Capable People, Capable Communities

Micro-Credentials in Software Development

Certificate in Software Development

MCSD51

Project Report

Assembly Job Buffer Board and Time Tracker

Abhishek Makker

March 2025

1. Contents

2. Executive Summary

In this project, I developed a web application to make it easier for a team supervisor/planner to plan upcoming jobs by sorting jobs to the days where capacity is available. The project is effectively a digitization project of an existing tool. The project uses HTML, CSS, Node.JS, and JavaScript. This is a self-sponsored, individual project developed in an iterative manner based on similar non-digital tools used at my work-place.

3. Introduction

A local manufacturing facility manufactures and assembles a variety of products of low to mid volumes. These products are assembled to order within a 5 day time period (buffer). It is the responsibility of the assembly manager and team leader to manage the incoming jobs and dispatch the products out before the deadline. Currently this is done using physical cards on a 5-day buffer board which is cumbersome to use and does not provide dynamic information. In addition, the amount of time taken to assemble each product of a job is not easily tracked and therefore makes planning more difficult.

4. Initial Proposal

Introduction

A local manufacturing facility manufactures and assembles a variety of products of low to mid volumes. These products are assembled to order within a 5 day time period (buffer). It is the responsibility of the assembly manager and team leader to manage the incoming jobs and dispatch the products out before the deadline. Currently this is done using physical cards on a 5-day buffer board which is cumbersome to use and does not provide dynamic information. In addition, the amount of time taken to assemble each product of a job is not easily tracked and therefore makes planning more difficult.

Scope

The home page will be a log-in page for the assembly team including the company logo. This is differentiate the functionality and privileges between the administrators (assembly managers and team leaders), and the assembly team members.

The completed project will consist of 2 major functional features:

1. A board with moveable cards with job information wherein each card contains the details of the job. The job number is a unique ID which provides information such as due date, part numbers, quantities, and approximate time to assemble. All jobs in the system populate a column based on due date.

The cards will be moveable between columns where the columns are dates for planning [from today to 5 days from today]. For functional testing, randomly generated jobs will be created to assign them to the columns and provide the user ability to drag and drop. The unique Job ID would be used to pull information about the job from the database.

The managers would be able to take the jobs from the board and assign them to separate rows/columns based on the assembly team or bench.

2. “Activating” or Starting a Job: When an assembly team starts a job – in this case, identified by a user moving the Job to the Active column and clicking Start, a different screen gets activated. This screen showcases the details of the job and the quantity remaining.

Each time a product is completed, the laser engraver is used to mark the job with the correct serial number. This signal is used to indicate to the program that a single part has been assembled. The program will then reduce the quantity of product remaining by 1 and record the time between each signal, thereby giving an average time/product over the course of the job [For the scope of this project, the laser signal is simulated using an RNG function within limits].

An additional feature to this part is that the average time/part is linked to the specific part. When that part shows up again, the database already has the moving average in the database for that specific part number.

If the pace of assembly is same or faster than the average pace of assembly for the product as retrieved from the database, the overall schema of the page will be positive and green whereas if the pace of assembly begins to lag behind the average pace of assembly, the overall schema showcases yellow and if it is very bad, then red.

A database is required for log-in.

A database is required to contain the Job ID, part number, quantities, and due dates.

Another linked table is required to store information about the Part Number, such as time to assemble and optimal bench/team to do so.

User stories and use cases

You cannot manage what you cannot measure. [1] *[https://www.facebook.com/taurangabusinesschamber/videos/members-oasis-engineeringoasis-engineering-2003-limited-recently-participated-in/1575133319761532/]*.

The concept of improvement is not possible or trackable if we cannot measure the parameters that could be improved. The rise of Industry 4.0, i.e. information in technology, means that any company not fully utilizing data to make data driven decisions will fall behind.

Giving team leaders and assembly managers the ability to dynamically plan their work and have visibility of current and upcoming workloads and assign the jobs to meet due dates as well as track whether each assembly team is on-track or not is important and allows them to understand and identify opportunities for process improvements. [2] *[https://www.callaghaninnovation.govt.nz/stories/oasis-embraces-industry-40/]*

Ethical and cultural impact

Industry 4.0, much like automation, is the next generation of technology available to industry. New Zealand’s economy is smaller than many other developed countries and faces a number of issues when it comes to competing on the global market. These issues include but are not limited to, lower economies of scale, higher costs and expenses, and a large geographical distance to global markets in the context of physical goods.

