Project Final Report:

Inventory Management Tool

CS 157A Team 14

Khoa Phan

Jing Hui Ng

Renad Morrar

1. **Overview:**

Many businesses are having a problem to manage their inventory in an effective way. Our tool is to provide a friendly and useful tool in order to help those businesses be able to manage their inventory more effectively . In detail, the software will help companies to manage the stock of products with various functions such as generating reports, maintaining the balance of the inventory. Moreover, it will provide accurate information about the products without any error. In a general sense, the inventory system will work by allowing the user to manage and keep track of the stock. The most important goal that we are aiming for is providing a customer a better way to manage their business’s inventory more effectively, which will lead to save a lot of money and time for the companies.

**2**. **System Environment:**

**A. A Structure Diagram of the System:**

****

**B. Hardware & Software:**

* Back-end: **Node.js version: 10.16.3** (Javascript runtime environment), **Express.js** v4.16.1(Web application framework for node.js), **npm v6.9.0**
* Front-end: **React v3.0.1** (Javascript library for building UI)**, Material UI v4.4.3**
* Database: **MySQL Workbench v8.0 CE**
* Misc: Any type of web browser (Chrome/Firefox/etc), Linux or Windows environment, and Wifi connect capability
* Visual Studio Code/Atom
* Git & Github
* Slack

**C. Programming/Application languages used:**

* Front-end:React, Javascript, Material UI
* Back-end: Javascript

**3. Functional Requirements:**

**A.** **Functionality/Features:**

User can edit the information of the item or remove it from the stock. They also can see a history of a particular item. The user can add a new item to the stock. The search function will be implemented to help the user find the item by type of the product. The users also need to log in in order to have access to the management system. The users can sign up for an account. Users can sort the item based on the product name, price and type.

**B.** **User Usage:**

1. Scenario: User login

1. Access to the login page

· When users open the webpage they will have access to the login screen

· Log in screen will have the ID and password boxes provided

2. Fill in the id and password

· Users will have an ID and password, they will be able to enter the text within a box and will be able to have access to the system after logging in successfully.

3. Navigate to the homepage with account logged in

· With the account logged in, the user will be able to navigate directly to the homepage

**II**. Scenario: Searching Item

1. Write the type of item that the user wants to find in the search bar

· Search bar is provided for the user on the top screen of the webpage

· User will be able to enter text through the search bar and be able to find the item they entered within the search box

· Each item will have their own unique name and number to distinguish the item from other items.

2. A list of items will be displayed based on the type of items.

· After hitting the enter button to locate the item the user entered, there will be a list of items related to the users search

**III**. Scenario: Edit the information of an item

1. Find the item on the product list

· The main page will display a list of items. User needs to find the item and click on the edit button of that specific item.

2. Edit the information

· User will be able to edit the information of an item by typing a new information into some text box such as product name, quantity, type and price.

3. Click the Submit Button

· A button with the sign “update” will be located at the bottom of the page in order to update the quantity the user has of the items

**IV**. Scenario: Remove Item

1. Search for the item that needs to be removed

· In the users inventory list, search for the item they wish to remove

2. Click the remove button

· Locate the item, and next to the item on the side their will be a “remove” button

3. Click the update button

· A button with the sign “update” will be located at the bottom of the page in order to update the removed item

**V.** Scenario: Records

1. Search for the item

· User should first search and locate the item they wish to view

2. Click view history button

· For that item, there will be a “View History” button located near the item

3. History

· This report will be able to view if there was any changes in quantity of an item, when it was changed, and or if the price has been changed.

**4. Non-functional Issues:**

1. **GUI:**

* It should be a Web Application with an user-friendly environment that will be built by using React & Material UI, CSS
* When the user first logged in, the user will be able to navigate directly to the homepage by clicking the special homepage button which will contain a list of items in the inventory.
* On the dashboard, the user will also see a simple search box with a magnifying-glass button that is displayed on the top middle. Also, search box will be provided on every page to ensure that the user is able to find the content they are looking for.
* Auto-suggest mechanism is provided when the user is typing in the search box by predicting what the user trying to look for to speed up the searching process.
* When the user selects on one item, it will show a detailed information of the item selected.
* User is able to edit the quantities of the selected item by changing the number and then simply clicking the update button.
* They should be able to see the pie charts and graphs when the click on the view report button on the web page in order to make stock control and inventory visibly easy, and allow the user to quickly see new trends, and help user to easily visualize the flow of the items.

5. **ER Diagram Design:**

****

**Employees:**

- This entity represents employees and it will have a name and a unique id as attributes.

- An employee is either a trainee or a has a permanent position (1-1 relation, each employee can only have one account).

**Account:**

* An account requires password and unique id to log in.
* An account allows employees to search and manage the inventory by adding, decreasing or updating the inventory.

