Grab

TASK 2
MEX Assistant – Insights

Kodang Koding Kiding

Grab

Kodang Koding Kiding

Meet Our Team

Matthew Alexander

Project Manager

Ivan Nathanael

AI Engine Developer

Gregorius Felix

Frontend Developer

Keisha Nalani

Data Report Specialist

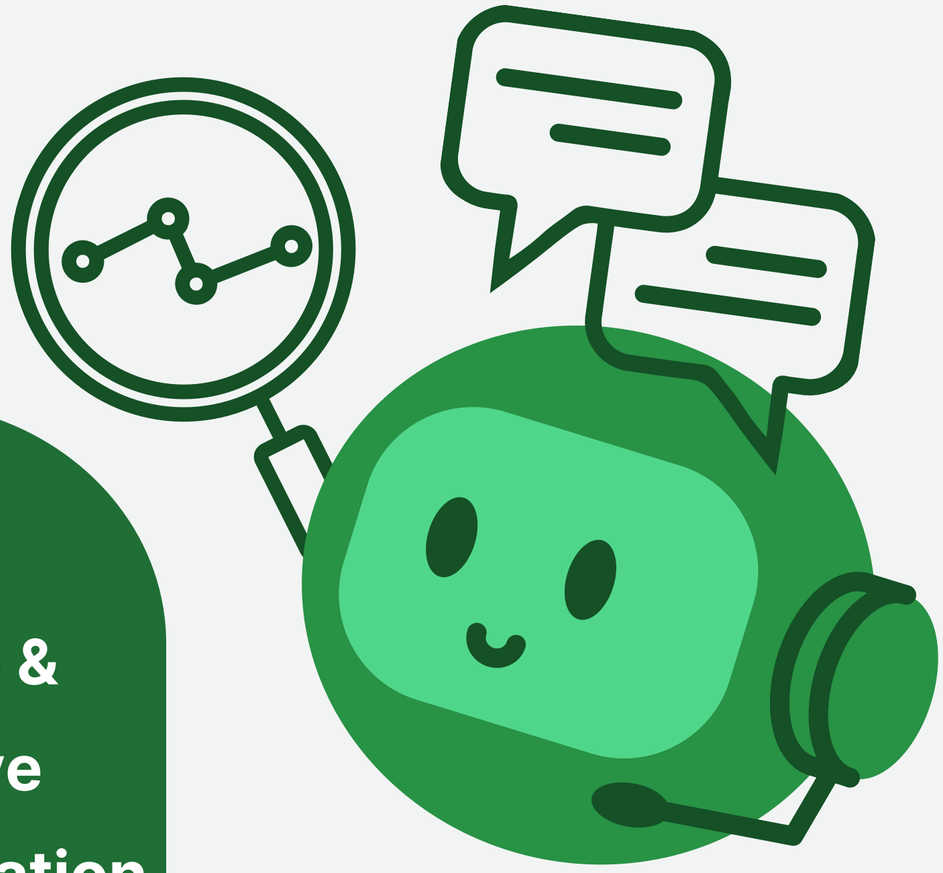
Abel Nathanael

Backend Lead

Kodang Koding Kiding

Task 2 - Problem

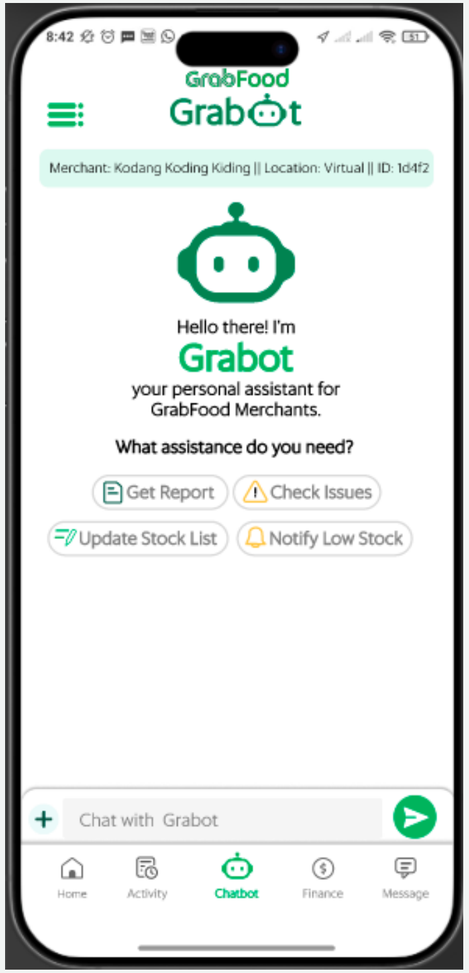
Developing a **Chat-Based Assistants** that helps **Grab's Merchant-Partner** run their business better by giving:

