COMP2030 - Human Factors for Interactive and Web-Based Systems

Web Solution - Collaborative Activity

2024 Semester 2

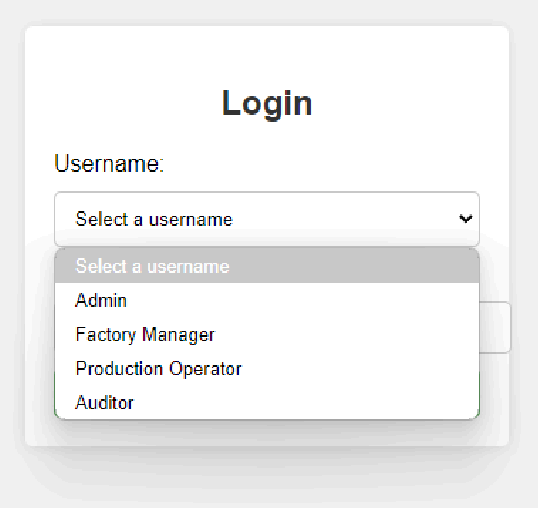
Abdul Ramahan (rama0043)
Anandan Somasundaram (soma0032)
Bailey Boyd (boyd0106)
Caleb Kemenyvary (keme0008)
Daniel Rosich (rosi0028)
Samuel Ngiri (ngir0003)

Table of Contents

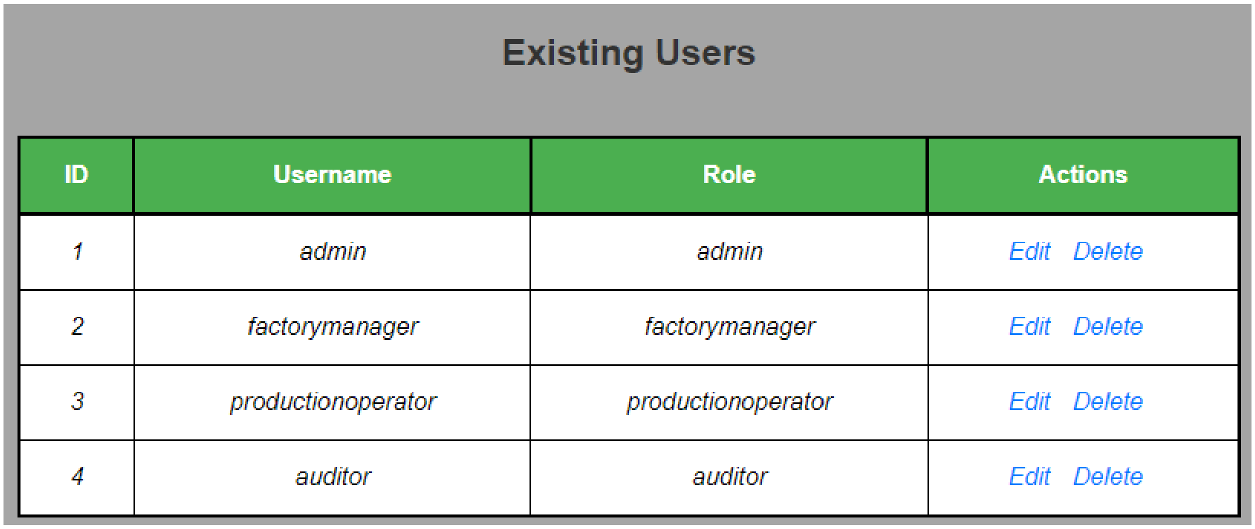
1. Website Design and Code Decisions

As the company has many employees, most having different roles and credentials, it’s important to ensure streamlined design so users can easily perform their required actions. We achieved this through a login system, where users are only able to access features that their credentials permit after inputting their password. After selecting a user and inputting the correct password, they are taken to an individual homepage for their respective role: (Admin, Factory Manager, Production Operator and Auditor), where they are shown all of their respective site features.