**Items:**

* Each item has its own item description and a unique item id.

**Records:**

* There will be reports that record the lists of change of the item.

**Has:**

* Each employee will have their own account to access to the inventory.

**Manage:**

* Accounts will have the manage permission to edit the inventory. This type of account can add, remove, or edit item.

**Search:**

* This allows users to use search tool in order to find a specific item based on the key term( category)

**View:**

* This relationship allows users to view the inventory without permission to edit anything.

**Report:**

* Each time a user makes a change in the inventory, the system will automatically report the change to the record, so that the users can keep track the change later.

**6. ER Diagram to Schemas:**

**I. Schemas for Non-Weak Entity Sets:**

Employees(EmpID, EmpName)

Accounts(Username, Password, userID)

History(ID, Event, pId, pName, Date)

Products(Quantity, pId, Price, pName, Type)

**II. Schemas for Non-Supporting Relationships:**

Has(EmpIDs, userID)

Manage(userID, pId)

View(userID, pId)

Search(userID, pId, Key term)

Sort(pId, userID)

Record(Id, pId)

**7. Normalization:**

**Relationship tables that are already in BCNF:** Has, manage, sort, search, view, record

**Functional Dependencies (all tables are in BCNF):**

Employees(EmpID, EmpName)

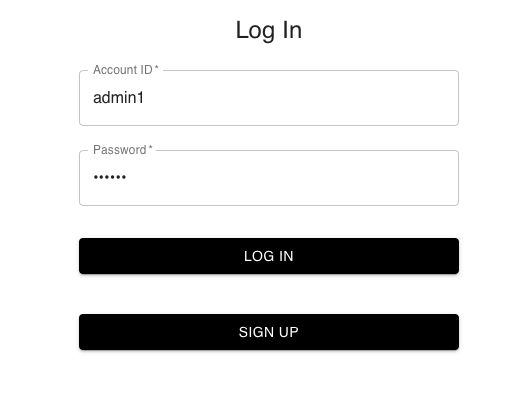
Accounts(Username, Password, userID)

History(ID, Event, pId, pName, Date)

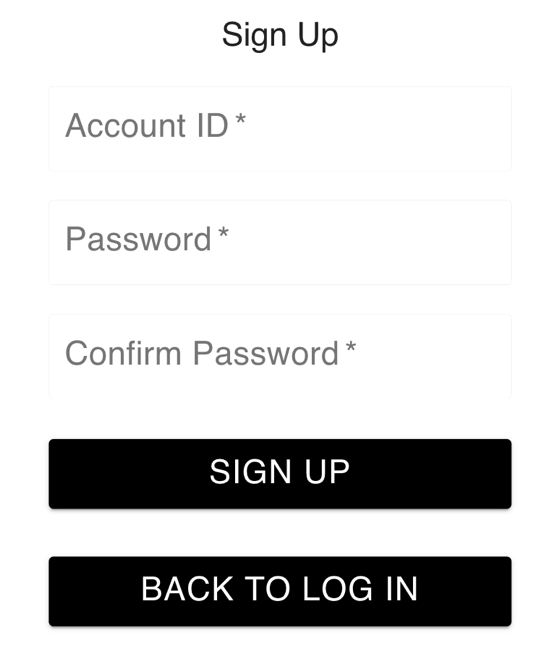
Products(Quantity, pId, Price, pName, Type)

**8. Screenshots:**

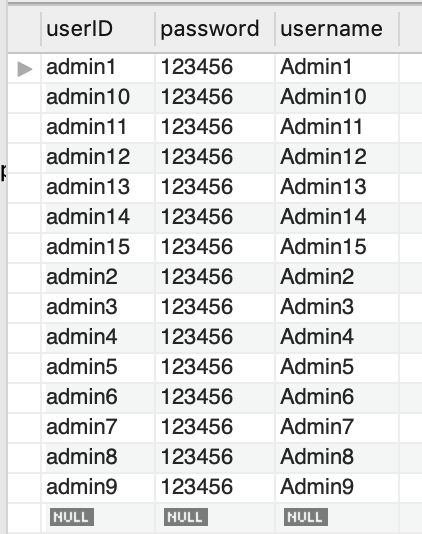
Login page:



Sign Up page:



