Lovable LLM Frontend Generation Prompt

AfroKen LLM - Citizen Service Copilot Frontend

📝 COMPLETE FRONTEND GENERATION PROMPT

Objective: Generate a production-ready, accessible, multilingual frontend for AfroKen LLM using React/Next.js, TypeScript, and Tailwind CSS.

Design Standard: WCAG 2.1 AA compliant, mobile-first, responsive (320px-1920px).

Target Users: 53 million Kenyans across all literacy levels, digital access levels, and abilities.

🎯 PROJECT REQUIREMENTS

Project Name

AfroKen LLM™ - Citizen Service Copilot for Inclusive Governance

Tech Stack

- **Framework:** React 18+ with TypeScript
- **Styling:** Tailwind CSS with custom design tokens
- **Build Tool:** Next.js 14+ for SSR/SSG
- **State Management:** Zustand
- **Data Fetching:** React Query (TanStack Query)
- **UI Components:** Shadcn/ui + custom components
- **Icons:** Lucide React (line-based, friendly style)
- **Charts:** Recharts (dashboards)
- **Maps:** Mapbox GL (geospatial)
- **Forms:** React Hook Form + Zod validation

Design System Reference

Use the following color palette and spacing system as foundation:

```
PRIMARY COLORS:
- Kenyan Green: #1B5E20 (primary actions, trust)
- Kenyan Teal: #00897B (secondary, progress)
- Kenyan Gold: #FFB300 (warnings, caution)
- Kenyan Red: #C41C3B (errors, critical alerts)

SEMANTIC COLORS:
- Success: #2E7D32
- Error: #D32F2F
- Warning: #F57F17
- Info: #0288D1
- Neutral Gray: #616161

TYPOGRAPHY:
- Headings: Poppins (28-32px H1, 22-24px H2, 18-20px H3)
- Body: Roboto (16px regular, 1.6 line height)
- UI/Buttons: Inter (16px bold)
- Multilingual: Noto Sans (full Unicode support)

SPACING (8px grid):
- XS: 4px, SM: 8px, MD: 16px, LG: 24px, XL: 32px, 2XL: 48px, 3XL: 64px

RESPONSIVE BREAKPOINTS:
- Mobile: 320px (feature phones)
- Tablet: 768px
- Desktop: 1024px (large desktop: 1280px+)
```

📱 5 MAIN INTERFACE PLATFORMS TO BUILD

1️⃣ WEB/DESKTOP INTERFACE (React/Next.js)

Primary Interface for citizen-facing access

Layout Structure

```
┌─────────────────────────────────────────────────┐
│ Header: Logo | Nav | Language Selector | Menu   │
├──────────────┬──────────────────────────────────┤
│  SIDEBAR NAV  │    MAIN CHAT AREA          │
│ (280px wide)  │  (70% of screen width)     │
│ • Home        │                            │
│ • Services    │ Message history (scroll)   │
│ • My Cases    │ Citation badges            │
│ • Help        │ Quick actions              │
│ • Settings    │                            │
│               │ Input: Text/Voice/Attach   │
└──────────────┴──────────────────────────────────┘
```

Key Components Needed

1. **Top Navigation Bar**
   - Logo (AfroKen with text)
   - Navigation links (Home, Services, Cases, Help)
   - Language selector (En, Sw, Sheng dropdown)
   - User menu (avatar, logout)
   - Sticky on scroll
2. **Left Sidebar Navigation**
   - Collapsible (icon only when collapsed)
   - Active state highlighting (Kenyan Green)
   - Smooth transitions
   - Responsive (hide on mobile)
3. **Main Chat Interface**
   - Message history area (scrollable, infinite scroll up)
   - User messages (right-aligned, green bubbles 🟩 #1B5E20)
   - Bot messages (left-aligned, gray bubbles with border)
   - System messages (center-aligned, blue background)
   - Citation badges (blue info badges with source)
   - Typing indicator (three animated dots)
   - Timestamp on messages (relative: "2 mins ago" or absolute: "14:32")
   - Quick action chips (inline suggestions)

4. **Input Area**
    - Text input (multiline, auto-expand to 5 lines)
    - Microphone button (voice input)
    - Attachment button (files, images)
    - Send button (arrow icon)
    - Show helper text: "Speak in Kiswahili, Sheng, or English"
5. **Service Panel (Right side, desktop only)**
    - Service details
    - Quick access buttons
    - Additional resources