We chose to store user login information in a single table to ensure easy data management, though leaving that as, it would prove a potential security risk, as if the system were to be hacked, a perpetrator would immediately have access to every user’s login credentials. We were able to remedy this by encrypting user’s passwords on the site’s backend, ensuring if it were to be compromised, login details would remain safe.



As we’re dealing with potentially very large data sets, accurate and reliable information storage is important, as to maintain efficiency. This is achieved through both having a unique naming system, but also assigning all database entries a unique identifier. This ensures that there's no overlap in database entries, unique names wont be required as long as each entry has a unique ID.

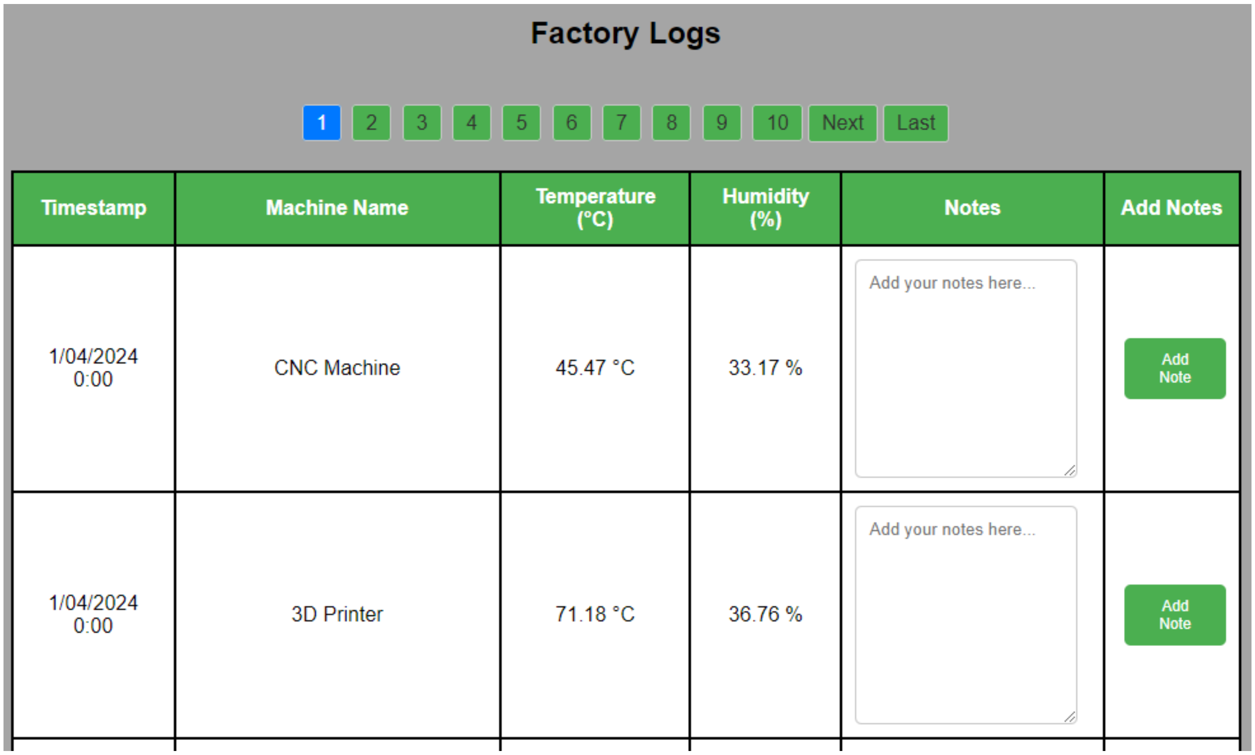


Existing Users

ID	Username	Role	Actions
1	admin	admin	Edit Delete
2	factorymanager	factorymanager	Edit Delete
3	productionoperator	productionoperator	Edit Delete
4	auditor	auditor	Edit Delete

As our database deals with large data sets, an issue with manually finding information presented itself. Presenting the tables as is (while somewhat sufficient), would result in one extremely long web page where a user would have to scroll for an unreasonable amount of time to find one specific entry. This would prove cumbersome, as users would have to make a mental note of the position of the data they are looking for's position, otherwise risking losing where it is, which would result in having to manually search for it again.

