1. **Project Proposal (Due 6/11/2025)**

**Title**: Retail Clothing Stock Manager

**Team Members**: Alex Secor (Sas24e) and Jose Solano (Jls23k)

**Description**: A backend system for retail stores to track stock by size, brand, and type. This is done with a CRUD interface, which means Create, Read, Update, and Delete. For security reasons, users are prompted for a login username and password to modify any stock information.

**Problem statement**: Some clothing stores still manually register their inventory count on paper; this creates a bigger risk of human error in the form of miscounting and bad management when finding what items need replenishing because of their low stock. A retail clothing stock management program will enhance work efficiency by keeping track of the clothing inventory stock and notifying workers of what items are low on stock, making the job of retail workers easier while diminishing the chances of human error, which can sometimes cost the retail store money.

**Proposed Features:**

● CRUD Interface

○ Estimated Work Hours (20hr)

● Login / password for web page usage.

○ Estimated Work Hours (10hr)

● Inventory management (Stock alerts)

○ Estimated Work Hours (10hr)

● Create / Read/ Update /Delete (CRUD) inventory.

○ Estimated Work Hours (20hr)

● Data: products, sales logs, admin entries, and sales records.

○ Estimated Work Hours (20hr)

**Tech stack:**

● Frontend: HTML, CSS and JavaScript

○ We chose HTML as the assignment requires a web page.

○ CSS will play a large role in styling the website

○ JavaScript will help expand the functionality of the website.

● Backend: PHP

○ PHP is well-supported by our hosting provider (InfinityFree) and allows for rapid development of server-side logic.

● Database: MySQL

○ Provides structured storage and efficient querying for clothing inventory, users, and sales logs.

● Hosting: InfinityFree

○ Free web hosting services allows us to host our database and supports web programs that utilize PHP files.

**Target users and benefits**: The target users are business owners and retail workers who want to track the stock and sales and modify the clothing information stored inside the database through the interface. Simplifies job tasks for retail audits for employees and alerts for clothes on low stock. The overall benefit of the system is that it makes inventory management easier and more accurate, which helps forecast stock shortages and overages. A virtual stock management system allows a business to analyze sales and determine what’s working and what’s not in terms of hot items, etc.

