Helix

Smart-Inventory Demo 2

Our Client

Amazon Warehouse Services

About Smart Inventory

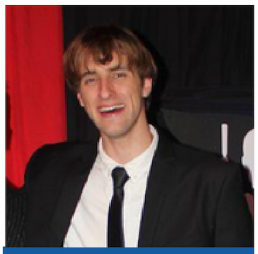
What is Smart Inventory?

Smart Inventory is a user-friendly, web-based application that streamlines and automates core inventory processes.

Who?

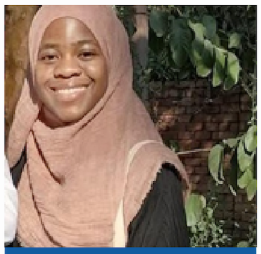
- Business Owners/Managers (Administrators):
- Employees (Inventory Controllers/End-Users):

Our Team

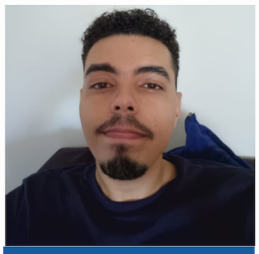


Tristan Sutherland
Project Manager

Bryce Sukhdeo



Hawa Ibrahim



Bouchaib Chraf

Thabang Kgaladi

Links To Documentation

ReadMe

Requirement Specification

Architectural Specification

Design Specification

User Manual

Coding Standards

Service Contract

Testing Policy


LIVE
Demo

Agenda

0 **Introduction**

1 **Documentation**

2 **Implementation Progress**

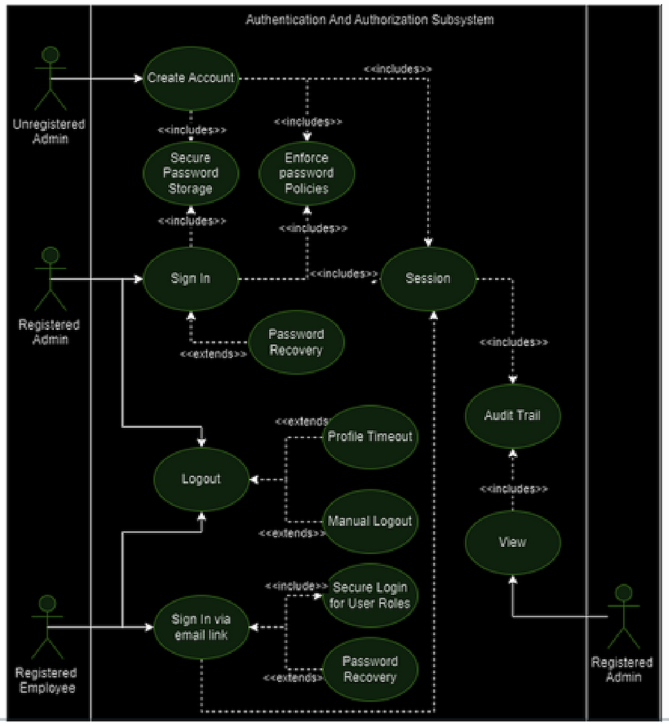
3 **Project Board**

4 **CI/CD**

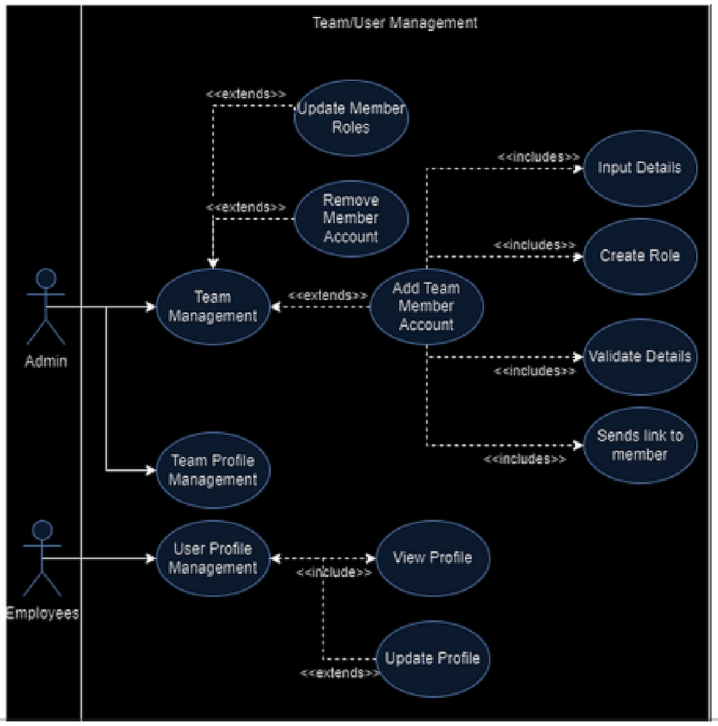
Subsystems

1. Authentication and Authorization Subsystem
2. Team/User Management Subsystem
3. Reporting Subsystem
4. Inventory Management Subsystem
5. Stock Request Subystem
6. Supplier Management Subsystem
7. Order Placement Subsystem