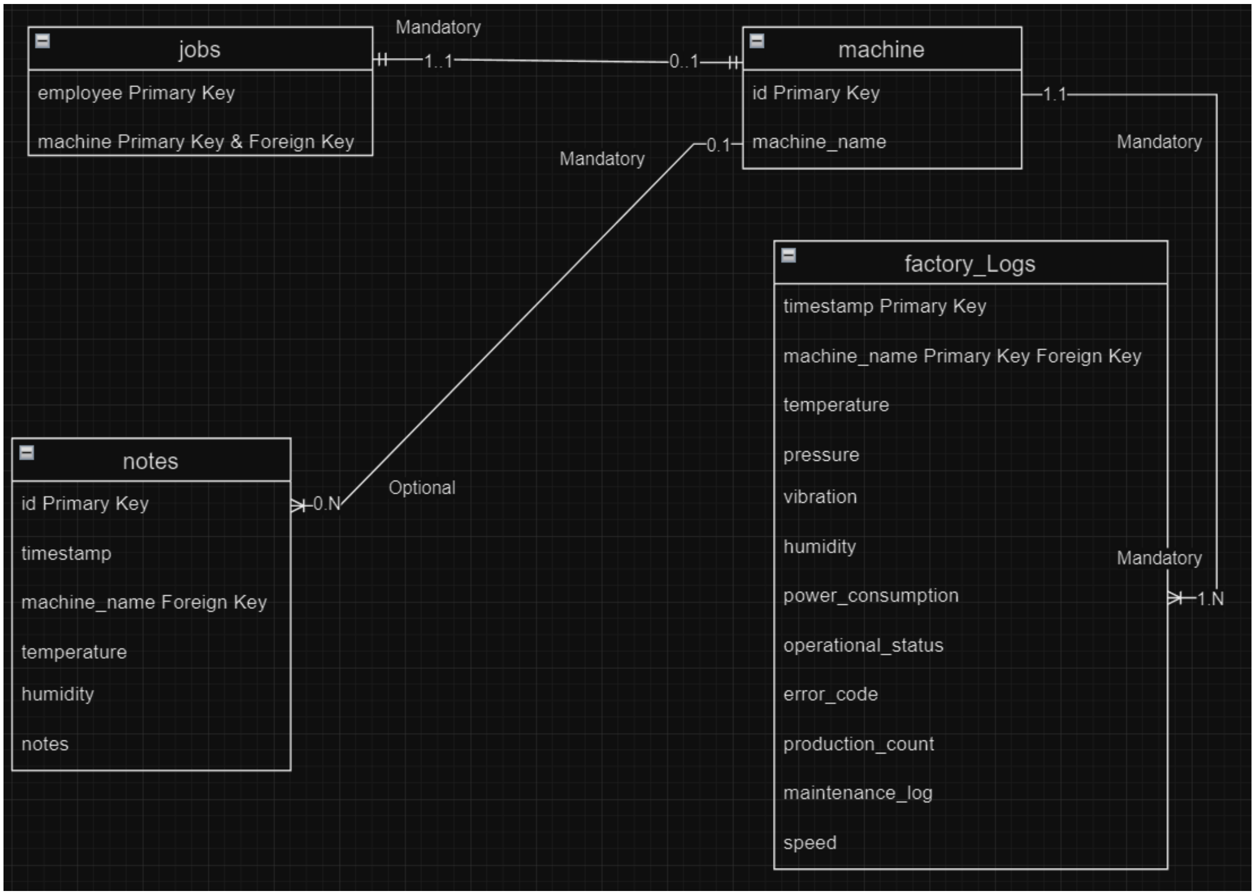
A solution was found through streamlining this data set. By dividing up the table into smaller pieces and putting them on separate pages, and adding a page index bar, which contains links to every page of the database, as well as a next page and last page button. This allows users to quickly switch between database sections, as they simply need to go to the page of the factory logs that the entry they are looking for is on, rather than manually scrolling through the entire database. This proves extremely useful, as users can make an estimate on where the entry they're looking for is located, based on the timeframe they search for, as it;s sorted by time of recording, with logs taken earlier in the day being at the start of the database, in descending order. For example, if they know they need to check on logs from the start of the day, it'll be on page one, and if they want to check on ones from the end of the day they will be on page 10.



