

A summary of new requirements previously absent, such as new User Stories, new code and tests, new functionality (e.g., a list of new code/test files, names/numbers of user stories, a short description of new functionality)

NEW USER STORY AND FUNCTIONALITY FOR SIGN IN

(Lance)

User Story 1:

As a user, I want to sign-in to my account.

Given: I am on the login page.

When: I enter my username and password,

Then: the application will verify my details and sign me in if it is correct.

Use Case	Sign in	
Actors	User/Customer	
Preconditions	The user must be on the login page	
Postcondition	The user signs into their profile.	
Flow of activities	Actor	System
	<ol style="list-style-type: none">1. The user first enters their username and password.2. The user selects login.	<ol style="list-style-type: none">1. The application handles the form changes.2. The application handles the username and password input and verifies if the user is in the database.3. The application finds the user in the database and proceeds to log them in.
Alternative Flows/ Exceptions	<ol style="list-style-type: none">1. The user's credentials cannot be found in the database (he/she is not registered).<ul style="list-style-type: none">- Proceed to sign up/register page.	

UPDATED AND NEW USER STORIES + NEW WIREFRAMES

(Laura)

NOTIFICATIONS + ALERTS

User Story 2:

As a user, I want to unsubscribe to the notifications of price drops and special deals.

Acceptance Criteria:

Scenario 1: Successfully unsubscribed to the notifications and alerts.

Given: I have previously received notifications or am a new user to the application.

When: I have clicked the option 'Account Notifications' from the 'Account' drop down box on the navigation bar,

Then: I am prompted to either enable or disable the notifications feature. Upon clicking the 'disable' button, I will successfully unsubscribe from receiving notifications. I can also enable the notifications function again at any time if I choose to accept them again.

Scenario 2: Successfully unsubscribed to the notifications and alerts.

Given: I have previously received notifications or am a new user to the application.

When: I have clicked the bell icon from the navigation bar,

Then: I will have successfully disabled notifications. If I click it again, I will have enabled the notifications again.

ITEM SEARCHING

User Story 4: As a user, I want to be able to view my liked items/ products.

Acceptance Criteria:

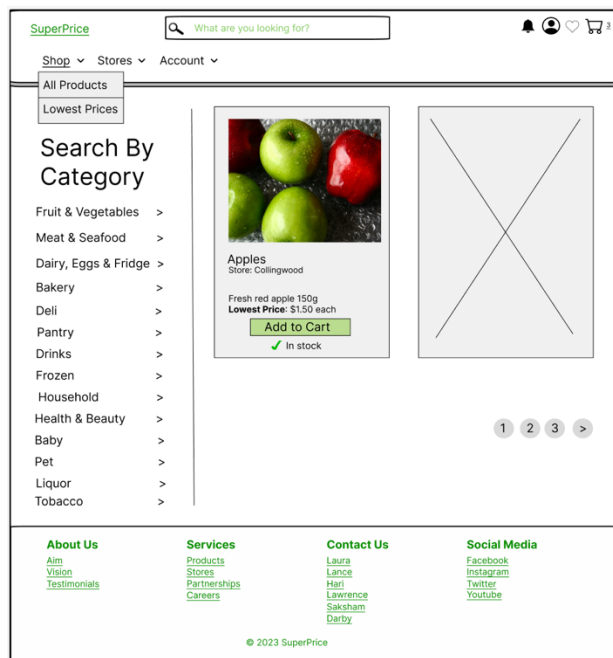
Scenario 1: I have successfully viewed my liked items/ products.

Given: I am on any page with view of the navigation bar,

When: I click on the heart icon in the top right corner of the navigation bar,

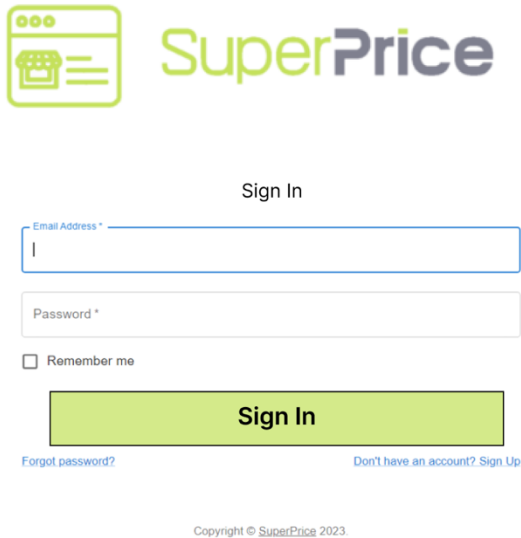
Then: I am taken to a page which displays for me all my liked items/ products, so next time I go to make a purchase, I can easily view my liked items and add them straight to my cart.

