

```
import React, { useState } from 'react';
import { Search, Bus, Clock, CreditCard, Menu, X, User, Info, PlusCircle, LayoutDashboard,
  LogIn, Settings } from 'lucide-react';
import { Card, CardContent } from '@components/ui/card';
```

```
// -----
// Design System (palette + tokens)
// -----
// Rationale:
// - Primary (teal) conveys trust & clarity for public transport
// - Accent (cyan) adds a tech-forward, fresh identity
// - High-contrast text grays for accessibility (WCAG > 4.5)
const PALETTE = {
  primary: '#009688', // Teal 600
  primaryDark: '#00796B', // Teal 700
  accent: '#5CE1E6', // Cyan 400
  bg: '#F9FAFB',
  surface: 'FFFFFF',
  text: '#1F2937',
  textMuted: '#6B7280',
  success: '#00C49A',
  error: '#D32F2F',
};
```

```

// Lightweight SVG map thumbnail to simulate a route preview without external embeds.
// In production, replace with a static tiles image + polyline overlay (Mapbox Static Images or
// your own tile server),
// or render a tiny Leaflet/MapLibre canvas.
function MapPreview({ color = '#2B7FFF', secondary = '#00C49A' }: { color?: string; secondary?:
string }) {
  return (
    <svg viewBox="0 0 320 120" className="w-full h-24 rounded-md bg-gray-100">
      {/* faint road grid */}
      <defs>
        <pattern id="grid" width="16" height="16" patternUnits="userSpaceOnUse">
          <path d="M 16 0 L 0 0 0 16" fill="none" stroke="#e5e7eb" strokeWidth="1" />
        </pattern>
        <filter id="shadow" x="-20%" y="-20%" width="140%" height="140%">
          <feDropShadow dx="0" dy="1" stdDeviation="1" floodOpacity="0.25" />
        </filter>
      </defs>
      <rect x="0" y="0" width="320" height="120" fill="url(#grid)" />

      {/* terminal markers */}
      <g filter="url(#shadow)">
        <circle cx="24" cy="96" r="6" fill={secondary} />
        <circle cx="296" cy="24" r="6" fill={secondary} />
      </g>

      {/* route polyline */}
      <polyline
        points="24,96 80,84 112,72 140,68 168,60 196,48 228,40 260,32 296,24"
        fill="none"
        stroke={color}
        strokeWidth="4"
        strokeLinejoin="round"
        strokeLinecap="round"
      />
    </svg>
  );
}

export default function DARoutesHomepage() {
  const [menuOpen, setMenuOpen] = useState(false);

  const routes = [
    {
      id: 1,

```

```

    name: 'T/Nyuki - Gerezani',
    via: 'via Bagamoyo Road, Muhimbili Road',
    color: '#2B7FFF',
    secondary: '#00C49A',
    number: '14',
    time: '05:00 - 21:00',
    fare: 'TSh 750'
  },
  {
    id: 2,
    name: 'Mabibo - M/mBusho',
    via: 'via Kawawa Road, Morogoro Road',
    color: '#7C3AED',
    secondary: '#00C49A',
    number: '14',
    time: '05:00 - 21:00',
    fare: 'TSh 750'
  },
  {
    id: 3,
    name: 'T/Nyuki - M/mBusho',
    via: 'via Bagamoyo Road',
    color: '#EF4444',
    secondary: '#00C49A',
    number: '14',
    time: '05:00 - 21:00',
    fare: 'TSh 750'
  }
];

```

```

return (
  <div className="min-h-screen" style={{ backgroundColor: PALETTE.bg, color:
PALETTE.text }}>
    { /* Header */ }
    <header className="flex items-center justify-between px-4 md:px-6 py-4 shadow-sm"
      style={{ backgroundColor: PALETTE.primary, color: '#ffffff' }}>
      <div className="flex items-center gap-3">
        { /* Hamburger (always visible) */ }
        <button
          aria-label="Open menu"
          onClick={() => setMenuOpen(true)}
          className="p-2 rounded-md focus:outline-none focus:ring-2 focus:ring-offset-2"
          style={{ backgroundColor: 'transparent' }}
        >