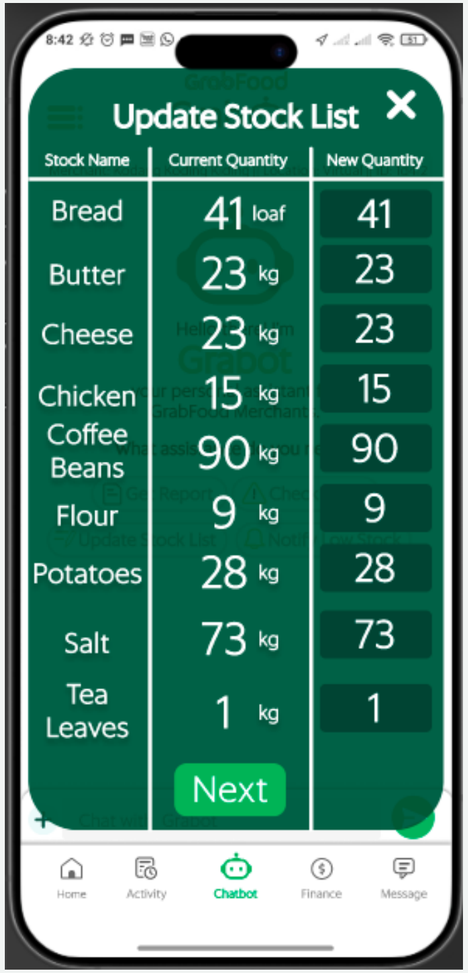


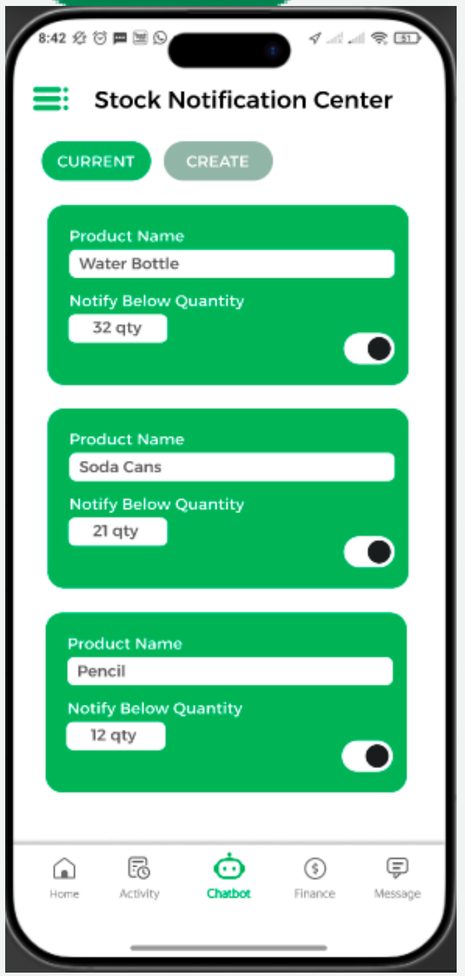
- **Real-Time Business Insights**
- **Personalized Merchant Guidance**
- **Intuitive & Effective Communication**