NEW PRODUCT LIST PAGE



This page displays the list of products, and has a search engine on the left hand side which makes it easy for the user to click which category they would like to search from. You will be able to see all products (under different categorical selection) on the page, which will display the product, the store its from, the description, its lowest price, if it's in stock, and then will give you the option to add it to cart.

NEW UI FOR SIGN IN



The image shows a web form for signing in to SuperPrice. At the top left is a logo consisting of a green square with a white shopping cart icon and three dots above it. To the right of the logo is the text "SuperPrice" in a green, sans-serif font. Below the logo and text is the heading "Sign In" in a small, black, sans-serif font. The form itself consists of two input fields: the first is labeled "Email Address *" and contains the letter "l"; the second is labeled "Password *". Below these fields is a checkbox labeled "Remember me". A large green button with the text "Sign In" in white is positioned below the checkbox. At the bottom of the form, there are two links: "Forgot password?" on the left and "Don't have an account? Sign Up" on the right. At the very bottom of the page, there is a small copyright notice: "Copyright © SuperPrice 2023."

Sign In

Email Address *

l

Password *

☐ Remember me

Sign In

[Forgot password?](#) [Don't have an account? Sign Up](#)

Copyright © SuperPrice 2023.

The sign in page is mostly functional at the moment. The Username that is currently working is: alice123 and the password is: password123. The links to "forgot password?" and "Don't have an account? Sign up", and the 'remember me' box currently do not work, but we will be working toward making these features functional for the next submission.

PAGE FUNCTIONALITY

At the moment, the only things that are working on the page are the links from the bell, heart, profile, and cart icons, as well as the shop, stores, and account pages from the navigation bar. These links are working and will take you to a separate page, but the pages that they take you to are currently non-functional. The links on the footer of the page do not link to anything either. However, the links to each of our group names do work and will show you a summary of what each of us has done for this assignment so far.

For the Main Content, you can sign in using a base username and password that's been defined in the database. Any incorrect combination gives you an error message. For the products list page, you can search by category from the sidebar, Sort By and Search from the FilterBar and use the Breadcrumbs menu for quick navigation. All the pagination buttons at the bottom work as well.

Database Implementation - Lawrence

Agile Epic: Design and Apply Database Schema for SuperPrice

User Story 1.

As a developer, I want to design and apply a database schema for SuperPrice so that the application can store, manage, and retrieve data efficiently.

Acceptance Criteria:

Given I have a clear understanding of the data requirements for SuperPrice,

When I design the database schema,

Then it should effectively represent all entities, relationships, and constraints necessary for the Super Price application.

Agile Epic: Flyway Used To Manage Database Migrations for H2 database

User Story 1.

As a database administrator, I want to use Flyway to manage database migrations so that I can ensure consistent and version-controlled changes to our H2 databases.

Acceptance Criteria:

Given I have an H2 database and a Flyway migration script,

When I execute Flyway commands,

Then the migration should be correctly applied to the H2 database

And I should be able to view the status of migrations using the Flyway info command.

Agile Epic: Implement H2-Console for Database Environment

User Story 1.

As a developer, I want to implement the H2-Console in the database environment so that I can easily interact with and monitor the H2 database via a web interface.

Acceptance Criteria:

Given I navigate to the H2-Console URL,

When I enter valid database credentials and attempt to login,

Then I should be able to access the H2 database and execute SQL statements.

Given I navigate to the H2-Console URL,

When I enter incorrect database credentials and attempt to login,

Then I should see an error message indicating a failed connection attempt.

Agile Epic: Input Sample Data into Database

User Story 1.

As a customer, I want to select products and place an order, So that I can purchase the items I need.

Acceptance Criteria:

Given a list of available products and my customer information,

When I select products with IDs 101 and 202, and place an order,

Then the database should record the order with the selected products, associate it with my customer information, and update the stock quantities of the purchased products.

Continuous Integration (CI/CD) - Lawrence

Agile Epic: Implement Continuous Integration

User Story 1.

As a developer, I want to have Continuous Integration (CI) implemented for our project so that my code is automatically tested and built whenever I push changes, ensuring that it integrates seamlessly with the existing codebase.

