



Mission

Create a unified platform connecting blood donors with recipients in real-time during emergencies



Vision

Reduce blood shortage emergencies through fast, efficient donor-recipient matching



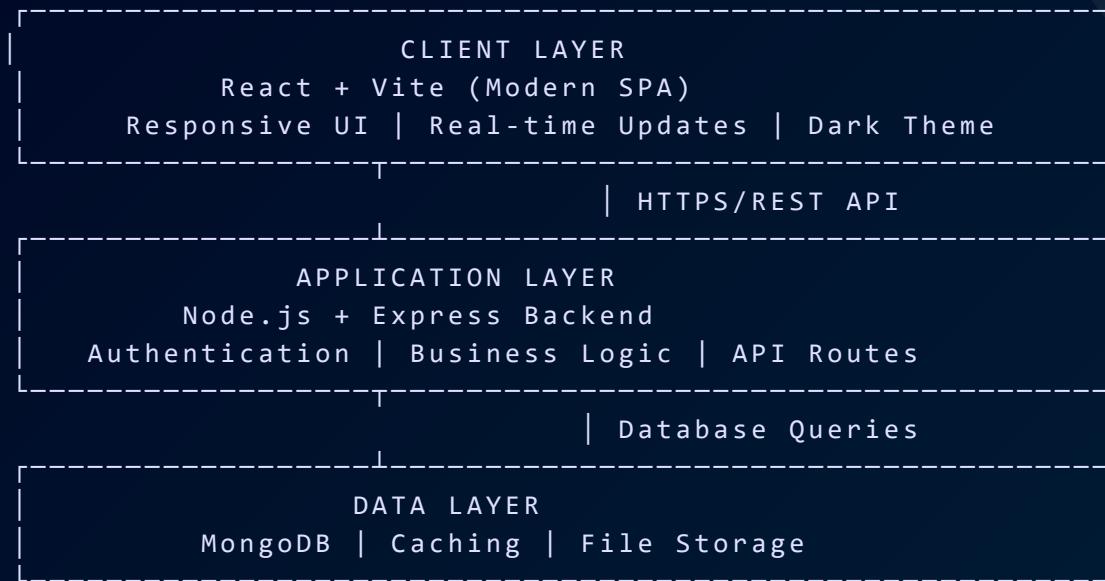
Functionality

Registration, Search, Profiles, Emergency Requests, Admin Analytics, Real-time Notifications



Users

Donors, Recipients, Hospital Admins, Emergency Coordinators, System Administrators



Microservices-Ready Architecture with Docker containerization

Technology Stack

- **Node.js 18+** - Runtime
- **Express.js** - REST API Framework
- **MongoDB** - NoSQL Database
- **JWT** - Authentication
- **Socket.io** - Real-time Comm

Core Features

- RESTful API Endpoints
- User Authentication & Authorization
- Real-time Notifications
- Database Optimization
- Error Handling & Logging

Node.js

Express

MongoDB

JWT

Socket.io

API ENDPOINTS STRUCTURE

```
/api/auth
POST /register           - User Registration
POST /login              - User Authentication
POST /logout             - Session Termination

/api/donors
GET /                   - List All Donors
POST /                  - Create Donor Profile
GET /:id                - Get Donor Details
PUT /:id                - Update Profile

/api/search
GET ?bloodType=0         - Search by Blood Type
GET ?location=city      - Search by Location

/api/emergency
POST /request            - Create Emergency Request
GET /active              - Active Requests
PUT /:id/fulfill        - Fulfill Request

/api/admin
GET /analytics           - Dashboard Analytics
GET /reports             - Generate Reports
```