A manufacturing facility that is Industry 4.0 capable has more information and real-time data that can provide increased efficiencies and better decision making capabilities to decision makers. This can provide the advantage New Zealand manufacturers need to up their game and make them

competitive in global markets by using data to make sustained incremental improvements to their business.

This project further provides another tool for manufacturers to use to improve their outputs and processes. [3] *[https://www.callaghaninnovation.govt.nz/products/in-focus-topic-pages/in-focus-industry-4-0/]*

SDLC

I have chosen to go with the Incremental Development Model for this project. The plan is to tackle one module at a time and make sure each module is working correctly before moving onto the next. The reason for this is that each module can functionally work by themselves however usefulness is only obtained once each page links to the next.

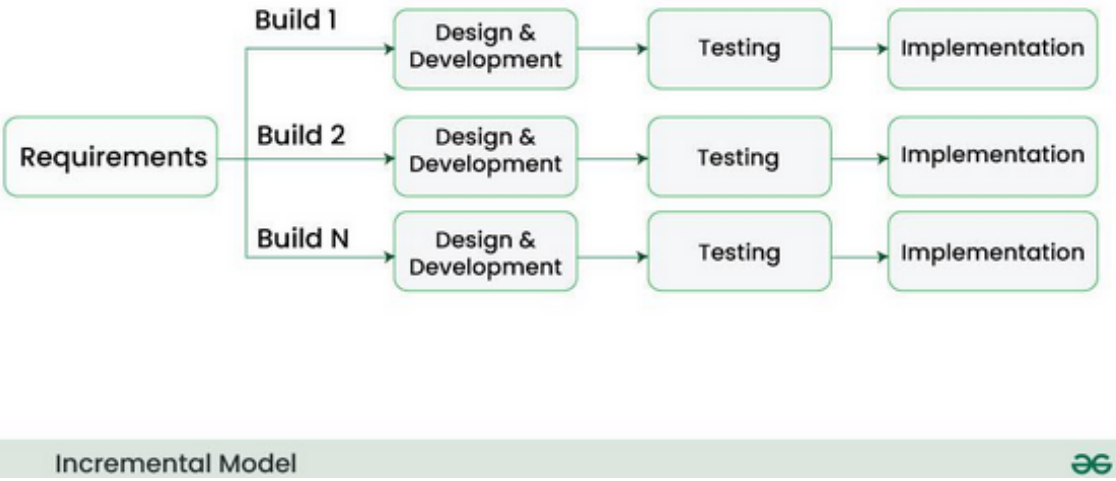


Figure 1: Incremental Model [4]

[Retrieved from https://media.geeksforgeeks.org/wp-content/uploads/20231030181201/incremental.jpg]

PROJECT TIMELINE

The project will proceed in modules with testing being conducted for each module. Each module will be tested and implemented before moving onto the next one.

The first module, creating a login page and database, will be implemented by the end of 2024. The second module is the most functionally heavy part of the project and will require the entire month of January to complete including functional testing. The third module for counting down the active job quantity will require the month of February including testing.

The month of March will be specifically for integration, de-bugging, improvements, and finally release.

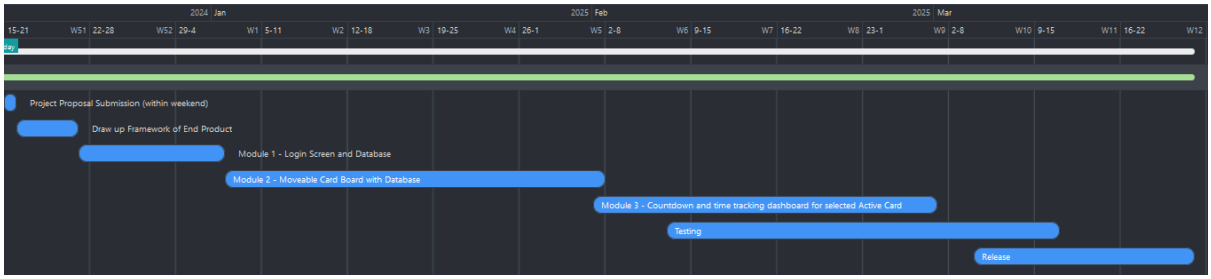


Figure 2: Gantt chart of project plan

DIAGRAMS

The below diagram shows the proposed layout of the project in terms of functional modules and databases. Some tables are linked and there are data streams between them. Tables may be updated live as someone moves a card from 1 column to another. This is indicative and features may change.