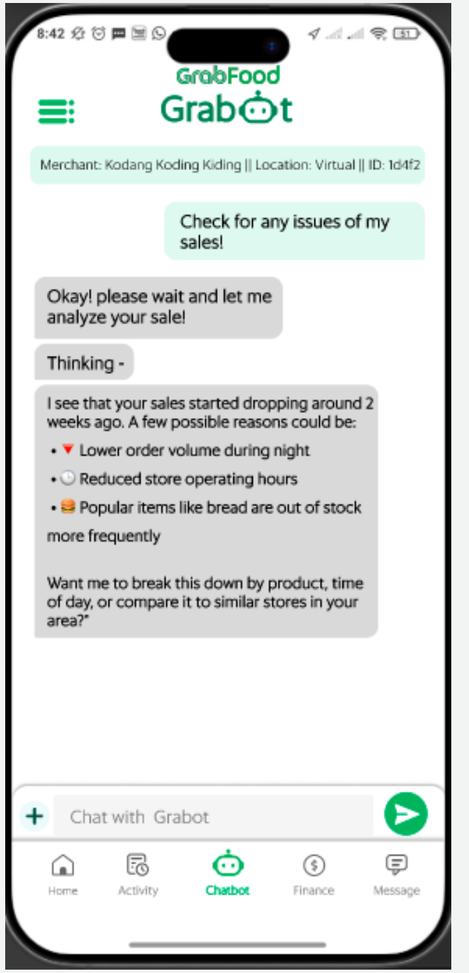
Kodang Koding Kiding

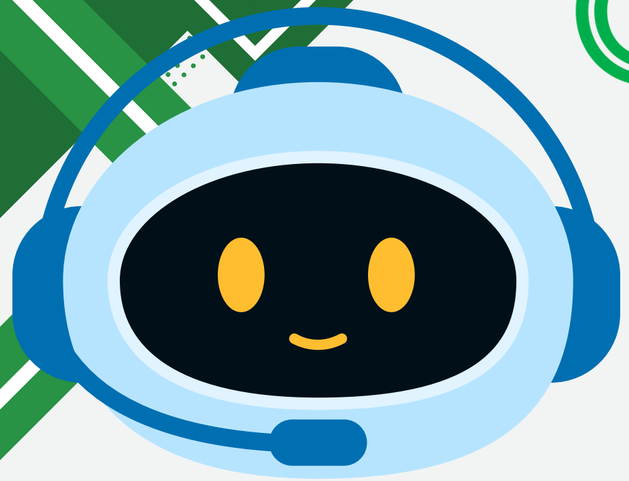
Task 2 - Product Grabot











Kodang Koding Kiding

Assistant Chatbot

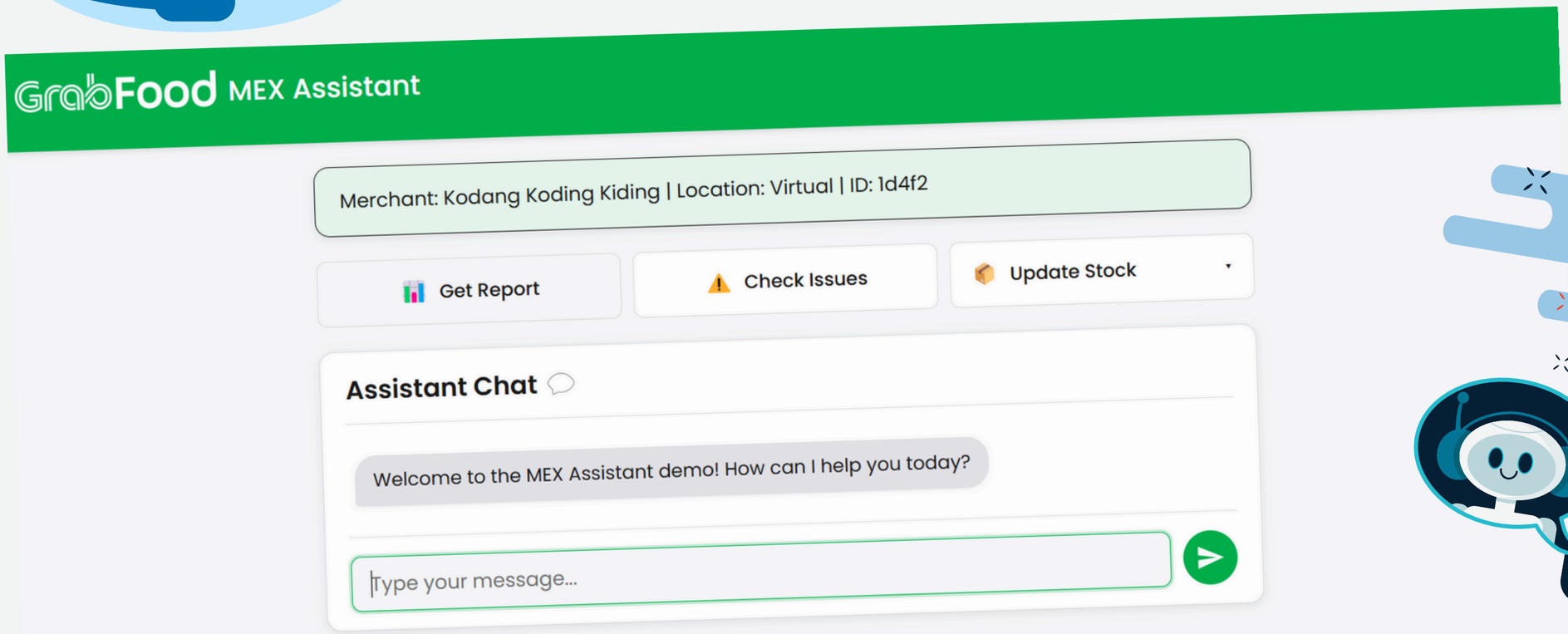
The chatbot acts as the main interface for the MEX Assistant, designed to help GrabFood merchants understand their business data, get reports, identify issues, and receive guidance