Factory Logs

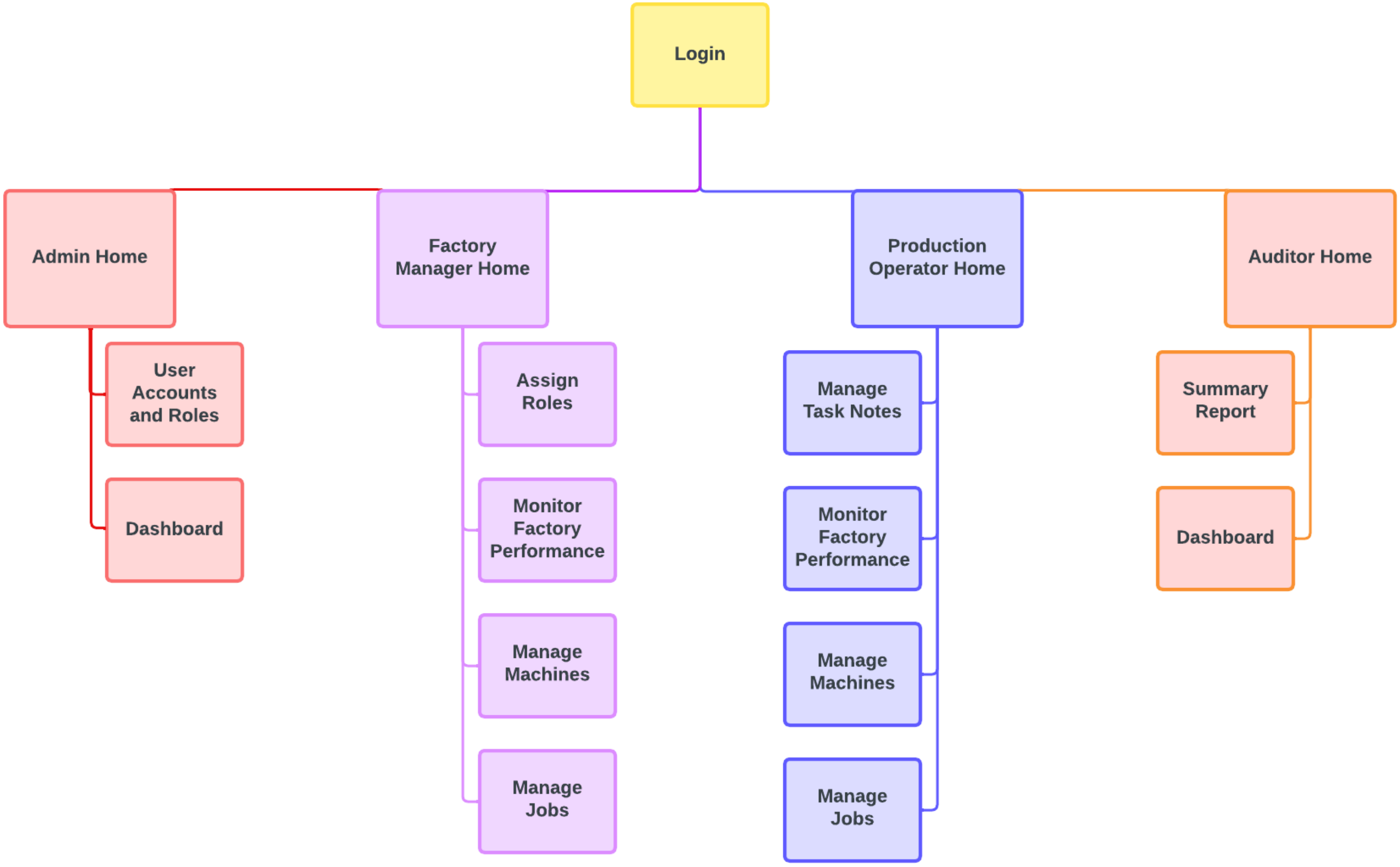
Timestamp	Machine Name	Temperature (°C)	Humidity (%)	Notes	Add Notes
1/04/2024 0:00	CNC Machine	45.47 °C	33.17 %	Add your notes here...	Add Note
1/04/2024 0:00	3D Printer	71.18 °C	36.76 %	Add your notes here...	Add Note

