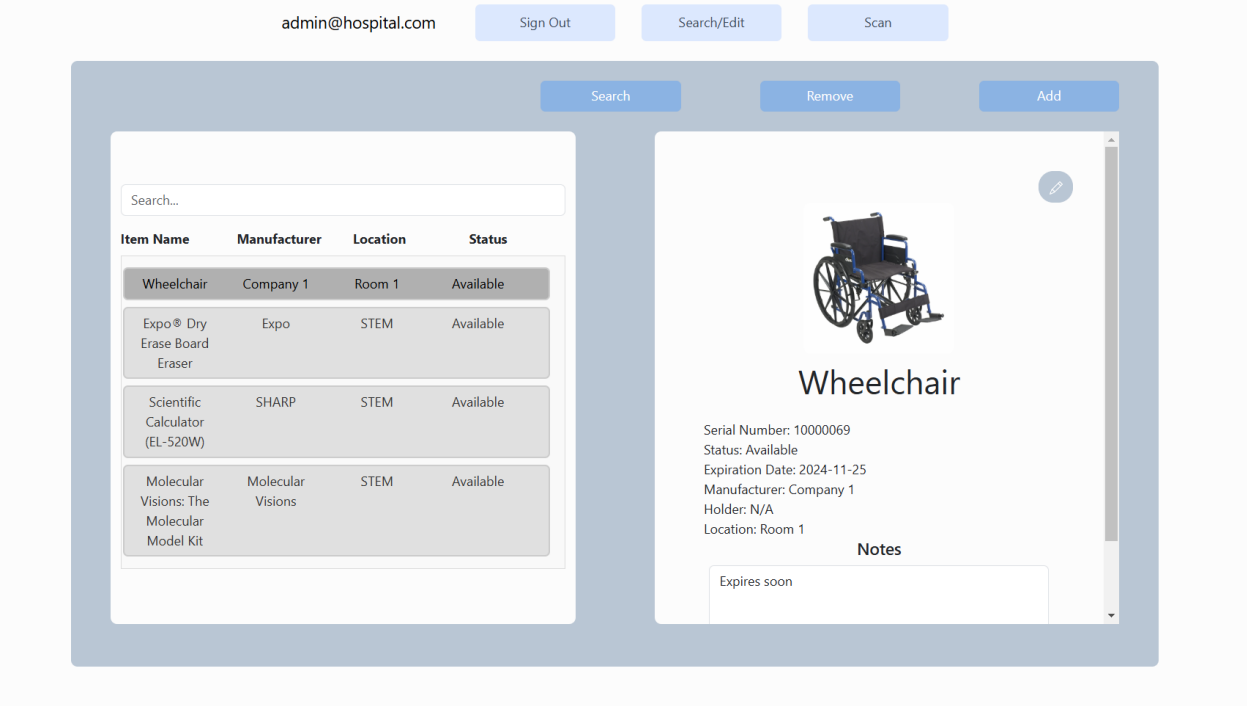
The admin dashboard can be accessed by clicking the Search/Edit button in the upper navigation bar. This will take you to the page shown in figure 2.0.2.

4.1.2 Searching for an Asset

Upon entering the admin dashboard, the AIM will default to asset search. To search for an asset, type the name of the asset into the search bar. Once the desired asset is found, additional

information can be viewed by clicking on the asset. After doing this, the asset information will be shown on the right-hand side of the screen.