Features

- ✓ Real-time message streaming
- ✓ Voice note transcription display
- ✓ Government verification badges
- ✓ Auto-save draft messages
- ✓ Search message history
- ✓ Export conversation as PDF
- ✓ Dark mode support
- ✓ Accessibility: Full keyboard nav, screen reader optimized

Accessibility Requirements

- ✓ ARIA labels on all buttons/interactive elements
- ✓ Focus indicators visible (3px outline)
- ✓ Semantic HTML (header, nav, main, section)
- ✓ Screen reader announcements for new messages
- ✓ Keyboard shortcuts (Enter to send, Shift+Enter for newline)
- ✓ High contrast mode toggle

2️⃣ MOBILE/WHATSAPP-LIKE INTERFACE (React/Next.js + React Native/Flutter)

Touch-optimized for 480px+ devices

Layout Structure

```
┌────────────────────┐
│ ◀ Chat Title ⋯ Menu │ (Compact header)
├────────────────────┤
│                    │
│ Full-screen messages │
│ (chat history)       │
│                    │
│ Quick actions inline │
│                    │
├────────────────────┤
│ [📎] [Type...] [🎤] [→]│ (Input bar)
├────────────────────┤
│ 🏠 💬 📊 ❓ ⚙️         │ (Bottom nav)
└────────────────────┘
```

Key Components Needed

1. **Compact Header**
   - Back button
   - Page title
   - Options menu (three-dot)
   - No top nav bar (space-efficient)
2. **Full-Width Message Stream**
   - User messages right-aligned, green
   - Bot messages left-aligned, gray with avatar
   - System messages center-aligned, blue
   - Message timestamps (compact: "14:32")
   - Swipe to reply (optional, if implementing)
3. **Input Bar**
   - Text input field (full width, 48px height min)
   - Attachment button (left side)
   - Voice record button (left side, press to record)
   - Send button (right side)
   - Character counter (optional)
4. **Bottom Navigation Bar**
   - 5 tabs: Home, Chat (active), Cases, Help, Menu

   - Icons + labels
   - Active indicator (underline, green)
   - Sticky on scroll
   - Safe area padding (notches)

5. **Service Shortcuts Page**
   - Grid of service cards (2 columns)
   - Quick access to: NHIF, KRA, ID, Business, County
   - Search bar at top
6. **Quick Actions Sheet (Bottom Sheet)**
   - Swipe up from bottom
   - More options for current service
   - Suggestions for next steps

Features

- ✓ Infinite scroll (load more messages up)
- ✓ Message search
- ✓ Share message
- ✓ Message reactions (emoji quick reactions)
- ✓ Draft auto-save
- ✓ Offline message queue (send when online)
- ✓ Haptic feedback on button press
- ✓ Status bar safe area awareness

Responsive Behavior

- Hide sidebar on devices <1024px
- Stack all content vertically
- Maximize touch target sizes (48x48px minimum)
- Optimize images for mobile bandwidth
- Service grid: 1 column (mobile), 2 columns (tablet), 3 columns (desktop)

3️⃣ USSD/SMS INTERFACE (Text-only, API-driven)

For feature phones with no smartphone/internet

Menu Structure

```
MAIN MENU (Text)
────────────────
1. National ID Services
2. NHIF Health Insurance
3. KRA Taxes
4. Business Services
5. Track Application
6. Help & Support
98. Change Language
99. Exit

Enter choice: _
```

Implementation Approach

1. **Backend-driven:** Text responses, no UI components
2. **Stateless:** Each interaction is a single USSD session
3. **Simple navigation:** Numeric choices (1-9, 0 for back)
4. **6 items max per menu:** Cognitive load management
5. **Plain text formatting:** No emojis, symbols, or special characters
6. **Auto-wrap text:** Max 160 characters per line (SMS standard)

Key Flows Needed

1. **Service Selection Menu**

```
  Main menu with 6 services
  User selects: 1 (ID Services)
  Sub-menu shows:
  1. How to apply
  2. Renew existing
  3. Check status
  0. Back
```

2. **Information Display**

```
Service name: NATIONAL ID
Requirements: Age 18+, Birth cert
Where: Huduma Centre
Cost: FREE
Time: 30 minutes
Next: Visit huduma.go.ke
1. Find center
2. More info
0. Back
```

3. **Status Check**

```
Enter ref number: REF-2025-123456
[Processing...]
Status: PROCESSING
Submitted: 10 Nov 2025
Next step: Verification (2-3 days)
1. More details
2. Contact support
0. Back
```