2. ER Diagram



The ER Diagram displays the relationships and data flow between the four entities, machines, jobs, factory logs and notes within the SMD_Database. The machine table allows each machine name to be identified by its unique id. The jobs table allows for creation of jobs for each employee and machine. This table includes a composite primary key of employee and machine. The machine also acts as a foreign key. The factory logs table creates logs of each machine within certain timestamps. This table uses data such as, temperature, humidity, production count, vibration, power consumption, pressure and speed when creating logs for each machine. The primary key for factory logs is a combination of timestamp and machine name for log entries. The notes table follows a similar design as it allows for production operators to create notes based on observations of specific logs created by the factory logs. The primary key for this table is id. Relationships across the table are evident as marked. Machine to jobs is one to one, machine to notes is one to many, machine to factory logs is one to many.

3. Site Map

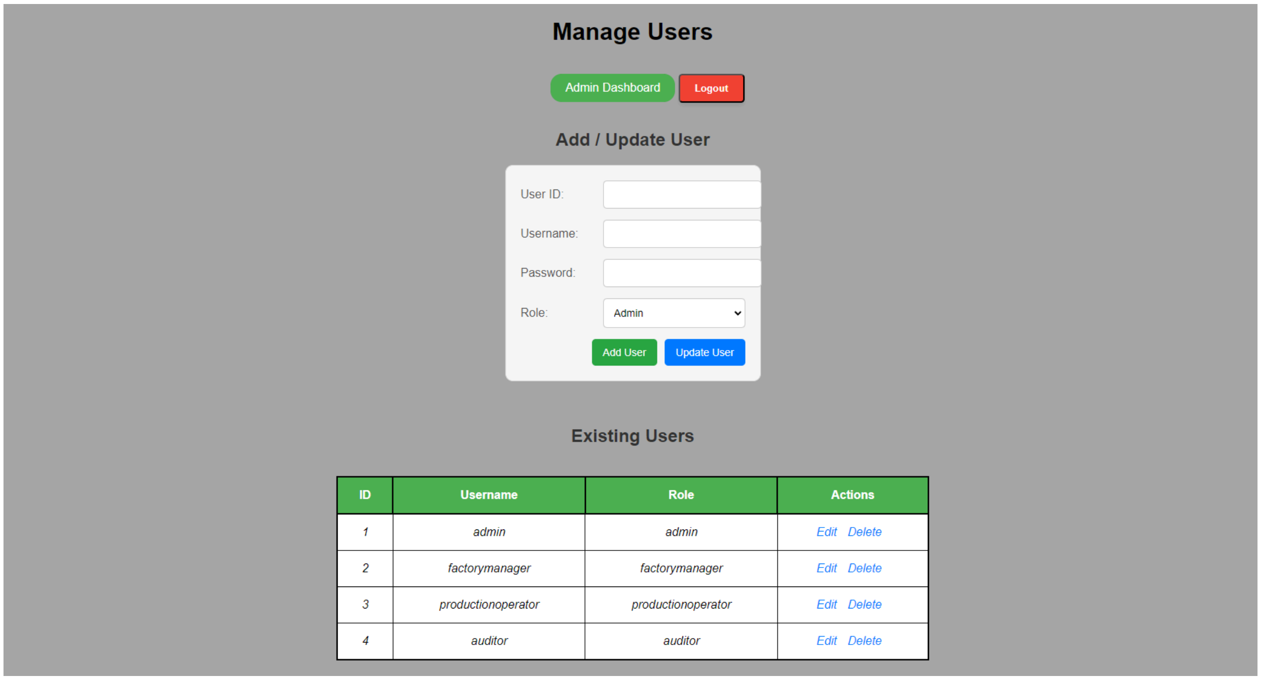


This site map illustrates the structure of a factory management web application, designed to provide different user roles with tailored functionalities. At the top, a unified login screen allows all users access to the system. Each role has their own functions that they can individually perform, however some functions can be performed by two types of roles. For example - both the administrator and auditor roles both have the capability to access the dashboard, yet only the auditor has access to the summary report. This way, from the login screen, each role will have access to only their own home page, and be able to access pages that only they have permissions to.

4. Justification of HTML, JS and PHP Used To Implement The Functionality

The implementation and functionality in the web application utilizes a combination of HTML, JavaScript and PHP documentation, which all serve various roles. The HTML is primarily used to be the structure of the content of each webpages, which are part of the user interface. We have HTML forms to gather user credentials, which provides a simple interface for input from the user. It also incorporates several external style sheets (.css) in the web application, one for a global stylesheet, a login & logout sheet, manage_machines and manage_users. Initially, we had one single .css stylesheet to be used on behalf of the entire website, however we quickly realized that it would be simpler for each main