Resteau – Full System Architecture (Frontend + Backend + Firebase)

A modern digital restaurant system with QR menus, real-time orders, and a staff dashboard.

🚀 Updated Feature Architecture

✅ Frontend (Next.js 15)

- Customer-facing menu UI
- Shopping cart
- Order placement
- Call-waiter feature
- Staff dashboard with real-time updates
- QR generator UI

✅ Backend (Node.js / Next.js API Routes OR Express Server)

You have **two options** for backend:

Option A — Next.js Server Actions + API Routes (Recommended)

- Built-in backend
- Fast, no need for extra server
- Works well with Firebase
- Good for small/medium restaurants

Option B — Dedicated Node/Express Backend

- Separate Node.js server
- Useful if you want:
    - POS system integration
    - Complex analytics
    - Microservices
    - Future mobile apps

Either way, the backend handles:

- Orders

- Menu management
- Table sessions
- Waiter calls
- Staff authentication

🔥 **Firebase Integration**

Firebase will give you all missing backend functionalities:

1. Firebase Authentication

- Staff login (admin, waiters, cooks)
- Optional customer session login
- Secure access to /staff dashboard

2. Firebase Firestore

Use Firestore collections:

menus/

categories/

orders/

orderId/

tables/

waiterCalls/

staff/

3. Firebase Storage

- Store images for menu items
  (food pictures, category icons)

4. Firebase Cloud Messaging

- Real-time notifications to staff when:
    - New order placed
    - Waiter button pressed
    - Order canceled

5. Firebase Hosting (Optional)

- Deploy frontend for free

Real-time updates come built-in through Firestore's onSnapshot().

🧩 **Updated Project Structure (Including Backend + Firebase)**

```
src/
├── app/
│   ├── menu/         # Customer-facing menu
│   ├── staff/        # Staff dashboard
│   ├── qr-generator/  # QR codes
│   ├── api/          # Server actions + backend API
│   ├── layout.tsx
│   └── page.tsx
├── lib/
│   ├── firebase.ts    # Firebase client init
│   ├── admin.ts       # Firebase Admin SDK (protected routes)
│   └── utils.ts       # Server & helper functions
├── components/
├── data/
│   └── menu.ts        # Optional local menu seed
├── types/
│   └── index.ts
└── styles/
    └── globals.css
```

Backend routes example:

/api/orders

/api/waiter-call

/api/menu

/api/auth (optional)

/api/staff

🔧 Backend Responsibilities

Below is what the backend will handle:

Orders

- Create order
- Update status: pending → preparing → ready → served
- Cancel orders
- Track timestamps and table numbers

Menu

- Staff can create / update / delete items
- Fetch menu by category
- Manage availability

Table Sessions

- Detect which table scanned the QR
- Assign orders to a session

Waiter Calls

- Customer presses “Call Waiter”
- Staff receives instant signal
- Staff can mark it as resolved

Admin Panel

- Add staff
- Role-based access control

🌐 Deployment Setup

Frontend

- Deploy on Vercel
  or Firebase Hosting
  or Netlify

Backend

Option A (Recommended): API Routes inside Next.js → automatically deployed
Option B: Deploy Node.js server to:

- Render
- Railway
- Heroku
- Firebase Functions

Database

- Firestore (real-time)

Storage

- Firebase Storage

Authentication

- Firebase Auth

💡 Updated Plan Summary

Component	Technology	Purpose
Frontend	Next.js 15, TS, Tailwind	UI, QR, menu, dashboard
Backend	Next.js API routes OR Express	Orders, menu, waiter calls
Database	Firebase Firestore	Real-time system
Auth	Firebase Authentication	Staff login
Storage	Firebase Storage	Menu photos
Notifications	Firebase Cloud Messaging	Instant waiter & kitchen alerts
Deployment	Vercel + Firebase	Production-ready

🛠️ What You Should Build First

To avoid complexity, follow this order:

Phase 1 — Core UI

- Menu UI
- Cart
- Staff dashboard (static)

Phase 2 — Firebase Integration

- Auth
- Firestore setup
- Orders collection
- Real-time listeners

Phase 3 — Backend Logic

- Order creation/update routes
- Waiter call system
- QR table session logic

Phase 4 — QR Generator

- Generate URL + table number
- Print/download

Phase 5 — Deployment

- Vercel + Firebase