Daimonji Sushi Restaurant

# Purpose and Objectives

The purpose of this website is to provide customers with a view of Daimonji sushi restaurant show casing some of the menu items including appetizers, main courses, drinks, and desserts.

# Frontend Design

Domain Name: [www.daimonjisushiseattle.com](http://www.daimonjisushiseattle.com) (not yet acquired)

## Audience and flow:

Currently there is no viable website for the restaurant Daimonji Sushi. They have been around for 10 years and currently the restaurant utilizes physical menus at their restaurant. Searching online for the restaurant, there is currently a yelp site and Google reviews. Customers will upload the physical menu on these review sites, but currently no up to date site exists.

As a user, customers will be able to view the website domain and check some of the various items that are listed on the menu including the appetizers, main courses, drinks, and dessert menu. Users will be able to search for the website via their favorite search engine. If a customer is at the restaurant, a QR code will be displayed on the table which will allow them to access the website via their phones.

## Sketch and Page Layout

The layout and plan is currently designed using the Figma app. A basic layout is implement show case both the PC/ desktop view as well as the mobile version if a user were to be viewing the website on their smartphone.

[Sushi Restaurant – Figma](https://www.figma.com/design/vsmRnNMIIx4EFmOy3eaNRW/Sushi-Restaurant?node-id=0-1&node-type=canvas&t=lSJ32B8yFc1rK4dj-0)

# User stories

As a user, I want to be able to view the latest menu items from Daimonji sushi restaurant on my phone or computer

As a user, I want to be able to filter the menu items based off of whether they are cooked or raw (stretch goal)

As a user, I want to be able to filter out sashimi items based off fish type (stretch goal)

As an admin, I want to be able to see the current inventory of my products and supplies

# Front End Color Scheme

Main page will have contrast colors of

* #FCFAEE
* #B8001F

Will also be implementing MaterialsUI for some components for the header/ footer.

Example of desktop version of site: [Sushi Restaurant – Figma](https://www.figma.com/design/vsmRnNMIIx4EFmOy3eaNRW/Sushi-Restaurant?node-id=0-1&node-type=canvas&t=UEF46CO6LqnoyiT4-0)

A screenshot of a computer screen

Description automatically generated

# Accessibility Features

In order to provide appropriate accessibility for all users, the website will include some of the following (in no particular order) ([Website Accessibility Checklist (15 Best Practices) | WebsiteSetup](https://websitesetup.org/web-accessibility-checklist/))

1. Appropriate color contrast
2. Implement a “light” and “dark mode” button
3. Images will have alternative text
4. Responsive design using larger buttons and easily visible links
5. Adding the focus css properties for buttons and labels to improve visual indicators

# Mobility Features

Currently, the app is web-based only and no android/ iOS features will be implemented. Responsive design will be utilized so that the webapp will be easily available and viewable on any hand-held device.

Example of responsive design: [Sushi Restaurant – Figma](https://www.figma.com/design/vsmRnNMIIx4EFmOy3eaNRW/Sushi-Restaurant?node-id=13-212&node-type=canvas&t=UEF46CO6LqnoyiT4-0)

A screenshot of a computer

Description automatically generated

# BackEnd

The backend for this system will utilize MongoDB for its database structure and table creation. The backend will primarily be used to allow the website owner to check basic inventory of ingredients that the restaurant currently has (or does not have). The backend system can also be used to determine if an item on the menu can be available/ unavailable as a result of the appropriate ingredients which can display to the user that the certain item is unavailable at this time.

Next.js will be used as both the front end and backend frameworks for this website. Next.js can be used to implement a simple REST API feature that can keep track of the ingredients / items on the menu.

Example of filestructure:

A screenshot of a computer

Description automatically generated

# Security Features

Aside from the security features that can be implemented from using a shared hosting site, this site will also include implementing features such as: ([An Overview of Best Practices for Security Headers | Okta Developer](https://developer.okta.com/blog/2021/10/18/security-headers-best-practices#x-xss-protection))

* Security headers, which ultimately secures the website and prevents the site from vulnerabilities that may harm the user.
  + Content-security- policy: tells the browser what items to load for the user
  + Permissions-Policy: to enable/ disable unwanted features and only require what is necessary
* Error Handling: Utilize appropriate error handling features to prevent unwanted information from being sent to the user incase of error in the code/ development.

# Hosting

Currently will utilize some sort of shared hosting (i.e., Bluehost, DreamHost, HostGator) as this would be the cheapest way to implement this website. With shared hosting, some of the benefits includes: ([What Is Shared Hosting and How Does It Work +Pros & Cons (hostinger.com)](https://www.hostinger.com/tutorials/what-is-shared-hosting#Features)

* Affordable
* Security features
  + Firewalls
  + Regular security updates
  + Malware scanning
  + SSL certificates