```

```

    <Menu size={22} className="opacity-90" />
  </button>
  <h1 className="text-2xl font-semibold tracking-wide">DARoutes</h1>
</div>

{/* Right-side actions: visible on md+; collapse into menu on mobile */}
<nav className="hidden md:flex items-center gap-6 text-sm font-medium">
  <a href="#explainer" className="hover:underline/50 opacity-95">Explainer</a>
  <a href="#login" className="hover:underline/50 opacity-95">Login</a>
</nav>
</header>

{/* Mobile Sidebar Menu */}
<div className={`fixed inset-0 z-50 ${menuOpen ? 'pointer-events-auto' :
'pointer-events-none'}`}>
  {/* Backdrop */}
  <div
    onClick={() => setMenuOpen(false)}
    className={`absolute inset-0 transition-opacity ${menuOpen ? 'opacity-40' : 'opacity-0'}`}
    style={{ backgroundColor: '#000' }}
    aria-hidden
  />
  {/* Drawer */}
  <aside
    className={`absolute left-0 top-0 h-full w-72 bg-white shadow-xl transition-transform
duration-200 ${menuOpen ? 'translate-x-0' : '-translate-x-full'}`}
  >
    <div className="flex items-center justify-between px-4 py-3 border-b">
      <div className="flex items-center gap-2">
        <Menu size={18} className="text-gray-500" />
        <span className="font-semibold">Menu</span>
      </div>
      <button aria-label="Close menu" onClick={() => setMenuOpen(false)} className="p-1
rounded hover:bg-gray-100">
        <X size={18} />
      </button>
    </div>
    <nav className="p-2 text-sm">
      <a href="#home" className="flex items-center gap-2 px-3 py-2 rounded
hover:bg-gray-100"><LayoutDashboard size={16}/> Home</a>
      <a href="#dashboard" className="flex items-center gap-2 px-3 py-2 rounded
hover:bg-gray-100"><LayoutDashboard size={16}/> My Dashboard</a>
      <a href="#contribute" className="flex items-center gap-2 px-3 py-2 rounded
hover:bg-gray-100"><PlusCircle size={16}/> Contribute a Route</a>
    </nav>
  </aside>
</div>

```

```

      <a href="#explainer" className="flex items-center gap-2 px-3 py-2 rounded
hover:bg-gray-100"><Info size={16}/> Explainer</a>
      <a href="#login" className="flex items-center gap-2 px-3 py-2 rounded
hover:bg-gray-100"><Login size={16}/> Login</a>
      <a href="#profile" className="flex items-center gap-2 px-3 py-2 rounded
hover:bg-gray-100"><User size={16}/> Profile</a>
      <a href="#settings" className="flex items-center gap-2 px-3 py-2 rounded
hover:bg-gray-100"><Settings size={16}/> Settings</a>
    </nav>
  </aside>
</div>

```

```

  {/* Search Bar */}
  <div className="flex justify-center mt-8 px-4">
    <div className="flex items-center bg-white border border-gray-200 rounded-full
shadow-sm px-4 py-2 w-full md:w-3/5 lg:w-1/2 max-w-2xl">
      <Search className="text-gray-400 mr-2" size={18} />
      <input
        type="text"
        placeholder="Search routes, stops, or terminals..."
        className="w-full outline-none text-sm text-gray-700"
      />
    </div>
  </div>

```

```

  {/* Filters */}
  <div className="flex justify-center mt-4 gap-2 md:gap-4 text-sm" style={{ color:
PALETTE.textMuted }}>
    <button className="px-3 md:px-4 py-1 bg-white border rounded-full hover:bg-teal-50">All
Routes</button>
    <button className="px-3 md:px-4 py-1 bg-white border rounded-full hover:bg-teal-50">By
Fare</button>
    <button className="px-3 md:px-4 py-1 bg-white border rounded-full hover:bg-teal-50">By
Terminal</button>
  </div>

