Group Member:

Salsabila Herlin Syahputri Lubis (001202400209)

 Diah Safitri Pane (001202400208)

Chensie Naila Angganici Kalalo (001202400203)

Food Delivery Website System Report

1. Overview

This food delivery website is designed to provide a streamlined platform for users to browse, order, and receive meals from partnered restaurants. The system also supports restaurant management, delivery coordination, and payment processing through a well-structured database and user-friendly interface.

The application flow guides users through the following key forms and processes:

- **Role Selection:** Before logging in, the user is presented with an option to choose their role: **Customer**, **Restaurant Owner**, or **Driver**. This choice directs them to the appropriate login or signup flow and subsequent interface.

Log in to the GoFood Clone

Email

budi.s@example.com

Password

••••••••••••••

Login

Forgot password?

Don't have an account? **Register now**

Log in as Restaurant Admin

Email
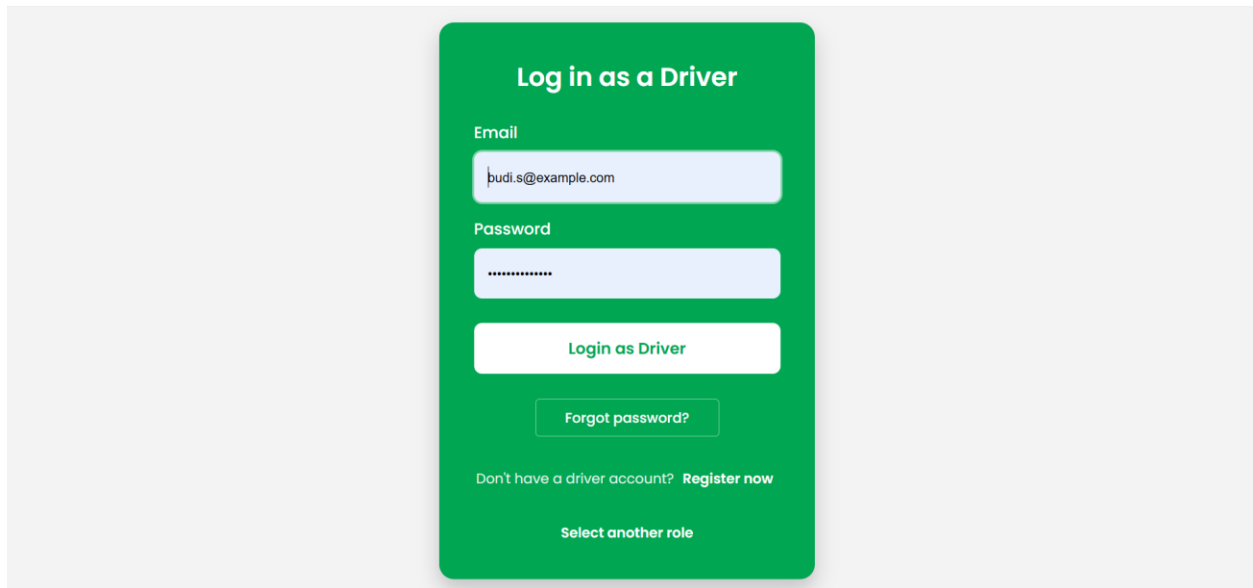
budi.s@example.com
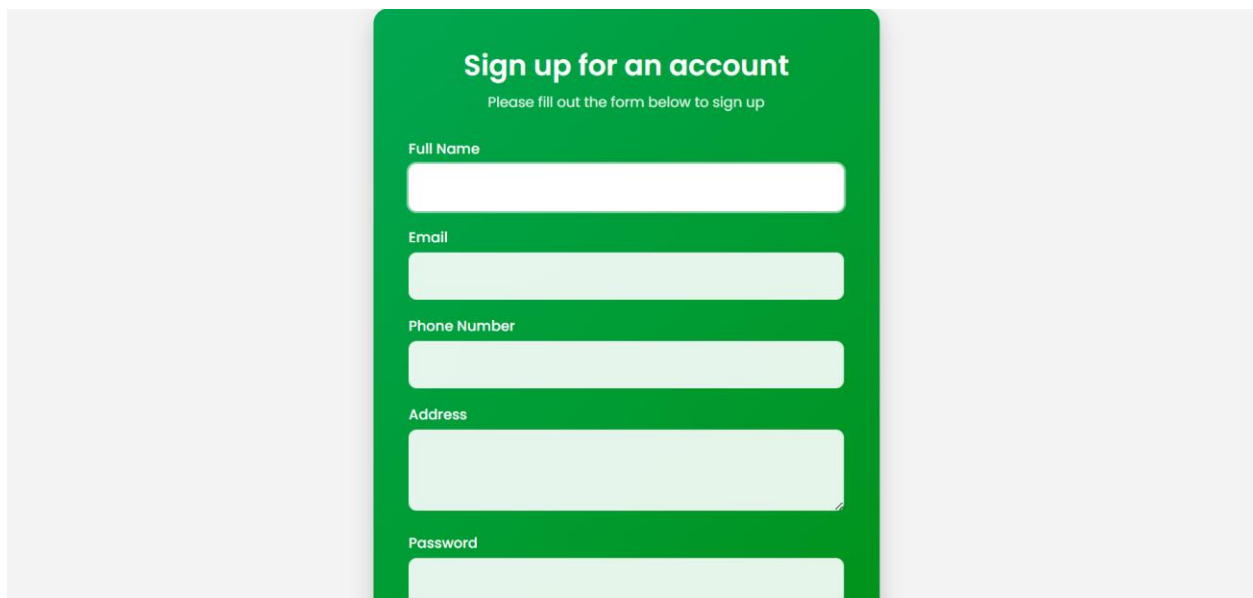
Password

••••••••••••••

Login as Restaurant

Forgot password?

