

Assignment: Presenting Website Recommendations

Josh Pahang 000762072



March 3, 2024

Cprg 219 A

# Project Background and Description

## Background:

My client has requested me to create a successful restaurant website capable of viewing information about the restaurant and capable of ordering food and the ability to have it delivered with a newly developed delivery system.

They have read my website audit and would like to work with me and go forward with the project. My client would love to enhance their customer experience by providing customers with an intuitive and seamless platform for both in-store and delivery orders. Customers will have access to this where they have easy access to their menu, order items with many customizable options, pay with a variety of payment options, and track all their orders live, all while maintaining flexibility and scalability in the future.

## Problem:

My client runs a restaurant and does not currently have an online ordering and delivery service available. Customers can only order in-person and by phone, which may lead to missed opportunities, lower customer engagement, and an overall lower reputation by not keeping up with modern trends.

The goal of my client is to provide a system that offers customers the convenience of ordering online, fully customizing their orders, tracking their deliveries live, viewing their past orders, managing their account profiles, and overall staying engaging with a loyalty program.

## Client Goals:

The following is a list of features required to achieve a successful restaurant delivery service application.

|  |  |
| --- | --- |
| * Home Menu | * Food Menu Page |
| * Address/Location Entry | * Order Pick-Up and Take-Out Page |
| * Search Functionality | * Item Cart |
| * Payment Window | * Order Confirmation and Tracking |
| * Customer Profile and Account Management | * Notification and Alert System |
| * Review and Feedback System | * Promotions and Rewards System |
| * Customer Support | * Order Customization |
| * Delivery Zone Management | * Reorder Past Orders |

# Solutions

## Possible Solutions and Steps

This section will give a brief overview of the solutions presented. The steps needed are also present and the priority of the step is also listed at the end in [square brackets].

### No-Code CMS

This option requires the use of an existing content management system (CMS) with eCommerce capabilities. Using a no-code CMS, like WordPress, will be ideal for quick development and easy to use for the client’s development team. Furthermore, with plugins like WooCommerce and WPForms, setting up pages for ordering and menu management, will be extremely useful.

### Steps Needed:

1. Brainstorm a suitable theme that reflects the core values of the client’s company. [optional]
2. Using any CMS platform (like WordPress), create a responsive website that is customized to the website’s branding. [high priority]
3. Using available plugins that are connected to the CMS, enable your website to have more added functionality. [high priority]
   1. These added functionalities include the ability to turn the website into an eCommerce website. [high priority]
   2. As well as easily creating quick drag-and-drop form page for a faster and more efficient build. [high priority]
4. Make sure the website has a clear entry point to enter in one’s address for delivery purposes. [medium priority]
5. Create an order page where we will be able to order items off the restaurant’s menu. [high priority]
6. Enable the website so the customer can customize any preferred ingredients of the items the customer would like to order. [medium priority]
7. Make sure the desired items are being tracked in the client’s cart. [high priority]
8. Integrate the payment gateways for the website, such as PayPal, Stripe, or Square, so the customer can pay for their order. [high priority]
9. Integrate a confirmation page/window so the customer can track the status of the order. [medium priority]
10. Allow the customer to view the location of a delivery driver. [medium priority]
11. Create a feedback page which shows the customer any potential loyalty points gained, allows the customer to rate their experience, and/or view a detailed list of their current or past order(s). [low priority]
12. Integrate testing of the site for speed and scalability. [medium priority]

### Custom Code CMS

This option requires the use of an existing content management system (CMS) that is built with popular web development frameworks, such as React and Node.js. A custom code CMS allows for more flexibility, scalability, and control. This would allow for a more personalized user experience.

### Steps Needed:

1. Brainstorm a suitable theme that reflects the core values of the client’s company. [optional]
2. Develop the front-end of the website using frameworks like React. [high priority]
3. Build the back end of the website where the data collected is stored in a database. [high priority]

\*The remaining steps are similar to the previous steps:

1. Make sure the website has a clear entry point to enter in one’s address for delivery purposes. [medium priority]
2. Create an order page where we will be able to order items off the restaurant’s menu. [high priority]
3. Enable the website so the customer can customize any preferred ingredients of the items the customer would like to order. [medium priority]
4. Make sure the desired items are being tracked in the client’s cart. [high priority]
5. Integrate the payment gateways for the website, such as PayPal, Stripe, or Square, so the customer can pay for their order. [high priority]
6. Integrate a confirmation page/window so the customer can track the status of the order. [high priority]
7. Allow the customer to view the location of a delivery driver. [medium priority]
   1. Using API’s can allow for the process to be integrated much easier.
8. Create a feedback page which shows the customer any potential loyalty points gained, allows the customer to rate their experience, and/or view a detailed list of their current or past order(s). [medium priority]
9. Integrate testing of the site for speed and scalability. [medium priority]

### Third Party Services

This option requires the use of an existing delivery service (DoorDash/UberEATS) to handle the delivery logistics. Integrating these services will allows the restaurant to focus more on preparing the food, rather than the delivery aspect.

### Steps Needed:

\*Like the previous solution, some steps may overlap with past steps.