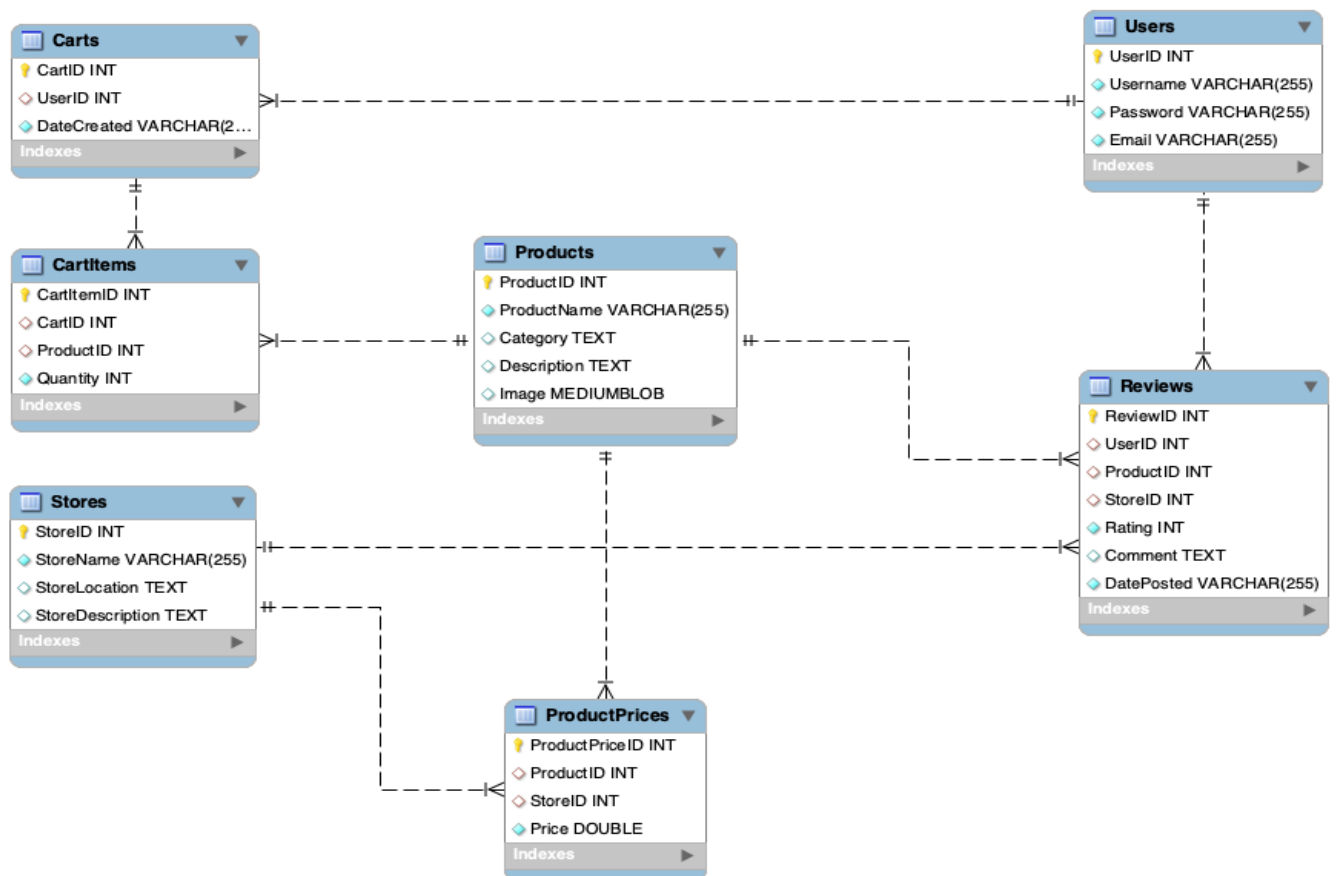
Acceptance Criteria:

Given that I have made code changes and pushed them to our version control repository,

When the changes are pushed,

Then the CI system should automatically trigger a build and test sequence for the latest changes.