4. **Language Selection**

```
CHANGE LANGUAGE
1. English
2. Kiswahili
3. Sheng
0. Back
Enter choice: _
```

Technical Implementation

- Use Africa's Talking or Twilio USSD API
- Store session state in Redis (expires after 10 mins)
- Menu navigation via simple string parsing
- Return plain text responses
- No images, videos, or multimedia
- Character limit: 160 chars per response (SMS width)

4️⃣ VOICE INTERFACE (Whisper ASR + Coqui TTS)

For visually impaired and low-literacy users

Layout Structure

```
┌─────────────────────────────┐
│ 🎤 VOICE ASSISTANT          │
├─────────────────────────────┤
│                             │
│     ○ ○ ○                   │
│   ○     ○                   │
│  ○  🎤  ○                   │
│   ○    ○                    │
│     ○ ○ ○                   │
│                             │
│ ~ ~~~ ~~~ ~~~~~ ~~~ ~ ~~~   │
│ (Animated waveform)         │
│                             │
│ Listening... (Please speak) │
│                             │
├─────────────────────────────┤
│ "Nataka kujua NHIF status"  │
│ [✓ Correct] [✗ Try] [Edit]  │
│                             │
│ [Continue] [Type Instead]   │
│ 🔊 Volume: [██████████]     │
└─────────────────────────────┘
```

Key Components Needed

1. **Microphone Button (Central)**
   - Large circular button (120x120px)
   - Glowing animation when listening
   - Tap to start/stop recording
   - Visual feedback (color change, animation)
2. **Waveform Visualization**
   - Real-time audio waveform display
   - Animated bars showing audio input
   - Updates every 100ms
   - Green while listening, red if error
3. **Transcription Display**
   - Shows recognized text in real-time
   - Confidence percentage (optional)

    - Allow user to confirm/edit before sending
    - "Try again" if recognition failed
4. **Response Delivery**
    - Text-to-speech (Coqui TTS)
    - Play button to replay response
    - Automatic playback on completion
    - Volume control slider
5. **Fallback to Text**
    - Option to type instead of voice
    - Text input appears if voice fails
    - Smooth transition between modes

Features

- ✓ Kenyan-accented speech recognition (Whisper fine-tuned)
- ✓ Auto-slower speech delivery for clarity
- ✓ Support Swahili, English, Sheng
- ✓ Background noise filtering
- ✓ Offline capability (local Whisper model)
- ✓ SMS follow-up option ("Send to WhatsApp")
- ✓ Accessibility: Screen reader compatible, keyboard nav

Audio Handling

- Record: Use Web Audio API or react-mic
- Process: Send to Whisper API/model
- TTS: Coqui TTS or Google Cloud TTS
- Stream response: Chunk-based audio playback
- Fallback: Text-only if audio fails

5️⃣ OFFICER DASHBOARD (Analytics & Admin)

For government ministry/county staff

Layout Structure (National Overview)

```
┌─────────────────────────────────────────────────────────────┐
│ AfroKen Admin | National | County | Service | Audit │
├─────────────────────────────────────────────────────────────┤
│                                                     │
│ KEY METRICS CARDS (4 columns)                       │
│ ┌─────────┐ ┌─────────┐ ┌─────────┐ ┌─────────┐     │
│ │ 58,234  │ │ 2.1M    │ │ 2.3 min │ │ 94.2%   │     │
│ │ Queries │ │ Citizens│ │ Response│ │ Accur.  │     │
│ └─────────┘ └─────────┘ └─────────┘ └─────────┘     │
│                                                     │
│ TOP SERVICES TABLE                                  │
│ ┌───────────┬─────────┬──────────┬──────────┐       │
│ │ Service   │Queries  │Complete  │Sentiment │       │
│ ├───────────┼─────────┼──────────┼──────────┤       │
│ │ NHIF Status │ 18,234 │  87%    │ ✓ Good   │       │
│ │ KRA Tax     │ 14,092 │  81%    │ ⚠ Warn   │       │
│ │ National ID │ 12,567 │  92%    │ ✓ Good   │       │
│ └───────────┴─────────┴──────────┴──────────┘       │
│                                                     │
│ GEOGRAPHIC HEATMAP (Mapbox visualization)           │
│ [Interactive map showing query volume by county]    │
│                                                     │
│ SENTIMENT TREND (30 days)                           │
│ [Recharts line chart: Positive, Neutral, Negative]  │
│                                                     │
│ TRENDING QUESTIONS                                  │
│ 1. "How to check NHIF status?" (2,847)              │
│ 2. "NHIF renewal process" (1,923)                   │
│ [Show 5, "See all"]                                 │
│                                                     │
└─────────────────────────────────────────────────────────────┘
```