**Benefits:**

● Data Analytics

● Stock Alerts

● Managing inventory levels

● Record keeping for future possible audits

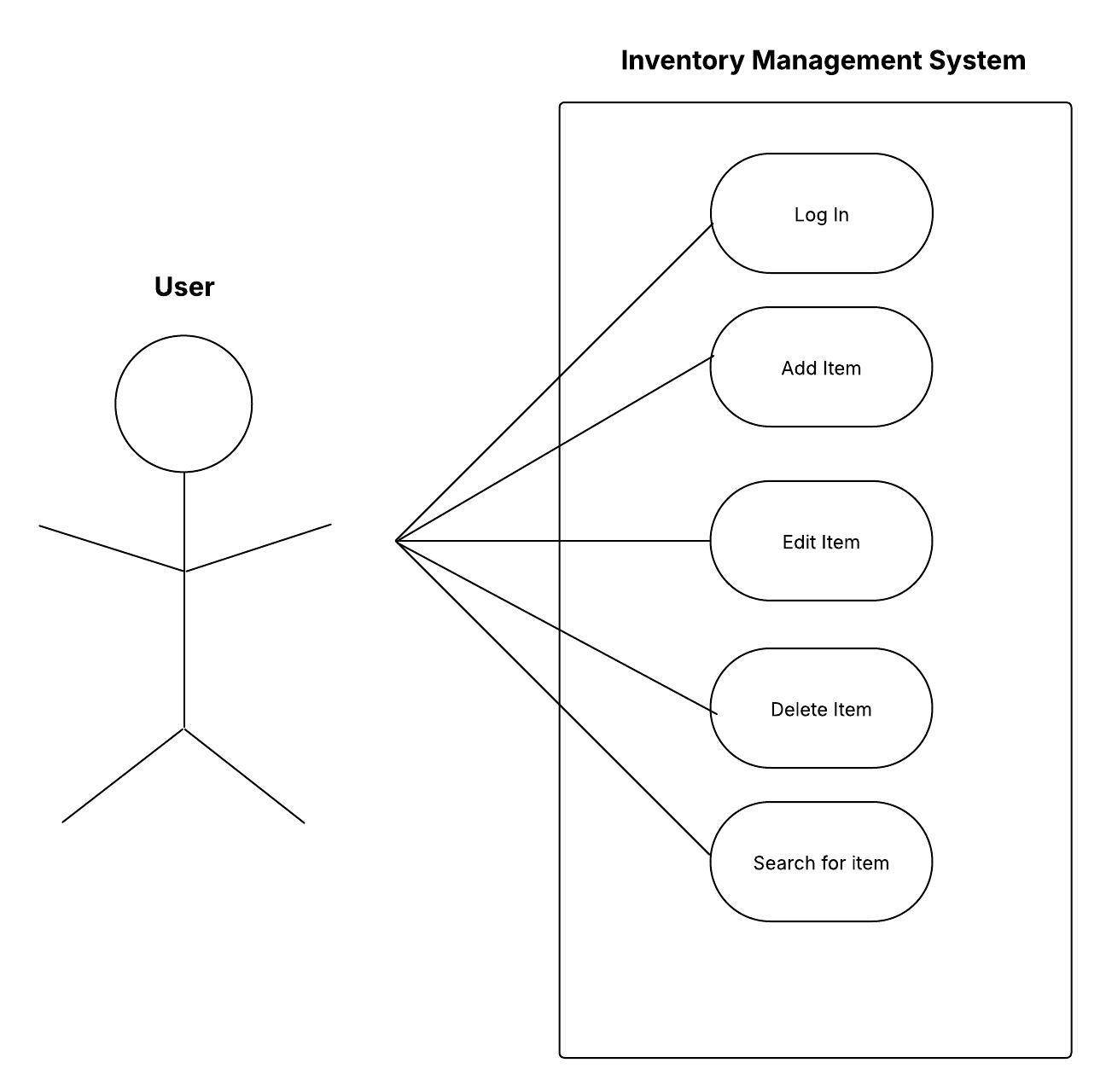
● Required for an online website

● Helps reduce customer questions about stock

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

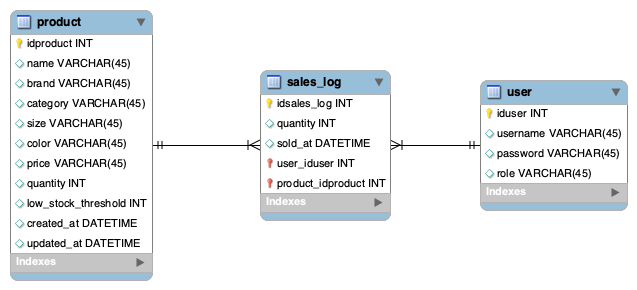
1. **System Design Document (Due 6/13/2025)**

● **Use Case diagrams/ Description**



**Description**: When the user successfully logs in he will be able to add, edit, delete and search for clothing items all through the CRUD interface. Specific use cases will be to search for items low on stock and perform audits on the clothing inventory by searching through the entire list of items.