section of the website to have its own style, as searching for one specific section to adjust only one or two points would become tedious behavior. Whilst using one stylesheet is better for performance, due to the loading time for each page, using less HTTP requests, using multiple .css files allows for ease of development, whereas having one per module makes it easier to locate. Since we are not going to be publishing this for public or commercial use, we decided to use multiple .css sheets, whereas we understand the performance practicality when only using one.

The PHP does play a crucial role in managing the backend functionality of our web application. In the add_user.php and login.php files, PHP is responsible for the server-side operations including database interactions, user authentication, and the session management. For instance, the PHP processes the login form, verifies the password securely using hashing techniques, manages user sessions and redirects each user based on their roles. This approach was used as opposed to simply clicking on your role to login without a password because it is an added security feature, preventing anyone from logging in as a role that they are not. Our approach was significantly more effective in preventing security breaches if this web application wasn't fictive. Our login process also supports incidental inputs from the user - if they incorrectly type in their password, they will receive a notification informing them that they did not input their credentials correctly. This saves the user from potentially being stuck in infinite loops of not being able to log in after a first incorrect attempt, because they were not informed they put in the wrong password. The PHP handles error logging, which is crucial for debugging and maintaining system integrity. Another key assumption we made when using PHP was the implementation of a logout function. Since factory workers often share devices across shifts, such as between day and night crews, it is of substantial importance to prevent users from accessing an account already logged in with elevated permissions beyond their own. This ensures a secure handover between shifts, protecting sensitive information and maintaining the integrity of individual user access.



Role: Admin

ID	Username	Role	Actions
1	admin	admin	Edit Delete
2	factorymanager	factorymanager	Edit Delete
3	productionoperator	productionoperator	Edit Delete
4	auditor	auditor	Edit Delete

Figure 1: The manage_user.php page - showing the logout button in red, is easily viewable and will prevent others from getting access to others' accounts.