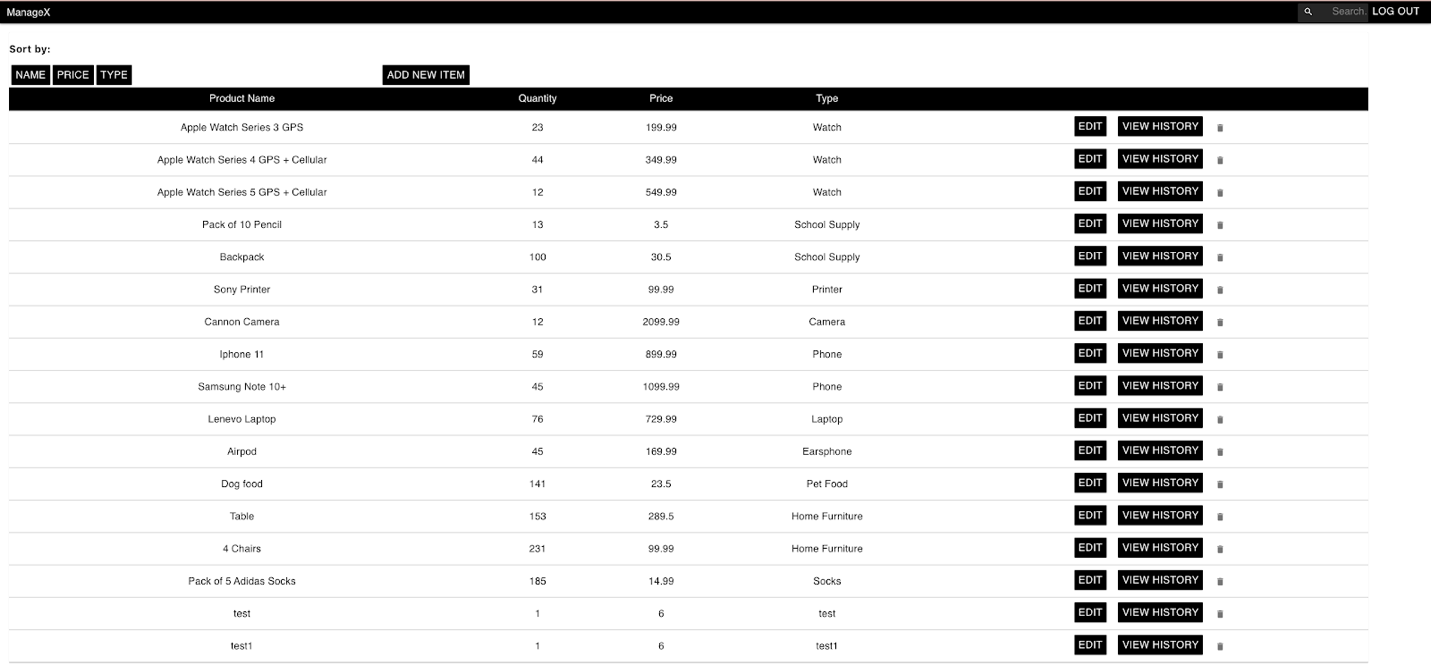
MySQL (before sign up):



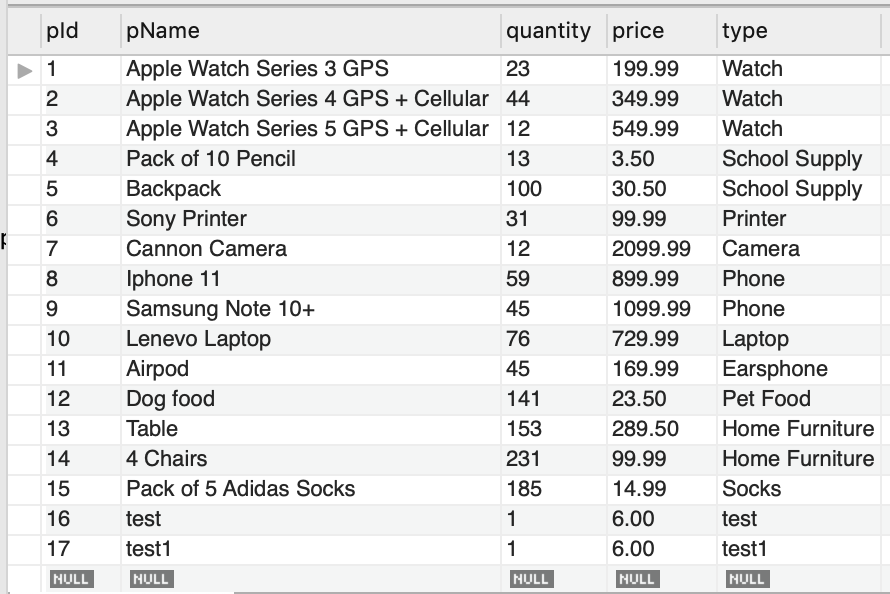
MySQL (after sign up):