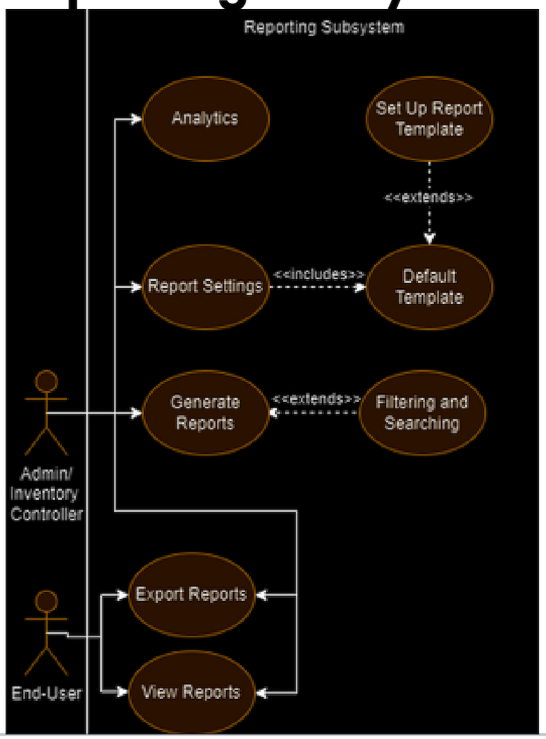
Authorization/ Authentication Subsystem



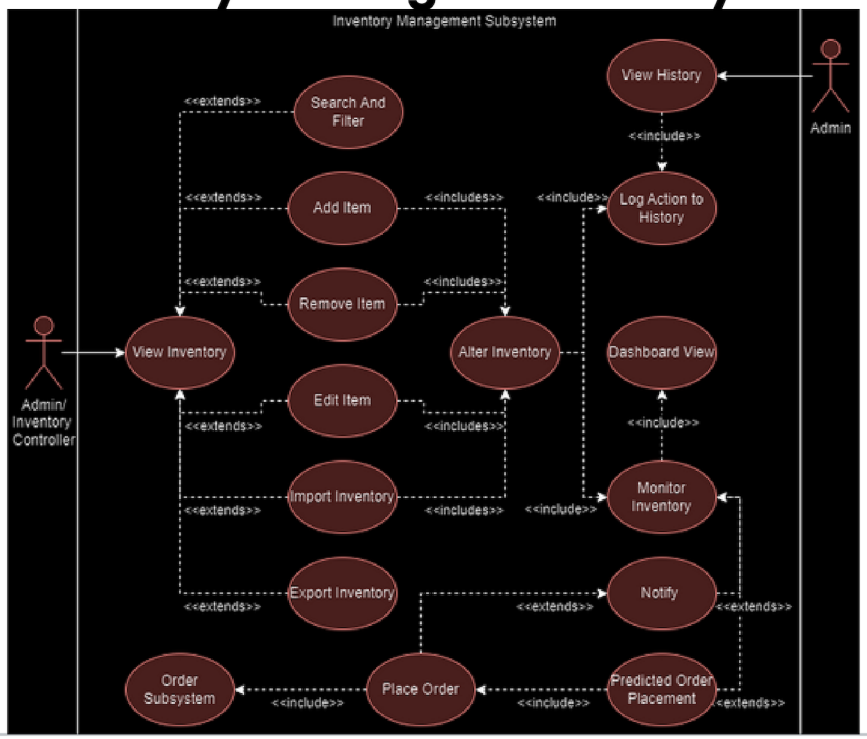
Team/User Management Subsystem