Database ERM Model



Use Case 6	Design and Apply Database Schema for SuperPrice	
Actors	Developer	
Preconditions	<p>Developer has access to the system where the application SuperPrice is being developed.</p> <p>Developer understands the requirements and data needs of the SuperPrice application.</p>	
Postconditions	<p>A database schema has been designed and applied.</p> <p>The application SuperPrice can efficiently store, manage, and retrieve data based on the schema.</p>	
Flow of activities	Actor	System
	<p>1. Analyzes the data requirements for SuperPrice.</p> <p>3. Designs the database schema based on the analyzed requirements.</p> <p>5. Applies the schema to the SuperPrice system.</p>	<p>2. Provides necessary data and interfaces for analysis.</p> <p>4. Accepts the schema design.</p> <p>6. Updates the database with the new schema.</p> <p>Confirms successful schema application.</p>
Alternative Flows/ Exceptions	<ul style="list-style-type: none"> ▪ The designed schema is found to be inefficient or not meeting the requirements. <ul style="list-style-type: none"> ○ The developer revises the schema and reapplies it to the system. ▪ Errors occur while applying the schema. <ul style="list-style-type: none"> ○ The developer debugs and corrects the schema before reapplying. 	

Use Case 7	Implement Continuous Integration for the Project	
Actors	Developer	
Preconditions	<p>Developer has access to the codebase of the project.</p> <p>The project has a version control system in place.</p> <p>Developer has knowledge of Continuous Integration (CI) tools and practices.</p>	
Postconditions	<p>Continuous Integration (CI) is set up and operational for the project.</p> <p>Code is automatically tested and built whenever changes are pushed, ensuring seamless integration.</p>	
Flow of activities	Actor	System
	<p>1. Sets up a CI server or uses a CI service.</p> <p>3. Configures the CI server or service</p>	<p>2. Registers and configures the CI setup.</p> <p>4. Observes the repository for</p>

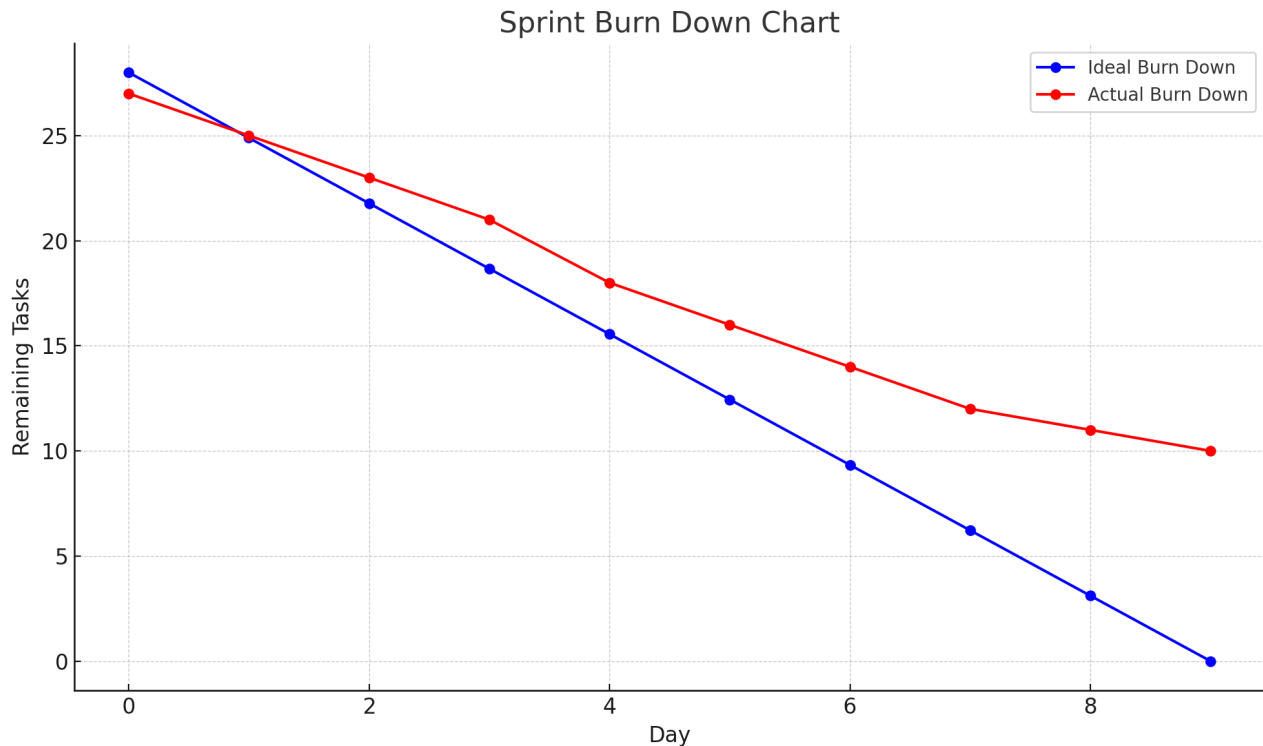
	<p>to watch the project's repository for changes.</p> <p>5. Pushes changes to the codebase.</p>	<p>changes.</p> <p>7. Triggers the CI process.</p> <p>Automatically tests and builds the project.</p> <ul style="list-style-type: none"> - Generates and provides CI reports.
Alternative Flows/ Exceptions	<ul style="list-style-type: none"> ▪ The CI process detects errors or failures in the code. <ul style="list-style-type: none"> ○ The developer reviews the error reports, fixes the issues, and pushes the corrected code. ▪ The CI process experiences issues or becomes non-operational. <ul style="list-style-type: none"> ○ The developer troubleshoots the CI setup and makes necessary adjustments. 	