Don't have a restaurant account?
Register now

Select another role

- **(Login Site):** Allows existing users (Customer, Restaurant Owner, Driver, Admin) to log in with their email and password after selecting their role.

- **(Sign Up Site):** Enables new users (primarily Customers, potentially others via specific flows) to create an account by providing their name, email, phone number, address, and password. The role selected initially will determine which type of account is created.

- **(Forgot Password):** Enables customers to change or reset the password if they forget their password.
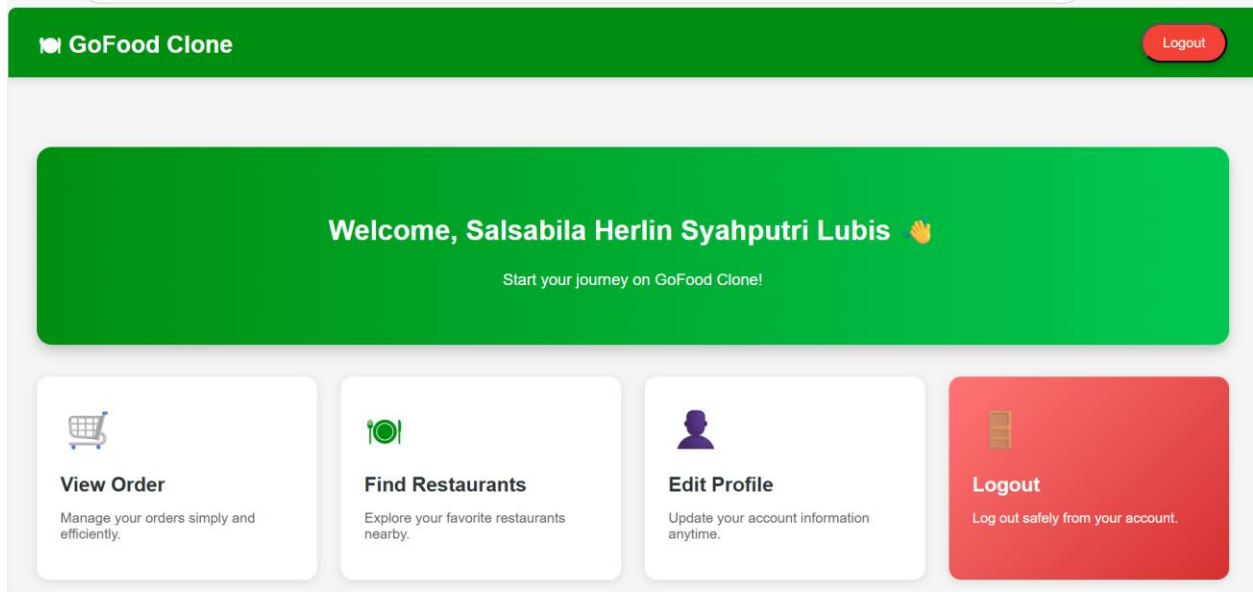
1.1 Customer dashboard
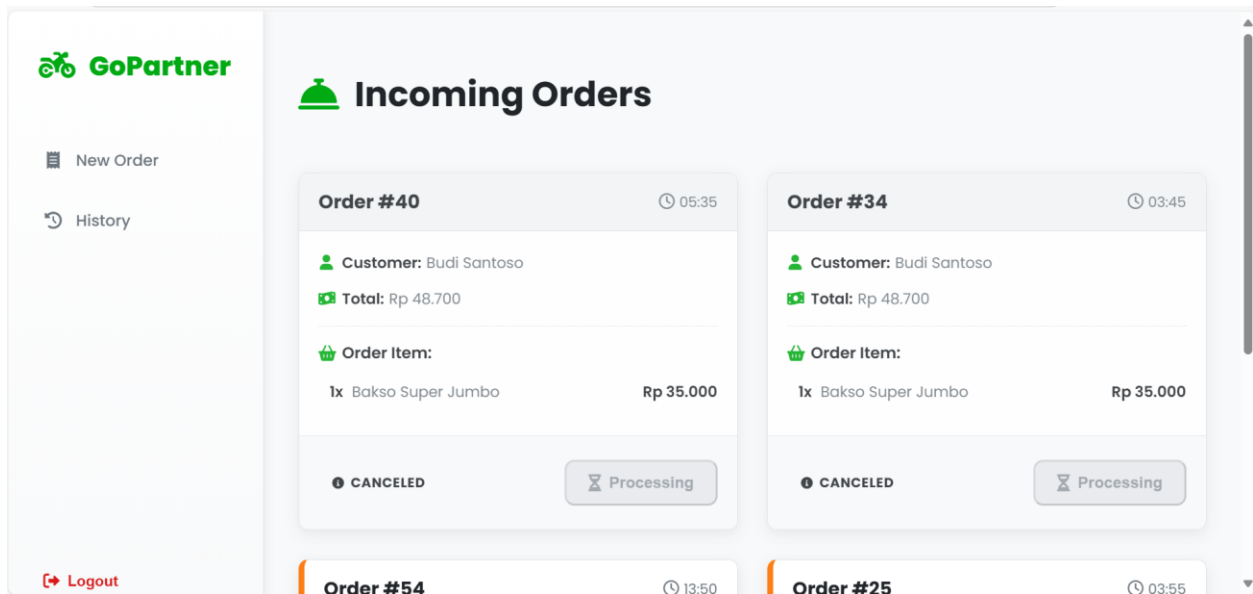
1.2 Restaurant dashboard
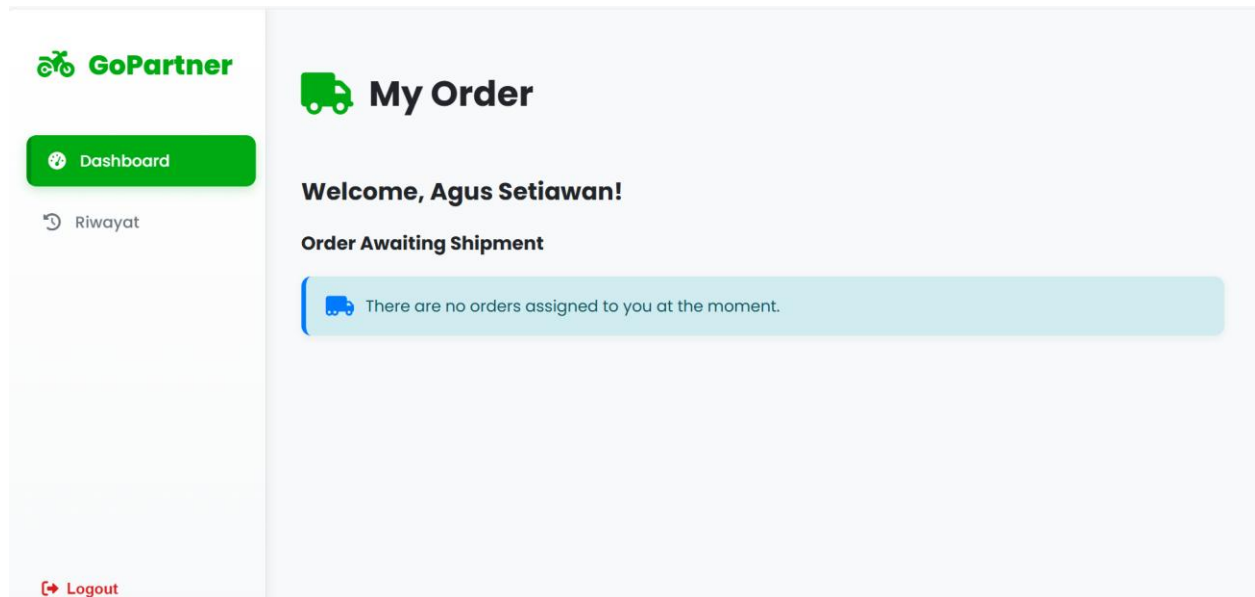
1.3 Driver Dashboard