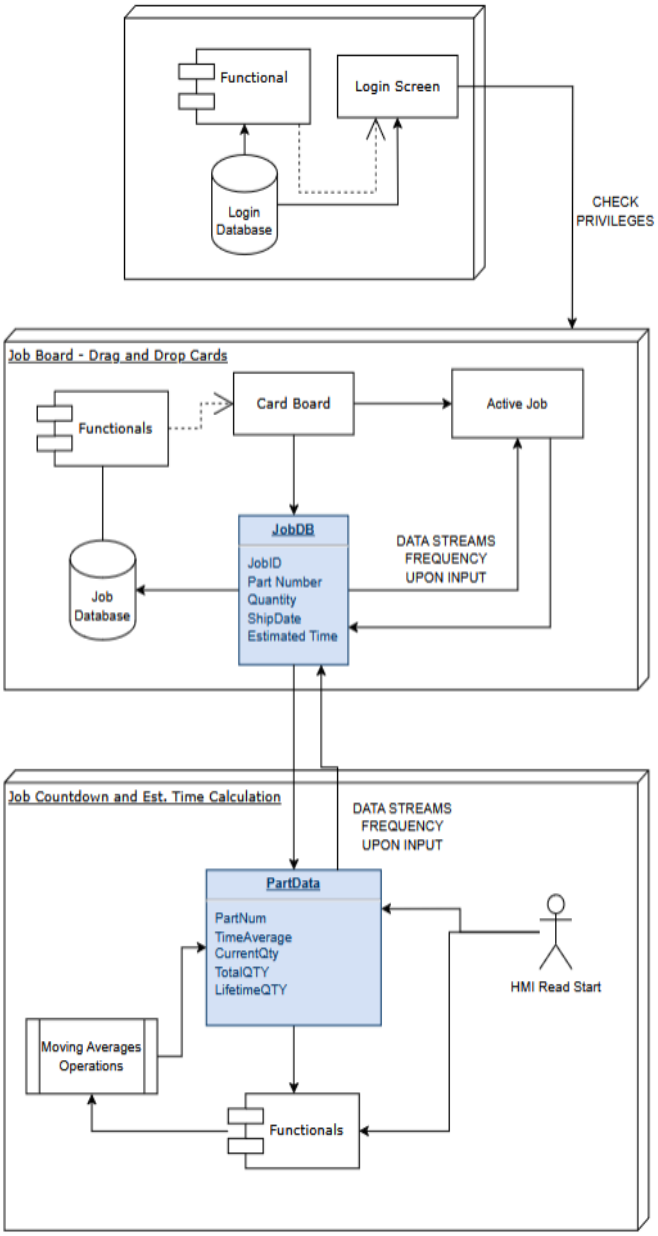


Figure 3: Functional diagram

References

[1] GeeksforGeeks. Iterative and incremental development (IID).
Retrieved from https://www.geeksforgeeks.org/iterative-and-incremental-development-iid/

[2] Callaghan Innovation, Oasis Engineering: Set up for new market success by setting up Industry 4.0 solutions (2020)
Retrieved from https://www.callaghaninnovation.govt.nz/stories/oasis-embraces-industry-40/

[3] Callaghan Innovation, In Focus: Industry 4.0
Retrieved from https://www.callaghaninnovation.govt.nz/products/in-focus-topic-pages/in-focus-industry-4-0/

[4] Tauranga Chamber of Commerce
Retrieved from https://www.facebook.com/taurangabusinesschamber/videos/members-oasis-engineeringoasis-engineering-2003-limited-recently-participated-in/1575133319761532/

5. Variations In User Requirements From The Initial Proposal

The initial proposal consisted of 2 primary functionalities, one related to the board visualization and movement of the job cards, and the other to provide another layer of analysis and information based on information gathered about the job based on hardware button presses (Human Machine Interactive elements).

The entire project in the initial proposal would have little to no data entry elements in the project in regards to adding, or deleting a row of data and would have no form entry apart from user registration. This was due to the initial approach being centred around the data being retrieved from an ERP system wherein the server would never have any write privileges to the ERP data structure.

Due to recommendations from the lecturers, the web application was modified to allow the users to add, edit, and remove jobs from the board.

Due to the time spent in getting all of the primary functionality up, in addition to the operations that were unplanned for, the second functionality was data capture and analysis was scrapped. The second functionality would require more time and is better suited as an add-on module to a polished core project and was not attempted.