React 19.2

Modern React with Hooks, Context API, and functional components



Vite 7.2

Lightning-fast build tool with HMR and optimized production builds



Styling

CSS Gradients, Animations, Responsive Design, Dark Theme

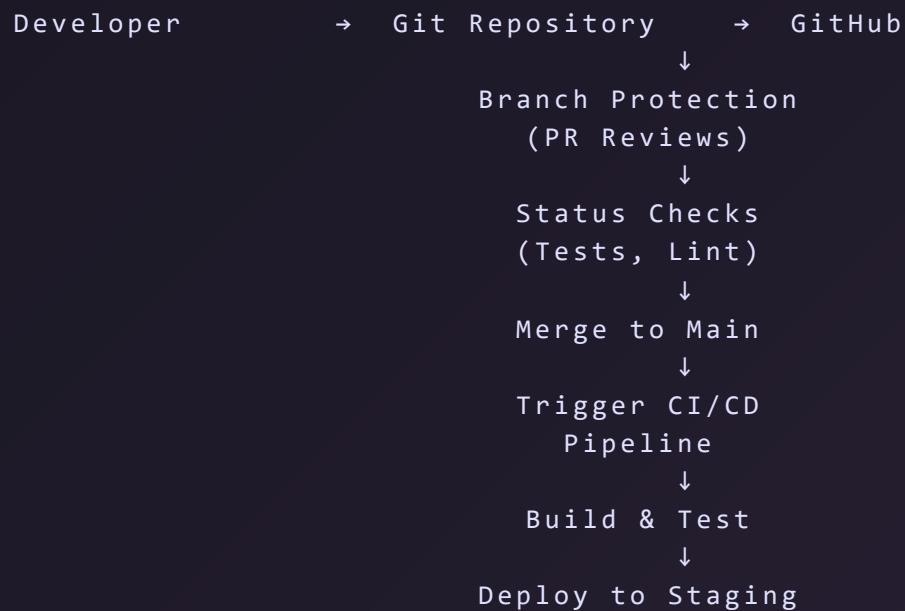
Components

- Donor Registration Form
- Advanced Search
- Donor Profiles
- Emergency Requests

Features

- Responsive Mobile UI
- Real-time Updates
- Admin Dashboard
- Beautiful Animations

GIT WORKFLOW



GitHub

Git Workflows

Branch Protection

PR Reviews



GitHub Actions

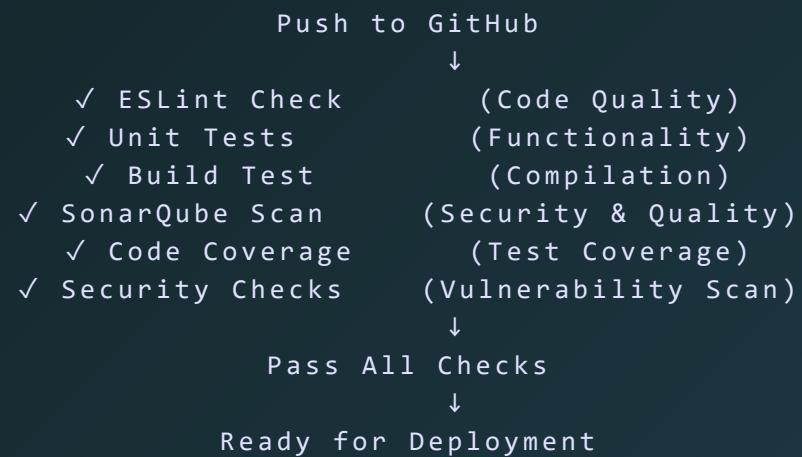
Automated testing, linting, and build verification on every push



SonarQube

Code quality analysis, security scanning, and coverage reporting

CI/CD PIPELINE STAGES



DOCKERFILE STRUCTURE

```
FROM node:20-alpine
  ↓
  WORKDIR /app
  ↓
  COPY package*.json ./
  ↓
  RUN npm ci --only=production
  ↓
  COPY . .
  ↓
  EXPOSE 5000
  ↓
  CMD [ "node", "server.js" ]
```

IMAGE: blood-donor-backend:latest

REGISTRY: Docker Hub / GitHub Container Registry

Benefits

- Consistent Environment
- Easy Scaling
- Version Control
- Quick Deployment

Orchestration

- Kubernetes Ready
- Docker Compose
- Multi-stage Builds
- Health Checks

Vercel Deployment

- **Auto-Deploy** on push to main
- **Node 20.x LTS Runtime**
- **SPA Routing** Configured
- **Edge Functions** Ready
- **Analytics** Enabled

Build & Optimization

- Vite Build Process
- Code Minification
- Asset Optimization
- CSS Optimization
- Performance Monitoring

Git Push → GitHub → Vercel → Build & Test → Deploy

Issue 1: CSS Contrast

Text not meeting WCAG accessibility standards

Solution

Updated button gradients with darker colors for proper contrast ratios

Issue 2: CSS Syntax Error

Duplicate closing braces breaking build

Solution

Removed malformed CSS blocks and validated all stylesheets

Issue 3: Vercel Config

Missing deployment configuration and Node version conflicts

Solution

Created `vercel.json` with proper Node 20.x, SPA routing, and build commands

```
=====
|| Blood Donor Finder Platform
|| Saving Lives Through Technology
|| 
|| Full Stack Development
|| CI/CD Pipeline Integration
|| Production Ready Deployment
|| Quality & Security Focused
=====
```