● **ER Diagram / description**

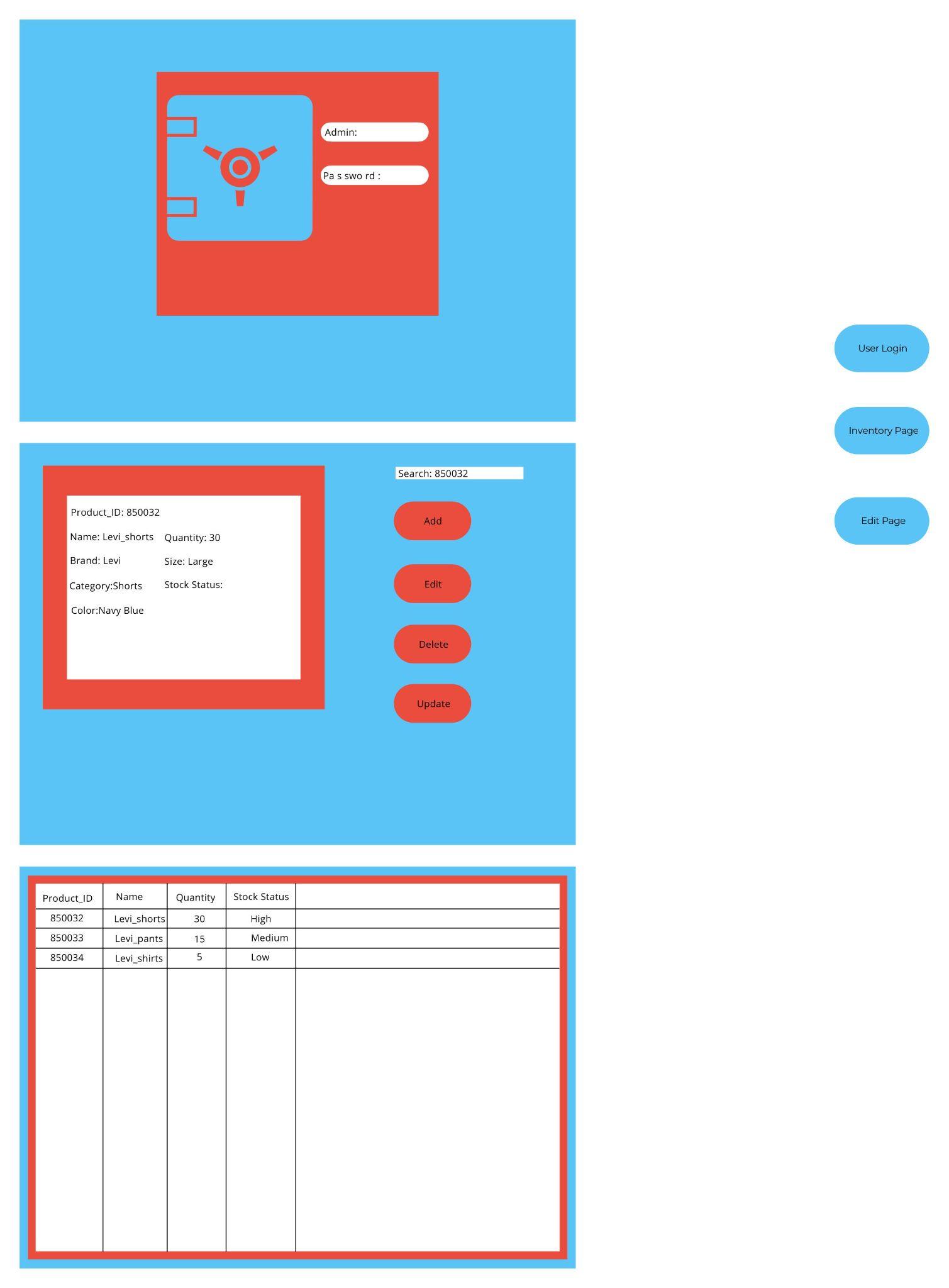


**Description**: Three sql tables all linked between each other for the management system to run as intended. The product table holds specific information about each type of clothing piece like the name, brand, category, price, quantity and a unique ID named idproduct which functions as the primary key.

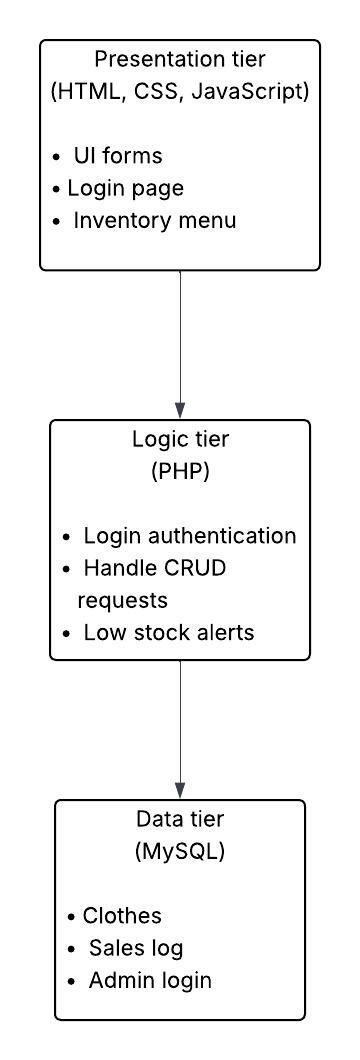
The sales\_log table keeps records of each item being sold, the quantity and the time that the item was sold, this is important as keeping track of sales shows how the quantity of stock on some items keeps diminishing therefore easier to track items that are low on stock. Sales\_log table uses the idsales\_log ID as a primary key to differentiate different sales being done. Also contains user\_iduser and product\_idproduct as foreign keys for better tracking between tables as the stock keeps updating.

The user table keeps track of the username, password and role of the user for improved security on who is able to access the management system. Sensible information like stock is crucial for businesses from an economic standpoint therefore the ability of editing and deleting stock should only be accessible to specific workers. The user table also includes a unique ID names iduser for each user registered inside the table with their included role.