6. Design Documents

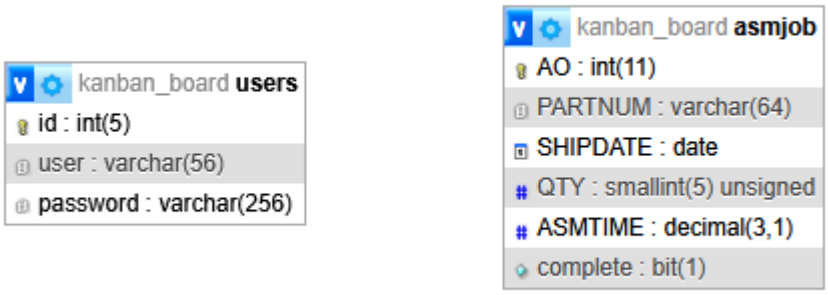


Figure 4: Database Schema

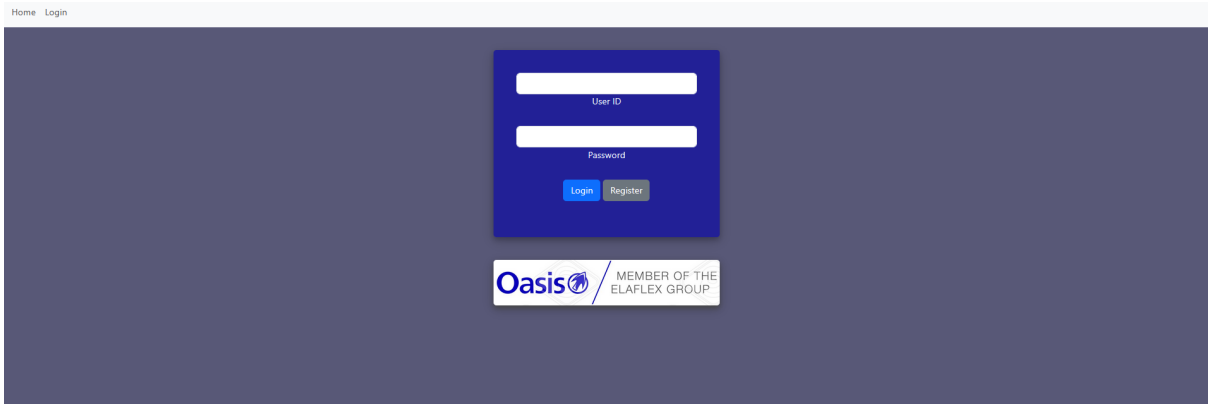


Figure 5: Homepage with nav bar and logo

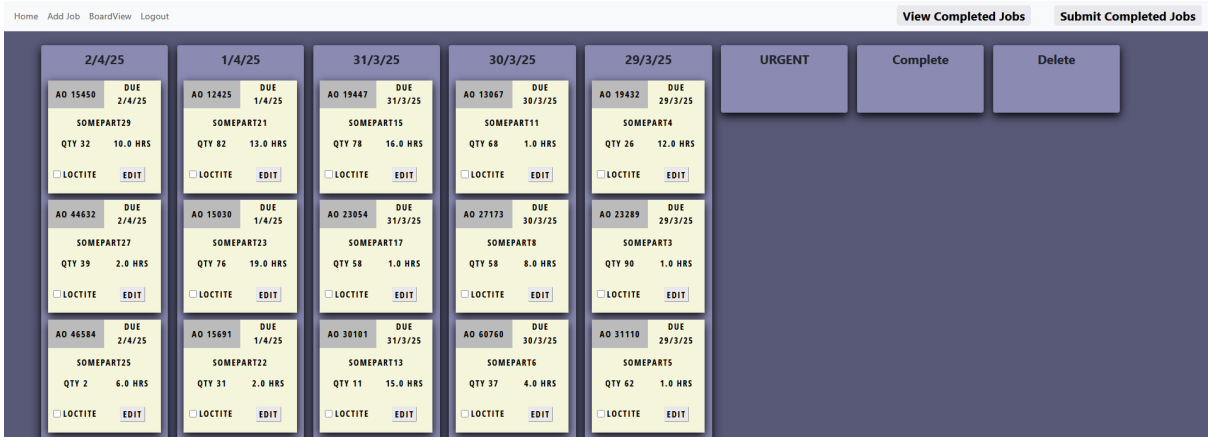


Figure 6: Boardview upon login - no urgent jobs as all dates are today and beyond. All cards are generated based on the SQL Database

