SREE NARAYANA GURU COLLEGE OF ENGINEERING AND TECHNOLOGY

Blood Report Fitness Evaluation System

Group Number: 6

Submitted by:

DHIYA MANOJ

SREEHARI T

VARSHA RAMACHANDRAN

VISMAYA PI

Guided by:

Ms Sruthi Krishnan

Date of Submission: January 27, 2026

System Specification

Project Summary

The Blood Report Fitness Evaluation System is a comprehensive web application designed to empower users to understand their health data. Utilizing advanced Optical Character Recognition (OCR) technology, the system scans and analyzes physical blood test reports, enabling users to digitize their medical records effortlessly. Beyond simple digitization, the platform acts as an intelligent health companion, providing personalized diet plans, fitness recommendations, and health insights based on specific blood biomarkers.

Key Features

- **Smart Blood Report Analysis:** Instantly scans and digitizes over 100 medical parameters from physical reports using Tesseract.js.

- **Personalized Diet Recommendations:** Generates custom meal plans tailored to specific health conditions (e.g., Anemia, Diabetes) identified in the report.

- **Fitness Guidance:** Suggests exercise routines appropriate for the user's current health status and fitness goals.

- **Digital Lens Technology:** Advanced image preprocessing to enhance OCR accuracy, correcting for low-quality or poorly lit images.

- **Offline-First Architecture:** Fully functional Progressive Web App (PWA) that works without an active internet connection, storing data locally.

- **AI Health Assistant:** Integrated chatbot to answer general health queries and explain medical terms.

Technology Stack

- **Frontend:** React 18.3.1 for a dynamic and responsive user interface.

- **Build Tool:** Vite 5.3.1 for fast development and optimized production builds.

- **OCR Engine:** Tesseract.js 5.0.3 running entirely client-side for privacy and speed.

- **Mobile Support:** Capacitor 6.0.0 allowing deployment as a native Android application.

- **Styling:** Vanilla CSS with modern Glassmorphism design principles.

- **PWA:** vite-plugin-pwa for offline capabilities and installability.

System Requirements

Hardware Requirements

For Web Application (PWA):

- Processor: Dual-core 1.5 GHz or higher
- RAM: Minimum 2 GB (4 GB recommended)
- Storage: 100 MB free disk space
- Camera: 5 MP or higher (for image capture)
- Display: 1280x720 resolution minimum

For Android Application:

- Android Version: 7.0 (Nougat) or higher
- RAM: Minimum 2 GB
- Storage: 150 MB free space
- Camera: Rear camera with autofocus

Software Requirements

Development Environment:

- Node.js: v16.0 or higher
- npm: v7.0 or higher
- Git: v2.30 or higher
- Code Editor: VS Code (recommended) or any modern IDE

Client-Side (Browser):

- Google Chrome: v90+
- Mozilla Firefox: v88+
- Safari: v14+
- Microsoft Edge: v90+
- JavaScript: ES6+ enabled

Optional Backend:

- Python: v3.8 or higher
- Flask/FastAPI: Latest stable version
- SQLite: v3.35 or higher

Functional Requirements

FR1: Blood Report Analysis

- The system shall accept blood report images in JPEG, PNG, or PDF format.

- The system shall extract text from the uploaded image using OCR technology.

- The system shall recognize and parse at least 100 different medical parameters.

- The system shall validate extracted values against physiological ranges.

FR2: Health Assessment

- The system shall compare extracted values against medical reference ranges.

- The system shall classify each parameter as Normal, Low, or High.

- The system shall predict potential health conditions based on extracted data.

- The system shall provide risk assessment (Low, Medium, High, Critical).

FR3: Personalized Recommendations

- The system shall generate diet plans based on identified deficiencies.

- The system shall suggest fitness activities appropriate for user's health status.

- The system shall provide food recommendations for each abnormal parameter.

FR4: Data Management

- The system shall store user profiles and report history locally.

- The system shall allow users to view, edit, and delete saved reports.

- The system shall support exporting reports as PDF documents.

- The system shall track weight and BMI over time.

FR5: User Interface

- The system shall provide a responsive interface for mobile and desktop.

- The system shall display results in color-coded format for easy interpretation.

- The system shall include an AI chatbot for health-related queries.

Non-Functional Requirements

NFR1: Performance

- OCR processing shall complete within 10 seconds for standard quality images.
- The application shall load the main interface within 2 seconds.
- The system shall handle concurrent analysis of multiple parameters efficiently.

NFR2: Accuracy

- OCR shall achieve minimum 90% accuracy on printed blood reports.
- Parameter extraction shall have less than 5% error rate after validation.
- Disease prediction rules shall be based on established medical guidelines.

NFR3: Usability

- The interface shall be intuitive requiring no training for basic operations.
- The system shall provide clear error messages and guidance.
- The application shall be accessible on devices with screen readers.

NFR4: Reliability

- The system shall function offline without network connectivity.
- Data shall be persisted locally to prevent loss on browser closure.
- The system shall gracefully handle invalid or corrupted images.

NFR5: Security & Privacy

- All data processing shall occur client-side (no server uploads).
- User data shall be stored only in the local browser/device.
- The system shall not transmit health data over the network.

NFR6: Compatibility

- The PWA shall be installable on Android, iOS, and desktop platforms.
- The application shall support multiple screen sizes (320px to 4K).
- The system shall function on browsers released within the last 2 years.

NFR7: Maintainability

- Code shall follow React best practices and component-based architecture.
- Medical knowledge base shall be easily updatable without code changes.
- The system shall include comprehensive inline documentation.