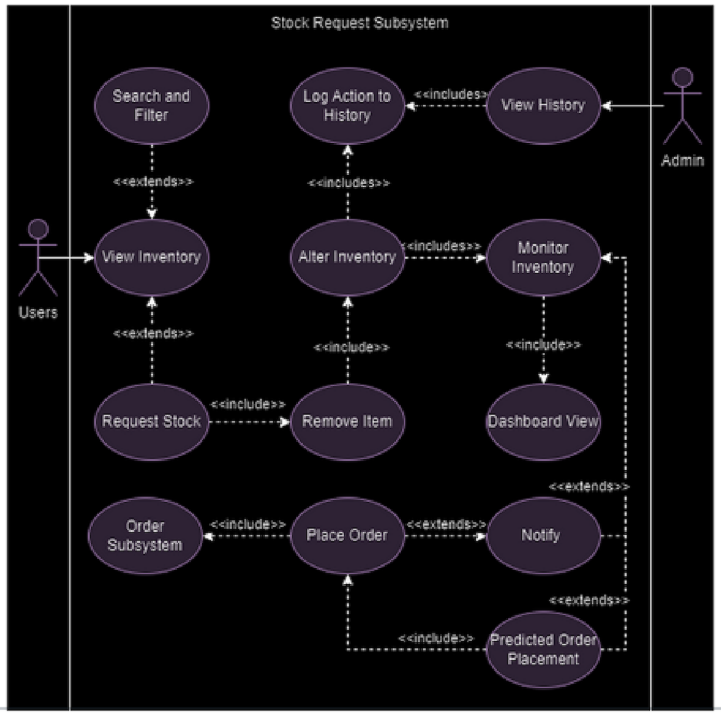
Reporting Subsystem



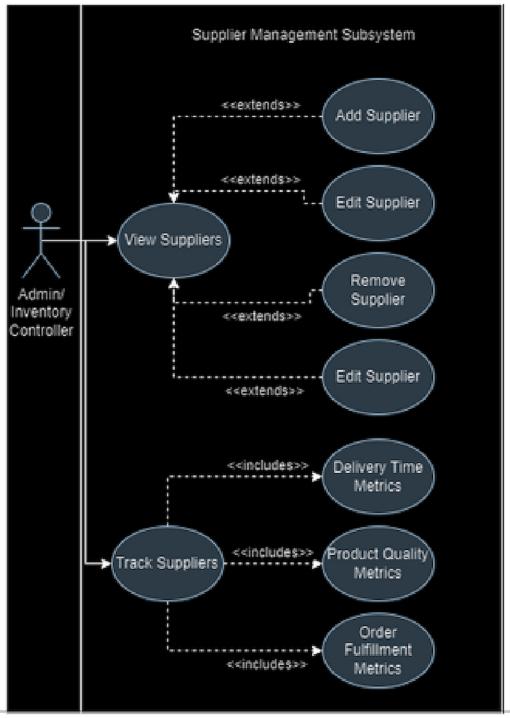
Inventory Management Subystem



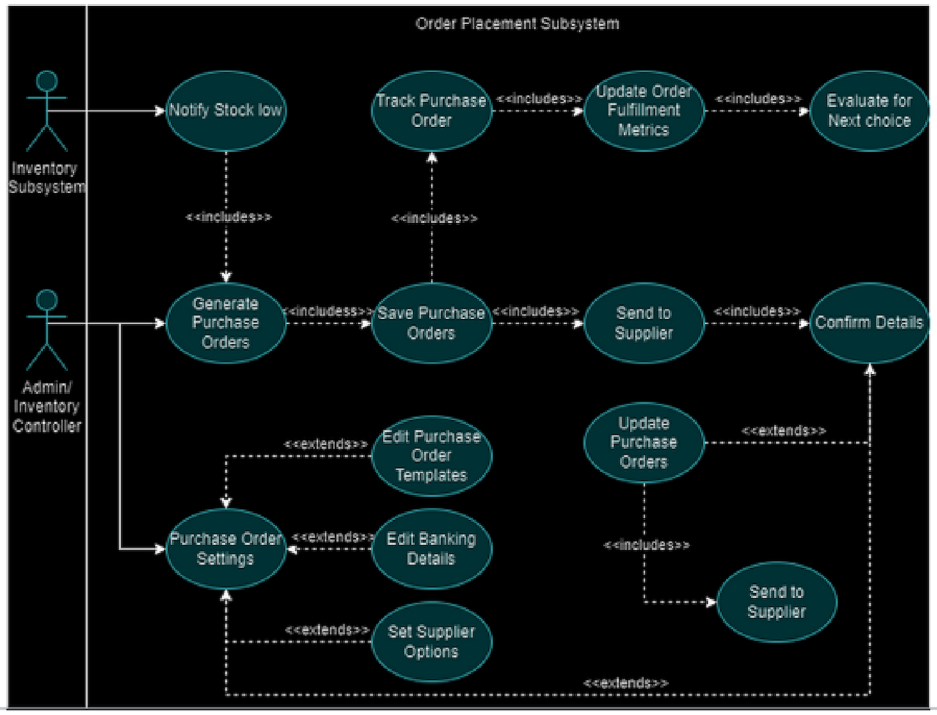