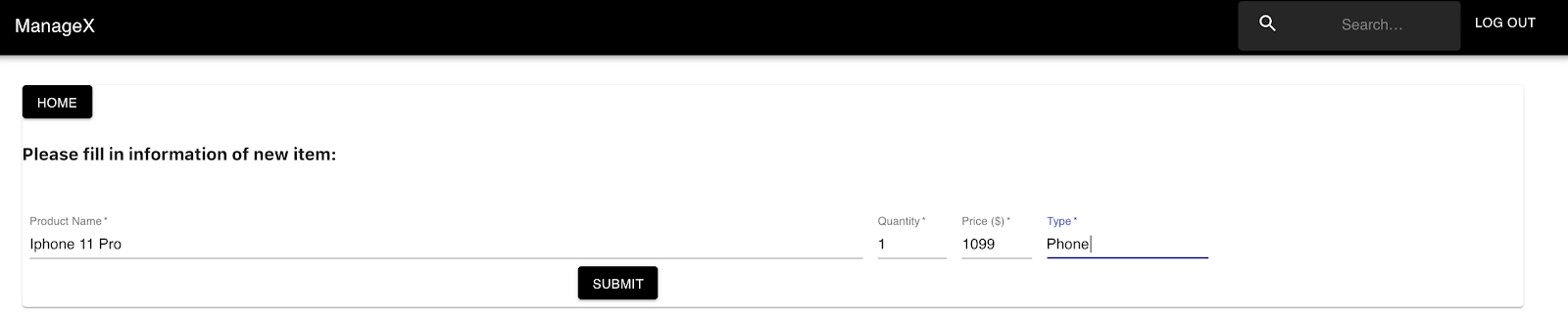
Home page:

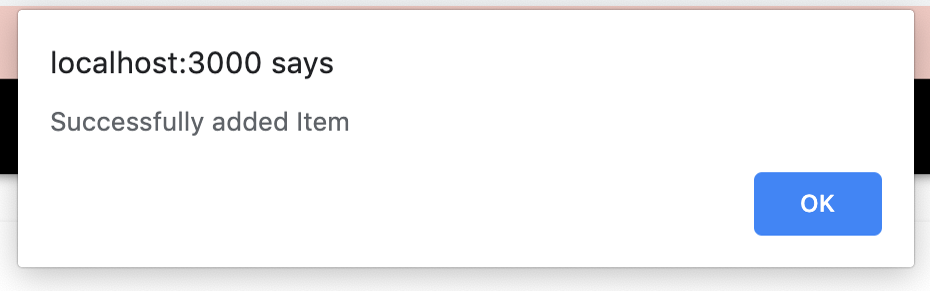


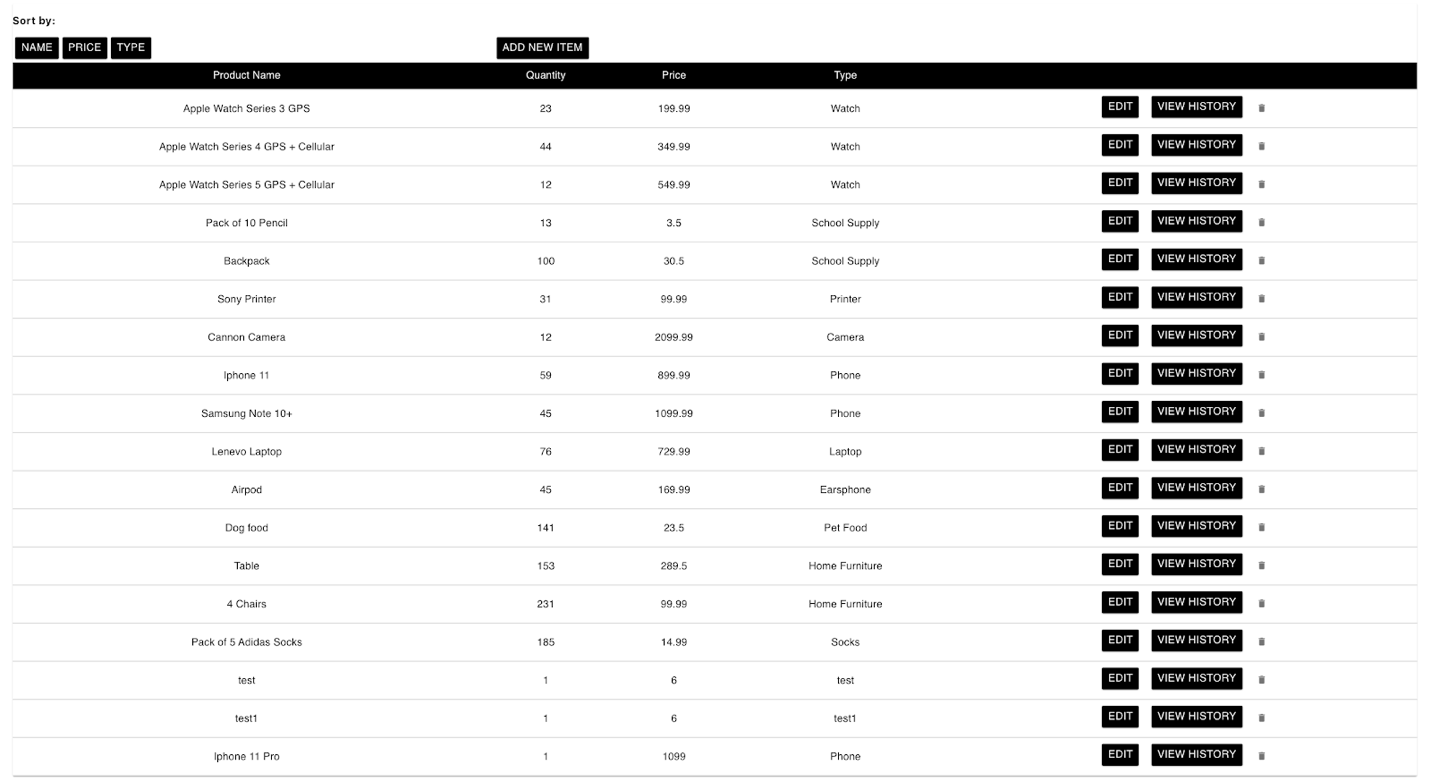
MySQL (before):