Module Description

1. Blood Analysis Engine (bloodAnalysis.js)

This is the core intelligence module of the system. It contains a comprehensive medical knowledge base covering over 100 blood parameters. The module defines reference ranges for normal, low, and high values for each parameter. It parses the raw text extracted by the OCR engine, validates the values against likely physiological ranges to prevent errors, and identifies potential health issues.

2. OCR Scanning Interface (BloodEvaluation.jsx)

The primary user interface for data entry. This module handles image capturing and uploading. It integrates the 'Digital Lens' algorithm which preprocesses images (converting to grayscale, adjusting contrast) before passing them to the OCR engine. It features a three-tier pattern matching system to accurately extract numerical values even from noisy text, distinguishing between decimals, integers, and malformed numbers.

3. Diet & Recommendation Generator (SpecializedDiet.jsx)

Based on the analysis results, this module constructs personalized action plans. It filters through a database of dietary recommendations to suggest foods that help correct specific deficiencies (e.g., iron-rich foods for low hemoglobin). It also generates a shopping list and provides lifestyle tips tailored to the user's unique blood profile.

4. Profile & Progress Dashboard (ProfileDashboard.jsx & WeightProgress.jsx)

Manages user data persistence and visualization. This module allows users to track their health trends over time, log parameters like weight and BMI, and view their report history. It utilizes LocalStorage to ensure data remains available even when offline, providing a seamless mobile app experience.

Architectural Diagram

Presentation Layer (React Frontend)

| Image Capture Input | Results Dashboard UI | Profile Manager |

Processing Layer (Client-Side Logic)

| Digital Lens Preprocessing | Tesseract.js OCR Engine | Regex & Keyword Parser |

Medical Intelligence Layer

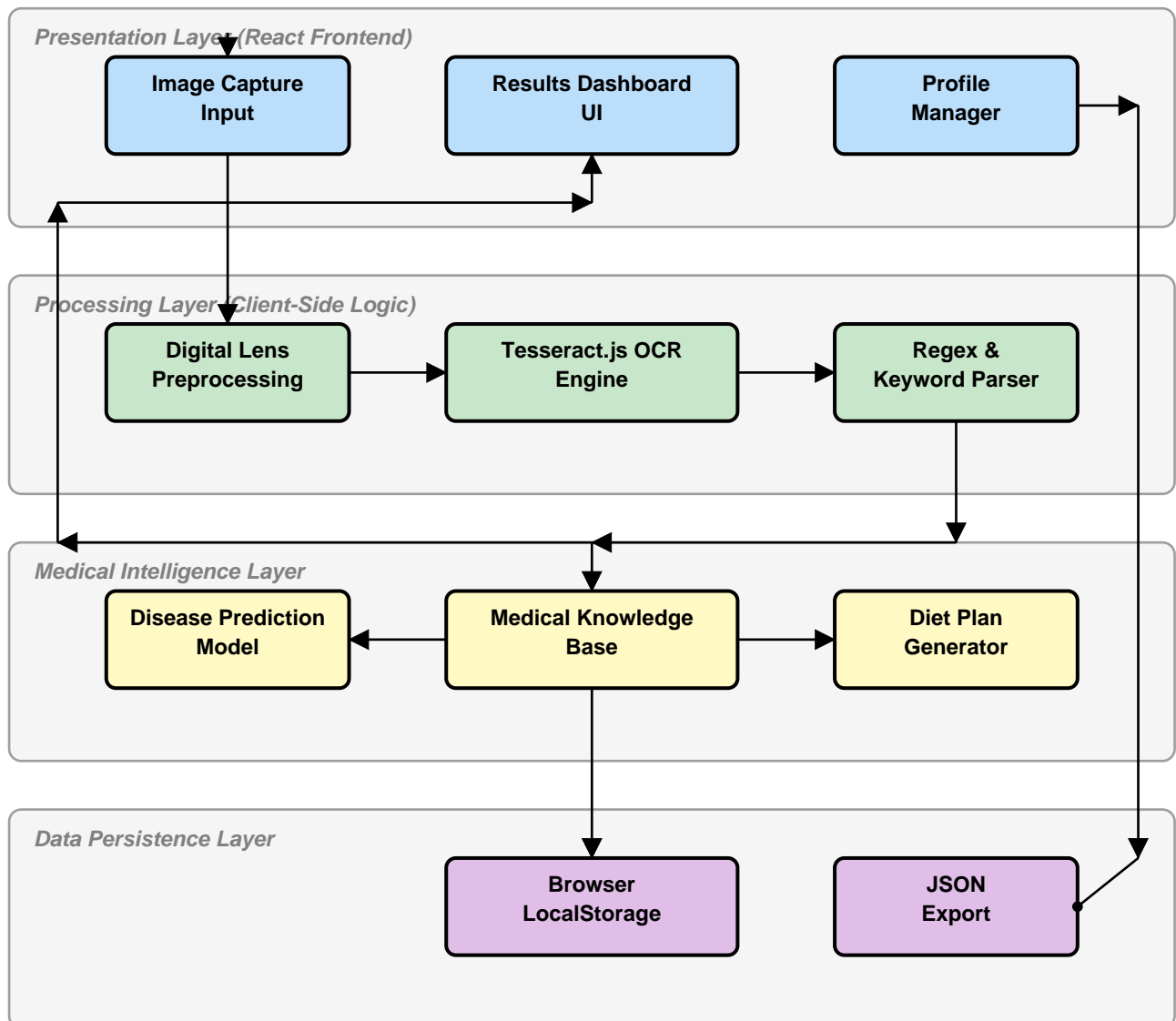| Disease Prediction Model | Medical Knowledge Base | Diet Plan Generator |

Data Persistence Layer

| Browser LocalStorage | JSON Export |

Figure 1: High-Level System Architecture and Data Flow