Key Components Needed

1. **Top Navigation Tabs**
    - National (overview)
    - County (select county from dropdown)
    - Service (detailed service performance)
    - Audit (logs, drift detection)
    - Settings (manage integrations)
2. **Metrics Cards (4-up grid)**
    - Daily Queries

- Unique Citizens Served
- Avg Response Time
- Accuracy Score
- Real-time updates via WebSocket

3. **Performance Table**
    - Sortable columns: Service, Queries, Completion %, Sentiment
    - Color-coded sentiment (green, amber, red)
    - Row details expandable
    - Export to CSV button
4. **Geographic Heatmap**
    - Mapbox GL integration
    - County boundaries color-coded by query volume
    - Hover to see county name + metrics
    - Interactive (zoom, pan)
    - Legend showing color mapping
5. **Charts & Visualizations**
    - Sentiment trend line chart (Recharts)
    - Service distribution pie chart
    - Response time box plot
    - Query volume by hour (heatmap)
6. **Bottleneck Detection**
    - Alert cards showing issues
    - "Birth Certificate: 76% complete rate"
    - Recommended actions
    - Color-coded by severity (red, amber, green)
7. **Trending Questions Feed**
    - List of top searches
    - Count/percentage
    - Click to drill down
    - "See all" pagination

8. **Audit Logs Section**
    - Searchable table of all system events
    - Timestamp, user, action, status
    - Filter by action type
    - Export to CSV

Features

- ✓ Real-time data via WebSocket
- ✓ Role-based access control (RBAC)
- ✓ Data export (CSV, PDF)
- ✓ Custom date range selection
- ✓ Drill-down analytics (click county → see details)
- ✓ Refresh button (manual + auto-refresh option)
- ✓ Dark mode support
- ✓ Performance optimized (lazy load charts)

County Dashboard Variant

- Filter to single county
- Different metrics (local services, wait times, feedback)
- Huduma Centre performance table
- Local language usage breakdown
- Regional customizations

🎨 GLOBAL DESIGN REQUIREMENTS

Design System Implementation

1. **Tailwind CSS Configuration**

```
javascript
```

```
// tailwind.config.js should include:
- Custom colors (Kenyan Green, Teal, Gold, Red)
- Custom typography (Poppins, Roboto, Inter, Noto Sans)
- Custom spacing (8px grid)
- Custom shadows (sm, md, lg)
- Dark mode support (class strategy)
- Extend with design tokens
```

2. **CSS Variables Approach**

```css
:root {
  --color-primary: #1B5E20;
  --color-secondary: #00897B;
  --color-accent: #FFB300;
  --color-error: #D32F2F;
  --spacing-sm: 0.5rem;
  --spacing-md: 1rem;
  --spacing-lg: 1.5rem;
  --font-primary: 'Poppins', sans-serif;
  --font-body: 'Roboto', sans-serif;
}
```

3. **Component Structure**

```
/src
├── components/
│   ├── common/        (Button, Input, Card, etc.)
│   ├── layout/        (Header, Sidebar, Footer)
│   ├── chat/          (ChatMessage, ChatInput, etc.)
│   ├── forms/         (FormField, FormGroup, etc.)
│   ├── dashboard/     (MetricCard, Chart, Table, etc.)
│   └── voice/         (VoiceButton, Waveform, etc.)
├── pages/             (page routes)
├── styles/            (global CSS, design tokens)
├── lib/               (utilities, hooks, constants)
└── types/             (TypeScript definitions)
```

Color Implementation

- Use Tailwind color classes: `bg-emerald-700`, `text-teal-600`
- Define custom colors in `tailwind.config.js` for exact hex values
- Maintain color consistency across all interfaces
- Support dark mode with `dark:` prefix

Typography Implementation

- Use custom font classes: `font-poppins`, `font-roboto`, `font-inter`
- Respect type scale: `text-3xl` (H1), `text-2xl` (H2), `text-base` (body)
- Line height: `leading-relaxed` (1.6) for body text
- Letter spacing: `tracking-tight` for headings

