

AquaNexus MVP Development Plan

Team: Caleb, Shelmith, Joyce
Stack: React + Vite (Frontend), Node.js + Express + PostgreSQL (Backend), Tailwind CSS
Duration: 5 Weeks

Phase 0: Setup & Landing Page Foundation (Week 1)

Goal: Initialize project, configure PostgreSQL, and build static landing page.

Todos

Task	Assignee	Details
Git Repo Setup	All	Create <code>aqua-nexus</code> repo with <code>main/develop</code> branches.
Backend Setup	Caleb	Initialize Node.js project. Install <code>pg</code> , <code>express</code> , <code>dotenv</code> .
PostgreSQL Setup	Joyce	Create ElephantSQL/Neon DB instance. Share connection string.
Frontend Setup	Shelmith	<code>npm create vite@latest</code> + Tailwind CSS configuration.
Landing Page Components	All	Divide sections: Caleb - Navbar/Footer, Joyce - Hero/HowItWorks, Shelmith - ServiceCards/TrustBadges

Database Schemas

```
-- users.sql
CREATE TABLE users (
  id SERIAL PRIMARY KEY,
  email VARCHAR(255) UNIQUE,
  password_hash TEXT,
  role VARCHAR(20) CHECK (role IN ('user', 'provider', 'admin')),
  created_at TIMESTAMP DEFAULT NOW()
);

-- providers.sql
CREATE TABLE providers (
  id SERIAL PRIMARY KEY,
  user_id INT REFERENCES users(id),
  service_type VARCHAR(50) CHECK (service_type IN ('rwh', 'borehole')),
  license_number VARCHAR(100),
```

```
service_areas TEXT[],  
UNIQUE(user_id)  
);
```

Phase 1: Core Backend - Auth & Profiles (Week 2)

Goal: Secure authentication and profile management.

Todos

Task	Assignee	Details
User Registration API	Caleb	POST <code>/api/auth/register</code> with role-based validation.
Provider Registration	Shelmith	Extend registration to capture <code>service_type</code> and <code>license_number</code> .
JWT Authentication	Joyce	Implement token generation/validation middleware.
Profile APIs	Caleb+Joyce	GET/PUT <code>/api/profiles/me</code> for users/providers.
Password Security	Shelmith	Integrate <code>bcrypt</code> for hashing.

Phase 2: Frontend Auth & Dashboard Shell (Week 3)

Goal: Connect frontend to auth APIs and build role-specific dashboards.

Todos

Task	Assignee	Details
Login/Register UI	Joyce	Build forms with role toggle (User/Provider).
Auth Context	Caleb	Create <code>AuthContext</code> to manage JWT and user state.
User Dashboard	Shelmith	<code>/dashboard/user</code> with "Post Project" CTA and active projects.
Provider Dashboard	Joyce	<code>/dashboard/provider</code> with available projects and bid history.

Protected Routing Caleb Restrict routes based on role (e.g., only providers can bid).

Phase 3: Project Posting & Bidding System (Week 4)

Goal: Implement core marketplace functionality.

Todos

Task	Assignee	Details
Project Posting API	Caleb	POST <code>/api/projects</code> with geolocation (lat/long).
Project Form UI	Shelmith	Multi-step form with cost estimator and permit guidance.
Bidding API	Joyce	POST <code>/api/projects/:id/bids</code> with validation (min_bid = project.budget * 0.7).
Bid Comparison UI	Joyce+Caleb	Side-by-side provider cards with ratings and cost breakdown.
Real-Time Updates	Shelmith	Implement WebSockets/Socket.IO for new bid notifications.

Phase 4: Escrow Payments & Reviews (Week 5)

Goal: Secure transactions and reputation system.

Todos

Task	Assignee	Details
M-Pesa Integration	Caleb	Use Safaricom API for escrow deposits.
Escrow Logic	Joyce	API to hold/release funds based on project milestones.
Payment UI	Shelmith	Build escrow dashboard with payment status and release controls.
Review System	All	Caleb - POST <code>/api/reviews</code> , Joyce - Star rating component, Shelmith - Review moderation

Compliance
Check

All

Verify alignment with Kenyan financial regulations.

Post-MVP Roadmap

1. Geolocation Matching (PostGIS integration)
 2. Admin Analytics Dashboard (Fraud detection, revenue reports)
 3. SMS Notifications (Twilio/Africa's Talking for bid alerts)
 4. Mobile App (React Native)
-

Workflow & Best Practices

1. Git Strategy:
 - Daily `git pull origin develop`
 - PRs require 2 approvals before merging
 - Test coverage for critical APIs
 2. Tools:
 - DB GUI: PGAdmin/TablePlus
 - API Testing: Postman (Shared workspace)
 - UI Prototyping: Figma (Link)
 3. Weekly Sync:
 - Mondays: Task assignment
 - Fridays: Demo progress + blockers
-

Critical Learning Resources

1. PostgreSQL with Node.js: <https://node-postgres.com/>
2. JWT Auth: <https://auth0.com/blog/complete-guide-to-react-user-authentication/>
3. M-Pesa API: <https://developer.safaricom.co.ke>

This plan prioritizes your water solution marketplace's unique needs while leveraging the team's strengths. Adapt as needed! 🚀