Commitment	Powered by Google's Gemini Generative AI model
Interaction	Messages are handled via the /chat endpoint in the Flask app
Query Engine	Receives user questions as well as builds detailed prompts for Gemini
Tool Capabilities	Can request the execution of backend tools to get information or perform tasks
Overall Function	An AI assistant that combines a large language model (Gemini) with specific business context and data access/processing tools to provide relevant insights and answers for merchants

Kodang Koding Kiding

Assistant Chatbot

MEX Assistant - Grabot



Kodang Koding Kiding

Get Daily Report

1

Sales Report

Total sales amount for the effective date.

2

Orders Report

Total number of accepted orders for the effective date.

3

Sales Trend Data

Data structure for sales trends (charts).

4

Item Stock Trend Data

Data structure for item quantity trends (charts).

5

Stock Forecast

List of items predicted to run out soon.

6

Low Stock Alert

List of items currently below the low stock threshold.

7

Top Products Pareto

Pareto analysis of top products by revenue for the day.

8

Words of Encouragement

Data structure for item quantity trends (charts).

Kodang Koding Kiding

Get Daily Report

Assistant Chat

Welcome to the MEX Assistant demo! How can I help you today?

Okay, I will get the latest report.

Okay, here's the planned structure for the Daily Report (Implementation In Progress):

Daily Report Template

Report Date: `[Date for which report applies]`

1. Basic Info

- **Merchant Name:** `[Your Merchant Name]`
- **Type:** `[e.g., Restaurant, Hawker]`
- **Location:** `[e.g., Central]`
- **Business Maturity:** `[e.g., 1.5 years]`
- **Scale Tier:** `[e.g., Silver Tier]`
  - *Note: Research needed for exact scale definition*

2. Performance Metrics

- **a. Sales Overview:** `[Summary of sales trends - e.g., Sales: $XXX.XX (N orders)]`
  - *[Chart: Sales over time]*
- **b. Item Sales:** `[Summary of items sold]`
  - *[Chart: Items sold over time]*
- **c. Top Products:** `[List of top items]`
  - *[Chart: Pareto analysis]*

3. Inventory Insights

- **Stock Run-out Forecast (1-3 Days):**
  - `[Item 1]: Est. [X] days left`
  - `[Item 2]: Est. [Y] days left`
  - *Based on recent sales velocity*

4. Word of Encouragement

- `[Motivational message based on performance]`

Status: Please note, the actual data calculation and population for this report are **still under development**. This is a template of the intended output structure based on project requirements.

Kodang Koding Kiding

Detecting Anomalies

- **Sales Drop (Day-over-Day):** Monitors daily sales, alerting you to major drops compared to the previous day and analyzing potential causes by category or item.
- **Low Acceptance Rate:** Monitors the percentage of orders you accept, notifying you if it falls below the recommended level.
- **Low Stock:** Identifies up to three items with critically low stock levels based on your set minimum quantity (below the set threshold).