Spacing Implementation

- Use Tailwind spacing scale: `p-4` (MD), `m-6` (LG), `gap-4` (spacing between items)
- Consistent padding for components: `px-4 py-3` for buttons
- Responsive spacing: `md:p-8` for larger screens

♿ ACCESSIBILITY REQUIREMENTS (WCAG 2.1 AA)

Mandatory Features

1. **Color Contrast**
   - All text: 4.5:1 contrast minimum (AA standard)
   - Large text (18pt+): 3:1 contrast minimum
   - UI components: 4.5:1 contrast for borders/outlines
   - Test with: WebAIM Contrast Checker, Coblis (colorblind simulator)
2. **Keyboard Navigation**
   - All interactive elements accessible via Tab key
   - Focus indicator visible (3px outline, blue or green)
   - Logical tab order (top to bottom, left to right)
   - Escape key closes modals/dropdowns
   - Enter key submits forms
   - Space bar activates buttons
3. **Screen Reader Support**
   - Semantic HTML: `<header>`, `<nav>`, `<main>`, `<section>`, `<aside>`
   - ARIA labels: `aria-label`, `aria-labelledby`, `aria-describedby`
   - ARIA roles: `role="button"`, `role="navigation"`, `role="banner"`
   - Live regions: `aria-live="polite"` for chat messages, alerts
   - Image alt text: Descriptive, not "image of..." or "picture"

- Form labels: Associated with inputs via `htmlFor` attribute

4. **Focus Management**
   - Focus visible on all interactive elements
   - Focus indicator: Solid outline (3px, high contrast)
   - No focus traps (Escape key works)
   - Skip to main content link visible on first Tab

5. **Mobile Accessibility**
   - Touch targets: 48x48px minimum (44px with padding acceptable)
   - Pinch-to-zoom enabled (no user-scalable="no")
   - Readable without horizontal scrolling
   - Sufficient space between touch targets (8px minimum)

6. **Multilingual Accessibility**
   - Language declared in HTML: `<html lang="sw">`
   - Language changes announced to screen readers
   - Proper diacritics: ā, ē, ī, ō, ū (Swahili)
   - No all-caps text (harder to read for dyslexia)
   - Line height 1.6+ for body text

Accessibility Testing Checklist

□ Test with NVDA (Windows) or JAWS
□ Test with VoiceOver (Mac/iOS)
□ Test with keyboard-only navigation
□ Run axe DevTools or WAVE browser extension
□ Check color contrast with WebAIM
□ Test with colorblind simulator (Coblis)
□ Test on mobile devices (touch targets, zoom)
□ Verify form labels & error messages linked
□ Check for focus traps
□ Test with screen magnification (200%)

🌍 INTERNATIONALIZATION (i18n)

Language Support (MVP)

1. **English** (Official language, fallback)

2. **Kiswahili** (70% population, DEFAULT language)

3. **Sheng** (Youth, colloquial, code-switched)

Implementation Requirements

1. **Language File Structure**

```
/public/locales/
├── en/
│   ├── common.json
│   ├── chat.json
│   ├── services.json
│   └── dashboard.json
├── sw/
│   ├── common.json
│   └── ...
└── sheng/
    ├── common.json
    └── ...
```

2. **i18n Library**
    - Use: next-i18next or i18next
    - Configure: Language detection, route-based language, localStorage persistence
    - Fallback chain: User preference → Browser locale → English
3. **Language Switcher**
    - Dropdown in header showing: English | Kiswahili | Sheng
    - Persist selection in localStorage + user profile
    - Change language instantly (no page reload)
    - Show flag + language name (not flags alone)
4. **Translation Considerations**
    - **Swahili strings are 10-15% longer:** Adjust UI layout accordingly
    - **Sheng uses code-switching:** "Nataka kujua my NHIF status" (Sw+En mixed)
    - **Avoid machine translation:** Use professional Kenyan linguists
    - **Test RTL later:** Not needed for current languages (all Latin-based)
    - **Decimal format:** "," or "." locale-specific
    - **Date format:** "14 Novemba 2025" (Swahili)
5. **Key Strings to Translate**
    - Navigation labels

- Button text
- Form labels & placeholders
- Error messages
- Success messages
- Help text & tooltips
- Chat responses (from API)
- Dashboard labels