Stock Request Subsystem

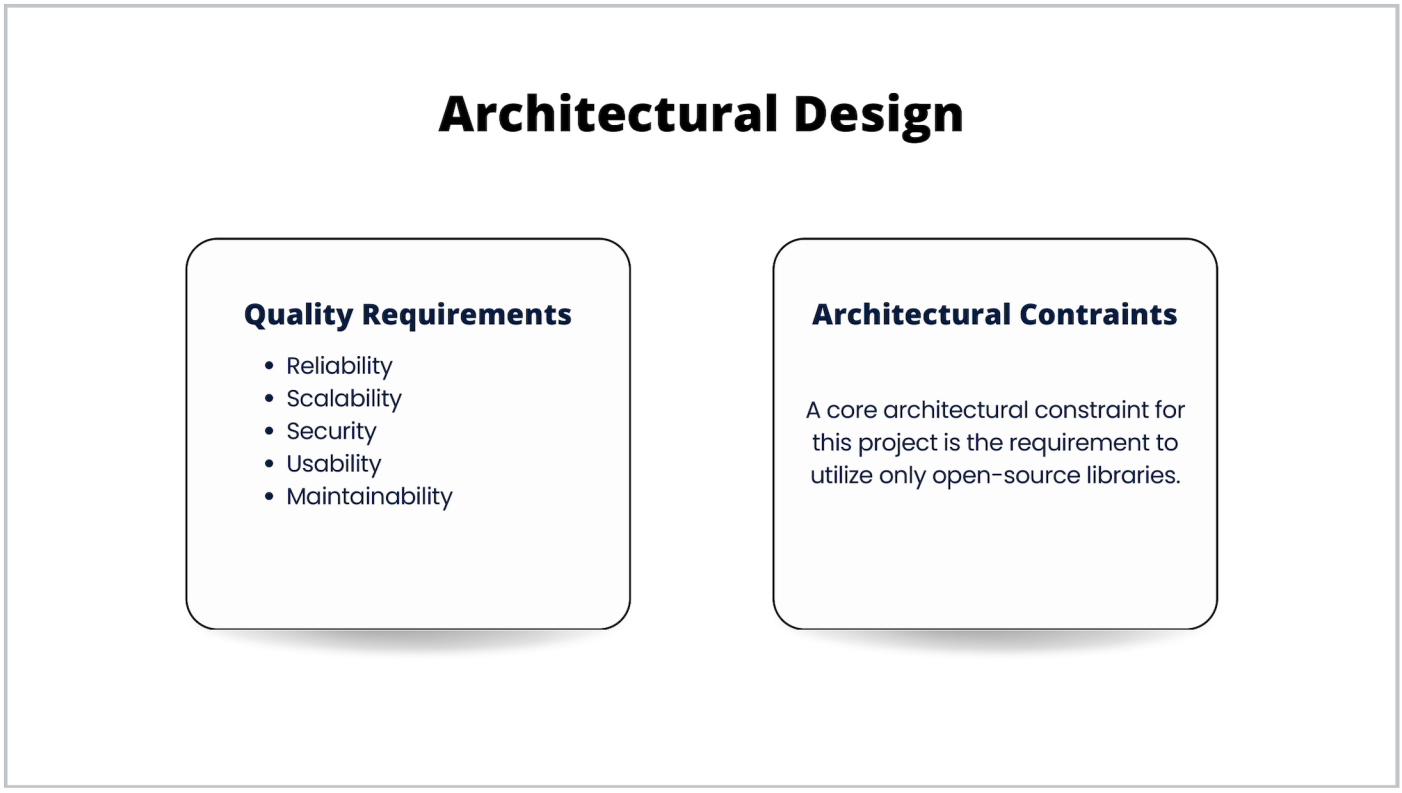


Supplier Management Subsystem



Order Placement Subsystem