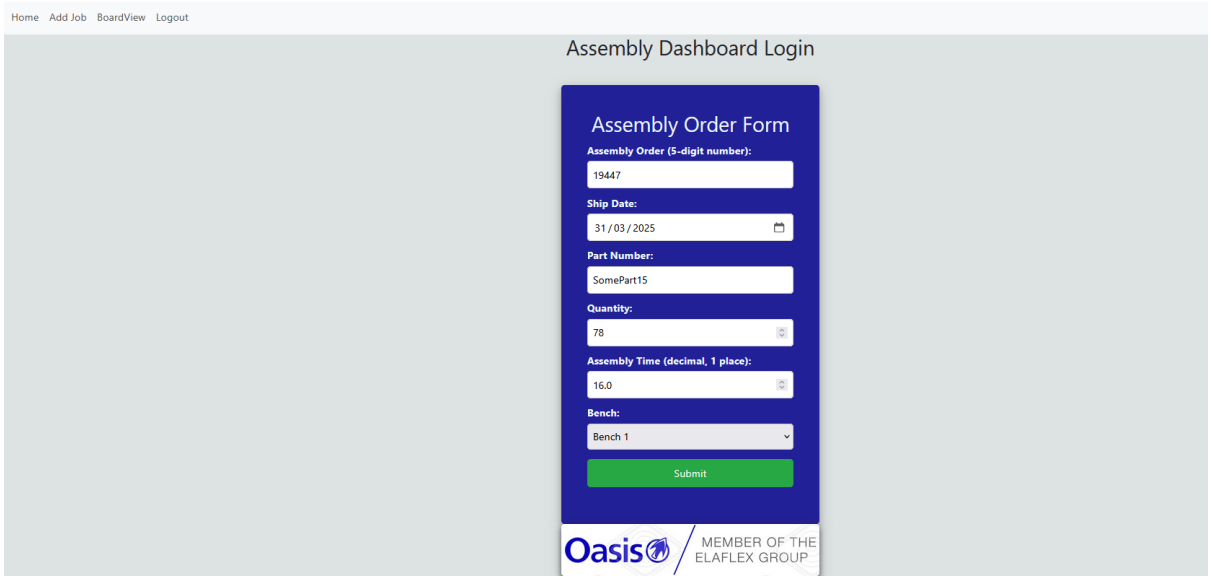


Figure 7: Clicking the Add Job or Edit button will take you to this page to add a new job or edit an existing job. The Bench Part of the form is a placeholder and does not perform or reflect any function. There will be a "Form Submitted Successfully" response under the form and the changes will be reflected in BoardView.

Figure 8: Clicking Submit Completed Jobs in BoardView will show "Jobs Submitted" and all Jobs in the Complete Column will disappear and their data will be updated in the SQL database.

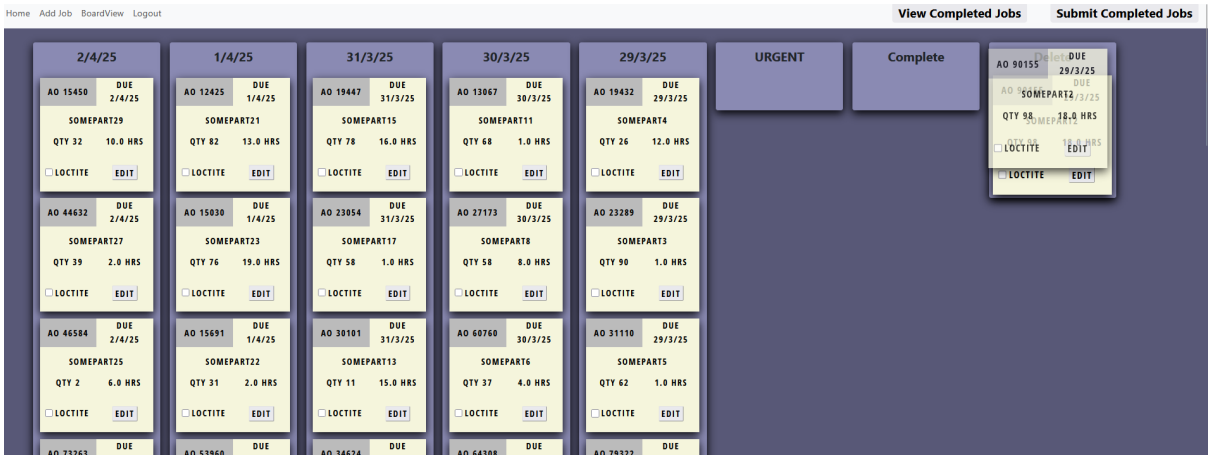


Figure 9: Dropping a card into the Delete column will cause the card to be visually eliminated from the board. It does not update the database so the card will return upon refresh. It's mainly added to reduce clutter if required but can be implemented better.