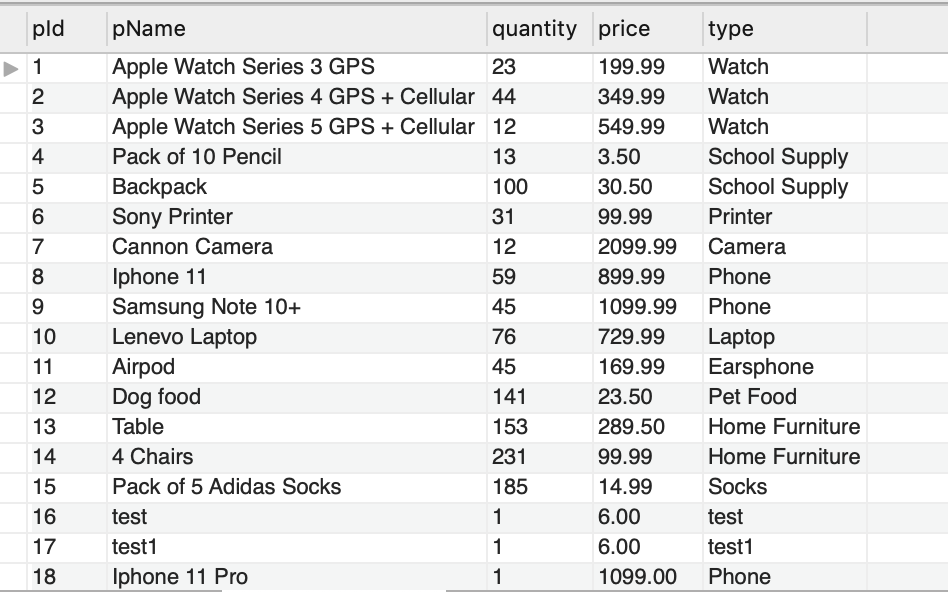


Adding an item (Iphone 11 Pro):