● **Basic wireframes or UI mockups**



● **Architecture diagram (showing 3-tier structure)**



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. Working Prototype / MVP (Due 6/22/2025)

• Functional front-end form pages

* Done. Functional Login Prompt and inventory check.

• At least one working end-to-end flow (e.g., user registration or request submission)

* Working Logic for inventory check / Login information validation.

• Hosting on local or cloud (e.g., localhost, Firebase, Vercel, Render, etc.)

* Hosting on local and on cloud \*\*Link to webpage: clothingstockcheck.wuaze.com
* # Disclaimer Safari is viciously protective use chrome!

Purpose: Early validation of functionality and integration.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Admin Panel (Due 7/6/2025)

An admin panel is a secure backend interface for managing and moderating content, users, or

requests.

**# Quick user access**

Employee Login:

Username: employee

Password: employee123

Admin Login:

Username: admin

Password: admin123

**# PLEASE USE THE LINK BELOW IF YOU"RE HAVING TROUBLE FINDING IT!**

http://clothingstockcheck.wuaze.com

Key Functionalities:

1. Admin Login (Authentication Required)

* Done. Fully functional Admin login with authentication.

1o Secure login with role-based access (admin only)

* Done. Administrators have access to unique properties.

o Optional: Password reset functionality

* Done. Administrators can reset passwords.

2. User Management

* Done. Users can be added / deleted and placed into active/ inactive status (no access)

o View all registered users

* Done. Admin’s can view and manage all users

o Option to activate/deactivate accounts

* Done. Admins can active and de-active accounts in user management tab.

o Edit/delete user details if needed

* Done. Admins can edit and delete user’s information.

3. Content Moderation

o View submitted requests/posts (e.g., food donations, friend profiles, meal plans)

* Done. Admins can approve or disapprove of employee’s items upon submission to be added.

o Approve/reject/edit entries before making them public

* Done. Admins hold the ability to deny or approve any entries!

o Report and flag inappropriate content

* Currently there is no outside user interactive meaning that anything and everything must first be admin approved. This feature will likely be put into place if say we allow user’s to post about items they bought or something however the app is for a corporation’s personal use not necessarily a public use yet.

4. Data Entry

o Admin can manually add content (e.g., categories, ingredients, FAQs)

* Done. Admins can manually add content, and it will go through automatically! This is because admin’s do not need approval to update the currently stocked item logs.

Additional things

* Updated DDL Script
* Updated DML Script.
* Added some CSS styling for dashboard this is a work in progress!
* Added the ability to add users.
* Added an employee dashboard with different restrictions.
* Code cleanup is also a work in progress! Need’s attention)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. Basic Analytics Dashboard (Due 7/13/2025)

A dashboard helps visualize trends, usage, and activity.

Key Metrics to Show:

1. User Statistics

o Total number of users

* Done. Admins can view total user’s.

o Daily/weekly/monthly sign-ups

* Done. Admins can view the daily / weekly / monthly sign-ups.

o Active vs inactive users

* Done. Admins can view how many active vs inactive users are present.

2. Activity Overview

o Total posts/requests submitted

* Done. Admins can view total products that have been approved / pending / rejected and

o Most common donation/meal/friend category

* Done. Admins can view the top 10 low stock items.

o Completion or matching rate

* Done. Admins can view Products by category quantity of category and total quantity of items in that category. (I believe this qualifies! Since it’s a total stock check.)

3. Graphical Insights (Charts/Tables)

o Bar or pie charts showing category breakdowns

* Done. Bar and pie charts are scattered depending on which analytic is desired.

o Time-based trends (line graph of usage)

* Done. Particularly used in the user statistics page (Signed up users).

o Most used features/pages

* Done. We have Activity reports page.

4. Search and Filter Tools

o Filter data by date range, user role, or category

* Done. Data such as dates and categories of clothes can be filtered.

Final Submission:

* Fully functional web app (hosted online)
  + Done. App hosted and functional through infinityfree
* Features reflecting real-life need (e.g., food matchmaker, friend recommendation,  
  dynamic meal plans)
  + Done. Features like inventory stock list, support ticket, low stock alerts added.
* Authentication and session handling
  + Done. User authentication and session handling through login prompts validate it.
* Data storage and retrieval using SQL/NoSQL
  + Done. SQL used to create a database that stores login information and product information like stock quantity, price, type and low stock threshold.
* Error handling and input validation
  + Done. Error handling and input validation for features like login and adding new items.
* Responsive UI (Bootstrap/Tailwind or plain CSS)
  + Done. Used plain CSS for our dashboards and used open source .js files to create our charts and graphs.