Members:

Bondoc, Jonas Jason D.

Chin, Yuan Carlo M.

Tulio, David John Emmanuel G.

Manalo, Gabriel Edward L.

ADMIN USER MANUAL

How to Run the System

Step 1: Install dependencies

Open the project folder in VS Code or terminal, then run:

npm install express cors

Step 2: Start the server

Run the command: node

server.js

You should see:

Server running on http://localhost:3000

Step 3: Access the Admin Pages Open a

browser and go to:

- **Dashboard** - http://localhost:3000

- **Products Management** - http://localhost:3000/product.html

- **Customer List** - http://localhost:3000/customer-list.html

- **Messages** - http://localhost:3000/customer-messages.html

Dashboard Overview:

Features:

- Sidebar Navigation – Easily move between different sections: Dashboard, Products, Customers, and Messages.

- Dashboard Cards – Each card represents a section:

o **Products**: Manage taho flavors, add new products, and update prices. o

Customers: View customer information and history. o **Messages**: Read messages from customers. o **Analytics**: Future section for sales tracking and data visualization.

Dashboard Overview View (dashboard.html):
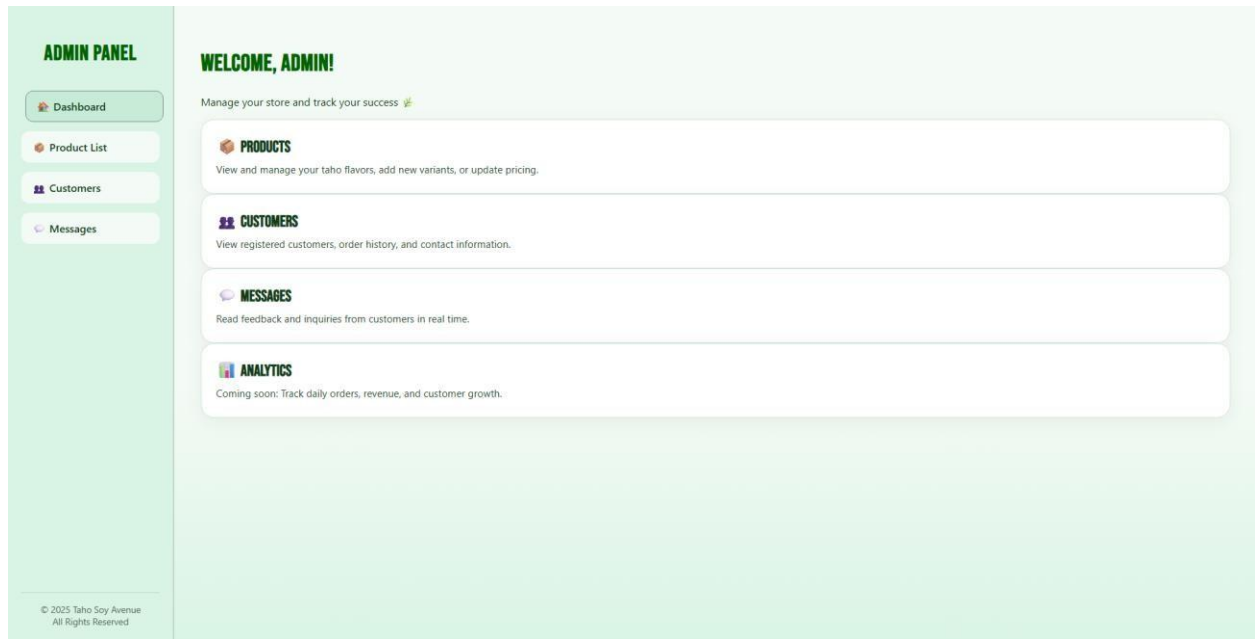
Product Management

Features

- Displays all products in responsive cards with:

- Product Name

- Category (Classic, Specialty, Toppings, Cookies)

- Size (if available)

- Price

- Description

- Includes a search bar to find products quickly by keyword.
  Includes a category filter to view only selected product types.
  Features an "**Add Product**" button that opens a modal form for adding new products.
  Each product card includes:

o **Edit button**: Opens a model with pre-filled product data for easy editing. o **Delete button**: Removes a product after admin confirmation.

o The modal form includes fields for name, description, price, size, and category, with a simple and modern design.

o Admin can easily copy product data for inventory tracking. o Automatically updates product list after add/edit/delete actions — no need to refresh the page.