Implementation Example

```typescript
// Using next-i18next
import { useTranslation } from 'next-i18next'

export const ChatHeader = () => {
  const { t } = useTranslation('chat')
  return (
    <header>
      <h1>{t('title')}</h1> {/* "Chat with AfroKen" or "Chafu na AfroKen" */}
    </header>
  )
}
```

📊 DATA & API INTEGRATION

Mock Data Structure

Create realistic mock data for:

- 5,000+ sample services (NHIF, KRA, ID, Business, County)
- 1,000+ recent queries with varied responses
- 100+ trending questions
- 47 counties with geographic data
- 10+ sample government agencies
- Dashboard metrics (daily query volume, response times, accuracy)

API Endpoints to Mock (using MSW or json-server)

```
GET /api/services - List all government services
GET /api/services/:id - Service details & guidance
GET /api/chat - Message history
```

```
POST /api/chat - Send message
GET /api/chat/:id/status - Check application status
GET /api/dashboard/metrics - National overview
GET /api/dashboard/counties/:id - County-specific metrics
GET /api/dashboard/services/:id - Service performance
POST /api/voice/transcribe - Voice transcription
POST /api/voice/synthesize - Text-to-speech
```

Chat Response Simulation

Create chat message components that simulate:

- Citations with source badges (Ministry of Health, eCitizen, etc.)
- Step-by-step guidance with numbered steps
- Status tracking with progress indicators
- Quick action buttons ("Find Center", "Book Appointment", etc.)
- Government verification badges (✓ Verified)
- Confidence indicators ("Based on verified eCitizen data")

📱 RESPONSIVE BEHAVIOR

Mobile (320px-480px)

□ Hide sidebar navigation
□ Stack all content vertically
□ Full-width input field
□ Bottom navigation bar (5 icons)
□ Touch targets 48x48px minimum
□ Service grid: 1 column
□ Font size: 16px minimum for inputs (prevent zoom on iOS)

Tablet (768px-1024px)

□ Sidebar visible (narrower, collapsed to icons)
□ Two-column layout (sidebar + main content)
□ Service grid: 2 columns
□ Dashboard: 2 columns for cards

Desktop (1280px+)

□ Full sidebar navigation
□ Three-column layout (sidebar + main + service panel)
□ Service grid: 3 columns
□ Dashboard: 4 columns for metric cards
□ Wider content: Max 1200px container width for readability

Common Responsive Patterns

```jsx
{/* Using Tailwind responsive classes */}
<div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-4">
  {/* Mobile: 1 column | Tablet: 2 cols | Desktop: 3 cols */}
</div>

<nav className="hidden lg:flex">
  {/* Hide sidebar on mobile */}
</nav>

<button className="p-4 md:px-6 lg:px-8">
  {/* Responsive padding */}
</button>
```

⚡ PERFORMANCE OPTIMIZATION

Core Web Vitals Targets

- **LCP (Largest Contentful Paint):** < 2.5 seconds
- **FID (First Input Delay):** < 100 milliseconds
- **CLS (Cumulative Layout Shift):** < 0.1
- **Test on:** 3G (1.5Mbps), 4G (4Mbps), WiFi

Optimization Strategies

1. **Image Optimization**
   - Use Next.js Image component (automatic optimization)
   - WebP format with JPEG fallback
   - Responsive images (srcset for different sizes)
   - Lazy load images below fold
2. **Code Splitting**
   - Dynamic imports for page-specific code
   - Route-based code splitting in Next.js
   - Load dashboard charts only when needed
3. **Caching**
   - Service Worker for offline capability
   - Browser caching headers (Cache-Control)

   - React Query stale-while-revalidate strategy

4. **Bundle Size**
   - Target: < 500KB JavaScript total
   - Monitor with bundlephobia or import-cost extension
   - Minimize large dependencies (use tree-shaking)

Lighthouse Audit

- Target score: ≥ 90 (Performance, Accessibility, Best Practices, SEO)
- Test in production build mode
- Simulate 3G throttling in Chrome DevTools

🔐 SECURITY & PRIVACY

Implementation Checklist

□ HTTPS only (no HTTP)
□ CSP headers (Content Security Policy)
□ No localStorage for sensitive data (use httpOnly cookies)
□ CSRF protection on forms
□ Sanitize user input (prevent XSS)
□ No API keys in frontend code (env variables)
□ Remove console.log in production

Kenya Data Protection Act Compliance

□ Privacy policy in English & Swahili
□ Clear consent before collecting data
□ Data minimization (collect only what's needed)
□ Encryption (TLS 1.3 for transit, AES-256 for rest)
□ User right to deletion
□ Transparent data usage

