Technical Documentation for Q-Commerce Website (Restaurant-Based)

Over view

This document outlines the architecture, technologies, and workflows used in developing a Q-commerce restaurant website.

Technology Stack

Frontend

Framework: Next.js

Hosting: Vercel

Content Management: Sanity CMS (dynamic restaurant menus, promotions, etc.)

UI/UX: Responsive design with Tailwind CSS and React components.

Backend

APIs:

Built with Node.js and Express.js.

Hosted on AWS.

Authentication:

MongoDB for secure user data storage.

Token-based authentication (JWT).

Order Management:

APIs for managing and tracking orders.

Integration with shipment tracking services.

Database

Type: MongoDB

Usage:

Storing user credentials.

Saving restaurant and menu details.

Logging orders and shipment statuses.

Payment Gateway

Service: Stripe

Features:

Secure payment processing.

Support for multiple currencies.

Real-time payment status updates.

Hosting and Deployment

Frontend: Deployed on Vercel for fast and seamless hosting.

Backend: Deployed on AWS for scalable and secure server management.

Key Features

User Authentication:

Secure signup and login.

Password reset functionality.

Restaurant Listing:

Dynamic content from Sanity CMS.

Categorized menus with search and filter options.

Order Tracking:

Real-time shipment tracking.

Integration with delivery services.

Payment Processing:

Integrated with Stripe for safe and quick transactions.

Admin Dashboard:

Manage restaurant listings.

Track orders and update statuses.

Scalability:

Optimized for high traffic with efficient database queries and CDN support.

Workflows

Data Flow:

User Interaction:

User accesses the website on a browser.

Frontend (Next.js) fetches content dynamically from Sanity CMS.

Authentication:

User credentials are securely stored in MongoDB.

JWT tokens are issued for session management.

Order Placement:

User selects menu items and proceeds to checkout.

Backend APIs handle order data and communicate with Stripe for payment.

Order Tracking:

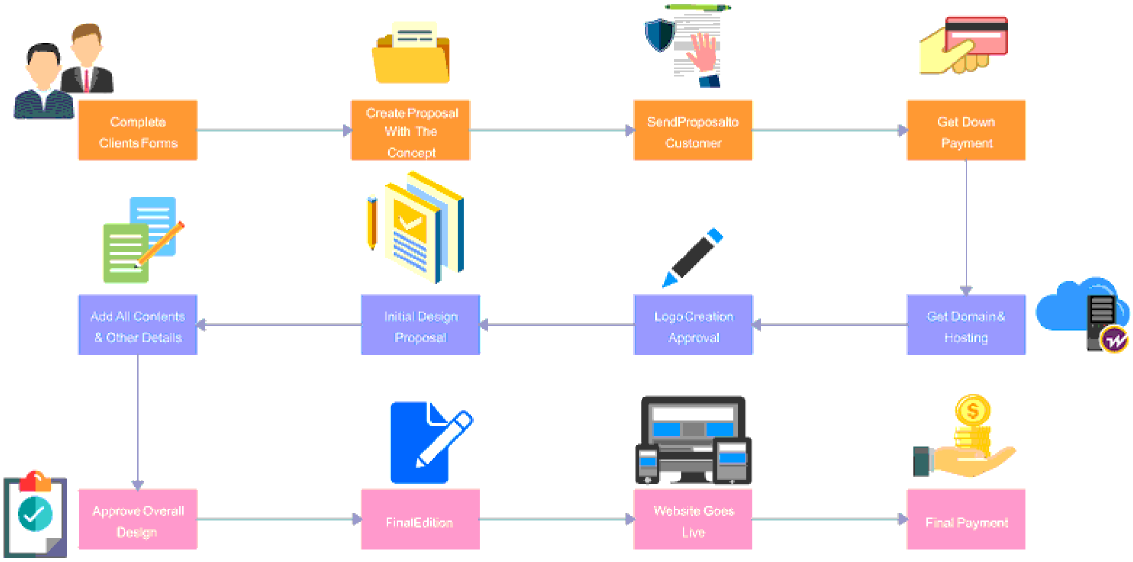
After payment confirmation, the order status is updated in MongoDB.

Shipment tracking is displayed using integrated APIs.

Deployment:

Frontend is continuously deployed on Vercel.

Backend scales on AWS with serverless architecture.


Complete Clients Forms
Create Proposal With The Concept
SendProposalto Customer
Get Down Payment
Add All Contents & Other Details
Initial Design Proposal
Logo Creation Approval
Get Domain & Hosting
Approve Overall Design
FinalEdition
Website Goes Live
Final Payment