BLOOM AFTER

Technical Requirements Document

Purpose

This document defines the technical approach for building Bloom After - a postpartum depression support web platform. It covers the system architecture, technology choices, database design, API structure, and development workflow. It is the reference document all engineers on the team work from.

System Overview

What We Are Building

Bloom After is a web application that gives Nigerian mothers and their caregivers access to trusted, clinically reviewed information on postpartum depression. The platform includes a resource hub, clinic and specialist directories, a moderated community story library, and NGO and media directories.

Architecture Layers

Layer	Responsibility
Frontend (HTML/CSS/JS)	What the user sees and interacts with
Backend API (Node.js + Express)	Handles requests, business logic, and talks to the database
Database (Supabase / PostgreSQL)	Stores all content, clinics, stories, and admin data
CMS (Strapi)	Lets non-technical editors write and publish content
File Storage (Cloudinary or Supabase Storage)	Hosts images and audio files

How the Layers Connect

The user opens the site in their browser (hosted on Netlify). The frontend makes requests to the Express API (hosted on Railway). The API queries the Supabase PostgreSQL database and returns data as JSON. Content editors log into Strapi to write articles, which are stored in the same database. Images and audio are uploaded to Cloudinary and referenced by URL. The clinic map is rendered using Leaflet.js with free OpenStreetMap tiles.

Technology Stack

Technology	What It Does	Why We Chose It
HTML / CSS / Vanilla JS	Frontend UI	No framework overhead; fast on low-end Android devices; specified in PRD
Node.js + Express.js	Backend API server	Full-text search built in; auth and storage bundled; open source; free tier
Supabase (PostgreSQL)	Database, auth, and storage	Full-text search built in; auth and storage bundled; open source; free tier
Strapi (Headless CMS)	Content management for editors	Non-technical team members can write and publish articles without touching code
Cloudinary	Image and audio file hosting	Reliable CDN; free tier is sufficient for MVP
Leaflet.js + OpenStreetMap	Clinic finder map	Fully open source; no API key or cost
Browser Geolocation API	User location for nearby clinics	Native browser feature; no external dependency
Netlify	Frontend hosting	Free tier; deploys automatically on git push
Render	Backend hosting	Simple Node.js deployment; supports environment variables

Frontend Architecture

How the Layers Connect

/client

/assets — images, fonts, icons

/css — stylesheets

/js — JavaScript modules (api.js, utils.js, etc.)

/pages — one HTML file per page

Pages

Page	Purpose
index.html	Landing page — mission, features overview, crisis footer
resources.html	Educational resource hub with search and filters
clinics.html	Clinic finder with map and list view
specialists.html	Specialist directory
stories.html	Community experiences library
submit-story.html	Story submission form
lifestyle.html	Lifestyle and medical intervention guides
media.html	Podcast and media directory

Page	Purpose
ngos.html	NGO and support network directory

Calling the API from the Frontend

All API calls are handled in a single `js/api.js` file. Other JS files import from it rather than making fetch calls directly.

Performance & Accessibility

- Page load target: under 3 seconds on a 4G connection.
- All pages must work on Android mobile without horizontal scrolling.
- Minimum body font size: 16px.
- WCAG 2.1 Level AA accessibility standard.
- All images must have alt text; icon-only buttons must have ARIA labels.
- Key articles cached via Service Worker for low-connectivity use
- Images or svg illustrations???

Backend Architecture

Folder Structure

/server

/routes — one file per feature (resources, clinics, etc.)

/controllers — business logic for each route

/middleware — JWT auth check, rate limiter, error handler

/models — Supabase query helpers

/utils — shared helpers (pagination, sanitisation)

/config — Supabase client setup, environment config

app.js — Express app setup

server.js — entry point

API Conventions

- All routes are prefixed with `/api/v1`
- Public routes require no authentication
- Admin routes require a valid JWT in the `Authorization: Bearer <token>` header
- All responses return JSON
- Errors return a consistent structure: `{ success: false, error: { code, message } }`
- List endpoints support `?page=` and `?limit=` for pagination (default: 20 per page)

Search

Full-text search is handled by PostgreSQL's built-in search functionality via Supabase. No external search service is needed. Search indexes are added to the `title`, `content`, and

`summary` columns of the resources table, and the `name` and `services` columns of the clinics table.

