

## Tech Stack, frameworks, and architecture

Layer	Technology	Justification
Frontend	<b>Next.js (React)</b>	Server-side rendering, API integration, routing, performance
Styling	<b>Tailwind CSS</b>	Utility-first, scalable, responsive design system
Backend	<b>Next.js API Routes</b> ( <i>or external API</i> )	Built-in backend endpoints, avoids context switching
Auth	<b>Auth.js / Clerk / Supabase Auth</b>	Easy role-based authentication, social login support
Database	<b>PostgreSQL</b> ( <i>via Prisma ORM</i> )	Relational, scalable, schema-safe with strong data typing
ORM	<b>Prisma</b>	Type-safe DB access, modern dev experience
Hosting	<b>Vercel</b>	Perfect fit for Next.js, instant deployment & CI/CD
Charts	<b>Recharts / Chart.js</b>	Render key analytics and impact data
Maps	<b>Mapbox / Leaflet.js</b>	Visualize pickup/drop-off locations and food activity
File/Image Upload	<b>Cloudinary / UploadThing</b> ( <i>optional</i> )	Media support for food photos

## Architecture Decisions

### Monorepo with Modular Structure

- **components/**: Reusable UI components (Forms, Cards, Tables)
- **pages/**: Route-based files using Next.js conventions
- **lib/**: Utilities (API clients, validation, auth helpers)
- **prisma/**: Schema + DB access
- **api/**: API routes for listings, transactions, users, analytics

### Role-Based Access

- Abstracted in middleware or per-page logic to enforce permissions
- Different dashboard experiences for grocers, restaurants, and food banks

### **Transactions & Food Lifecycle**

- Listings have a lifecycle: **AVAILABLE** → **CLAIMED** → **COMPLETED**
- Transactions table links donors and food banks for traceability

### **Analytics Pipeline**

- Aggregation queries powered by Prisma or scheduled CRON jobs
- Admin dashboard reads precomputed metrics or live calculations

## **Development Roadmap (Broken into Tasks)**

### **Phase 1: Project Bootstrap**

- Initialize Next.js app with TypeScript + Tailwind CSS
- Configure ESLint, Prettier, and Git hooks
- Set up auth (Clerk/Auth.js/Supabase)
- Set up PostgreSQL + Prisma schema

### **Phase 2: Authentication & User Roles**

- Sign-up page with role selection (Grocer / Restaurant / Food Bank)
- Protect routes based on role
- Create role-based dashboards
- Store location/address per user

### **Phase 3: Core Features**

#### **For Grocers & Restaurants**

- Food listing form (name, quantity, expiry, type, pickup/delivery)
- “My Listings” dashboard
- Edit/delete listing flow



### **For Food Banks**

- Browse listings (filters: food type, distance, availability)
- Claim listing / schedule pickup
- View “My Orders” page



### **Phase 4: Shared Infrastructure**

- Pickup/Drop-off location management
- Display interactive map with location pins
- Confirm transaction flow (from CLAIMED to COMPLETED)



### **Phase 5: Analytics & Admin Tools**

- Track and display metrics (pounds saved, top categories)
- Graphs and charts on dashboards
- Admin panel for resolving disputes or moderation (optional)



### **Phase 6: Polish & Deployment**

- UI polish (accessibility, mobile responsiveness)
- Environment variables & production configs
- Deploy to Vercel
- Launch checklist