Project Documentation – Akasa Food Order Platform

1. Project Overview

Domain: Full Stack Web Development
Tech Stack: React, TypeScript, Tailwind CSS, Node.js, Express, MongoDB, Brevo API

Description:
Akasa Food Order is a full-stack food ordering platform designed to provide users with a seamless and modern experience in browsing, selecting, and purchasing food items online. The platform offers real-time cart management, secure user authentication, order tracking, and automated email confirmations through the Brevo transactional email service.

2. Objectives

- To design and implement a responsive and interactive food ordering system.
- To integrate secure user authentication with JWT tokens.
- To enable users to browse items, filter by categories, and manage their cart dynamically.
- To build a scalable backend with RESTful APIs connected to MongoDB.
- To implement automated email notifications for password resets and order confirmations.

3. System Architecture

Frontend (React + TypeScript + Tailwind CSS)

Backend (Node.js + Express + JWT)

Database (MongoDB Atlas – Cloud)

Email Service (Brevo API)

Frontend:

- Built using **React (Vite + TypeScript)** for a fast, modular, and maintainable UI.
- Styled using **Tailwind CSS** with responsive and dark-mode designs.

Backend:

- Developed using **Node.js and Express.js** with a RESTful API structure.
- Secured with **JWT-based authentication** for protected routes.

Database:

- Cloud-hosted **MongoDB Atlas** used for storing user, product, and order data.

Email Service:

- **Brevo API (Transactional Email)** used for sending password reset and order confirmation emails securely over HTTPS.

4. Functional Requirements

User Authentication

- Register with email and password.
- Login using valid credentials.
- Password reset via secure tokenized email link.
- JWT-based authentication for session management.

Food Menu Management

- Browse available food items categorized as Fruits, Vegetables, Breads, Non-Veg, Dairy, etc.
- Search and filter food items dynamically.
- Product images fetched dynamically from Unsplash API.

Cart and Order Management

- Add, remove, and update items in the cart.
- View cart summary with total price.
- Proceed to checkout with real-time stock validation.
- Create new orders and view order history with timestamps.
- Each order assigned a unique tracking ID.

Email Functionality (Brevo)

- Send password reset emails with secure token URLs.

- Send order confirmation emails with invoice summaries.
- All email operations handled via Brevo HTTPS API (for Render compatibility).

5. Non-Functional Requirements

- **Security:**
  - Passwords encrypted with bcrypt.
  - JWT for secure route access.
  - Helmet + Rate Limiter for API protection.
- **Scalability:**
  - Modular architecture with separate routes and controllers.
  - API-first approach allows easy frontend or mobile app integration.
- **Performance:**
  - Optimized React rendering with lazy loading.
  - Backend API caching (optional future enhancement).
- **Reliability:**
  - MongoDB Atlas for cloud-hosted resilience.
  - Graceful error handling across API endpoints.

6. System Modules

Module	Description
Authentication	Handles signup, login, JWT generation, and password reset.
Menu Module	Displays available items by category, with images and descriptions.
Cart Module	Manages add-to-cart, quantity updates, and cart persistence per user.
Order Module	Processes orders, generates tracking IDs, and updates inventory.
Email Service	Uses Brevo API to send confirmation and password reset emails.
Admin (Future Scope)	Add/edit/delete food items, monitor orders, and manage users.

7. Tools & Technologies Used

Category	Tools / Libraries
Frontend	React, Vite, TypeScript, Tailwind CSS
Backend	Node.js, Express.js, JWT, bcrypt
Database	MongoDB Atlas (Cloud NoSQL Database)
Email Service	Brevo Transactional Email API
Deployment	Render (Backend), Vercel (Frontend)
Version Control	Git + GitHub
Others	Helmet, CORS, Express Rate Limiter

8. Deployment Details

Frontend:

- Hosted on **Vercel**
- Accessible at: https://akasa-food-order.vercel.app *(replace with your actual link)*

Backend:

- Hosted on **Render**
- API Base URL: https://akasa-backend.onrender.com/api

Database:

- MongoDB Atlas Cluster (M0 tier) connected via MONGO_URI environment variable.

Environment Variables (Backend)

PORT=10000

MONGO_URI=<your-atlas-uri>

JWT_SECRET=<secure-secret>

EMAIL_SERVICE=brevo

EMAIL_USER=<verified-sender@domain.com>

BREVO_API_KEY=<your-brevo-api-key>

FRONTEND_URL=https://akasa-food-order.vercel.app

9. Current Status

Feature	Status
Frontend UI	Completed
Backend API	Completed
MongoDB Integration	Completed
Authentication (JWT)	Completed
Email (Password Reset + Order Confirmation)	Pending API key fix on Render
Deployment	Done (Frontend + Backend)

10. Future Enhancements

- Admin dashboard for menu and order management.
- Integration of real payment gateway (Razorpay or Stripe).
- Push notifications for order updates.
- Live order tracking using WebSockets.
- AI-based food recommendations.

11. Conclusion

Akasa Food Order successfully demonstrates a **complete full-stack web application** with secure authentication, modular API design, and robust frontend integration. The project highlights best practices in modern web development — from JWT security to cloud deployment — and stands as a production-ready prototype for scalable food delivery systems.