API Endpoints

Search

Method + Endpoint	Auth	Description
GET /api/v1/resources	Public	List published resources. Supports ?theme=, ?type=, ?q= filters
GET /api/v1/resources/:id	Public	Get a single resource
POST /api/v1/admin/resources	Admin	Create a resource
PUT /api/v1/admin/resources/:id	Admin	Update a resource
PATCH /api/v1/admin/resources/:id/publish	Admin	Toggle published status
DELETE /api/v1/admin/resources/:id	Admin	Delete a resource

Clinics

Method + Endpoint	Auth	Description
GET /api/v1/clinics	Public	List verified clinics. Supports ?city=, ?state=, ?free=true
GET /api/v1/clinics/nearby	Public	Proximity search. Requires ?lat=, ?lng=, ?radius=
GET /api/v1/clinics/:id	Public	Get clinic details
POST /api/v1/admin/clinics	Admin	Add a clinic
PUT /api/v1/admin/clinics/:id	Admin	Update clinic details
PATCH /api/v1/admin/clinics/:id/verify	Admin	Mark as verified

Specialists

Method/Endpoint	Auth	Description
GET /api/v1/specialists	Public	List verified specialists. Supports ?state=, ?language=

GET /api/v1/specialists/:id	Public	Get specialist profile
POST /api/v1/admin/specialists	Admin	Onboard a specialist
PATCH /api/v1/specialists/:id/availability	Admin	Toggle accepting_patients status

Community Stories

Method + Endpoint	Auth	Description
GET /api/v1/stories	Public	List approved stories. Supports ?theme= filter
POST /api/v1/stories/submit	Public	Submit a story (goes to pending queue)
GET /api/v1/admin/stories/queue	Admin	View pending submissions
PATCH /api/v1/admin/stories/:id/status	Admin	Approve, decline, or request edits

Other Endpoints

Method + Endpoint	Auth	Description
GET /api/v1/ngos	Public	NGO directory listing

GET /api/v1/media	Public	Podcast and media directory
POST /api/v1/media/suggest	Public	User suggestion (goes to moderation queue)
POST /api/v1/auth/admin/login	—	Admin login, returns JWT
GET /api/v1/admin/analytics	Admin	Page views, top searches, top visited clinics

API Endpoints

The database runs on PostgreSQL hosted by Supabase. All tables use UUID primary keys. Timestamps are stored in UTC. Row-Level Security (RLS) is enabled on all tables.

Resources

Column	Type	Notes
id	UUID	Primary key
title	TEXT	Required
content	TEXT	Full article body. Required

summary	TEXT	Short version for cards (300–500 chars)
theme	TEXT	Symptoms / Causes / Treatment / Recovery
content_type	TEXT	article / infographic / audio / myth-busting
language	TEXT	Default: en
source_url	TEXT	Link to originating clinical guideline
reviewed_by	TEXT	Name of clinical reviewer
published	BOOLEAN	Default: false
created_at	TIMESTAMPTZ	Default: now()
updated_at	TIMESTAMPTZ	Default: now()

Clinics

Column	Type	Notes

id	UUID	Primary key
name	TEXT	Required
address	TEXT	Required
city	TEXT	Required
state	TEXT	Nigerian state. Required
phone	TEXT	—
email	TEXT	—
services	TEXT[]	Array of services offered
opening_hours	TEXT	e.g. Mon–Fri, 8am–5pm
fee_range	TEXT	e.g. Free / 1,000–5,000 NGN
has_free_care	BOOLEAN	Filter flag. Default: false
rating	DECIMAL(2,1)	Aggregate rating 1.0–5.0

verified	BOOLEAN	Admin-verified. Default: false
created_at	TIMESTAMPTZ	Default: now()