📋 DELIVERABLES CHECKLIST

Components to Build

□ Button (Primary, Secondary, Tertiary, Icon, Loading states)
□ Input (Text, Textarea, Select, Checkbox, Radio, Date, File)
□ Card (Service, Status, Citation, Alert)
□ Chat (UserMessage, BotMessage, SystemMessage, Citation)
□ Navigation (TopNav, Sidebar, BottomNav, Breadcrumb)
□ Forms (FormField, FormGroup, validation)

☐ Modals (Dialog, Alert, Confirmation)

☐ Toasts (Success, Error, Warning, Info)

☐ Loaders (Skeleton, Spinner, Progress bar)

☐ Tables (Sortable, Filterable, with pagination)

☐ Charts (LineChart, BarChart, PieChart, Heatmap)

☐ Maps (Mapbox integration for county data)

Pages to Build

☐ Landing page (Hero, quick services, CTA)

☐ Chat page (Main conversation interface)

☐ Services explorer (Grid of all services)

☐ Service details page (Steps, guides, resources)

☐ Status tracking page (Application progress)

☐ Voice interface page (Microphone, transcription, TTS)

☐ Settings page (Language, preferences, profile)

☐ Dashboard - National (Metrics, charts, trends)

☐ Dashboard - County (Regional performance)

☐ Dashboard - Audit (Logs, system events)

☐ 404 page (Not found)

☐ Loading page (Initial load state)

Features to Implement

☐ Real-time message streaming

☐ Voice recording & transcription (Whisper)

☐ Text-to-speech response (Coqui)

☐ Auto-save draft messages

☐ Message search

☐ Offline message queueing

☐ Language switching (instant update)

☐ Dark mode toggle

☐ User preferences (language, theme)

☐ Export conversation as PDF

☐ Share conversation link

☐ Message reactions (emoji)

☐ Quick actions (chips/buttons)

☐ Geographic data visualization (Mapbox)

☐ Analytics dashboard (Recharts)

☐ Real-time metrics (WebSocket)

Testing to Conduct

☐ Unit tests (components, utils, hooks)

☐ Integration tests (page flows)

- □ E2E tests (critical user journeys)
- □ Accessibility tests (axe, WAVE, screen readers)
- □ Performance tests (Lighthouse, WebPageTest)
- □ Cross-browser testing (Chrome, Firefox, Safari, Edge)
- □ Mobile testing (iOS Safari, Chrome Android)
- □ Network throttling tests (3G, 4G, WiFi, offline)
- □ Responsive design tests (320px-1920px)
- □ i18n tests (English, Swahili, Sheng)

🎯 PRIORITY & PHASED ROLLOUT

Phase 1 (MVP - Week 1-2)

Minimum Viable Product - Core Functionality

- □ Landing page + hero section
- □ Chat interface (web + mobile)
- □ Basic message sending/receiving (mock API)
- □ Service explorer (grid view)
- □ Language selector (English, Swahili)
- □ Basic responsive design
- □ WCAG 2.1 AA accessibility

Phase 2 (Week 3-4)

Enhanced Features

- □ Voice input (Whisper integration)
- □ Voice output (TTS integration)
- □ Service details page (step-by-step guidance)
- □ Status tracking page
- □ Dark mode support
- □ Message search & history
- □ Quick actions/suggestions
- □ Sheng language support

Phase 3 (Week 5-6)

Dashboards & Analytics

- □ Officer dashboard (national overview)
- □ County dashboard variant
- □ Metrics cards & real-time updates
- □ Geographic heatmap (Mapbox)
- □ Performance charts (Recharts)
- □ Trend analysis

□ Bottleneck detection alerts

Phase 4 (Beyond)

Advanced Features

□ USSD/SMS text interface

□ Offline-first capability (Service Worker)

□ PWA installability

□ Advanced analytics & reporting

□ Admin management interface

□ Integration with government APIs

□ Multi-county support

□ Advanced search & filters

🚀 FINAL INSTRUCTIONS FOR LOVABLE LLM

Code Quality Standards

- Use TypeScript (strict mode)
- Follow React best practices (hooks, composition)
- Use Tailwind CSS for all styling (no inline styles)
- Semantic HTML (accessibility-first)
- Proper error handling & validation
- Environment variables for API endpoints
- Comments on complex logic
- Meaningful variable/function names