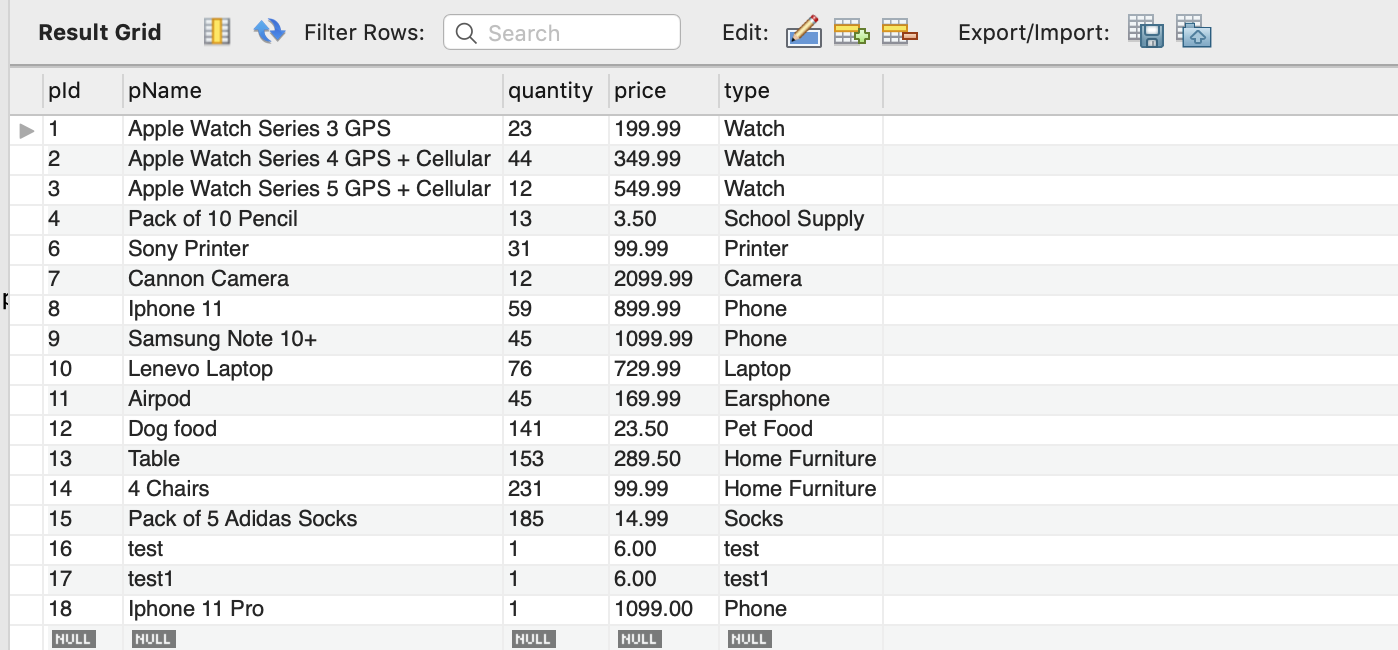




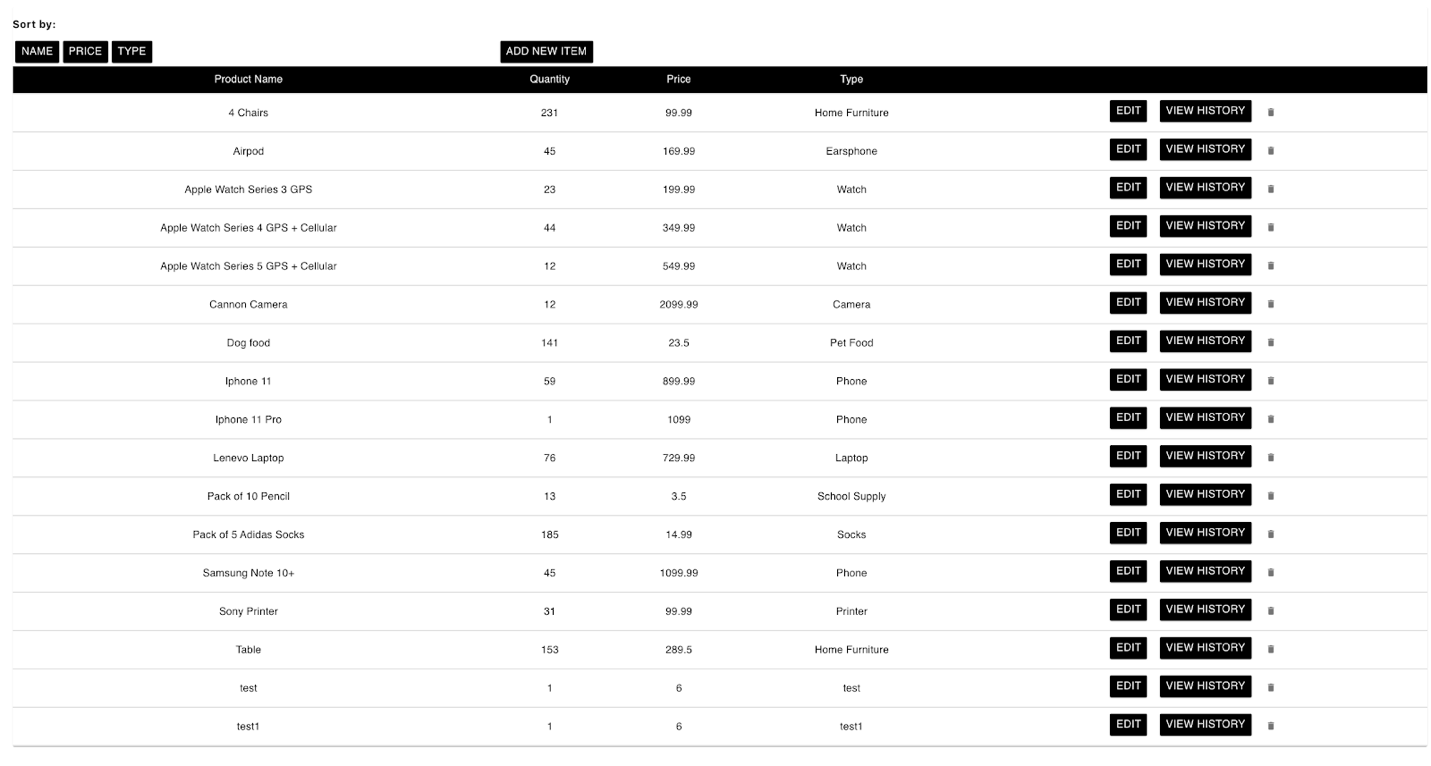