Specialists

Column	Type	Notes
id	UUID	Primary key
full_name	TEXT	Required
credentials	TEXT	e.g. MBBS, MRCPsych. Required
specialty	TEXT	Areas of expertise. Required
languages	TEXT[]	Languages spoken
whatsapp	TEXT	Optional
email	TEXT	Optional
phone	TEXT	Optional

fee_range	TEXT	Consultation fee range
accepting_patients	BOOLEAN	Default: true
city	TEXT	Practice location
state	TEXT	Nigerian state
verified	BOOLEAN	Admin-verified. Default: false
created_at	TIMESTAMPTZ	Default: now()

Community_stories

Column	Type	Notes
id	UUID	Primary key
narrative	TEXT	Full story text. Required
timeline	TEXT	When PPD started/ended

what_helped	TEXT	What aided recovery
what_hindered	TEXT	What made things harder
location	TEXT	Optional city/state
is_anonymous	BOOLEAN	Default: true
author_name	TEXT	Null if anonymous
theme_tags	TEXT[]	Thematic categorisation
status	TEXT	pending / approved / declined. Default: pending
created_at	TIMESTAMPTZ	Default: now()

Other tables

Table	Purpose
ngos	NGO directory - name, mission, services, geographic reach, contact info

media_resources	Podcast and YouTube directory - title, platform, language, link
clinic_reviews	Moderated user-submitted clinic ratings and reviews
admin_users	Admin accounts, linked to Supabase Auth

Authentication & Security

Admin Authentication

- Admin accounts are created in Supabase Auth (email + password)
- On login, the API returns a signed JWT valid for 8 hours
- All `/admin/*` routes check for a valid JWT before proceeding
- Three roles: super_admin, content_editor, moderator
- There are no public user accounts in MVP — the platform is publicly browsable without login

Security Practices

- HTTPS enforced across all environments
- All form inputs are sanitised server-side to prevent SQL injection and XSS attacks
- Rate limiting is applied to all POST endpoints
- CORS is configured to allow only the production frontend domain

- All secrets (API keys, JWT secret) are stored as environment variables — never in source code
- Supabase Row-Level Security enabled on all tables

Privacy

- No third-party analytics or advertising trackers on the platform
- Community story anonymisation is permanent and irreversible once published
- User location data (for clinic proximity search) is never stored
- Privacy policy is linked in the footer on every page

Repository & Workflow

Structure

/bloom-after

/client — frontend

/server — backend API

/docs — TRD and other documentation

/scripts — seed scripts and helpers

README.md

CONTRIBUTING.md

.env.example — template for required environment variables

.gitignore

Branching Strategy

Branch	Purpose
prod	Stable, production-ready code. Only the Engineering Lead merges here.
dev	Integration branch. All features are merged here first for testing.
feature/*	Individual feature work. Branch from dev. e.g. feature/clinic-finder
fix/*	Bug fixes. e.g. fix/map-loading-error

Pull Request Rules

- All PRs target `dev`, never `prod` directly
- At least one peer review approval is required before merging to `dev`
- The Engineering Lead reviews and approves all merges from `dev` into `prod`
- PR descriptions must include: what changed, how to test, and screenshots for any UI changes

Environment Variables

Variable	Description

SUPABASE_URL	Supabase project URL
SUPABASE_ANON_KEY	Public key used by the frontend
SUPABASE_SERVICE_KEY	Private key used by the backend only — never expose this
JWT_SECRET	Secret used to sign admin tokens
CLOUDINARY_CLOUD_NAME	Cloudinary account name
CLOUDINARY_API_KEY	Cloudinary API key
CLOUDINARY_API_SECRET	Cloudinary secret — backend only
STRAPI_URL	Strapi CMS base URL
PORT	Server port. Default: 3000

Repository & Workflow

Week	Focus	Engineering Lead	Team
Week 1	Foundation	TRD finalised, repo structure defined, PR rules set, DB schema reviewed	Repo setup, design system, landing page built and deployed
Week 2	Core Features	PR reviews for resource hub, clinic finder, specialist directory	Resource hub, clinic finder with map, specialist directory
Week 3	Community & Admin	PR reviews for submissions, moderation dashboard, remaining directories	Story submission form, admin dashboard, podcast and NGO directories
Week 4	Polish & Launch	Final prod merge, performance and accessibility review	Responsive QA, seed data, performance optimisation, live deployment