File Organization

```
/src
├── app/              (Next.js pages)
├── components/       (Reusable components)
├── hooks/            (Custom React hooks)
├── lib/              (Utilities, constants)
├── styles/           (Global CSS, design tokens)
├── types/            (TypeScript interfaces)
├── public/           (Static assets, i18n)
└── __tests__/        (Test files)
```

Key Instructions

1. **Design System First:** All colors, typography, spacing from the provided palettes

2. **Accessibility is Mandatory:** WCAG 2.1 AA compliance is required, not optional
3. **Mobile-First Responsive:** Design for 320px first, enhance for larger screens
4. **Multilingual from Day 1:** Support English, Swahili, Sheng with proper i18n setup
5. **Performance Matters:** Target <2.5s LCP on 3G networks
6. **Real Data:** Use realistic mock data (services, chats, metrics)
7. **Offline First:** Service Worker caching for offline capability
8. **Dark Mode:** Full dark mode support with Tailwind dark: classes
9. **No External CDN:** All fonts, icons, libraries from npm (self-hosted)
10. **Semantic HTML:** Use `<header>`, `<nav>`, `<main>`, `<section>`, `<article>`

Testing Requirements

- Run Lighthouse audit (target ≥90)
- Test with axe DevTools (0 accessibility issues)
- Keyboard navigation test (all interactive elements accessible)
- Screen reader test (NVDA or VoiceOver)
- Mobile device test (iOS Safari, Chrome Android)
- Network throttling test (simulate 3G)
- Dark mode verification
- Language switching test (no broken layouts)

✅ FINAL CHECKLIST BEFORE SUBMISSION

□ All 5 interfaces built (Web, Mobile, USSD, Voice, Dashboard)
□ Design system implemented (colors, typography, spacing)
□ WCAG 2.1 AA compliant
□ Responsive (320px-1920px)
□ Multilingual (English, Swahili, Sheng)
□ Dark mode supported
□ Accessibility tested (axe, WAVE, screen readers)
□ Performance tested (Lighthouse ≥90)
□ Mobile-tested (iOS & Android)
□ Keyboard navigation working
□ No console errors
□ Code quality (TypeScript, best practices)
□ Mock APIs working

□ Documentation complete
□ Ready for production deployment

📞 SUPPORT RESOURCES

Design Reference Documents:

1. `AfroKen_LLM_Frontend_Design_Brief.md` - Complete design system
2. `AfroKen_Frontend_Design_Integration_Summary.md` - Architecture & topology
3. `AfroKen_Frontend_Quick_Reference.md` - Visual quick reference
4. Figma Design Kit (figma.com/afroken-llm) - Interactive components

Technology Recommendations:

- React 18 + Next.js 14
- TypeScript (strict mode)
- Tailwind CSS + Shadcn/ui
- React Query for data
- Zustand for state
- next-i18next for i18n
- Recharts for dashboards
- Mapbox GL for maps
- Whisper for voice input
- Coqui for text-to-speech

🎯 SUCCESS CRITERIA

Your generated frontend is successful when it meets ALL criteria:

✅ Functionality

- All 5 interfaces fully functional (web, mobile, USSD, voice, dashboard)
- Mock APIs integrated and responding correctly
- Message sending/receiving working
- Service navigation working
- Dashboard metrics updating

✅ Design

- Matches provided color palette exactly
- Typography follows type scale
- Spacing uses 8px grid consistently
- Responsive layout at all breakpoints
- Dark mode fully supported

✅ **Accessibility**

- WCAG 2.1 AA compliant (axe DevTools 0 issues)
- Keyboard navigation on all interactive elements
- Screen reader compatible (semantic HTML + ARIA)
- High contrast ratios (4.5:1 minimum)
- Touch targets 48x48px minimum

✅ **Performance**

- Lighthouse score ≥90
- LCP <2.5s on 3G
- FID <100ms
- CLS <0.1
- Images optimized (WebP)

✅ **Internationalization**

- English, Swahili, Sheng fully supported
- Language switching instant (no reload)
- No layout breaking on longer text (Swahili)
- Proper diacritics (ā, ē, ī, ō, ū)

✅ **Code Quality**

- TypeScript (strict mode)
- Semantic HTML
- No console errors
- Proper error handling
- Meaningful naming

Ready to generate? Use this prompt with Lovable LLM and reference the design documents provided.

Good luck! 🚀