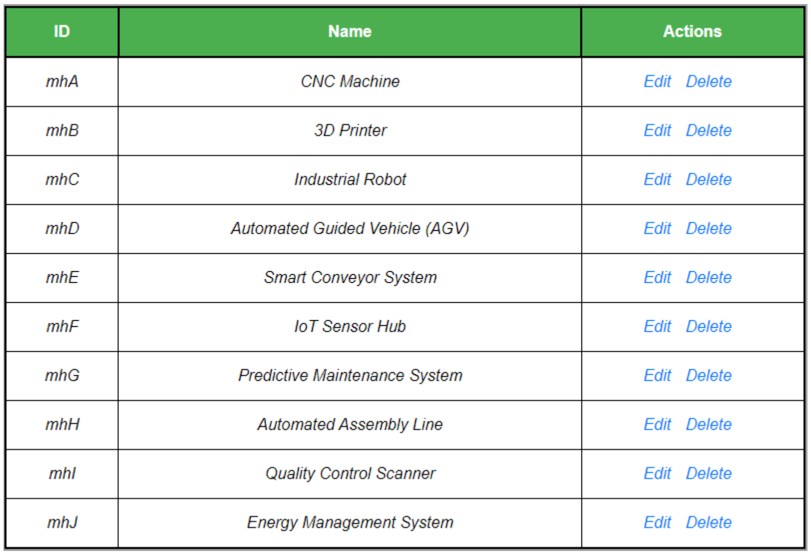
Within the Smart Manufacturing Dashboard database (smd_database.sql) we created extra tables to fit some of the tasks that each role of the SMD required. For instance, within the tasks performed by factory managers, they should be able to monitor the overall factory performance. This should be able to be done by simply showing the factory_logs.csv file as a database viewable for the user, without the necessity to include extra tables. However, another one of the factory manager's tasks - add, update and delete machines - will require an additional table to be created. Within the factory_logs.csv file, we see that there are only 10 different machines, all having their information taken at 30 minute intervals. We can create an ID for each machine, followed by the names of each machine, added into a separate table, to help factory managers easily view, manage, and update machines. This is done through the 'machine' table, which only includes those two columns.

The manner in which we used each page to serve a specific purpose ensures a seamless, secure and a user-friendly experience. Our decision making for each task that the roles must be able to perform has allowed us to create effective functionality across said tasks, and each technology used is leveraged where it can be, enhancing the functionality and the performance of our web application.

5. Creative Inclusions

After logging into the Smart Manufacturing Dashboard in a specific role, two types of buttons will be present on the page. A green task button, used for each role performing a task for their job, and a red logout button [as seen in figure 1]. The decision to make the logout button a different colour is due to the button's function to be different from every other. This will sign out of your account and redirect you to the SMD's login page. It is also red for users to mentally distinguish the difference between those functions, and red, tends to be a colour more associated with caution, so users are more inclined to be wary of using it, only when it is necessary. When hovering over these buttons, the green buttons change hue slightly, and the logout button slightly raises above the screen, giving the impression of it jumping out of the screen, this is also a tactic to make sure users are more aware of what buttons they are hovering over, to make sure they don't accidentally click on it.

We decided to change the background of all pages (except the login page) to be a greyer colour because we believed that using a blank white background would eventually cause mild eye strain if used for extended periods of time. The colour we chose was a mix between a bright and dark shade of grey, so we could ensure we achieved a happy medium of users who do and don't prefer a 'dark mode' styled display [see figure 1].



ID	Name	Actions
mhA	CNC Machine	Edit Delete
mhB	3D Printer	Edit Delete
mhC	Industrial Robot	Edit Delete
mhD	Automated Guided Vehicle (AGV)	Edit Delete
mhE	Smart Conveyor System	Edit Delete
mhF	IoT Sensor Hub	Edit Delete
mhG	Predictive Maintenance System	Edit Delete
mhH	Automated Assembly Line	Edit Delete
mhI	Quality Control Scanner	Edit Delete
mhJ	Energy Management System	Edit Delete

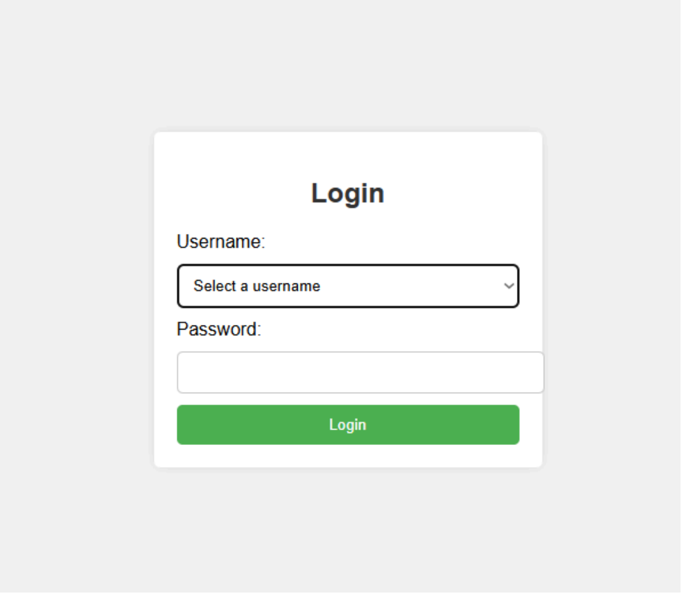
Figure 2: the Manage Machines table in the Manage Machines Home page

