FoodTuck Website Documentation

Overview

This documentation provides a detailed guide for the development and functionality of the FoodTuck website. The website is designed to showcase a food truck business, offering information about menu items, chef details, special offers, and online ordering capabilities.

Project Structure

The project uses Next.js for the frontend framework, Tailwind CSS for styling, and Sanity CMS for content management. The backend API handles product data and integrates seamlessly with the frontend.

Folder Structure

- src
  - app
    - components
      - Header.tsx
      - Footer.tsx
      - Menu.tsx
      - ChefSection.tsx
      - SpecialOffers.tsx
    - pages
      - index.tsx
      - menu.tsx
      - about.tsx
      - contact.tsx
    - styles
      - globals.css
  - sanity

- schemas
  - food.ts
  - chef.ts
  - offer.ts
- api
  - products
    - [id].ts
    - index.ts

Features

1. Home Page

- A visually appealing hero section with a background image of the food truck.
- Highlighted menu categories and special offers.
- Testimonials from customers.

2. Menu Page

- Dynamic rendering of food items using Sanity CMS data.
- Filters for categories such as appetizers, main courses, desserts, and beverages.
- Detailed information for each menu item, including:
  - Name
  - Description
  - Price
  - Image

3. About Page

- Information about the food truck's history, mission, and values.
- Details about the chefs, sourced from the chef.ts schema in Sanity CMS.

4. Contact Page

- Contact form for customer inquiries.
- Integration with Google Maps to display the food truck's location.

5. Special Offers Section

- Displays time-sensitive deals fetched from the offer.ts schema.
- Includes a countdown timer for limited-time offers.

6. API Integration

- **Endpoints for fetching food items, chefs, and offers.**
- **Dynamic routing for individual product details using [id].ts.**

Sanity CMS Schemas

food.ts

```
{
 name: 'food',
 title: 'Food',
 type: 'document',
 fields: [
  { name: 'name', type: 'string', title: 'Name' },
  { name: 'category', type: 'string', title: 'Category' },
  { name: 'price', type: 'number', title: 'Price' },
  { name: 'originalPrice', type: 'number', title: 'Original Price' },
  { name: 'tags', type: 'array', title: 'Tags', of: [{ type: 'string' }] },
  { name: 'image', type: 'image', title: 'Image' },
  { name: 'description', type: 'text', title: 'Description' },
  { name: 'isAvailable', type: 'boolean', title: 'Availability' }
 ]
}
```

chef.ts

```
{
 name: 'chef',
 title: 'Chef',
 type: 'document',
 fields: [
```

```
    { name: 'name', type: 'string', title: 'Name' },
    { name: 'specialty', type: 'string', title: 'Specialty' },
    { name: 'image', type: 'image', title: 'Image' },
    { name: 'bio', type: 'text', title: 'Biography' }
  ]
}
```

offer.ts

```
{
  name: 'offer',
  title: 'Offer',
  type: 'document',
  fields: [
    { name: 'title', type: 'string', title: 'Title' },
    { name: 'description', type: 'text', title: 'Description' },
    { name: 'discount', type: 'number', title: 'Discount (%)' },
    { name: 'expiryDate', type: 'datetime', title: 'Expiry Date' }
  ]
}
```

API Endpoints

Fetch All Products

Endpoint: /api/products Method: GET Response:

```
[
  {
    "id": "1",
    "name": "Cheeseburger",
```

```
  "price": 8.99,
  "category": "Main Course",
  "image": "image_url",
  "description": "A juicy cheeseburger with fresh lettuce and tomato."
 }
]
```


Fetch Product by ID

Endpoint: /api/products/[id] Method: GET Response:


```
{
 "id": "1",
 "name": "Cheeseburger",
 "price": 8.99,
 "category": "Main Course",
 "image": "image_url",
 "description": "A juicy cheeseburger with fresh lettuce and tomato."
}
```


Styling

- **Global Styling: Tailwind CSS utility classes for responsive design.**
- **Custom CSS: Specific styling for hero sections, menu items, and contact forms.**
- **Fonts and Colors:**
  - **Font: 'Times New Roman' for text.**
  - **Colors: Soft blues and whites with subtle shadows for a clean look.**

Deployment

- **Platform: Vercel**
- **Steps:**
  1. **Connect the GitHub repository to Vercel.**
  2. **Deploy the project.**
  3. **Set environment variables for Sanity CMS and API endpoints.**

Future Enhancements

- **Add user authentication for online ordering.**
- **Enable payment gateway integration.**
- **Add a blog section for recipes and food stories.**

Umama Rajput