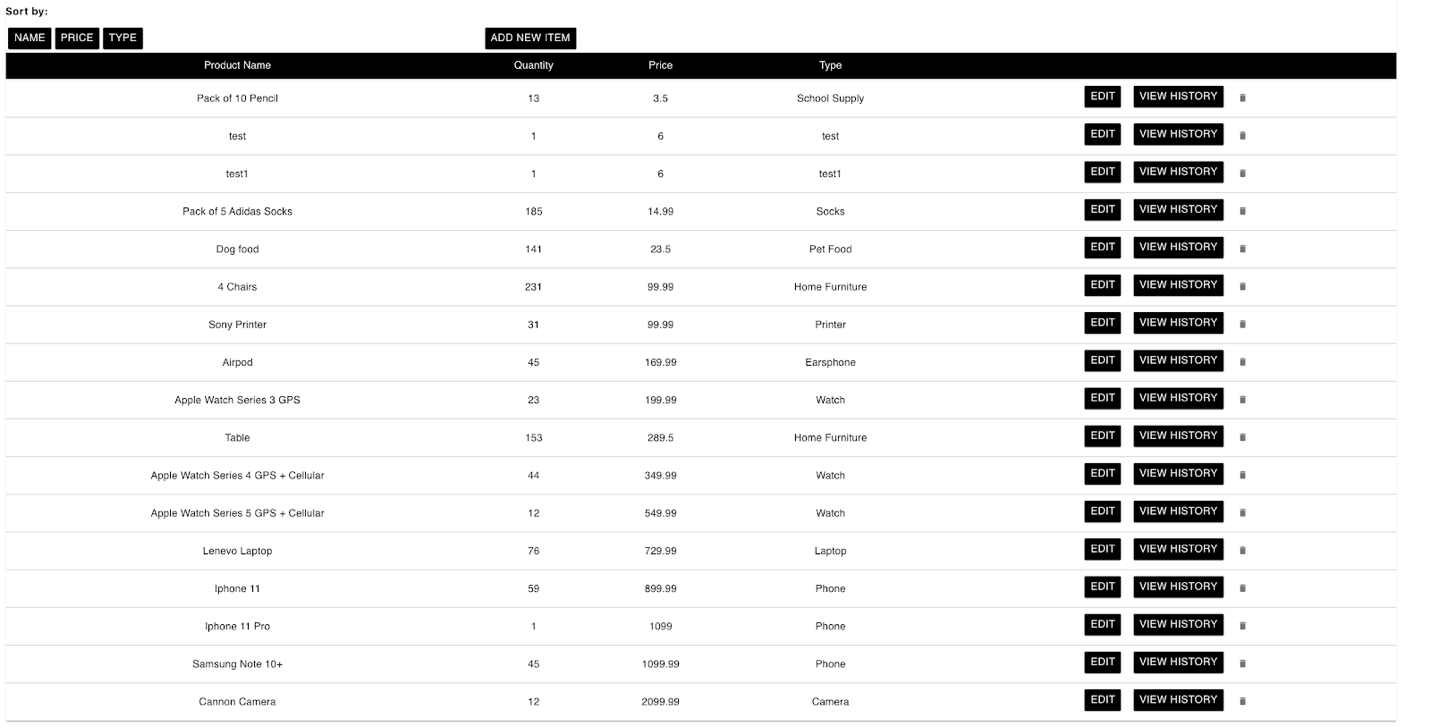
Removing an item(backpack with pId 5 has been removed):