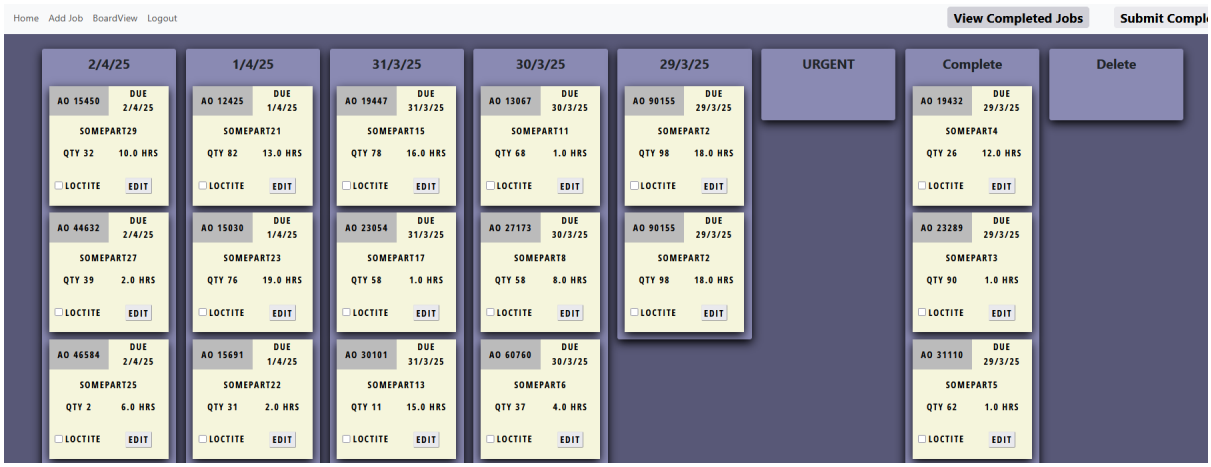


Figure 10: Clicking View Completed Jobs will show the completed jobs in the Complete jobs column.

7. Developer Documentation

8. Tools and Technologies Used (in alphabetical order)

- Bootstrap
- ChatGPT
- CSS
- EJS
- Express.js
- HTML
- JavaScript

- MySQL
- Node.js
- phpMyAdmin
- XAMPP
- Visual Studio Code

9. Overview of software and its source code

Software and source code is available on the github.

https://github.com/amak905/AssemblyDashboard-v1

The readme iterates this but the web application is made to work based on a dynamic data stream as the current date is very important.

You will need to download the xlsx, open it, convert it to csv then import into the SQL to get desired results!! In a real environment, the application would be connected to a data stream.

Provide the entire source code as Appendix A at the end.

10. Known bugs and limitations

Nothing obvious so far..

11. SDLC

The proposed approach was incremental, however a combination of spiral and incremental approaches worked well for this project since parts of the project could be done fairly quickly but others required more time even for incremental improvements.

The transition to spiral was fairly natural and the overall approach is detailed below.

The spiral methodology worked well for this project. The initial spiral through the project was the development of the initial visual framework and user interface for the primary board view. The board view was developed with a basic layout and drag and drop functions were completed with a basic text element.

The spiral then went on to create the basic login page and the login routing.