- **(Home Page or Dashboard):** The main landing page after login, providing role-specific access. Customers can explore restaurants, view orders, and manage profiles. Restaurant Admins and Drivers would access their respective dashboards here.
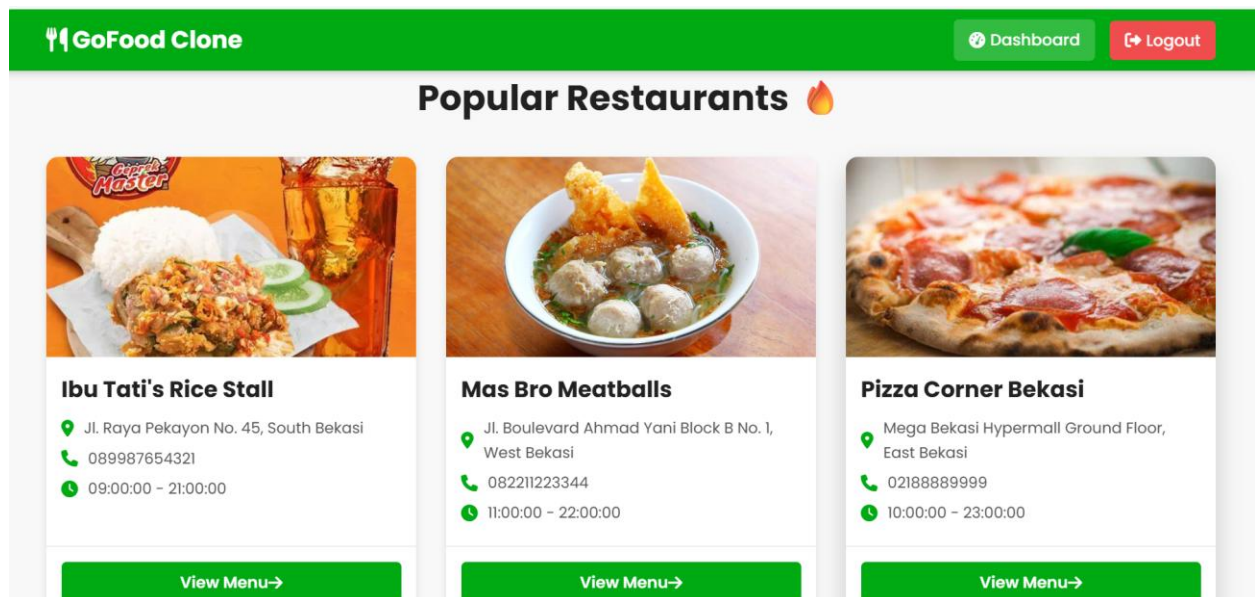
- **(View Restaurant):** Allows customers to browse available restaurants, search.