Other clickable links we decided to implement were the edit and delete buttons for the update machines and update jobs page tables. The 'edit' and 'delete' buttons were highlighted to the hyperlink blue color to indicate to the user that they could be clicked to edit or delete that specific machine [figure 2]. When hovering over the buttons, they are marked with an underline to further clarify to the user they were links. This allows the user of the Smart Manufacturing Dashboard to understand how they can update tables within their respective page.

Overall, we chose not to overly stylize each page, only minimal decorations for practical purposes, not just because we intended on creating a simple functioning Smart Manufacturing Dashboard, but because for a commercially used application, it would be better not to style each page to look pretty and appealing. Its purpose is functional, not cosmetic.

6. Justification Of Validation and Security Functionality

As this website prototype deals with large data sets as well as personal employee information, it’s important to maintain proper site integrity and security. This is done in a few ways, the primary one being the site’s log in and log out functionality. To access any part of the site, you must select a user, (Admin, Factory Manager, Production Operator and Auditor) and input their respective passwords. This only gives the user access to that roles’ sections of the website, which means that they are limited to only performing actions that their role permits, such as when you log in as a Production Operator, you’re limited to Monitoring the factory performance, managing jobs, managing machines and assigning roles. To access anything else, you must log out and sign in again.



This functionality ensures system integrity, as employees are restricted to only perform actions they have the required permissions for while also ensuring users have a streamlined user experience, only seeing the site features they require. While this does have potential issues in the circumstance with the use of shared devices, where people may leave the device logged into their user, having a logout option button on all pages of the site keeps it simple and straightforward, so after an employees finished using any device, they should be able to return it to the login page incredibly easily.

If we were to flesh this out further, adding a function where the site automatically logs you out after leaving the device idle, most likely showing a popup that would ask “Are you still using this device?”, logging out the user when they don't confirm they are after a long enough amount of time. This would ensure that people aren't able to access features of the site that they don't have the correct login for.

Additionally, databases are shared throughout the website, as we use SQL as the base. So, all updates and changes done are maintained consistently for all users. This would reduce the chance for conflict errors, as all users will see the same changes as they are implemented.

Additionally, to ensure the safety of user’s login credentials, we chose to encrypt the user passwords on the database side, as it’s stored in a single table. This retains the benefit of simple data storage, while ensuring that if the site were to be compromised that data would remain safe.

Appendix

- Lin, J.C.W., Djenouri, Y. and Srivastava, G., 2021. Efficient closed high-utility pattern fusion model in large-scale databases. *Information Fusion*, *76*, pp.122-132.
- nanquim. (2017, May). Single or Multiple CSS. *freeCodeCamp*.
- Dowden, M. and Dowden, M., 2020. *Architecting CSS: The programmer's guide to effective style sheets*. Apress.
- Blue Dolphins (2024) Smart Manufacturing Dashboard [source code]. https://github.com/KazzzzzyBazzzzzy/COMP-2030---Blue-Dolphins
- Nissen, A., Riedl, R. and Schütte, R., 2024. Users' reactions to website designs: A neuroimaging study based on evolutionary psychology with a focus on color and button shape. *Computers in Human Behavior*, *155*, p.108168.
- Singh, P. and Kaur, K., 2015, February. Database security using encryption. In *2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE)* (pp. 353-358). IEEE.