By: Nir Froiman and Alex Segal
Led By: Ronen Zilber

המכללה האקדמית להנדסה בראודה בכרמיאל

CartGenie

Project Code: 25-2-D-20

The Problem

There is a significant disconnect between daily grocery shopping and personal health requirements.

- **Generic Advice** - Doctors often provide vague dietary recommendations that are difficult for patients to translate into specific product choices at the supermarket.
- **Lack of Integration** - Current grocery apps focus on manual lists and convenience, lacking the ability to integrate medical data (like blood tests).
- **Unhealthy Choices** - Without accessible, personalized guidance, users unknowingly purchase foods that may be detrimental to their specific medical conditions.

The Need

There is a critical demand for a smart platform that bridges the gap between medical diagnosis and food consumption.

- **Personalized Automation** - Users need a system that analyzes objective health data (blood work) to automatically generate medically appropriate shopping lists.
- Actionable Guidance - A tool is required to convert complex medical data into simple, specific product recommendations.
- **Feedback Mechanism** - There is a need to validate dietary compliance by scanning receipts, ensuring that the food actually purchased aligns with the user's health goals and medical advice.

Main Requirements

Main Functional Requirements (What it does)

- User Profiles - Create and manage health profiles (Age, BMI, conditions).
- Document Upload - Upload or scan blood tests and grocery receipts (PDF/Image).
- Smart Analysis - Extract text via OCR and analyze data using an AI agent (Gemini).
  - Blood Tests - Detect abnormal values and biomarkers.
  - Receipts - Classify food items as Healthy, Neutral, or Risky based on the user's health.
- Feedback - Generate explained health reports and allow exporting them to PDF.

Main Non-Functional Requirements (How it performs)

- Performance - Process documents in under 10 seconds with 95% uptime.
- Accuracy - Support English & Hebrew OCR and match at least 80% of receipt items.
- Tech Stack - Use Tesseract for OCR and an AI Agent for analysis models.
- UX & Privacy - Intuitive mobile design with dark mode; users can delete data at any time.

Project Metrics, Goals & Compliance (KPIs)

1. Technical Performance

- **Barcode Latency** - Achieved ~2.5 seconds (Goal: < 4s), ensuring a fast shopping experience.
- **Receipt Processing** - Achieved ~6-8 seconds (Goal: < 10s) using the optimized Gemini model.
- **Stability** - Maintained 99.9% uptime due to a robust Node.js and TypeScript architecture.

2. AI Accuracy & Reliability

- **Health Recommendations** - ~95% accuracy in identifying harmful ingredients against medical profiles (Goal: > 90%).
- **OCR** - ~90% success rate in parsing items and prices from receipts (Goal: > 85%).
- **Consistency** - While textual phrasing varies slightly, the critical safety verdicts (Safe/Avoid) remain 100% consistent.

3. User Experience (UX)

- **Onboarding** - Complete profile setup takes under 2 minutes.
- **Cross-Platform** - Fully functional on both iOS and Android with seamless access to native features (Camera, Haptics).
- **Conclusion** - CartGenie successfully met all critical KPIs. The system effectively balances the heavy computational load of Generative AI with a responsive, accurate, and user-friendly mobile experience.

The Solution

1. Frontend: Mobile Application

The client side is built using React Native and Expo, enabling a high-performance, native experience on both iOS and Android from a single codebase.

- Core Logic: Developed in TypeScript to ensure type safety and prevent runtime errors.
- Device Integration: Utilizes Expo SDKs to access hardware features like the Camera (for scanning) and Haptics (for tactile feedback).
- Routing: Implements file-based routing via Expo Router for seamless navigation.

2. Backend & Database

The server acts as a secure gateway between the user, the database, and the AI models.

- Server: Built on Node.js and Express.js, providing a scalable RESTful API that handles user authentication, data processing, and external API orchestration.
- Database: Uses MongoDB (with Mongoose). Its NoSQL document structure is optimized for storing complex, nested data such as dynamic medical profiles and detailed product ingredients, which often vary in structure.

3. AI Engine: The Core Intelligence

Unlike traditional apps that use static databases, CartGenie employs Google Gemini as a reasoning engine.

- Gemini (Text & Logic) - Receives raw product data and compares it against the user's specific health constraints ("High Sodium vs. Hypertension"). It outputs a structured JSON response with a safety classification (Safe/Avoid) and a human-readable explanation.

The Technolgies

Tesseract OCR

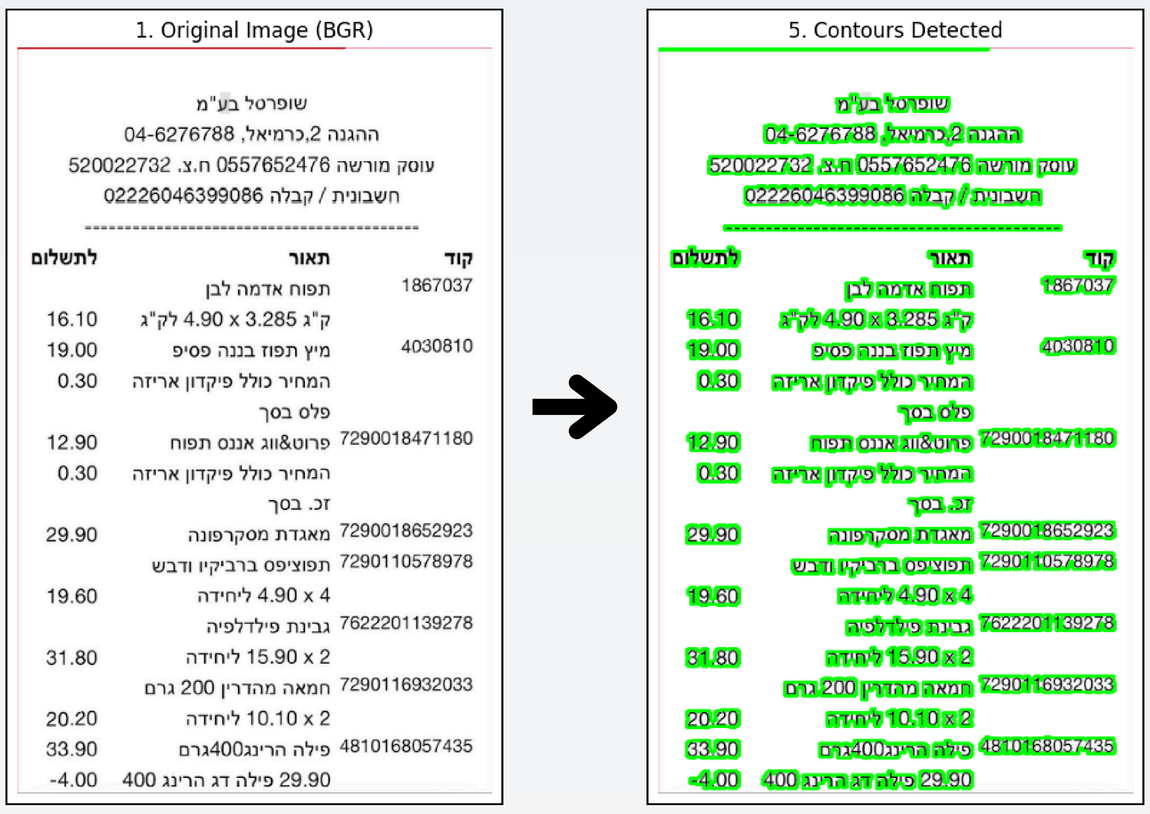
Gemini

JavaScript

mongoDB

Sign in with Google

Tesseract OCR example



Scan Me!!

(Available only for Android)