By monitoring sales trends, acceptance rates, and stock levels, this feature **provides actionable alerts**. It enables merchants to **address potential problems early**, ensuring smoother operations and minimizing negative impacts on their business.

Kodang Koding Kiding

Detecting Anomalies

Assistant Chat

Welcome to the MEX Assistant demo! How can I help you today?

Okay, I will check the system for anomalies.

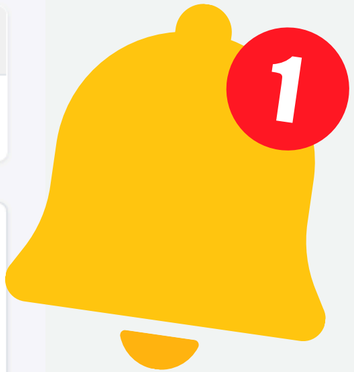
Hi! I checked the system for anomalies in your recent operations (comparing yesterday to the day before) and didn't find anything unusual. Everything seems to be running smoothly!

Notifications

Manage Low Stock Notifications

Current Notification Rules

Product Name	Notify Below Quantity	Status	Actions
Bread	10 (loaf)		



Create New Notification Rule

Product Name: -- Select a Product --

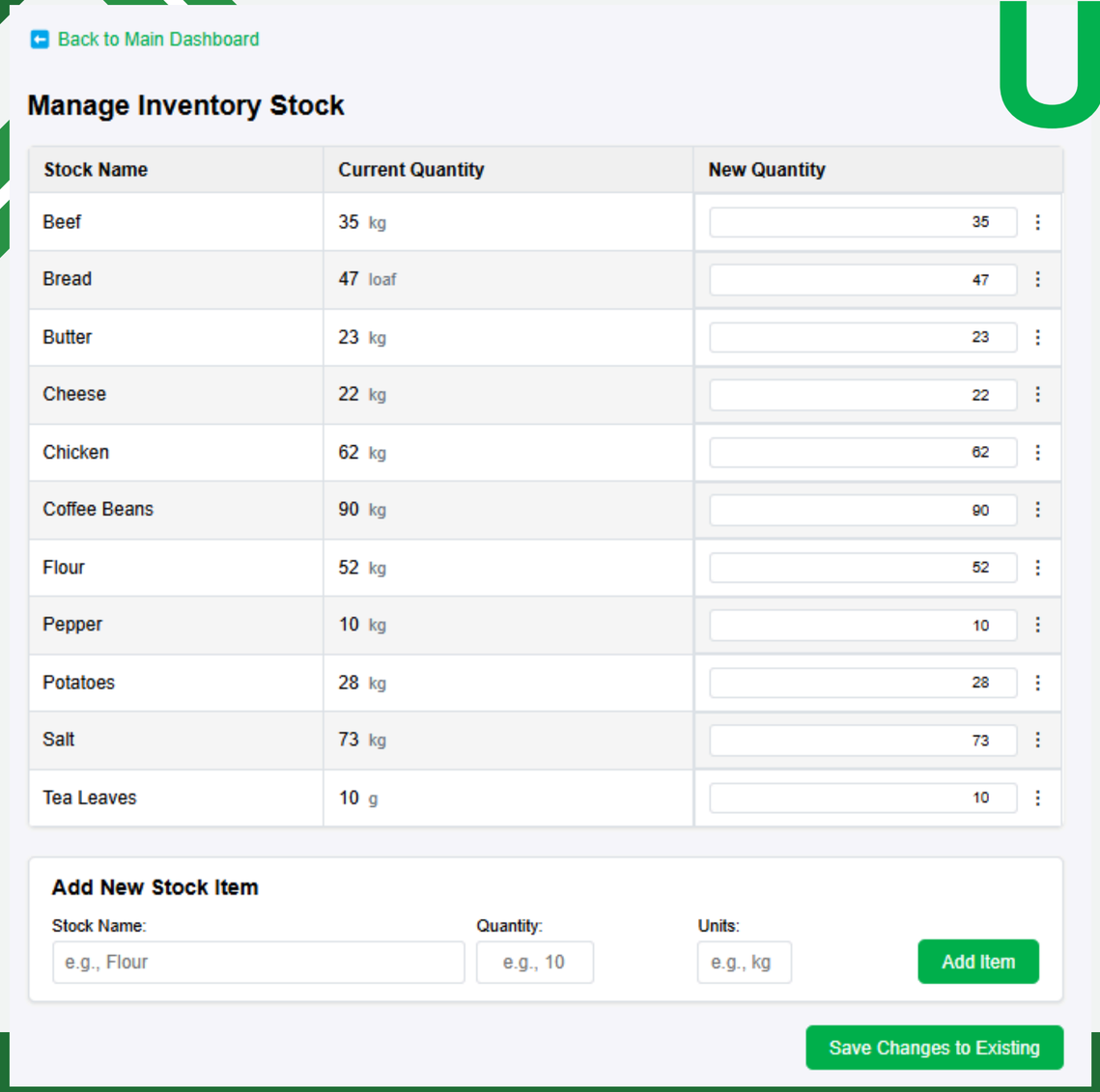
Notify below quantity: e.g., 5

Enable Notification:

Add Rule

Kodang Koding Kiding

Update Stock



Automatically retrieves and displays the merchant's current inventory in another tab.

How the inventory works?

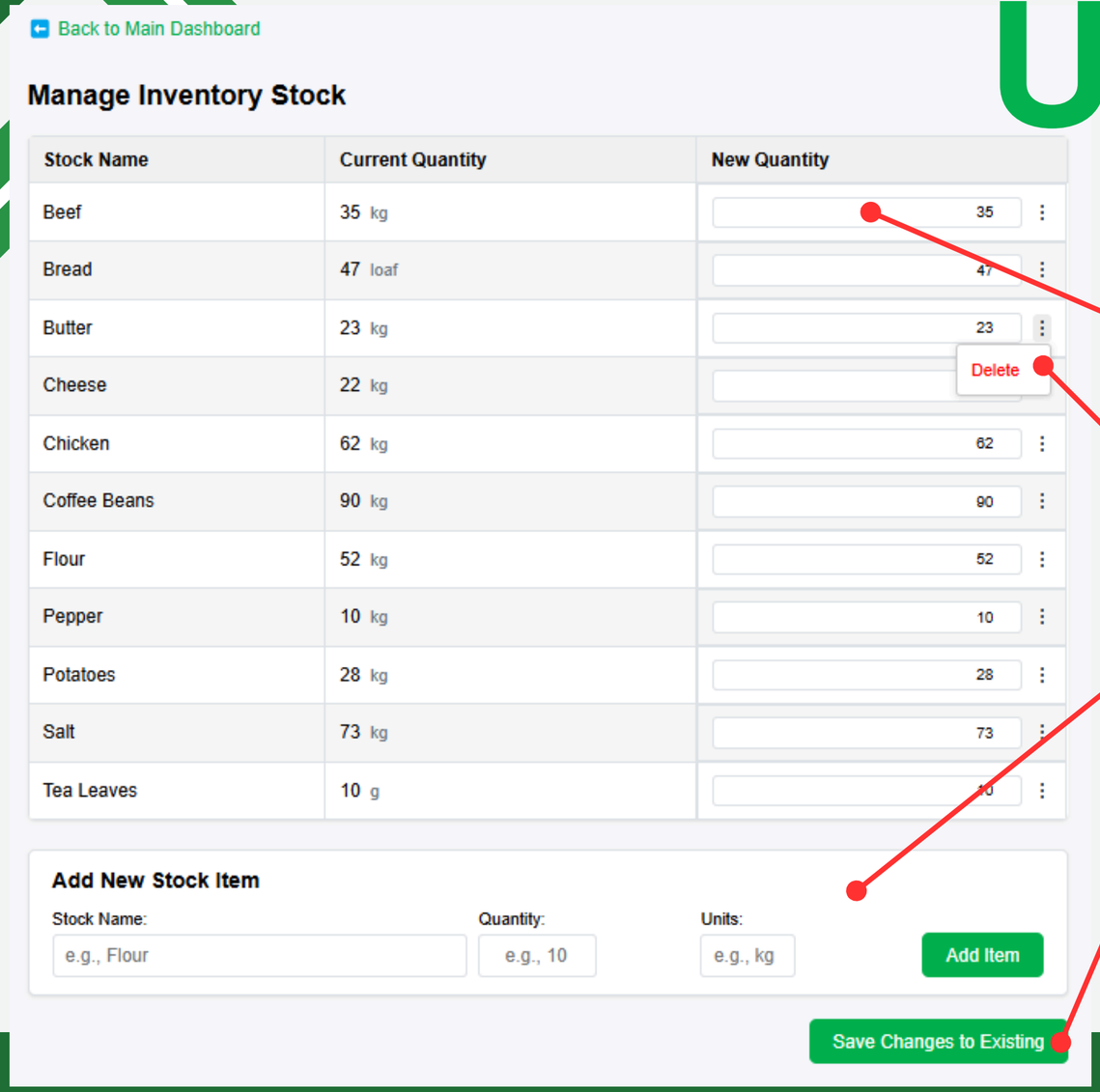
Looks through the merchant's entire inventory history, calculating the most recent quantity for each item based on the latest update date.

