DAY: 3

API INTEGRATION AND DATA MIGRATION IN PROJECT

Project Overview

The Restaurant Website Project is a comprehensive platform developed using **Next.js** and **Sanity CMS** to provide an engaging and user-friendly experience. This website is designed to showcase the restaurant's food menu, chef profiles, and other essential information. The integration of dynamic data through external APIs and manually curated datasets ensures that the content remains fresh and easily manageable.

Objectives

- Develop a responsive and interactive website for a restaurant.
- Display categorized food items with images, descriptions, and pricing details.
- Highlight chef profiles with their experience, specialty, and availability.
- Integrate dynamic data management using Sanity CMS and Next.js API routes.

Technologies Used

- **Next.js:** A React-based framework for building the frontend.

- **Sanity CMS:** A headless content management system for managing dynamic content.
- **Tailwind CSS:** A utility-first CSS framework for responsive and modern UI design.
- **REST API:** For seamless data integration between the CMS and frontend.

Key Features

1. Food Menu Display

- A visually appealing showcase of food items categorized into **Drinks**, **Snacks**, **Main Course**, **Appetizers**, and **Desserts**.
- Detailed product descriptions with images, current and original pricing, discount offers, and availability status.

2. Chef Profiles

- Comprehensive profiles of chefs displaying their **name**, **position**, **years of experience**, **specialty**, and **availability**.
- High-resolution images and engaging biographies to introduce the chefs.

3. Dynamic Data Management

- Easy content updates through Sanity CMS without needing to modify the code.
- API routes in Next.js fetch and display data dynamically, ensuring the latest content is always live.

4. Responsive and Accessible Design

- Fully optimized design for desktops, tablets, and mobile devices.
- Intuitive navigation and user-friendly interface for all users.

API Integration Process

1. Understanding the API

- Reviewed API documentation to understand key endpoints like /products, /categories, and /orders.
- Identified how data aligns with the Sanity CMS schema.

2. Schema Adjustments

- Compared Sanity CMS schema with API data fields.
- Adjusted fields for compatibility, e.g., mapping product_title to name.

3. Data Migration Steps

- **Manual Import:** Used JSON format to manually upload small datasets via Sanity's import tools.
- **API Script:** Developed scripts in Next.js API routes to fetch and transform external data.
- **Validation:** Ensured data consistency through validation checks.

4. API Integration in Next.js

- Created utility functions for data fetching from Sanity CMS and external APIs.
- Rendered fetched data into components dynamically.
- Tested endpoints using **Postman** and browser developer tools.

Project Workflow

1. Frontend Development

- Built responsive UI components using Next.js and styled with Tailwind CSS.
- Designed dynamic components to display real-time data.

2. Backend Integration

- Developed Next.js API routes for fetching data.
- Integrated Sanity CMS for real-time content updates.

3. Testing and Quality Assurance

- Validated API endpoints with **Postman**.
- Tested responsiveness across various devices and browsers.

4. Deployment

- Deployed the website on **Vercel** for live access and performance optimization.

Challenges Faced

- Mapping Sanity CMS schema to match external API data.
- Handling API errors and ensuring reliable data fetching.
- Maintaining design consistency across different devices.

Achievements

- Successfully integrated APIs and CMS for dynamic content management.
- Developed a scalable, user-friendly website.
- Simplified content updates with Sanity CMS.

Future Enhancements

- Add user authentication for personalized experiences.
- Implement online ordering and reservation functionality.
- Optimize performance for high-traffic handling.

Conclusion

This project demonstrates the successful use of **Next.js** and **Sanity CMS** to create a dynamic, responsive restaurant website. Seamless API integration and user-friendly design ensure an engaging and easily manageable platform.

Prepared by: Nibras Nadeem
Project: Restaurant Website
Technologies: Next.js, Sanity CMS, Tailwind CSS