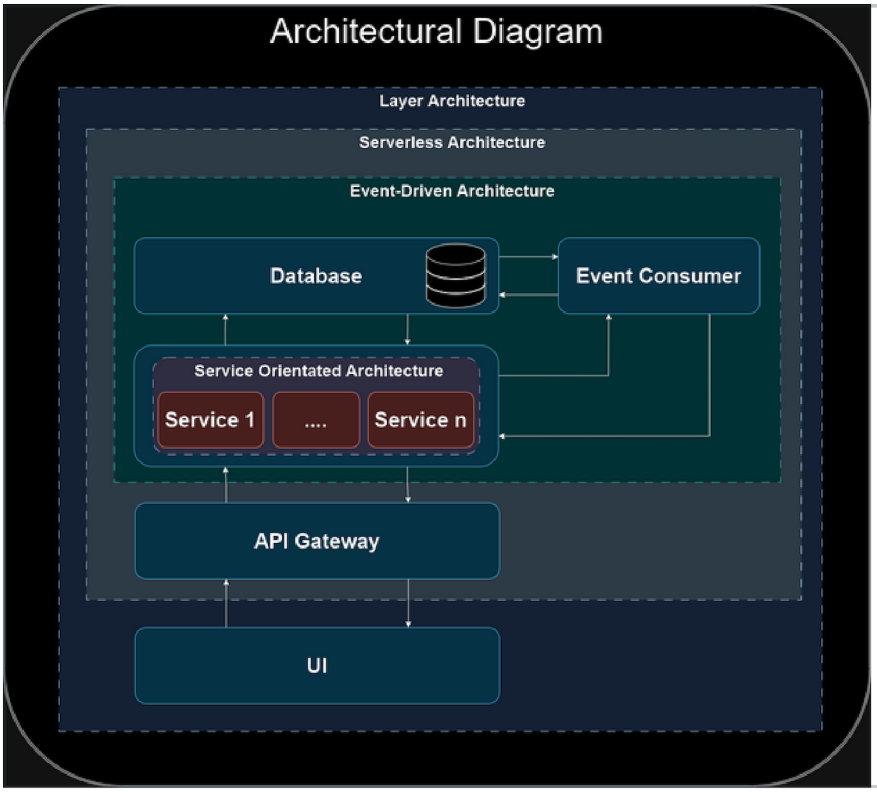
How to quantify security
* Role based access control should be prevelant in all ends of the app including help and settings.