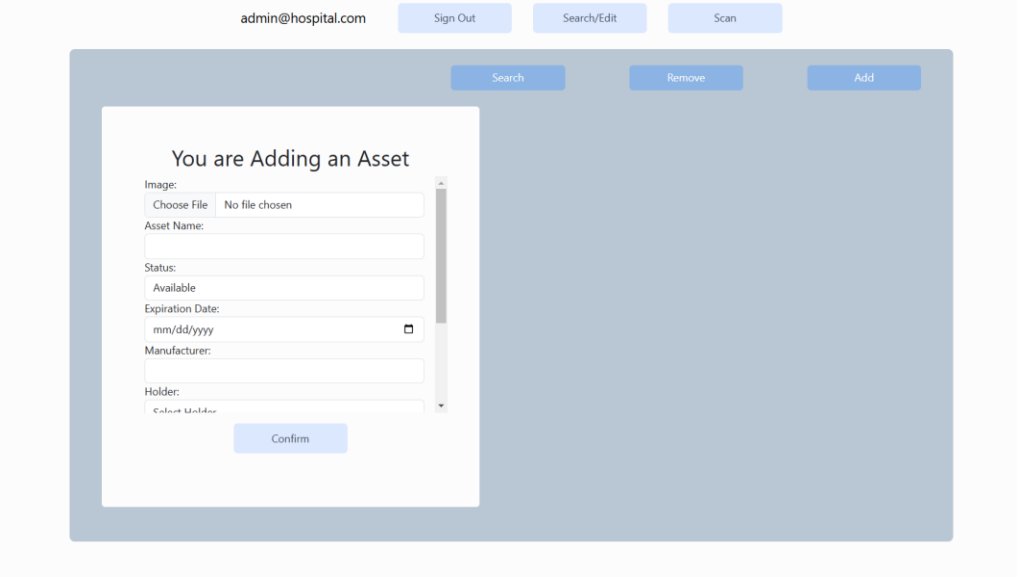
Figure 4.1.2.0 - Additional Information Shown for Asset



4.1.3 Adding an Asset

To add an asset, click the add button in the lower navigation bar. This will bring the user to a page allowing for the input of asset information, which includes an image, name, status, expiration date, manufacturer, holder, location, and notes.

Figure 4.1.3.0 - Add Page

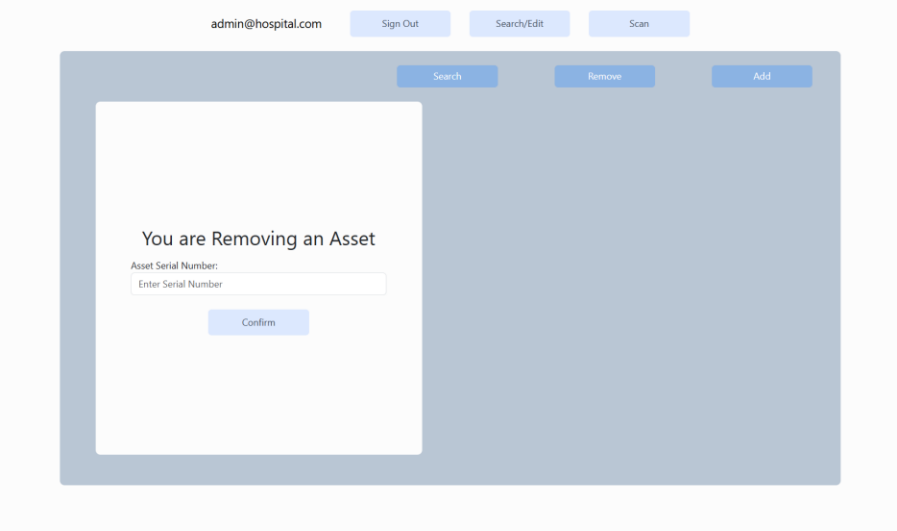


Note that only a name field is required to add the asset. After completing the fields, click the confirm button at the button of the screen. A pop-up will indicate that the asset has been added successfully.

4.1.4 Removing an Asset

To remove an asset, click the remove button in the lower navigation bar. This will navigate the user to a page allowing for the removal of all asset information.