Entities in The Inventory

- **Stock Name** → Product Name
- **Current Quantity (w/ unit)** → Initial Product Count
- **New Quantity** → Updated Product Count

Kodang Koding Kiding

Update Stock



How The Inventory is Operated

Modify: Allows you to adjust the recorded quantity for an item that is already part of your inventory list.

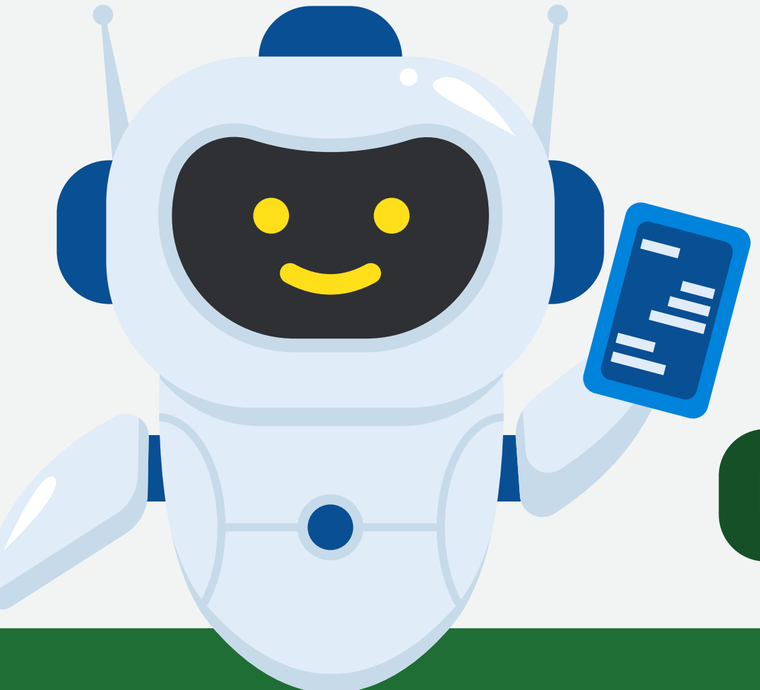
Delete: Permanently removes the selected stock item and its details from the inventory list.

Add New: Input a new type of stock item, along with its initial quantity and unit, into the inventory system.

Save Changes: Updates the system's record of current stock levels, applying any quantity adjustments you made in the "New Quantity" fields for existing items.

Grab

Thank you



Kodang Koding Kiding

```
getWeather = async
  const apiCall = fetc
  .then(res => res.json()
  .then(res => {
    console.log(res);
    let city = res.name;
    let country = res.sys.country
    let weatherDescription = res.wea
    let currentTemp = res.main.temp;
    let maxTemp = res.main.temp_max;
    let minTemp = res.main.temp_min;

    console.log(weatherDescription);
    console.log(currentTemp);
    console.log(maxTemp);
    console.log(minTemp);

    this.setState({
      city: city,
      country: country,
      weatherDescription: wea
      currentTemp: curre
      maxTemp: maxTem
```