Product Backlog (Updated)

ID	ITEM	OWNER	PRIORITY	EFFORT	STATUS	DEFINITION OF DONE	NOTES	EPIC
1	Frontend-Backend Integration	Backend	High	13	In Progress	Frontend and backend communicate smoothly	Requires thorough testing and debugging	Application CI/CD
2	Delivery Management System	Backend	High	13	Not Started	Users can schedule and track deliveries	Requires integration with third-party	Product Delivery
3	Price Comparison Algorithm	Backend	High	13	In Progress	Application compares prices across stores	Requires complex algorithm development	Product Search & Comparison
4	Product Search Functionality	Backend	High	8	Completed	Users can search for products	Implement search algorithms and filters	Product Search & Comparison
5	Product Database Setup	Backend	High	8	Completed	Database stores product information	Integrate with chosen database (e.g., MySQL)	Product Search & Comparison

6	User Registration and Login	Backend	High	5	In Progress	User can register and log in securely	Requires encryption and validation	User Management
7	User Interface Design	Frontend	High	8	In Progress	Application has a user-friendly interface	Collaborate with UI/UX designers	UX/UI Design
8	Category Browsing	Frontend	High	5	Completed	Users can browse products by category	Integrate with backend for category data	Product Search & Comparison
9	User Reviews and Ratings	Frontend	Medium	8	Not Started	Users can leave reviews and ratings	Integrate with backend for review storage	User Management
10	User Profile Management	Backend	Medium	5	In Progress	Users can edit profile settings	Allow users to update personal information	User Management
11	Notifications and Alerts	Backend	Medium	8	Not Started	Users receive price drop notifications	Implement notification system	Alerting & Notifications
12	CI/CD Implementation	DevOps	Medium	8	Completed	Automatic testing and deployment setup	Integrate with GitHub Actions	Application CI/CD
13	Performance Optimization	DevOps	Medium	8	Not Started	Application is optimized for performance	Fine-tuning of backend and frontend	Application Enhancement
14	Delivery Options Setup	Backend	Medium	5	Not Started	Users can choose delivery options	Implement flexible time slot selection	Product Delivery
15	Deployment and Hosting	DevOps	Medium	5	Not Started	Application deployed to recommended env	Utilize recommended hosting environment	Application CI/CD
16	User Documentation	Documentation	Low	5	Not Started	Comprehensive user guide and documentation	Essential for user onboarding	User Management

Sprint 2 Burndown Chart



Sprint 3 Planning (Estimate)

Project Sprint Planning Notes

Team: 7

Sprint: Sprint 3

Date: 18/09/2023 – 15/10/2023

Attended: (TBC)

Scrum Master: Lawrence Dawood

Product Owner: Prakash Adhikari

Development team: Darby, Laura, Lance, Saksham, Lawrence, Krisanahari

1. Goal

To enhance the frontend user interface by adding features like user reviews and ratings, and to further optimize the price comparison functionality to include real-time data and promotions.

2. Duration of the sprint

4 weeks

3. What is the team's vision for this sprint?

For this sprint, we envision:

- Expanding the frontend UI to display user reviews and ratings for products.
- Improving the price comparison functionality by incorporating real-time data and promotional prices.
- Enhancing the overall user experience by introducing features like sorting, filtering, and a more intuitive layout. By the end of the sprint, users should be able to not only compare prices but also gauge product quality through reviews, and take advantage of any current promotions.

4. Estimation in story points

- **Frontend design for user reviews and ratings:** 10 points.
 - **Justification:** Incorporating user reviews and ratings requires both frontend design changes and backend data integration. The team must ensure that the reviews are displayed in a readable manner and that users can easily submit their reviews.
- **Optimization of price comparison feature:** 15 points.
 - **Justification:** Incorporating real-time data and promotional prices adds a layer of complexity to the price comparison feature. It requires real-time API calls, error handling, and ensuring that promotional prices are clearly flagged to users.
- **Enhanced user experience features:** 7 points.
 - **Justification:** While features like sorting and filtering are standard, their seamless integration into the existing UI requires careful design and testing.