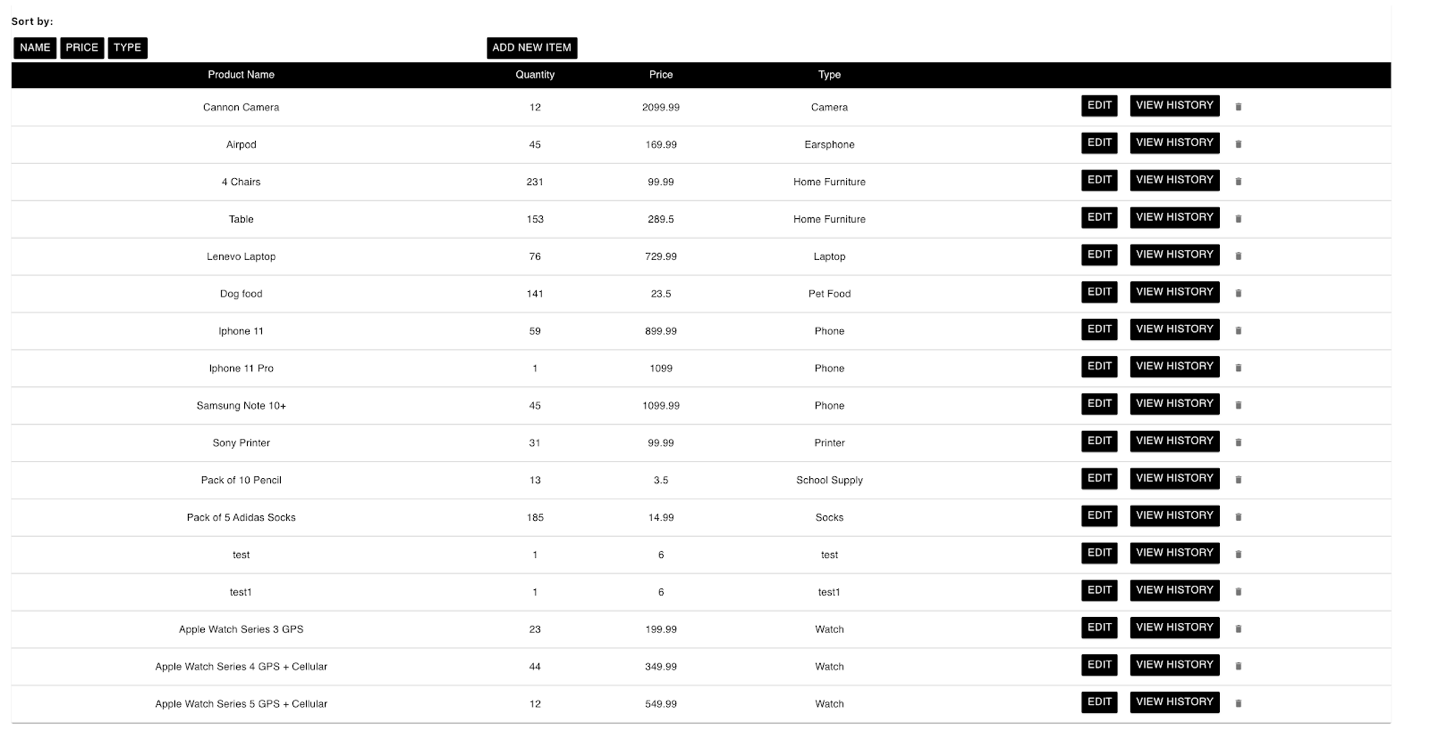
Sort by name:



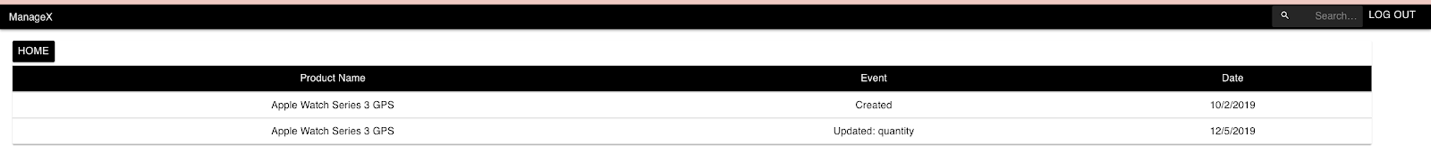
Sort by price:



Sort by type:



View history:



**How to run:**

1. Git clone <https://github.com/CS157A-Team14/DatabaseApplicationProject.git>
2. Open a query tab and paste the mySQL code (db\_script\_updated) from DatabaseApplicationProject/mySQL folder
3. Run the queries
4. Cd to client, npm install, then npm start
5. Cd to server, npm install, then npm start
6. Localhost tab will be opened and you are all set

**Lesson Learned:**

**Jing Hui Ng:** Throughout the semester, I learned to apply theoretical knowledge taught in the lecture in our database application using Javascript and mySQL. Most importantly I learned to communicate with teammates and to sketch the ER diagram so that our tables in the database will be correct. However, It was a struggle for me to update and alter the data in the mySQL database.

**Khoa Phan:** This project helps me have a better understanding on how to do a full stack web application. I am now quite similar using ExpressJS, ReactJS and MySQL to develop any application. I also have a chance to apply what I learn about MySQL to my project.

**Renad Morrar:** I have learned a lot in this database class and working with a group on a large project. I have never worked with databases before so learning all the different techniques and features for databases was interesting. I hope to continue to work with databases in the future and broaden my knowledge on the concepts. Every project will have obstacles and I did struggle with some aspects of our project, which is also how I learned, by struggling and trying to work through the struggles.