Kizuno is a home-cooked meals delivery service that connects stay-at-home mothers and home cooks with people who want healthy, homemade food. Cooks post their menus the day before, and customers can order meals that are delivered to their workplace or study place the next day.

Business Model:

How It Works

01 Platform Overview

A digital marketplace where home cooks (stay-at-home parents, homemakers, etc.) post their daily menus.

Customers can browse through various home-cooked meal options and place pre-orders

02 Pre-Order System

Cooks post their menus a day in advance, specifying what meals will be available for the next day.

Customers have until a set cut-off time to place orders.

Meals are prepared fresh in the morning and delivered by lunchtime to customers' work or study locations.

03 Revenue Model

Commission-based: The platform earns a percentage of each transaction between cooks and customers.

Delivery fee: An additional fee for delivering the meals to customers.

Subscription plans (future feature): Regular customers can subscribe to meal plans for discounted pricing and priority ordering.

04 Target Audience

Primary Users: Students and professionals who seek healthy, home-cooked alternatives to restaurant or fast food.

Suppliers: Stay-at-home mothers, homemakers, and passionate cooks with a desire to earn income while working from home.

05 Logistics

Cooks prepare the meals in their homes.

Consumers can chose either delivery or pick-up based on the cook’s convenience

A network of delivery partners ensures the meals reach customers during lunch hours

To create the website for Kizuno, you can follow these main steps and pages to ensure the functionality aligns with the business model you’ve outlined:

### Steps for Creating the Website:

1. \*\*Plan the Structure\*\*:

- Identify key features based on the business model: menu posting by cooks, pre-orders by customers, delivery logistics, user accounts, and payment integration.

- Choose tech stack: HTML, CSS, PHP, SQL for backend, and possibly JavaScript for front-end interactions.

2. \*\*Design Wireframes/Mockups\*\*:

- Sketch the main pages (homepage, user dashboard, order flow, etc.) to visualize the layout.

3. \*\*Set Up Your Database\*\*:

- Create tables for users (cooks, customers), orders, menus, and transactions in SQL.

4. \*\*Develop the Frontend\*\*:

- Use HTML and CSS for layout and design.

- Implement JavaScript for dynamic content updates (e.g., updating menus, displaying order status).

5. \*\*Backend Development\*\*:

- Use PHP to handle user registrations, login, menu posting, order placement, and payments.

- Connect the website to the SQL database to fetch/store data dynamically.

6. \*\*Test Functionality\*\*:

- Test all features like user registration, placing orders, viewing menus, payment, and delivery tracking.

7. \*\*Deploy the Website\*\*:

- Deploy it to a server or hosting platform for live access.

---

### Pages to Include:

1. \*\*Homepage\*\*:

- Introduction to Kizuno.

- How it works (Platform Overview, Pre-Order System).

- Links to login/sign-up.

- Featured meals of the day (dynamic section showing available menus).

2. \*\*Sign Up/Login Page\*\*:

- Separate forms for customers and cooks.

- Create user roles: Cooks (to post menus) and Customers (to order).

3. \*\*Cook Dashboard\*\*:

- Menu posting interface: Cooks can upload their daily menu (with descriptions, price, and availability).

- Order management: Track orders placed by customers.

4. \*\*Customer Dashboard\*\*:

- Menu browsing page: List of available meals posted by cooks.

- Search filters (e.g., meal type, cook’s location, dietary preferences).

- Pre-order interface: Select meals, add to cart, and choose delivery or pick-up.

- Order tracking: View order status (processing, on the way, delivered).

5. \*\*Checkout & Payment Page\*\*:

- Customers can confirm their order, choose delivery or pick-up.

- Payment gateway integration for secure transactions.

- Confirmation email after the payment.

6. \*\*Order Summary/Tracking Page\*\*:

- Shows order details, expected delivery time, and delivery partner info (if applicable).