Mas Bro Meatballs

Address: Jl. Boulevard Ahmad Yani Block B No. 1, West Bekasi

Telephone Number: 082211223344

R Rudi Setiawan 3 ★ ★ ★ ☆ ☆

The tendon meatballs are a bit tough.

Purchased in 2025-04-15

Review made in 2025

B Budi Santoso 5 ★ ★ ★ ★ ★

The food was good. Highly recommended

Purchased in 2025-04-26

Review made in 2025

- **(View Review):** Displays the review and rating from other customers

4.0 ★ ★ ★ ★ ☆ View review 🍱 Delivery: Arrive in 25-35 minutes (1.05 km) Super Partner

🍴 Menu

Bakso Super Jumbo

Large beef meatballs filled with minced meat

Rp 35.000

[-] 1 [+]

Mie Ayam Pangsit

Chicken noodles with boiled and fried dumplings

Rp 28.000

[-] 1 [+]

2 item
Rp 63.000
Mas Bro Meatballs 🛒

- **(Menu Option):** Displays the menu of a selected restaurant for customers to browse and select items.

Payment summary

Price	Rp 63.000
Delivery and Processing Fees	13.700
Total payment	**Rp 76.700**

Delivery

○ Express 25 min	Rp 15.700
◉ Reguler 29-39 min	Rp 13.700
○ Ekonomis 39-49 min	Rp 5.500

- **(Payment and Delivery Option):** Allows customers to choose delivery options and proceed to payment.