How to quantify usability
* A success rate of at least 70% will be expected of our app when tested.
* A time of acheiving a simple task should take no more than 2 minutes for a new user to acheive.
* A 1-5 scale of how satisfied a user was using the app. A average higher than 3 is the goal.

How to quantify reliability
* A mean time between failures of two times within the same 30 days for two hours.
* Mean time to repair of lesss than five hours is our expectation.

How to quantify scalability
* The system is designed to handle 40 user requests per second.
* The system must also have at least an 80% success rate with more than 20 users.

How to quantify Maintainability
* fixing a bug should not cause more than two hours of un-needed time due to un-maintainable code.
* Adding a new feature should not cause more than two hours of un-needed time due to un-maintainable code.



Architectural Styles

- Layered Architecture
- Service-Oriented Architecture (SOA)
- Event-Driven Architecture (EDA)
- Serverless Architecture

Latest Features

1. Team Management
- Add team member.
- Remove team member.
- Assign roles.
- Edit roles.
- Edit member details
- Export Table
- Search and sort table

2. Inventory Management
- Add Item
- Remove Item
- Remove Multiple Items
- Edit Item
- Export Table
- Search and sort table
- Request Items

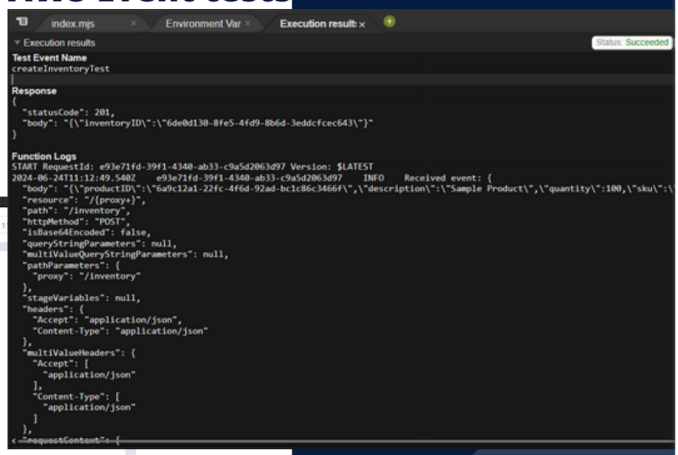
4. Supplier Management
- Add Supplier
- Remove Supplier
- Edit supplier details
- Export Table
- Search and sort table

5. Dashboard
- Add Widget(graph, summary, table)
- Remove widget
- Reposistion widget
- Save state

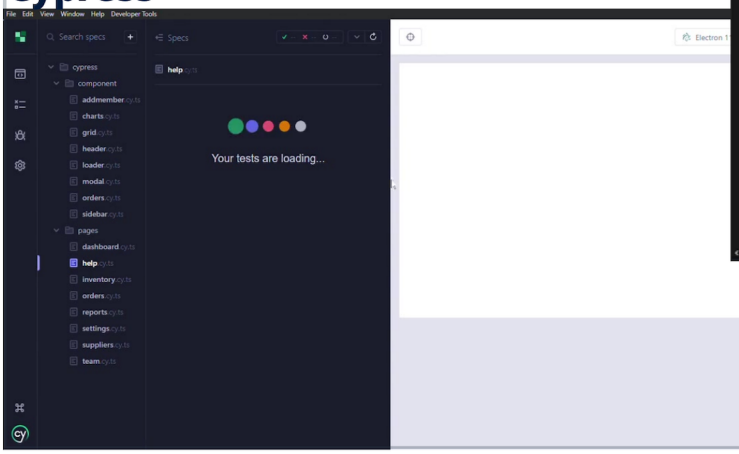
6. Settings
- Change Password
- Change details(name, surname, email)
- Delete Account