7. \*\*About Us Page\*\*:

- Information about the Kizuno platform and its mission to provide healthy, home-cooked meals.

8. \*\*Contact Us Page\*\*:

- Contact form and customer support details.

---

### Additional Features:

- \*\*User Profiles\*\*: Cooks and customers can view/edit their profiles.

- \*\*Reviews and Ratings\*\*: Customers can leave reviews for meals/cooks.

- \*\*Delivery Integration\*\*: Show delivery status in real-time (if using a delivery network).

- \*\*Subscription Plans (Future)\*\*: Develop a system for customers to subscribe to weekly/monthly meal plans.

---

By following this guide, you can structure and build a website for Kizuno that covers the essential aspects of your business model.

CREATE TABLE users (

user\_id INT AUTO\_INCREMENT PRIMARY KEY,

full\_name VARCHAR(100) NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

phone\_number VARCHAR(15),

password\_hash VARCHAR(255) NOT NULL,

user\_type ENUM('cook', 'customer') NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE cooks (

cook\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

bio TEXT,

specialty VARCHAR(255),

rating DECIMAL(3, 2) DEFAULT 0, -- Average rating, like 4.5 out of 5

total\_reviews INT DEFAULT 0,

FOREIGN KEY (user\_id) REFERENCES users(user\_id) ON DELETE CASCADE

);

CREATE TABLE customers (

customer\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

address VARCHAR(255),

FOREIGN KEY (user\_id) REFERENCES users(user\_id) ON DELETE CASCADE

);

CREATE TABLE meals (

meal\_id INT AUTO\_INCREMENT PRIMARY KEY,

cook\_id INT,

name VARCHAR(255) NOT NULL,

description TEXT,

price DECIMAL(10, 2) NOT NULL,

available\_date DATE NOT NULL,

image\_url VARCHAR(255), -- URL to the image of the meal

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (cook\_id) REFERENCES cooks(cook\_id) ON DELETE CASCADE

);

CREATE TABLE orders (

order\_id INT AUTO\_INCREMENT PRIMARY KEY,

customer\_id INT,

total\_amount DECIMAL(10, 2) NOT NULL,

order\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

delivery\_address VARCHAR(255),

status ENUM('pending', 'delivered', 'cancelled') DEFAULT 'pending',

FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id) ON DELETE CASCADE

);

CREATE TABLE order\_items (

order\_item\_id INT AUTO\_INCREMENT PRIMARY KEY,

order\_id INT,

meal\_id INT,

quantity INT DEFAULT 1,

price DECIMAL(10, 2) NOT NULL, -- Price at the time of order

FOREIGN KEY (order\_id) REFERENCES orders(order\_id) ON DELETE CASCADE,

FOREIGN KEY (meal\_id) REFERENCES meals(meal\_id) ON DELETE CASCADE

);

CREATE TABLE reviews (

review\_id INT AUTO\_INCREMENT PRIMARY KEY,

customer\_id INT,

meal\_id INT,

rating INT CHECK (rating BETWEEN 1 AND 5), -- Rating between 1 and 5

comment TEXT,

review\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id) ON DELETE CASCADE,

FOREIGN KEY (meal\_id) REFERENCES meals(meal\_id) ON DELETE CASCADE

);

CREATE TABLE subscriptions (

subscription\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

subscription\_plan VARCHAR(255),

start\_date DATE,

end\_date DATE,

status ENUM('active', 'expired', 'cancelled') DEFAULT 'active',

FOREIGN KEY (user\_id) REFERENCES users(user\_id) ON DELETE CASCADE

);

CREATE TABLE payments (

payment\_id INT AUTO\_INCREMENT PRIMARY KEY,

order\_id INT,

payment\_method ENUM('credit\_card', 'paypal', 'cash\_on\_delivery'),

payment\_status ENUM('completed', 'pending', 'failed') DEFAULT 'pending',

payment\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (order\_id) REFERENCES orders(order\_id) ON DELETE CASCADE

);