Kilometros de Vida

> Connecting food surplus with those in need through volunteer-driven logistics

A full-stack MERN application that reduces food waste by connecting restaurants, markets, and food donors with volunteer drivers who distribute surplus food to communities in need.

Contents

Features

For Donors

- Register food donations with pickup details
- See donation locations on interactive map
- Edit or delete donations from profile
- Secure Google OAuth authentication

For Volunteers

- Sign up as volunteer driver
- Specify availability preferences
- Manage volunteer profile

For Everyone

- View real-time impact statistics
- Interactive map showing donation locations
- Fully responsive mobile design
- Accessible interface (WCAG AA)

Tech Stack

Frontend

- **React 19** - UI library

- **Vite** - Build tool and dev server
- **React Router** - Client-side routing
- **Tailwind CSS 4** - Styling
- **Framer Motion** - Animations
- **React Leaflet** - Map visualization
- **Axios** - HTTP client

Backend

- **Node.js 25** - Runtime environment
- **Express.js 5** - Web framework
- **MongoDB Atlas** - Database
- **Google OAuth 2.0** - Authentication
- **Nominatim API** - Geocoding

Prerequisites

Before you begin, ensure you have: - **Node.js** (v18 or higher) - **npm** (v9 or higher) - **MongoDB Atlas account** (free tier works) - **Google Cloud Console account** (for OAuth)

Prerequisites

Before you begin, make sure you have installed:

- **Node.js** (version 14 or higher)
- **npm** (comes with Node.js)
- A **MongoDB Atlas** account (free)
- A **Google Cloud Console** account (for OAuth)

Step-by-Step Setup

1. Clone the Repository

```
git clone https://github.com/alvarogalloc/km-de-vida.git
cd km-de-vida
```

2. Set Up MongoDB Atlas

1. Go to MongoDB Atlas
2. Create a free account if you don't have one
3. Create a new **Cluster** (select the free tier option)
4. Go to **Database Access** and create a database user:
    - Username: `your-username`

- Password: `your-secure-password`
- Permissions: **Read and write to any database**

5. Go to **Network Access** and add your IP:
   - Click **Add IP Address**
   - Select **Allow Access from Anywhere** (0.0.0.0/0) for development
6. Go to your cluster and click **Connect**
7. Select **Connect your application**
8. Copy the **connection string** (it will look like this):

   ```
   mongodb+srv://user:<pass>@cluster.mongodb.net/
   ```

9. Replace `<password>` with your actual password

3. Set Up Google OAuth

1. Go to Google Cloud Console
2. Create a new project or select an existing one
3. Enable the **Google+ API**:
   - Go to **APIs & Services** > **Library**
   - Search for "Google+ API" and enable it
4. Create OAuth 2.0 credentials:
   - Go to **APIs & Services** > **Credentials**
   - Click **Create Credentials** > **OAuth client ID**
   - Application type: **Web application**
   - Name: `Km de Vida Local`
   - **Authorized JavaScript origins**:
     - `http://localhost:5173`
     - `http://localhost:5050`
   - **Authorized redirect URIs**:
     - `http://localhost:5173`
   - Click **Create**
5. Copy the **Client ID** and **Client Secret**

4. Configure Environment Variables

Backend (project root):

1. Copy the template file:

   ```
   cp .env.template .env
   ```

2. Edit `.env` and add your credentials:

   ```
   ATLAS_URI=yourmongoconnection
   PORT=5050
   DB_NAME=kmdevida
   ```

```
GOOGLE_CLIENT_ID=your-client-id.apps.googleusercontent.com
GOOGLE_CLIENT_SECRET=your-client-secret
```

Frontend (`client` folder):

1. Copy the template file:

   ```
   cd client
   cp .env.template .env
   ```

2. Edit `client/.env` and add your Google Client ID:

   ```
   VITE_GOOGLE_CLIENT_ID=your-client-id.apps.googleusercontent.com
   ```

   IMPORTANT: This must be the **same Client ID** used in the backend.

5. Install Dependencies

Backend:

```
npm install
```

Frontend:

```
cd client
npm install
cd ..
```

6. (Optional) Seed the Database with Sample Data

If you want to start with sample data:

```
node seed.js
```

This will add 4 sample donations to your database.

Run the Application

You need **TWO terminals** open:

Terminal 1 - Backend:

```
npm run dev
```

The backend server will be running at **http://localhost:5050**

Terminal 2 - Frontend:

```
cd client
npm run dev
```

The frontend will be running at **http://localhost:5173**

Seeding Sample Data

To populate the database with sample donations:

```
node seed.js
```

This adds 4 sample donors with addresses in Guadalajara, Mexico.

Production Build

```
cd client
npm run build
```

Build output will be in `client/dist/`

Project Structure

```
km-de-vida/
|-- client/                 # Frontend React application
|   |-- src/
|   |   |-- components/    # Reusable components
|   |   |-- context/       # React Context
|   |   |-- pages/         # Page components
|   |   |-- App.jsx        # Main app component
|   |   |-- main.jsx       # Entry point
|   |   |-- index.css      # Global styles
|   |-- public/            # Static assets
|   |-- vite.config.js     # Vite configuration
|-- middleware/            # Express middleware
|-- server.js             # Express server
|-- seed.js               # Database seeding script
|-- ARCHITECTURE.md       # System architecture docs
|-- WORKFLOW.md           # User workflow diagrams
|-- DESIGN_DOC.md         # Design documentation
|-- Dev Log.md            # Development log
|-- README.md             # This file
```

API Documentation

Authentication

- `POST /api/auth/google` - Verify Google OAuth token

Donations

- `GET /api/data` - Get all donations (public)
- `GET /api/my-donations?email=<email>` - Get user's donations
- `POST /join/giver` - Create new donation
- `PUT /api/donations/:id` - Update donation
- `DELETE /api/donations/:id` - Delete donation

Volunteers

- `POST /join/driver` - Register as volunteer
- `GET /api/my-volunteer-shifts?email=<email>` - Get user's shifts

See Architecture.pdf for detailed API documentation.

Team

Regina Beltrán López - Frontend Design & UX
Daniela Terán Martija - Full-Stack Integration & Deployment
Álvaro Gallo Cruz - Backend Architecture & Database

License

This project is licensed under the ISC License.

Acknowledgments

- **OpenStreetMap** for free geocoding via Nominatim
- **Google** for OAuth authentication
- **MongoDB** for database hosting
- **Vite** for amazing developer experience

Built with love to reduce food waste and fight hunger