Figure 4.1.4.0 - Remove Page



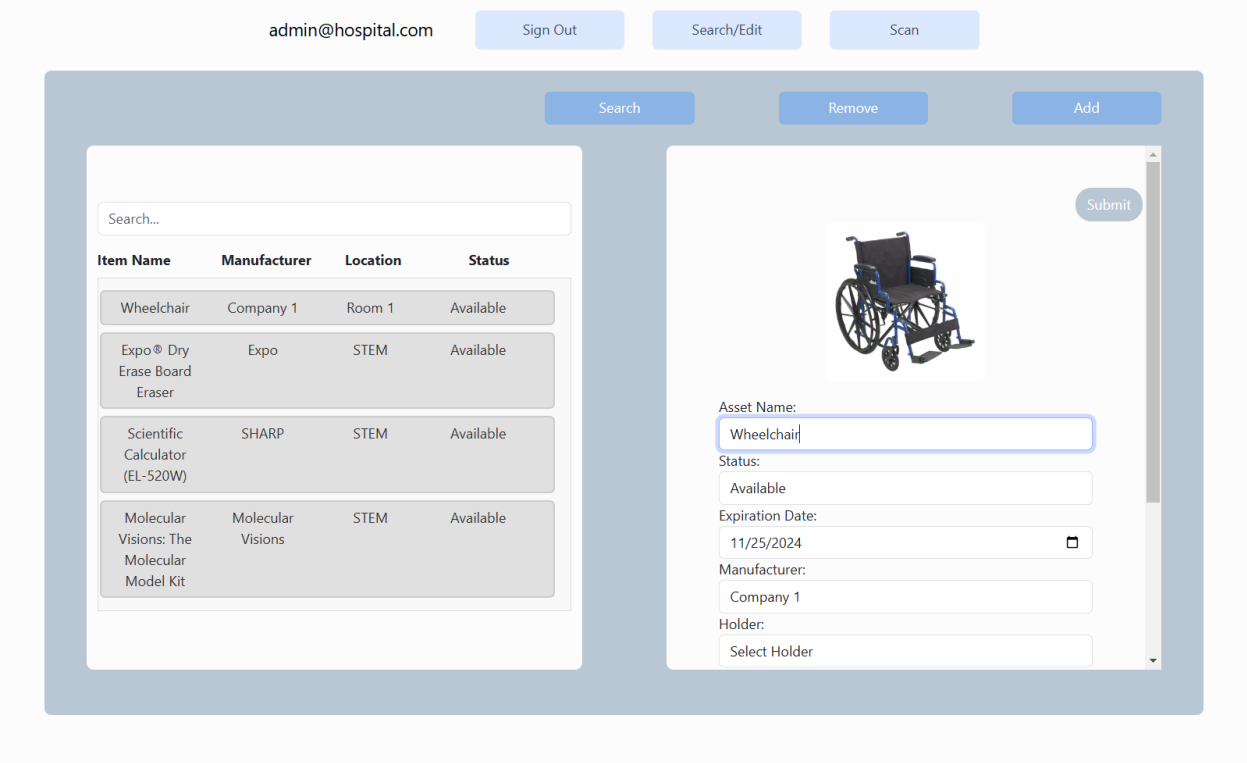
To remove an asset, enter the serial number and confirm the removal. To verify the serial number of the asset, navigate back to the search page by clicking the Search button and find asset details

by searching as indicated in Figure 4.1.2.0. After clicking the Confirm button, a pop-up will appear to confirm the deletion.

4.1.5 Editing an Asset

To edit an asset, click the Seach button in the lower navigation to access the search page. Search for an asset and access its additional information as detailed in 4.1.2. Click the pencil icon in the upper-right corner to allow fields of the asset to be editable.

Figure 4.1.5.0 - Editing an Asset



Once the edits have been completed, click the Submit button which replace the pencil icon button. This will confirm changes.

4.1.6 Scan Page

The scan page can be accessed by clicking the Scan button in the upper navigation bar. This will take you to the page as shown in figure 2.0.3. Note: Navigating to the scan page will lock the program until the unlock button is clicked. Unlocking will require the administrator to sign back into the system before use.

4.1.7 Sign in Assets

To sign in an asset, the user must have access to their badge, the asset, and the corresponding barcodes for each. Upon entering the scan page, the system will default to signing

items out; however, this setting can be adjusted by clicking the Sign In button or using the shortcut CTRL + (CTRL + SHIFT + =). The page will prompt the user to begin by scanning their badge. Once this is completed, the instructions will update to either explain an error or direct the user to scan the barcode on their asset using the barcode scanner. After the asset is scanned, the final step is to scan the badge again to confirm the sign-in. If these steps are completed correctly, the system will notify the user that the scan has been successfully completed. Note: The system will reset after 5 seconds, allowing the next user to scan items. Should the user wish to cancel the process at any time before completion, this can be done by clicking the Cancel button or using the shortcut CTRL + X.