Payment Method

- ◉ GoPay
- ○ Cash
- ○ Credit Card

Delivery Address

Default address
JL.Movie land

Alternative address

Buy and deliver now

- **(Payment Method and delivery address):** Facilitates the selection of a payment method, delivery address and order confirmation by the customer.
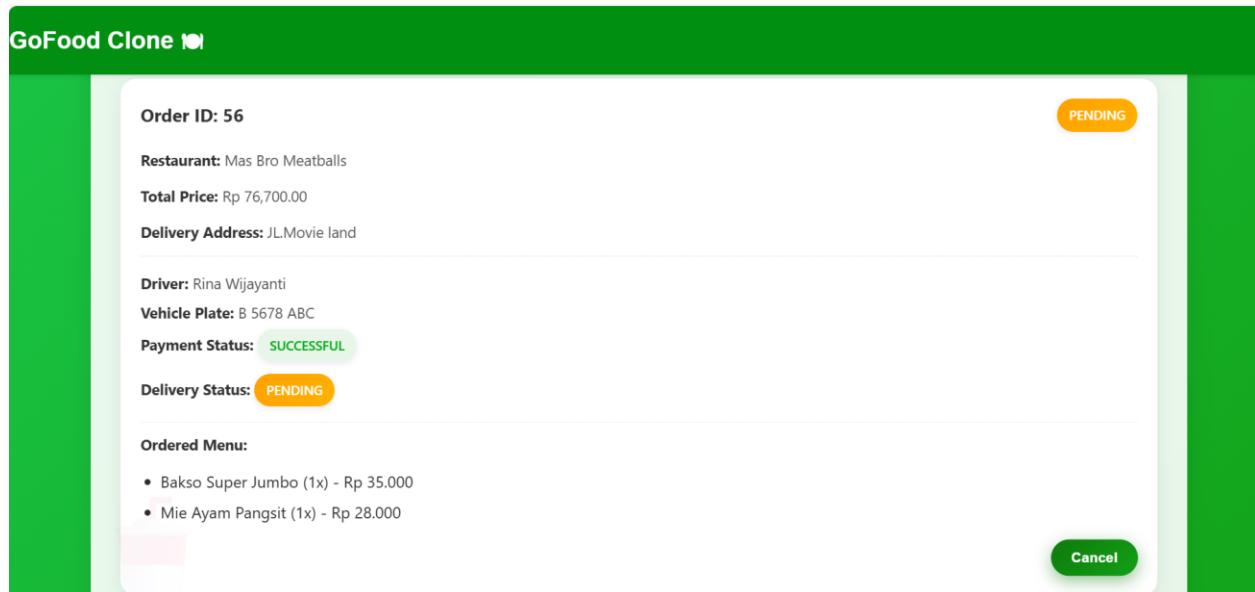
- **Active Order:** The customer can track their active order and see the driver's information such as name and vehicle plate and also the customer can cancel the order if the status is still pending.
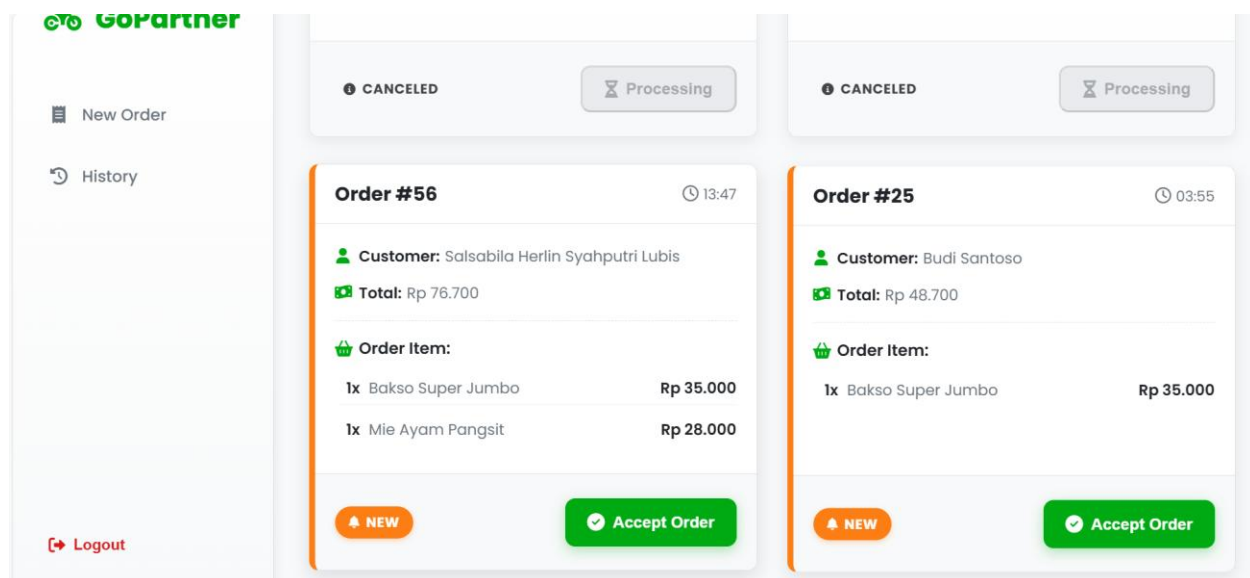
- **Order Processing (Restaurant Admin):** After a customer places an order, the **Restaurant Admin** receives the order notification. They can click the Accept Order button to confirm and process the order. The status of the order changes from **Pending** to **Processing** or **In Process**.
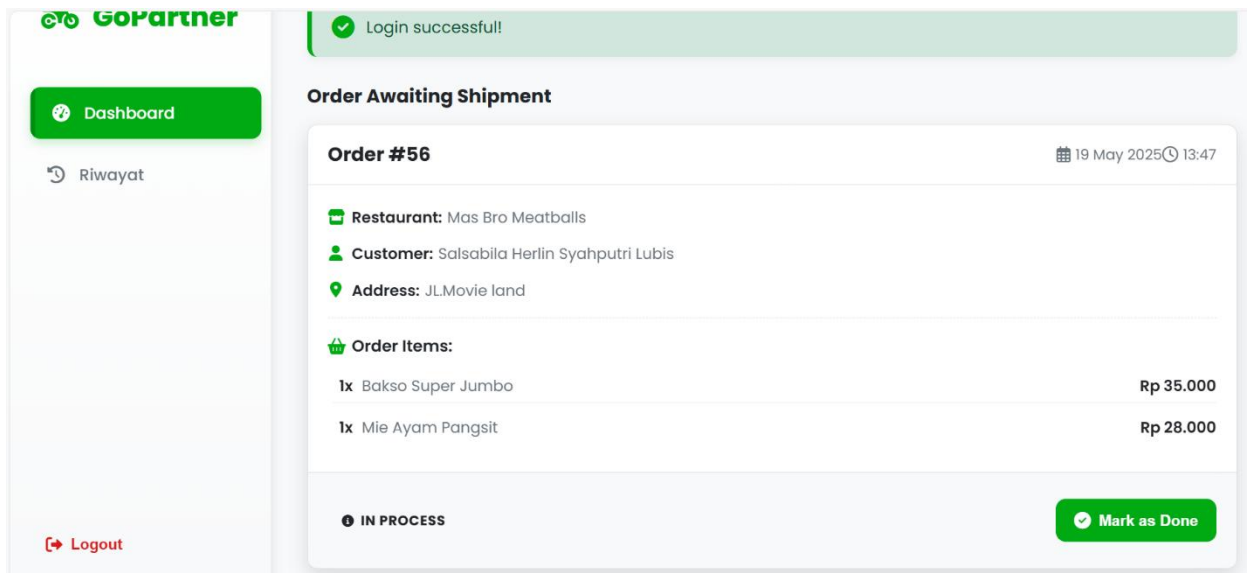
- **Order Completion (Driver):** Once the food is prepared ,the **Driver** transports the food to the customer's specified address. After delivering the food to the customer, the **Driver** uses their interface (web view) to **click a button like "Mark as Done"**. This action updates the status of the order to **Completed** or **Delivered**.
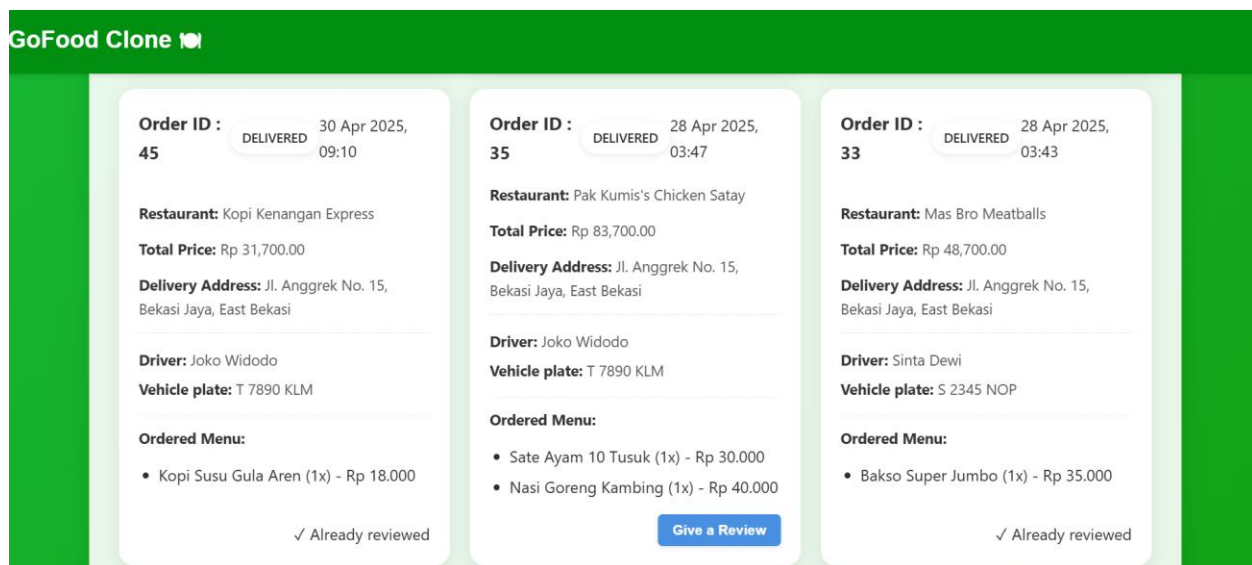
- **(View Order History):** Customers can access this page to see the history order.
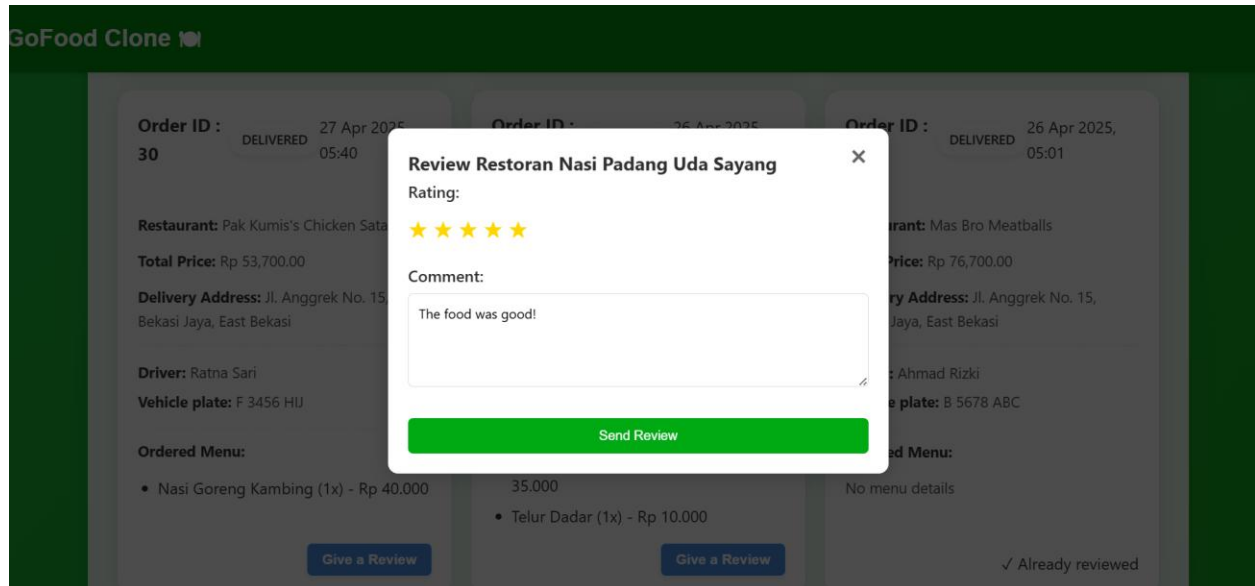
- **(Restaurant Review)**: Enables customers to submit ratings and reviews after completing an order (once the status is **Completed** or **Delivered**).