1. Brainstorm a suitable theme that reflects the core values of the client’s company. [optional]
2. Develop the front-end of the website using either a No-Code CMS, or a Custom Code CMS. [high priority]
3. Build the back end of the website where the data collected is stored in a database, if using a Custom Code CMS. [high priority]
4. Create a page where it will redirect the user to the third-party’s integrated service. [high priority]
5. Make sure the website has a clear entry point to enter in one’s address for delivery purposes. [medium priority]
6. Ensure that the integration has fast and seamless order flow between the restaurant and the service. [medium priority]
7. Integrate testing of the site for speed and scalability. [medium priority]

# Limitations

This section will go over the limitations of the previous 3 solutions.

1. **No-Code CMS**
   1. Limited Customization:
      1. The custom features like real-time order tracking, custom search functionality, and any complex user customization would require custom development.
   2. Scalability:
      1. If web traffic increases during peak hours or overall increases overtime, there may be performance issues that are present.
2. **Custom Code CMS**
   1. Costly and resourceful:
      1. Building a website from scratch requires a lot of time and developer resources. This can lead to long development cycles which can cost a lot of money and add up over time.
   2. Maintenance:
      1. Since this website has been built from scratch, there is a need for frequent maintenance, making this option more complex and time-consuming.
3. **Third-Party Integration**
   1. Limited Control:
      1. By using a third-party service, the client will have limited control over the delivery service and would have to abide with the policies of the integrated service.
   2. Fees:
      1. Similar to the previous limitation, these services may require a fee where they take a portion of the profits from the delivery and transaction.

**Alternative & Additional Services**

* Integrating scalable database services like AWS and Firebase will allow the website to handle more transactions, ensuring seamless quick transactions through multiple users.
* Instead of relying on the third-party services to handle the delivery service and take away potential profits, using custom-built tracking systems, like the Google Maps API, would provide my client with more control and flexibility with functions like ‘Real-time order tracking’.

# Features Options and Priority List

## Priority List of Features

## High Priority

The high priority list are must-haves and are fully required in order to have a basic functioning website with a ordering and delivery system attached.

1. Home Menu
   1. Essential for customers to view information about the store, operating hours, contact info, and social media.
2. Food Menu Page
   1. Essential for customers to browse food options and filter by dietary preference.
3. Address/Location Entry
   1. Critical entry field for customers for accurate delivery.
4. Order Pick-Up and Take-Out Page
   1. Essential function for making an order.
5. Order Customization
   1. This is necessary as it allows customers to add/remove toppings and adjust their order to their preference.
6. Item Cart
   1. Essential for managing, customizing, and viewing orders and the costs.
7. Payment Window
   1. A secure and necessary step when the customer pays.
8. Order Confirmation and Tracking
   1. Allows customers to track their delivery in real-time.

## Medium Priority

A medium priority list greatly helps with user experience and although not fully required for a functioning website, these functions are basically required for a successful website in modern times.

1. Search Functionality
   1. Helps customers quickly find specific items or services.
2. Customer Profile and Account Management
   1. Enables customer interaction and allows them to save preferences.
3. Notification and Alert System
   1. Useful for keeping customers informed about their order.
4. Review and Feedback System
   1. Useful for the restaurant and other customers.
5. Promotions and Rewards Program
   1. Encourages customer loyalty through discounts and rewards.
6. Customer Support
   1. A critical feature for solving issues.

## Low Priority

These options are more quality-of-life functions that would be nice to have.

1. Delivery Zone Management
   1. Necessary for operational purposes, but the entire application can run without it.
2. Reorder Past Orders
   1. Useful for customers who order the same thing but not useful for first-time users.

## Examples:

A screenshot of a website

AI-generated content may be incorrect.Feature of ‘Address/Location Entry’. Screenshot taken from DoorDash’s website for the restaurant ‘Kinjo Sushi & Grill’

A screenshot of a food menu

AI-generated content may be incorrect.

Using Third-Party Services (DoorDash, UberEATS) on the home page. Screenshot taken from <https://redheadscafe.com/>

A screenshot of a phone

AI-generated content may be incorrect.

Feature of ‘Order Tracking’. Example taken from a screenshot of an UberEATS order.

# Time Estimates

\*These estimates are based on using a Custom Code CMS as solution.

1. Initial UI/UX Design & Wireframing ~ 20 hours
   1. Designing the layout, creating user flows, and creating wireframes for main web pages (Home, Food Menu Page, Cart, Payment, etc.).
2. Database & Backend Setup ~ 25-30 hours
   1. Setting up a secure database and backend that handles and stores orders, customer account information, and handles the payment process.
   2. Also integrated to handle user authentication.
3. Frontend Development ~ 40-50 hours
   1. Using frameworks like React, building the UI and functionality that revolves around the home page, food menu page, cart, order customization, etc.
4. Integrating a Payment Gateway ~ 10-15 hours
   1. Integrating PayPal, Stripe, or Square to handle secure payments of multiple variations.
5. Testing and Debugging ~ 20-30 hours
   1. All these features must be testing to ensure proper working order, whether that is payment processing, order tracking, and responsiveness of the UI on different sized devices.
6. Launch and Post-Launch Support ~ 10 hours
   1. Deploying the website to live servers and providing users with support during the initial launch period.
   2. Data and bug collecting may happen so future development can help optimize its performance.

Overall integrating this system will take around 3-4 weeks to fully implement. To account for any unseen circumstances, it will be best to have a budget that compensates for 5 weeks’ worth of work.