Uses a Node.js (Express) backend with **/api/products** endpoints to handle:

- **GET → Display all products or filter**

- **POST → Add a new product**

- **PUT → Update existing product**

- **DELETE → Remove product**

Product Management View (product.html):

Customer List:

Features:

- Displays: o Customer ID o Full Name o Email Address o Phone Number o Date Joined
- Includes a **search function** to quickly find a customer by name or contact number.
- Helps staff manage customer loyalty programs or verify information.
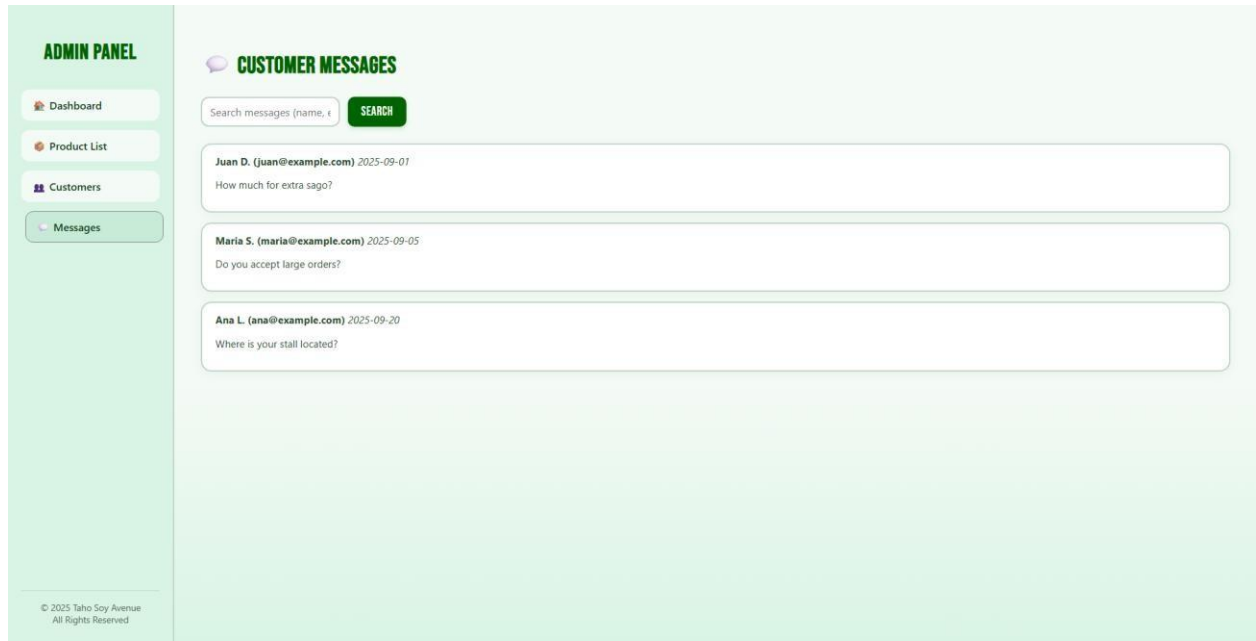
Customer List View (customer-list.html):
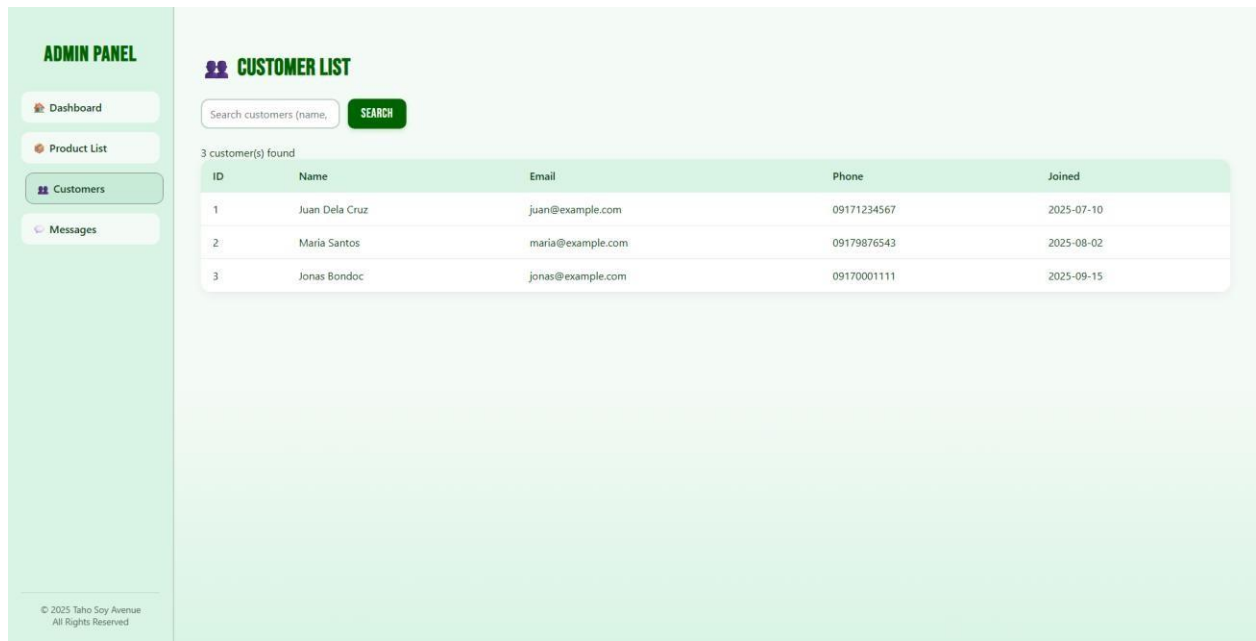
Customer Messages:

Features:

- Displays: o Sender Name o Email o Message Content o Date Sent
- Admin can view questions such as product availability or location inquiries.
- Helps improve customer response efficiency.