Tests - Cypress, Postman and AWS test event

AWS Event tests



Cypress



Deployment

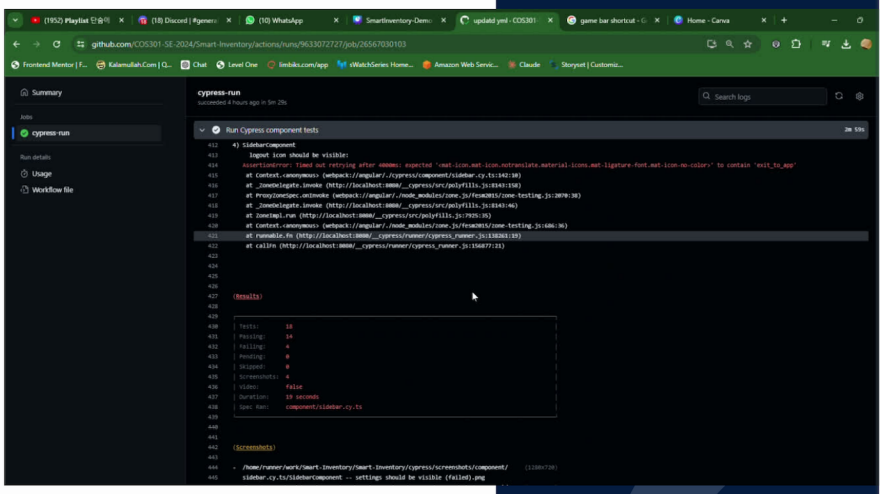
Deployment

- Set Trigger to deploy on push to main.
- Preview deployments through amplify sandbox to help enhance API and web app testing, creation and maintenance.

Testing

Tests

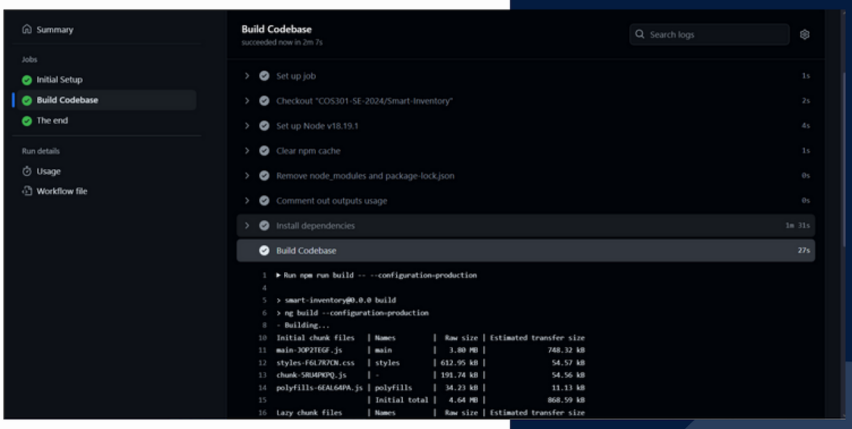
- Set Trigger on pushes to all branches.
- When the trigger event is triggered it will run unit tests and e2e tests.



Linting

Linting

- Set Trigger on merge into main and develop.
- The automated linter ensures:
  - No syntax errors.
  - No coding standards violations.
  - Styling is readable.