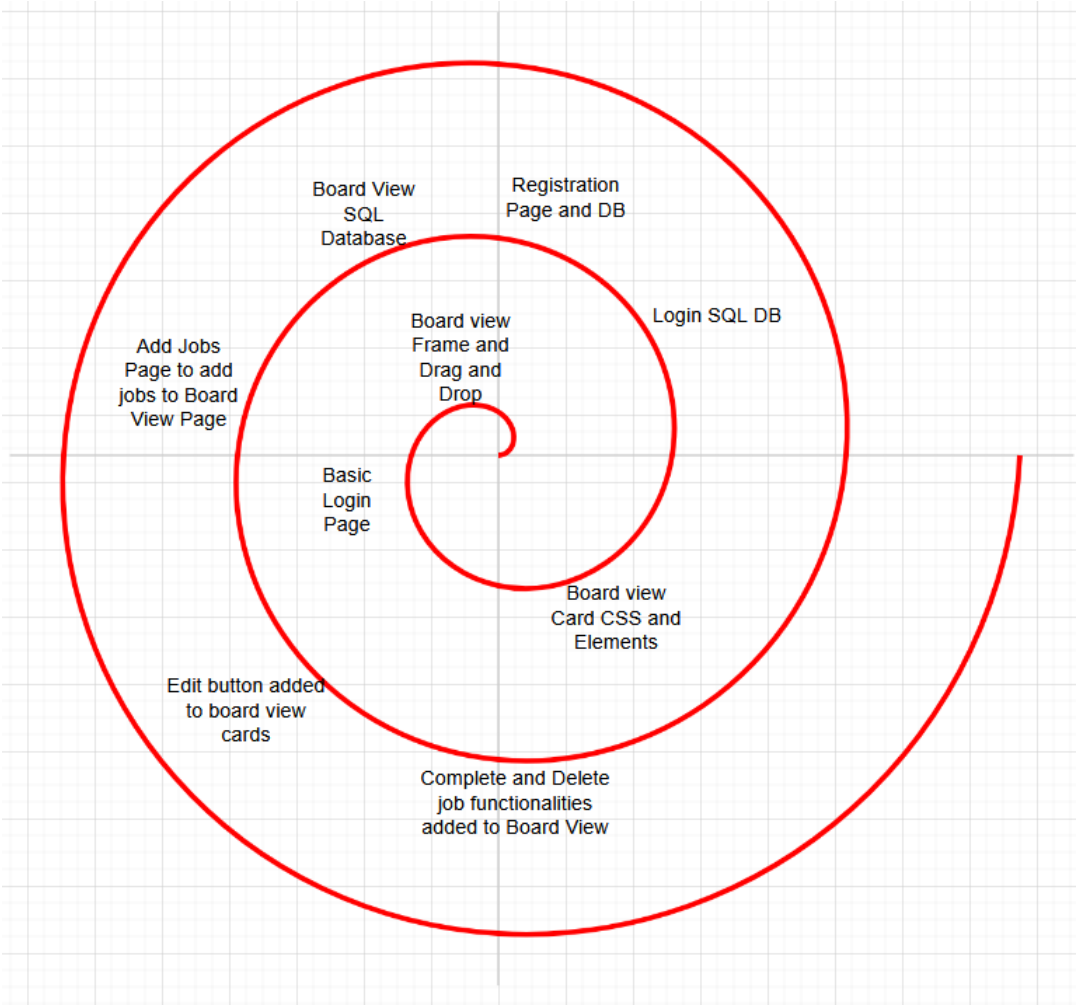
I then went back to the board view to create the CSS for the card and the required elements within the card.

Once all of this was done, the SQL was done for the login page and the routing through the login page was verified.

From the login page, the registration page was developed.

Then the project went on to add the SQL database that the cards on the board view would be getting data from. Then finally a page was added to add jobs to the Jobs database and an edit button was designed onto the cards to allow for editing of jobs.

An image is shown showing the outline of the project progress.



12. Project Timeline

The planning was fairly optimistic in terms of the number of hours that would be available for coding the project. The initial timeline does show an overlap between Module 3 and testing. The time allocated for Module 3 was not available due to addition of other features and tidying up the primary functionality of the project.

13. Ethical and Cultural Impact

As per the initial proposal, there is little to no ethical/cultural impact apart from the enhanced productivity granted by digitisation of manual systems. There is the possible privacy impact of login-ID and password security, however in an intranet environment, the login/password may not be related to a person at all but related to the position instead in which case privacy is not relevant (in an intranet environment in this specific use case).

14. Test Document

In this section, document the test cases and the test results.

Test Case Description	Test Steps	Test Data	Expected Results	Actual Results	Pass/fail
Check Customer Login with Valid Data	Try logging in with a registered ID	Username: user Password: pass	Logged in and sent to BoardView	Logged in and sent to BoardView	Pass
Check user login with invalid data	Try logging in with invalid data at the homepage	Any random inputs	Sent to a please log in page	Sent to a please log in page	Pass
Check user registration works	Click the Register button and fill the form	Any suitable inputs with password confirmation	“Thank you for registering”	“Thank you for registering”	Pass
Check board view drag and drop functionality	Login and go to BoardView	Try dragging, dropping, and sorting cards between columns	Working functionality	Working	Pass
Check board view loads data from SQL Database and sorts cards as per due date	Login and go to boardview	Check data from SQL table generates cards in the BoardView	Boardview should show entries that are incomplete, within 5 days of today, and urgent jobs where the due date has passed	Working as intended	Pass
Check loaded cards where the due date has passed are in the “URGENT” column	Login and go to boardview	Check cards in the Urgent column have dates that are already passed	Check cards in the Urgent column have dates that are already passed	Working as intended	Pass
Check dragging a card into the Delete column makes it disappear	Login, go to boardview, and move a card over to the delete column	-	Card moved over to the delete column should disappear and reappear into the correct column when refreshed	Working as intended	Pass