4.1.8 Sign out Assets

To sign out an asset, the user must have access to their badge, the asset, and the corresponding barcodes for each. Upon entering the scan page, the system will default to signing items out; however, this can also be accessed by clicking the Sign Out button or using the shortcut CTRL -. The page will prompt the user to begin by scanning their badge. Once this is completed, the instructions will update to either explain an error or direct the user to scan the barcode on their asset using the barcode scanner. After the asset is scanned, the final step is to scan the badge again to confirm the sign-in. If these steps are completed correctly, the system will notify the user that the scan has been successfully completed. Note: The system will reset after 5 seconds, allowing the next user to scan items. Should the user wish to cancel the process at any time before completion, this can be done by clicking the Cancel button or using the shortcut CTRL + X.

5 Troubleshooting & Support

As the AIM system is run locally, the only changes that could occur after the system has been installed successfully that would cause errors would have to be done manually which is incredibly rare. As such, most errors will occur during the installation phase.

During the installation phase, if any command including “py” or “py -m” is not understood by the command prompt, first run “py –version” to see if Python is installed correctly. If that shows that py is unrecognized, then try running “python --version”. If that works, replace “py” with “python” every time its seen in this manual. If the command doesn’t work, reinstall Python.

When trying to run the system for the first time and the website is giving a full screen error, these are the steps to take. First, end the backend server and run “py -m pip install –r requirements.txt”, “py manage.py makemigrations”, and “py manage.py migrate” before starting the server again. Next, end the frontend server and run “npm install” before running it again. This should ensure all dependencies are installed.

If there is still an error in the system, there are two courses of action, The first is to run “py manage.py flush”. This will delete all items saved in the database which may fix the problem if there is an item in the database that can’t be processed by the backend. However, if many assets are saved in the database, it would not be ideal to lose them all.

The second course of action is to redownload the AIM ZIP folder from the Github repository and follow the installation process again. Once you’ve downloaded the new AIM, find the db.sqlite3 file which is in folders django_backend, then backend. Copy and paste it in the same location that it was in the previous AIM folder. Try running this server. If this still doesn’t work, contact support (section 5.4).

5.1 Error Messages or Behaviors

If there are no assets showing up in the admin dashboard or barcodes that should be in the system are not recognized by the system, this means there is a network error. This typically means the backend is not running as it's supposed to. This is fixed by restarting the backend server. The same is true if you encounter errors trying to add or remove assets in the system.

5.2 Special Considerations

The installation process is made for Windows systems and may not work with other operating systems.

5.3 Maintenance

As the system is run locally, there is no real maintenance that needs to be performed unless an update is downloaded. If that's the case, then for the backend, run "py -m pip install –r requirements.txt", "py manage.py makemigrations", and "py manage.py migrate" before starting the server. For the frontend, run "npm install" before running.

5.4 Support

If there are persisting errors, you can email aim_helpdesk@hospital.com to report the problem and receive guidance. Provide as much detail on the error message as possible, including but not limited to the error message produced by the system, screenshots of what's going wrong, and what you've already done to try to fix the error.

6 Product Documentation

6.1 <Subsystem 1 of prototype>

6.1.1 BOM (Bill of Materials)

Table 3. Bill of Materials

Item Name	Description	Units of Measure	Quantity	Unit Cost	Extended Cost	Link
VS Code	Source code editor capable of coding in multiple languages through its extensive library of languages and features.	Free download	5	0	0	https://code.visualstudio.com/
Barcode Scanner	Scanner to use to test scanning integration and functionality	Online purchase	1	$39.99	$39.99	https://amazon.ca/dp/B088QV215Y

6.1.2 Equipment list

6.1.3 Instructions

6.2 Testing & Validation

The first set of testing that was done was related to the functionality of the prototype 1. The critical assumption was that the system would be able to run efficiently with many assets and users, and two tests were conducted to verify this assumption. These tests were performance measurements that were tested against the criteria of the system's functionality when given a certain number of assets and users. In essence, two tests were conducted where a script was created to generate random users and assets and inserted them into the system. The number of users and assets was noted for each separate test when a system failure was noticed where there was visible lagging or a full crash. The minimum values to exceed were 2000 users and assets, and a system breaking point was displayed at 30 000 assets and 5 000 users. These results validated the critical assumption and ensured that the system would be able to handle many assets and users and still run efficiently.