- **(Edit Password), (Profile Information), (Change Password), (Delete Account):** Customers can manage and update their profile details, password,or delete their account.

2. Business Functions

The system supports the following key business functions accessible to users:

- **User Registration & Login:** Create an account and sign in securely for various roles.
- **Browse Restaurants & Menus:** View available restaurants and their menu items

(Customer).
- **Place Orders:** Select food items, customize orders, and confirm checkout (Customer).
- **Manage Orders:** Accept, process, and update order status (Restaurant Admin). View and manage assigned deliveries (Driver).
- **Track Orders:** Monitor order status (preparing, dispatched, delivered) (Customer, Driver, Admin).
- **Make Payments:** Pay via Gopay, e-wallet, or other supported methods (Customer).
- **Delivery option:** Choose between Express, Regular, and economic delivery types (Customer).
- **Rate & Review:** Submit feedback on orders and restaurant service (Customer).
- **Order History:** View the past orders (Customer).

3. Data Requirements

The system requires the following data entities, stored in the database tables:

- User Account Information (email, username, full name, phone number, password, and role)
- Restaurant Information (email, password, restaurant name, description, location, phone number, operating hours, owner ID, logo/image, and role)
- Driver Information(name,vehicle plate,password,email, and role)
- Menu Item Information (item name, description, price, category, restaurant ID)
- Order Information (order ID, customer ID, restaurant ID, order status, order time, delivery address, delivery_driver_id, delivdery_time, delivery_costa, and total amount)
- Order Item Details (order ID, menu item ID, quantity, item price at time of order)
- Payment Information (payment method, status, linked order ID)
- Delivery Information (order ID, delivery personnel ID, delivery time, delivery status)
- Review and Rating Information (review text, rating value, order ID, customer ID, restaurant ID, review date/time)

4. Business Rules

The application operates under the following business rules:

- **User Roles & Access:**
  - A user must register with a valid email and password to access system features.