Sprint 3 Backlog (Estimate)

PBI ID	PBI NAME	TASK ID	TASK	ASSIGNED TO	NOT STARTED	IN PROGRESS	COMPLETED	NOTES
PBI-1	User Interface Enhancements	#28	Sign-In Page UI	Saksham		✓		
PBI-1	User Interface Enhancements	#29	Product Details Webpage	Saksham		✓		
PBI-2	Frontend and Backend Integration	#32	Integrating Frontend with the Backend	Saksham		✓		

PBI-3	User Navigation	#33	Breadcrumbs Menu for easy navigation	Saksham		✓		
PBI-3	User Navigation	#34	FilterBar to search through list of products	Saksham		✓		
PBI-3	User Navigation	#48	Create links on nav bar to different pages	Laura	✓			
PBI-4	User Reviews & Ratings	#49	Frontend design for user reviews and ratings	Saksham	✓			
PBI-4	User Reviews & Ratings	#50	Backend API for user reviews and ratings	Lawrence	✓			
PBI-5	Price Comparison	#53	Optimization of price comparison for real-time data	Lawrence/Hari	✓			
PBI-5	Price Comparison	#54	Integrate promotional prices into price comparison	Hari	✓			
PBI-6	Improved User Experience	#56	Enhanced sorting feature for product list	Saksham	✓			
PBI-6	Improved User Experience	#57	Enhanced filtering feature for product categories	Saksham	✓			
PBI-6	Improved User Experience	#58	Improve overall site layout for	Saksham		✓		

			intuitive navigation					
PBI-7	Backend Database	#59	Database schema refinement	Lawrence		✓		
PBI-8	Backend Optimization	#62	API response caching	Hari	✓			
PBI-9	Backend Security	#65	Encrypt sensitive user data	Lance	✓			
PBI-10	Backend	#66	Implementation of Delivery Database	Lawrence	✓			
PBI-11	Frontend	#67	Checkout Page, Home Page, Delivery Page, Profile Page, Sign up Page, External API (AusPost, etc),	Saksham	✓			

Sprint Retro

Project Sprint Planning Notes

Team: 7

Sprint: 0

Date: 20/08/23 + 17/09/23

Attended: Darby, Laura, Lance, Saksham, Lawrence, Krisanahari

Scrum Master: Lawrence Dawood

Product Owner: Prakash

Development team: Darby, Laura, Lance, Saksham, Lawrence, Krisanahari

1. Things That Went Well

What went well? What the team is happy about?

- We had good communication throughout the sprint.
- The team had clear distinction of the tasks allocated to each member.
- Everyone was concerned about each other's progress and were helping each other when needed.
- The allocation of tasks for frontend, backend, and database was clearly agreed upon.
- The end goal for the sprint was clear and straightforward.

2. Things That Could Have Gone Better

What could have gone better? What the team could improve?

- Everyone having a good and clear understanding of how the application is built (i.e., how everything is connected) could have brought out a better outcome for this sprint.
- The team could revise more on each of their shortcomings in terms of technical skills and communication skills.
- Although we already have good communication, it could still be improved by maintaining a constant/consistent schedule or routine.

3. Things That Surprised Us

What wasn't expected?

- We were surprised by the amount of preparation to setup the base code before actually implementing the application.
- We were also surprised by the amount of trial and errors before having everything running smoothly and properly.

4. Lessons Learned

What have you learned from the above points?

- We could have saved more time by fully understanding how the application works and how everything is connected first before diving straight into implementing it.

- We could have also planned out the implementation of the application better by agreeing on important aspects to avoid spending too much time on fixing errors that could have been easily avoided.

5. Final Thoughts

Things to Keep

- Meetings and constant communication are necessary to keep for the team to achieve the goal.
- Constant use of the project board and communication tools to update everyone on their progress.

Things to Change

- Inconsistent routine or schedule for meetings and communication between team members.
- Better planning of the tasks to be done and allocation of participation in the project.

Summary

Overall, the sprint went well as the goal was achieved. Good communication was practised throughout but could be improved by having a consistent routine or schedule and response time between team members to avoid blockage of tasks. Planning the activities needed to be done beforehand could have also been improved to save time that could have been spent on more important things which could have led to a better outcome for this sprint.