Kilometros de Vida - Design Document

1. Project Overview

Kilometros de Vida is a web platform connecting food donors (restaurants, markets) with volunteer drivers to distribute surplus food to those in need. The goal is to reduce food waste and alleviate hunger through efficient logistics.

2. Architecture

The project follows the **MERN Stack** architecture: - **Frontend**: React, Tailwind CSS, Framer Motion. - **Backend**: Node.js, Express.js. - **Database**: MongoDB (Atlas). - **Authentication**: Google OAuth 2.0. - **Third-Party API**: Geocoding API.

3. Database Schema

Users Collection (`users`)

- `_id`: ObjectId
- `email`: String (Unique)
- `name`: String
- `picture`: String (URL)
- `lastLogin`: Date

Givers Collection (`givers`)

- `_id`: ObjectId
- `userId`: String (Linked to User email/ID)
- `orgName`: String
- `contactPerson`: String
- `donorEmail`: String
- `donorPhone`: String
- `foodType`: String
- `pickupTime`: String
- `address`: String
- `createdAt`: Date

Drivers Collection (`drivers`)

- `_id`: ObjectId
- `volunteerName`: String
- `volunteerEmail`: String
- `volunteerPhone`: String
- `availability`: String
- `createdAt`: Date

4. API Endpoints

Authentication

- `POST /api/auth/google`: Verifies Google ID token and creates/updates user.

Donations (Givers)

- `POST /join/giver`: Create a new donation.
- `GET /api/data`: Fetch all donations (public view).
- `GET /api/my-donations`: Fetch donations for a specific user (authenticated).
- `PUT /api/donations/:id`: Update a donation.
- `DELETE /api/donations/:id`: Delete a donation.

Volunteers (Drivers)

- `POST /join/driver`: Register a new volunteer.

5. Frontend Design

Key Components

- **Navbar**: Responsive navigation with User Profile dropdown.
- **Home**: Landing page with "Impact Map" showing live donation locations.
- **Giver**: Form to register donations with address input.
- **Profile**: Dashboard for users to manage their donations (Edit/Delete).
- **DonationMap**: Google Maps integration to visualize data.

UX/UI

- **Design System**: Modern, clean aesthetic using a primary blue/teal palette and serif fonts for headings.
- **Interactivity**: Animations using Framer Motion (fade-ins, counters).
- **Responsiveness**: Fully responsive layout for mobile and desktop.

6. Security

- **Authentication**: Google Identity Services (OAuth 2.0) ensures secure login without storing passwords.
- **Authorization**: Backend endpoints for modifying data should ideally verify the user's identity (currently implemented via client-side checks and email matching for the prototype).
- **Environment Variables**: API Keys and DB URIs are stored in `.env` files and not committed to version control.

7. Challenges & Solutions

- **Challenge**: Linking donations to users without a complex auth system.
  - **Solution**: Used Google Auth to get a verified email and linked records via that email/ID.
- **Challenge**: Visualizing data.
  - **Solution**: Integrated Google Maps API to geocode addresses on the client-side and display markers, providing immediate visual feedback of the project's impact.
- **Challenge**: Making sure addresses are real and can be mapped.
  - **Solution**: Added Google's geocoding to check addresses automatically.
- **Challenge**: Working on phones and computers.
  - **Solution**: Used responsive design to adapt to any screen size.
- **Challenge**: Stopping duplicate registrations.
  - **Solution**: Check emails to prevent the same person signing up twice.
- **Challenge**: Keeping the site fast with lots of donations.
  - **Solution**: Only load donations in batches instead of all at once.
- **Challenge**: Hosting the website - Render kept crashing and was slow.
  - **Solution**: Switched to Vercel which handles the frontend much better.
- **Challenge**: Protecting user privacy on the public map.
  - **Solution**: Show location markers without exact addresses or personal info.
- **Challenge**: Connecting donors with drivers for pickup.
  - **Solution**: Share contact info only after driver registration for coordination.