Customer Messages (customer-messages.html):

API Endpoints Summary

Endpoint	Method	Description
/api/products	GET	Fetch all product data
/api/customers	GET	Fetch customer list
/api/messages	GET	Fetch contact messages
/health	GET	Server health check

/api/products (View):

Pretty-print ☑

```
{
  "count": 20,
  "data": [
    {
      "id": 1,
      "category": "classic",
      "name": "Classic Arnibal",
      "size": "Large",
      "price": 95,
      "desc": "Traditional silken tofu with sweet arnibal syrup and chewy sago pearls"
    },
    {
      "id": 2,
      "category": "specialty",
      "name": "Biscoff Taho",
      "size": "Large",
      "price": 150,
      "desc": "Classic taho topped with Biscoff cookie spread with whipped cream and real Biscoff crumble"
    },
    {
      "id": 3,
      "category": "specialty",
      "name": "Cookies & Cream Taho",
      "size": "Large",
      "price": 150,
      "desc": "Classic taho topped with Cookies & Cream spread with whipped cream and crumbled cookies"
    },
    {
      "id": 4,
      "category": "specialty",
      "name": "Salted Toffee",
      "size": "Large",
      "price": 150,
      "desc": "Silky tofu with salted toffee syrup with whipped cream and chewy sago pearls"
    },
    {
      "id": 5,
      "category": "specialty",
      "name": "Strawberry",
      "size": "Large",
      "price": 105,
      "desc": "Fresh strawberry syrup with soft tofu and sago pearls"
    },
    {
      "id": 6,
      "category": "specialty",
      "name": "Pandan Taho",
      "size": "Large",
      "price": 105,
      "desc": "Creamy Pandan syrup with fresh buko and silken tofu and sago pearls"
    },
    {
      "id": 7,
      "category": "specialty",
      "name": "Matcha Taho",
      "size": "Large",
      "price": 105,
      "desc": "Japanese matcha syrup with soft tofu and sago pearls"
    },
    {
```

/api/customers (View):

localhost:3000/api/customers

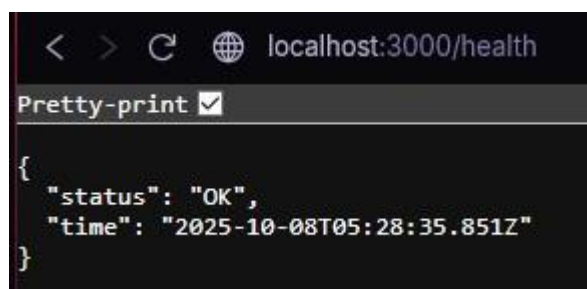Pretty-print ☑

```json
{
  "count": 3,
  "data": [
    {
      "id": 1,
      "firstName": "Juan",
      "lastName": "Dela Cruz",
      "email": "juan@example.com",
      "phone": "09171234567",
      "joined": "2025-07-10"
    },
    {
      "id": 2,
      "firstName": "Maria",
      "lastName": "Santos",
      "email": "maria@example.com",
      "phone": "09179876543",
      "joined": "2025-08-02"
    },
    {
      "id": 3,
      "firstName": "Jonas",
      "lastName": "Bondoc",
      "email": "jonas@example.com",
      "phone": "09170001111",
      "joined": "2025-09-15"
    }
  ]
}
```

/api/messages (View):

localhost:3000/api/messages

Pretty-print ☑

```json
{
  "count": 3,
  "data": [
    {
      "id": 1,
      "name": "Juan D.",
      "email": "juan@example.com",
      "message": "How much for extra sago?",
      "dateSent": "2025-09-01"
    },
    {
      "id": 2,
      "name": "Maria S.",
      "email": "maria@example.com",
      "message": "Do you accept large orders?",
      "dateSent": "2025-09-05"
    },
    {
      "id": 3,
      "name": "Ana L.",
      "email": "ana@example.com",
      "message": "Where is your stall located?",
      "dateSent": "2025-09-20"
    }
  ]
}
```

/health (View):


```
localhost:3000/health
Pretty-print ✓

{
  "status": "OK",
  "time": "2025-10-08T05:28:35.851Z"
}
```