Table 4. Prototype 1 Testing

Test Number	Reason for Prototype	Evaluation Criteria/Determine Measurables	Level of Prototype
1	Performance measurement	System Functionality (Number of Assets)	Medium/High Fidelity, Comprehensive
2	Performance measurement	System Functionality (Number of Users)	Medium/High Fidelity, Comprehensive

Kind of Prototype	Metrics	Test Description
Analytical	Assets: # before system failure	How many assets the prototype can support before the system fails
Analytical	Users: # before system failure	How many users the prototype can support before the system fails

	Analysis Method	Minimum	Expected Physical Maximum	System Breaking Point	Interpretation
# of Assets	Add assets until the system fails	2000	4000	30,000	This passes easily, the number of assets will not be an issue.
# of Users	Add users until the system fails	2000	3000	5000	This passes easily, the number of users will not be an issue.

The second set of testing was related to the functionality of the barcode scanner for prototype 2. Prototype 2 involved the integration of the barcode scanner into the system to ensure that the system would be able to read the information output by the barcode scanner. The critical assumption was that barcode scanner would work for all barcodes and that the system would be able to read the information output by the scanner. To validate this assumption, one test was conducted that aimed to analyze the system and barcode scanner's functionality when given random barcodes. In essence, a set number of barcodes were generated from the internet and scanned using the barcode scanner. If the barcode scanner was able to accurately read the information, and if the system was able to read the information scanned by the scanner, then that was considered as a success. After going through 10 barcodes, it was determined that the critical assumption was validated, meaning that the scanner worked and that the system was able to read the scanner's output.

Table 5. Prototype 2 Testing

Test Number	Reason for Prototype	Evaluation Criteria/Determine Measurables	Level of Prototype
1	Performance measurement	System/Barcode Functionality	Medium/High Fidelity, Comprehensive

Kind of Prototype	Metrics	Test Description
Analytical	Yes or No Value	Can the barcode scanner read the barcode and have this information sent to our system?

Analysis Method	Minimum	Interpretation

Scan the barcode and observe whether it outputs the text/value.	2	Our barcode scanner can read the barcodes that we give it, and it accurately outputs a value from the barcode.

7 Conclusions and Recommendations for Future Work

Throughout the course of the project, many lessons were learned. The coding between the front-end and back-end was a challenge, given the nature of the syntax between the two. Both JavaScript and React use non-conventional object-oriented patterns. Additionally, challenges arose when ensuring that the system would work across various inputs and screen sizes. Bootstrap also had to be learned to ensure adherence to the design-for-scalability DFX principle.

The most productive avenues going forward revolve around the deployment of the server on the cloud, specifically reducing deployment time. While the server can be deployed online currently, it takes a significant amount of time to do so. Decreasing this time adds more functionality to the AIS and adds to its possible use cases. Future groups can also seek to enhance the aesthetics of the AIS to their liking, or if user preferences change in the future.

If given more time, we would likely implement the cloud solution. Additionally, we would add all the assets and users that are currently present in the AIS that the BCHC to make the transition between systems smoother. The only item that had to be abandoned was the cloud deployment, in favor of ensuring that the local solution functioned as effectively as possible.

8 Bibliography

APPENDICES

9 APPENDIX I: Design Files

https://makerepo.com/rlangley/2112.gng-2101-b11-scanngo

Table 6. Referenced Documents

Document Name	Document Location and/or URL	Issuance Date

10 APPENDIX II: Other Appendices

You can include other critical and important work here. Maybe they are not important in the structure of this document but need to be included.