Check the edit button functionality on a card and check that it is reflected in the board view	Login, go to boardview, click edit on any card, and edit what is required	Any	Edited info should show in boardview.	Working as intended	Pass
Check the 'Submit Completed Jobs' functionality	Login, go to boardview, move cards into the "Complete" Column and press the "Submit Completed Jobs" button	-	Completed jobs should disappear from the board and not return upon refreshing	Working as intended	Pass
Check the 'Add Jobs' functionality	Login, go to boardview, click Add Jobs in the Nav Bar	Add suitable data as per form	BoardView should reflect the new job if it falls within 5 days or the due date is already passed	Working as intended	Pass
Check the Logout button works	Click logout in the nav bar when logged in	-	Displays login page	Working as intended	Pass

15. User Documentation

The most important step is to setup the SQL Database correctly.
The provided database is outdated!

1. Download database.xlsx and open it to update the dates.
2. Export database.xlsx as a csv.
3. Import the csv into the asmjob table of the sql database.

This will show the functionality of the application.

From the home page, the user should click the register function and register a user.
The user can then login with the credentials.
The user can go to the BoardView and drag and drop job cards into various columns.
Functionality is explained in the Design Documents section with screenshots.

16. Configuration Guide

Do not skip this step.
Data is required for checking all functionality of the project so new data will have to be entered into the database.

An excel sheet is provided – export as CSV and import it into the asmjob table, skipping the 1st Row as that is the header.
Alternatively, manual data entry can be done through the Add Job button of the web application.

17. Future Software Enhancement List

In the next version, we can add rows or colours to cards to identify which jobs are in progress. The additional functionality of gathering data from jobs in progress would be the future enhancement to add and would round out the web application.

The final addition would be to link the web-application to a database managed by the ERP software so that the data is always up to date.

18. Reflection

I started this course with little to no knowledge of web development. I had gone into it expected application development in the lines of C++/.NET so it was a bit of a surprise. The instructors were excellent and I was able to pull a project together with their guidance and inputs.

I had some background in C++ and other OOPs so working in Javascript and HTML/CSS was a bit to wrap my head around as the way variables are called is quite different and you don't require an explicit declaration of the variable or type!

The tools learnt here have been extremely valuable however I found the report documentation tedious and repetitive. Having to submit a weekly "What have you learnt" was also extremely redundant for someone with good attendance, active class participation, and consistent implementation of last weeks work in the project. I would recommend that the weekly assignment be a screenshot or github of the final project progress to keep everyone on their toes and not leave starting the project for the end stretch.

Adding references to code from websites is.. although understandable from an "introduction to academia" perspective, not a very useful exercise to list out the things that were googled. All of the code can be learned from the code and library documentation. Not sure what the intent here was.

If I were to do the same project again, I would use more functions and structure the code better. I don't like messy and tedious code.

References

Future Skills Academy. August 2022. JavaScript.

CSS Tooltip. (n.d.). W3Schools Online Web Tutorials. https://www.w3schools.com/css/css_tooltip.asp

How to set input type date's default value to today? (n.d.). Stack Overflow. https://stackoverflow.com/questions/6982692/how-to-set-input-type-dates-default-value-to-today#:~:text=var%20today%20%3D%20moment().,way%20to%20solve%20this%20issue

- *HTML: Hypertext Markup Language | MDN*. (2022, August 10). MDN Web Docs. https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/date

Nurullah, M. (2021, August 30). *Create registration and login form in Node.js & MySQL*. CodingStatus. https://codingstatus.com/create-registration-and-login-form-in-node-js-mysql/

SQL - SELECT from multiple tables with MS SQL server - GeeksforGeeks. (2021, June 14). GeeksforGeeks. https://www.geeksforgeeks.org/sql-select-from-multiple-tables-with-ms-sql-server/

Drag and Drop - https://www.w3schools.com/html/html5_draganddrop.asp

LayOutIt! Layout tool - https://grid.layoutit.com/

19. Appendix: Source Code

https://github.com/amak905/AssemblyDashboard-v1