Technical Design Document: Mombasa Vibe Hotel v2.0

1. System Overview

The website is a Single Page Application (SPA) built with a "Static-First" philosophy. It uses a decoupled data layer (JSON) to allow for rapid rebranding across different restaurant clients in the Mombasa region.

- Architecture: Jamstack (JavaScript, API-less, Markup)
- Hosting: Netlify / Vercel (Global Edge Network)
- Integration: WhatsApp Business URL Scheme (The "Pseudo-Backend")

---

2. Technical Stack

Layer	Technology	Rationale
Frontend	Vanilla JS (ES6+)	Eliminates framework overhead (React/Vue) for <1s TTI.
Styling	Tailwind CSS	Utility-first; purged for minimal CSS payload.
Data	JSON (Flat-file)	Local, version-controlled, no SQL database required.
Media	WebP / AVIF	60-80% smaller than JPEG; critical for 4G performance.
Icons	Lucide-SFC / SVGs	Resolution independent; zero HTTP requests.

---

3. Data Schema & Logic Flow

3.1 The "Liquid Menu" Engine

The site uses an asynchronous `fetch()` to pull from `menu.json`. This allows you to update prices for a client by editing a single file without rebuilding the entire UI structure.

JavaScript

```
// Logic Fragment: Menu Renderer
async function loadMenu() {
  const response = await fetch('./data/menu.json');
  const data = await response.json();
  renderCategories(data.categories);
}
```

3.2 The WhatsApp Booking Bridge

Instead of a database, user data is serialized into a URI-encoded string. This offloads the "Order Management" to the restaurant's existing WhatsApp workflow.

Serialization Formula:

$$URL_{final} = base\_url + phone + encode(Template + UserData)$$

---

4. Performance Optimization (The "Safaricom 4G" Protocol)

To meet the < 3s load time constraint in Mombasa's network environment, the following optimizations are enforced:

- Critical CSS: Inlined in the `<head>` to prevent Flash of Unstyled Content (FOUC).
- Image Lazy Loading: Native `loading="lazy"` on all menu items.
- Zero External Dependencies: No jQuery, no heavy Google Analytics (use lightweight Plausible or Vercel Analytics).
- Font Display: `font-display: swap;` ensures text is visible while *Playfair Display* loads.

---

5. Security & Reliability

- SSL: Mandatory via Let's Encrypt (provided by Netlify).
- No-Database Security: Since there is no database, there is no risk of SQL injection or user data breaches.
- Form Validation: Client-side regex for Kenyan phone formats (`^(\+254|0)[17]\d{8}$`).

---

6. Deployment Workflow

The site is built to be "clonable" for new clients in under 30 minutes.

- Clone Template: `git clone restaurant-vibe-demo`
- Update Config: Change `client_phone` and `colors` in `tailwind.config.js`.
- Update Content: Replace images in `/assets` and items in `menu.json`.
- Push: `git commit -m "Launch Nyali Branch"` -> Auto-deploys to production.

---

7. Analytics Strategy

To prove the "Revenue Math" to the client, we track "Intent Events" rather than just page views:

- Menu_Open: User is interested in food.
- WhatsApp_Redirect: The primary conversion (Confirmed lead).
- Call_Click: Secondary conversion for older demographics.