- A user can only have one role per account: Customer, Restaurant Owner, Delivery Personnel(driver).
  - Restaurant Owners can only manage their own restaurant's data and orders.
  - Delivery Personnel can only view and update the status of orders assigned to them.
- **Menu Management:**
  - Menu items must be linked to a specific restaurant.
  - Only available menu items can be ordered.
  - Prices must be greater than zero and cannot be modified on active orders.
- **Order Placement:**
  - Customers can only place orders from restaurants that are currently open.
  - An order must contain at least one menu item.
  - Once an order is placed, it cannot be edited
  - Customers can cancel the order before the order is confirmed by restaurant admin.
- **Payment Handling:**
  - Orders must be paid before being marked as ready for dispatch, unless cash-on-delivery is selected.
  - Each payment transaction must be recorded with a unique reference number.
  - Failed payments must not update the order status.
- **Delivery Operations:**
  - Each order must be assigned to one delivery personnel(driver).
  - Delivery personnel can only view orders assigned to them.
  - Order status must be updated in real time (e.g.,Delivered).
- **Reviews and Ratings:**
  - Only customers who have completed an order can submit a review for that order.
  - Each order can only be reviewed once.
  - Reviews must be linked to both the customer and the restaurant.

5. CRUD Processes Implementation

This section outlines where Create, Read, Update, and Delete (CRUD) operations are performed within the application flow, with a focus on user, restaurant admin, and driver account management, as well as core order processing.

5.1 Create (C) Operations

- **User Registration (Customer, Restaurant Admin, Driver):**

- **Process:** A new user (customer, or potentially restaurant admin/driver via a specific sign-up flow or admin creation) completes **(Sign up site)** or an equivalent interface after selecting their role.
- **Database Action:** Inserts a new row into the User table with provided name, email, phone, address, password, and the appropriate role ('user' for customer, 'restaurant owner', 'driver'). For restaurant owners, a corresponding entry in the Restaurant table would also be created or linked. For drivers, an entry in the Driver table would be created or linked.
- **Business Rule Applied:** "A user must register with a valid email and password to access system features. "A user can only have one role per account"
- **Order Placement:**
  - **Process:** A customer selects menu items (**Menu Option**), chooses delivery (**Payment and Delivery Option**), and confirms payment on **(Payment Method)**.
  - **Database Action:** Inserts a new row into the Orders table and multiple rows into the OrderDetail table.
  - **Business Rules Applied:** "Customers can only place orders from restaurants that are currently open.", "An order must contain at least one menu item.", "Orders must be paid before being marked as ready for dispatch."
- **Restaurant/Menu Item Creation:**
  - **Process:** (Performed by Restaurant Owners/Admins via a backend interface).
  - **Database Action:** Inserts new rows into the Restaurant and Menu tables.
- **Review Submission:**
  - **Process:** User submits feedback on **(Restaurant Review)** after order completion.
  - **Database Action:** Inserts a new row into the RestaurantReview table.
  - **Business Rules Applied:** "Only customers who have completed an order can submit a review", "Each order can only be reviewed once."

5.2 Read (R) Operations

- **User Login:**
  - **Process:** User attempts login on **(Login Site)** after selecting their role.
  - **Database Action:** Reads user credentials (email, password, role) from the User table for verification and role-based access.
- **Browsing Restaurants:**
  - **Process:** User accesses **(View Restaurant)**.
  - **Database Action:** Reads data from the Restaurant table.

- **Viewing Menus:**
  - **Process:** User accesses **(Menu Option)** for a selected restaurant.
  - **Database Action:** Reads data from the Menu table, filtered by restaurant_id.
- **Viewing Order Status & History:**
  - **Process:** User accesses **(View order)**. Restaurant Admins and Drivers access their respective dashboards.
  - **Database Action:** Reads data from the Orders and OrderDetail tables, filtered by user_id (for customers), restaurant_id (for restaurant owners), or delivery_driver_id (for drivers).
- **Viewing Profile Information (User):**
  - **Process:** User accesses **(Profile Information)** or a role-specific profile page.
  - **Database Action:** Reads user data (name, email, phone, address, role) from the User table, filtered by user_id. Restaurant owners might also read related Restaurant data, and drivers related Driver data.
- **Displaying Reviews:**
  - **Process:** (When viewing restaurant details).
  - **Database Action:** Reads data from the RestaurantReview table, linked to User and Restaurant.

5.3 Update (U) Operations

- **User Profile Information Update (Customer):**
  - **Process:** User updates details (like name, email, phone, address) on **Form (Profile Information)** or a role-specific profile page.
  - **Database Action:** Updates existing rows in the User table for the specific user_id.
- **Password Change (Customer, Restaurant Admin, Driver):**
  - **Process:** User updates their password on **(Edit Password)** or **(Change Password)** or a role-specific password change interface.
  - **Database Action:** Updates the password column in the User table for the specific user_id with a new hashed password.
- **Order Status Update (Restaurant Admin, Driver):**
  - **Process:** Restaurant Admin accepts/processes order. Once the order is completed Driver marks order as delivered.
  - **Database Action:** Updates the status (e.g., to 'Processing', 'Delivered'), delivery_status (e.g., 'Delivered'), delivery_time, delivery_driver_id columns in the Orders table.
  - **Business Rule Applied:** "Order status must be updated in real time."