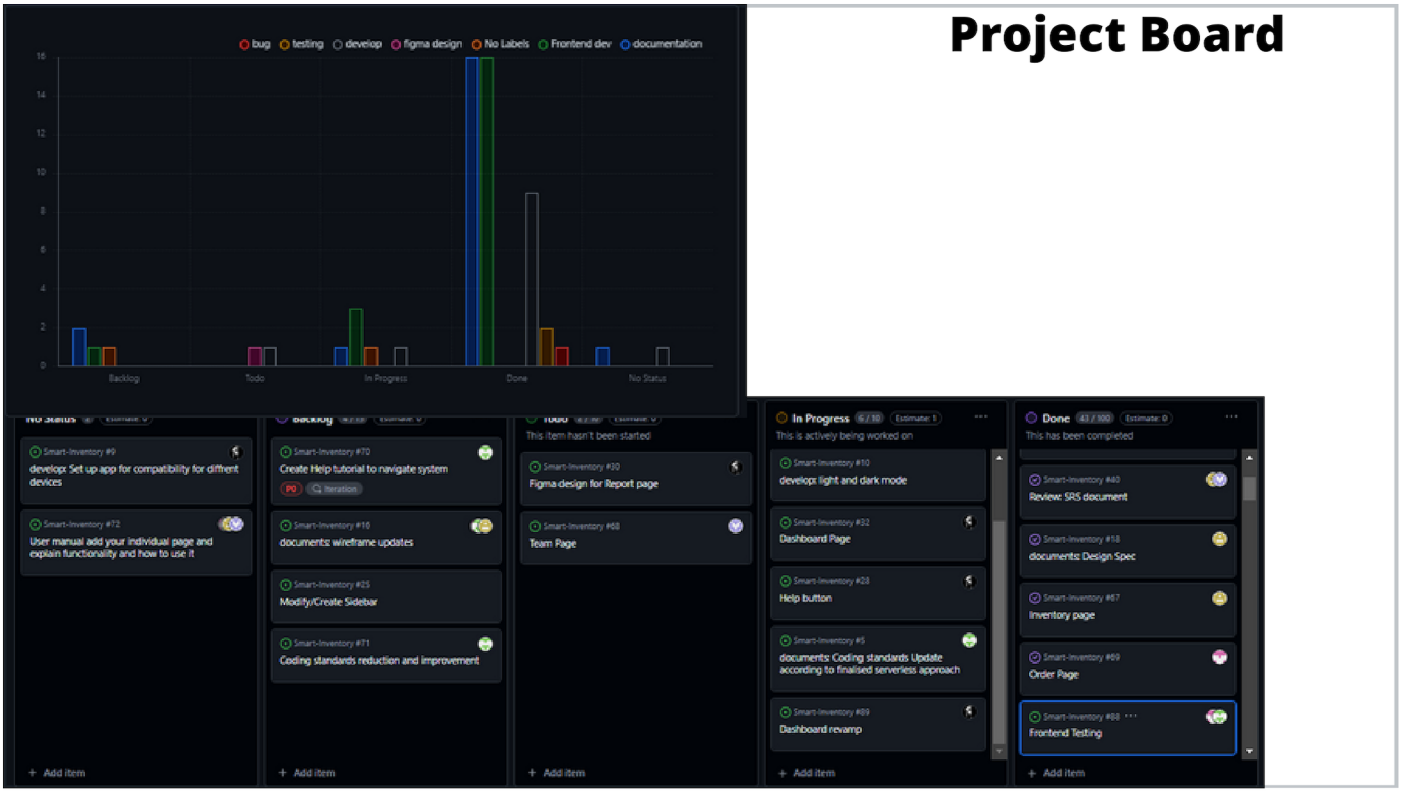


Building

Building

- Set Trigger to build on push to main or develop branch.
- This helps ensure that no issues are around before the web app is deployed.

Project Board



THANK YOU!

QUESTIONS?