Wedding Website: Gabriel & Milleny

Architecture & Implementation Guide based on PDF Specification

1. Setup & Installation

As per the PDF (Section 6.2), we use the `create-next-app` command with specific flags.

```
npx create-next-app@latest casamento-gabriel-milleny --typescript --tailwind --eslint
cd casamento-gabriel-milleny
npm install lucide-react date-fns qrcode.react zod
```

2. Folder Structure (File Tree)

Organize your files exactly as described in **Section 2.3** of the report.

```
casamento-gabriel-milleny/
├── app/
│   ├── layout.tsx       # Global font import (Playfair/Lato) & Meta tags
│   ├── page.tsx         # Main Home (Hero)
│   ├── rsvp/
│   │   └── page.tsx     # RSVP Form Route
│   ├── presentes/
│   │   └── page.tsx     # Gift Registry Route
│   └── api/
│       └── rsvp/        # Server Route for Supabase/Resend
├── components/          # Reusable UI
│   ├── Navigation.tsx   # Bottom/Top Bar
│   ├── Hero.tsx         # Names & Date
│   ├── GiftCard.tsx     # Individual gift item
│   └── Countdown.tsx    # Client-side only timer
├── lib/                 # Business Logic (The "Motor")
│   ├── pix.ts           # CRC16 & Payload Logic (Critical)
│   └── utils.ts         # Helper functions
├── public/              # Static Assets
│   └── noise.svg        # The grain texture
└── tailwind.config.ts   # Design System
```

3. Key File Implementations

A. Design System (`tailwind.config.ts`)

Configuring the palette from **Section 4.1**.

```
import type { Config } from "tailwindcss";

const config: Config = {
  content: ["./app/**/*.{ts,tsx}", "./components/**/*.{ts,tsx}"],
  theme: {
    extend: {
      colors: {
        wedding: {
```

```
          bg: '#F5F1E6',
          card: '#EBE5CE',
          primary: '#5D4037', // Brown text
          accent: '#A67B5B',  // Terracotta buttons
          muted: '#8C7C71',
        }
      },
      fontFamily: {
        serif: ['var(--font-playfair)'],
        sans: ['var(--font-lato)'],
      }
    },
  },
  plugins: [],
};
export default config;
```

B. The PIX Engine (`lib/pix.ts`)

The PDF emphasizes a manual implementation of the CRC16-CCITT-FALSE algorithm to avoid heavy dependencies.

```
// lib/pix.ts

// Polynomial: 0x1021 | Initial Value: 0xFFFF
const calculateCRC16 = (payload: string): string => {
  const polynomial = 0x1021;
  let crc = 0xFFFF;

  for (let i = 0; i < payload.length; i++) {
    let c = payload.charCodeAt(i);
    crc ^= (c << 8);
    for (let j = 0; j < 8; j++) {
      if ((crc & 0x8000) !== 0) {
        crc = (crc << 1) ^ polynomial;
      } else {
        crc = crc << 1;
      }
    }
  }
  return (crc & 0xFFFF).toString(16).toUpperCase().padStart(4, '0');
};

interface PixParams {
  key: string;
  name: string;
  city: string;
  amount: number;
  txid?: string;
}

export const generatePixPayload = ({ key, name, city, amount, txid = '***' }: PixParams
   // Implementation logic matches the provided React file
   // 1. Format Amount (toFixed(2))
   // 2. Normalize Strings (remove accents)
   // 3. Construct TLV Strings (IDs 00, 26, 52, etc.)
   // 4. Append CRC16
};
```

4. Deployment (Section 7)

1. Push the code to a **GitHub** repository.
2. Connect the repo to **Vercel**.
3. Add Environment Variables in Vercel settings:
    - `NEXT_PUBLIC_PIX_KEY` : (e.g., email@address.com)
    - `SUPABASE_URL` : (For RSVP database)
    - `SUPABASE_ANON_KEY` : (For RSVP API)

5. PDF Specific Features Checklist

- [x] **Grainy Texture**: Implemented via CSS `mix-blend-overlay` in `layout.tsx` .
- [x] **Google Maps**: Used CSS filters `grayscale(100%) sepia(20%)` to match the beige theme.
- [x] **Mobile First**: Navigation bar fixed at bottom for mobile, top for desktop.
- [x] **PIX Copy/Paste**: Uses `navigator.clipboard` for native mobile interaction.