5.4 Delete (D) Operations

- **Account Deactivation/Deletion (Customer):**
  - **Process:** (initiated by the user via a profile setting).
- **Database Action:** The **Delete Account** feature allows a registered user to permanently remove their account from the food delivery website.
- **Order Cancellation:**
  - **Process:** User cancels an order (usually via **View order**).
  - **Database Action:** Updates the status column in the Orders table to 'Canceled' (logical delete).
  - **Business Rule Applied:** "only canceled before preparation begins."

6. Database Table Structure

The system utilizes the following database tables:

User Table

Column Name	Data Type	Constraints	Description
user_id	INT(11)	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for each user.
name	VARCHAR(100)	NOT NULL	Full name of the user.
email	VARCHAR(100)	NOT NULL	User's email address (for login).
phone	VARCHAR(15)	NOT NULL	User's phone number.
address	TEXT	NOT NULL	User's primary address.
password	VARCHAR(255)	NOT NULL	Hashed password.
role	VARCHAR(255)	NOT NULL, DEFAULT 'user'	User's role (customer, owner, driver, admin).

Orders Table

Column Name	Data Type	Constraints	Description
order_id	INT(11)	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for each order.
user_id	INT(11)	NOT NULL, FOREIGN KEY REFERENCES User(user_id)	Customer who placed the order.
restaurant_id	INT(11)	NOT NULL, FOREIGN KEY REFERENCES Restaurant(restaurant_id)	Restaurant the order is from.
order_time	TIMESTAMP	NOT NULL, DEFAULT CURRENT_TIMESTAMP	Timestamp when the order was placed.
status	VARCHAR(50)	NOT NULL	Current status of the order.
payment_method	VARCHAR(50)	NOT NULL	Method used for payment.
payment_status	VARCHAR(50)	NOT NULL	Status of the payment.
payment_total	DECIMAL(10,2)	NOT NULL	Total amount of the order.
delivery_driver_id	INT(11)	NOT NULL, FOREIGN KEY REFERENCES Driver(driver_id)	Driver assigned to the order.
delivery_time	DATETIME	DEFAULT NULL	Actual time of delivery.
delivery_type	VARCHAR(50)	NOT NULL	Type of delivery (Express, Regular, Affordable).
delivery_status	VARCHAR(50)	DEFAULT NULL	Current status of the delivery.
delivery_cost	DECIMAL(10,2)	NOT NULL	Cost of the delivery.

OrderDetail Table

Column Name	Data Type	Constraints	Description
order_detail_id	INT(11)	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for each order item entry.
order_id	INT(11)	NOT NULL, FOREIGN KEY REFERENCES Orders(order_id)	Links to the main order.
menu_id	INT(11)	NOT NULL, FOREIGN KEY REFERENCES Menu(menu_id)	The specific menu item ordered.
quantity	INT(11)	NOT NULL	Number of units of the menu item.
price	DECIMAL(10,2)	NOT NULL	Price of a single unit at the time of order.

Restaurant Table

Column Name	Data Type	Constraints	Description
restaurant_id	INT(11)	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for each restaurant.
name	VARCHAR(100)	NOT NULL	Name of the restaurant.
address	TEXT	NOT NULL	Physical address of the restaurant.
phone	VARCHAR(15)	NOT NULL	Contact phone number for the restaurant.
open_time	TIME	NOT NULL	Daily opening time.
close_time	TIME	NOT NULL	Daily closing time.
role	VARCHAR(255)	NOT NULL, DEFAULT 'restaurant admin'	Role associated with the restaurant (Consider linking to User.user_id).

Menu Table

Column Name	Data Type	Constraints	Description
menu_id	INT(11)	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for each menu item.
restaurant_id	INT(11)	NOT NULL, FOREIGN KEY REFERENCES Restaurant(restaura	Links the menu item to its restaurant.
Category_id	INT(11)	NOT NULL	Identifier for the menu item's category.
Category_name	VARCHAR(100)	NOT NULL	Name of the food category.
name	VARCHAR(100)	NOT NULL	Name of the menu item.
description	TEXT	DEFAULT NULL	A brief description of the item.
price	DECIMAL(10,2)	DEFAULT NULL	Price of the menu item.
Image_url	VARCHAR(512)	DEFAULT NULL	URL to the image of the menu item.

Driver Table

Column Name	Data Type	Constraints	Description
Driver_id	INT(11)	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for each driver.
name	VARCHAR(100)	NOT NULL	Driver's full name.
phone	VARCHAR(15)	NOT NULL,	Driver's contact

			phone number.
Vehicle_plate	VARCHAR(15)	NOT NULL	Vehicle license plate number.
role	VARCHAR(255)	NOT NULL, DEFAULT 'driver'	Role associated with the driver

RestaurantReview Table

Column Name	Data Type	Constraints	Description
Review_id	INT(11)	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for each review.
User_id	INT(11)	NOT NULL, FOREIGN KEY REFERENCES User(user_id)	The customer who submitted the review.
Restaurant_id	INT(11)	NOT NULL, FOREIGN KEY REFERENCES Restaurant(restaurant_id)	The restaurant being reviewed.
comment	TEXT	DEFAULT NULL	Textual feedback or comment.
Review_date	DATE	NOT NULL, DEFAULT CURRENT_DATE	Date when the review was submitted.
rating	INT(11)	DEFAULT NULL, CHECK (rating BETWEEN 1 AND 5)	Numerical rating (1 to 5 stars).

| | | | |
| | | | |