```

```

  {/* Routes Grid */}
  <section className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6 px-4 md:px-6 mt-8
md:mt-10">
    {routes.map((route) => (
      <Card key={route.id} className="hover:shadow-md transition-shadow cursor-pointer
border border-gray-100">
        <div className="flex items-center justify-between p-4 border-b">
          <div>

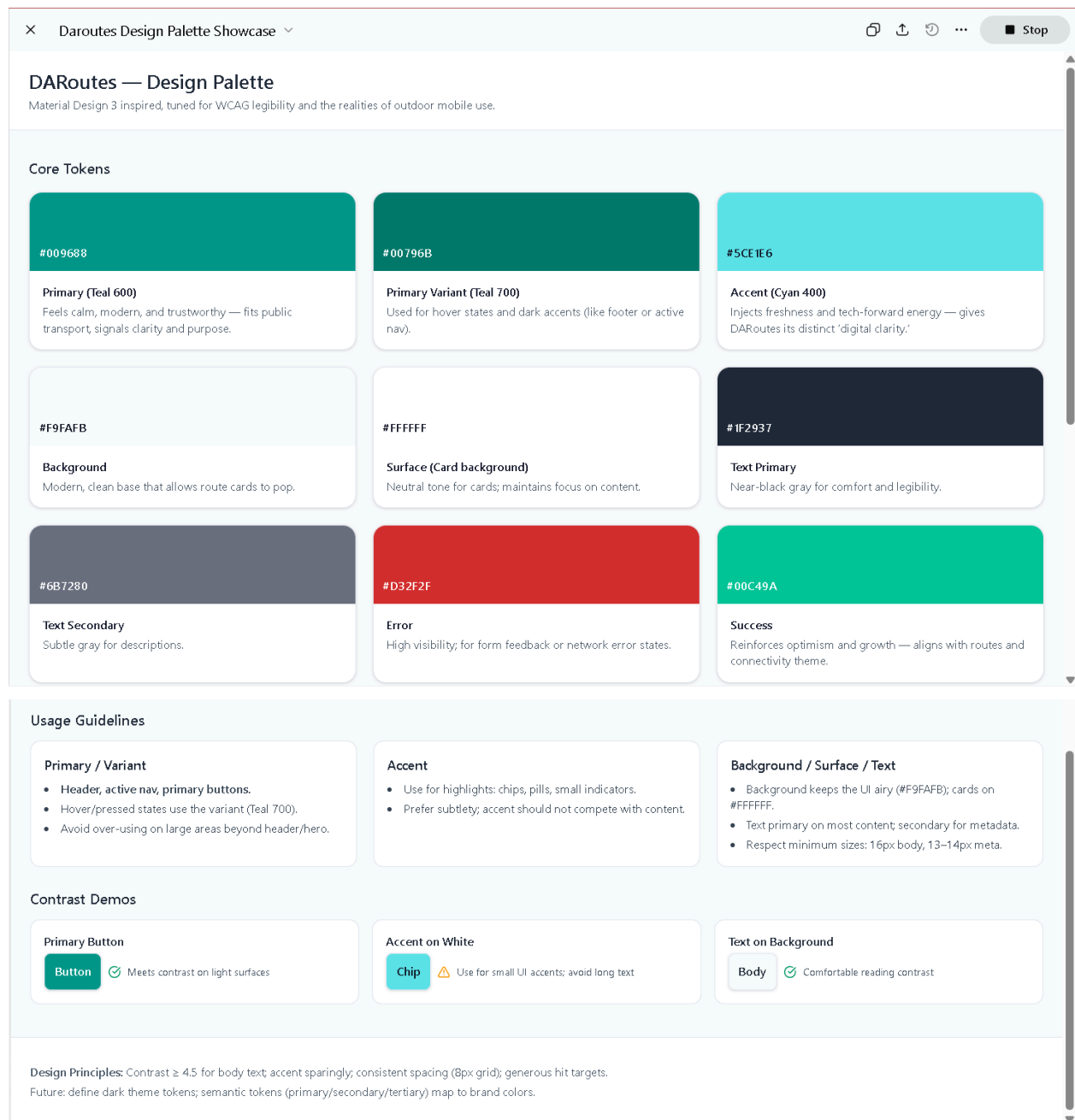
```

```

        <h2 className="font-semibold text-lg" style={{ color: PALETTE.text
}}>{route.name}</h2>
        <p className="text-sm" style={{ color: PALETTE.textMuted }}>{route.via}</p>
        </div>
        <div className="flex gap-1">
            <div className="h-4 w-4 rounded-sm" style={{ backgroundColor: route.color
}}></div>
            <div className="h-4 w-4 rounded-sm" style={{ backgroundColor: route.secondary
}}></div>
        </div>
        </div>
        <CardContent className="p-4 flex flex-col gap-2">
            {/* Map Preview (SVG) */}
            <div className="overflow-hidden rounded-md">
                <MapPreview color={route.color} secondary={route.secondary} />
            </div>
            <div className="flex justify-between text-sm mt-2" style={{ color: PALETTE.textMuted
}}>
                <span className="flex items-center gap-1"><Bus size={14} />
{route.number}</span>
                <span className="flex items-center gap-1"><Clock size={14} /> {route.time}</span>
                <span className="flex items-center gap-1"><CreditCard size={14} />
{route.fare}</span>
            </div>
        </CardContent>
    </Card>
    )}
</section>

    {/* Footer */}
    <footer className="mt-12 md:mt-16 text-center text-sm py-6 border-t" style={{ color:
PALETTE.textMuted, backgroundColor: PALETTE.surface }}>
        <p>DARoutes © {new Date().getFullYear()} — Mapping Dar es Salaam's Public Transport
Network</p>
        <p>Design palette: Primary {PALETTE.primary}, Accent {PALETTE.accent}</p>
    </footer>
</div>
);
}

```



```
import React from "react";
import { CheckCircle, AlertTriangle } from "lucide-react";

// DARoutes Design Palette Showcase
// Material Design 3 inspired + WCAG-aware tokens
const TOKENS = [
  { role: "Primary (Teal 600)", hex: "#009688", rationale: "Feels calm, modern, and trustworthy — fits public transport, signals clarity and purpose." },
  { role: "Primary Variant (Teal 700)", hex: "#00796B", rationale: "Used for hover states and dark accents (like footer or active nav)."},
```

```

    { role: "Accent (Cyan 400)", hex: "#5CE1E6", rationale: "Injects freshness and tech-forward energy — gives DARoutes its distinct 'digital clarity.'" },
    { role: "Background", hex: "#F9FAFB", rationale: "Modern, clean base that allows route cards to pop." },
    { role: "Surface (Card background)", hex: "FFFFFF", rationale: "Neutral tone for cards; maintains focus on content." },
    { role: "Text Primary", hex: "#1F2937", rationale: "Near-black gray for comfort and legibility." },
    { role: "Text Secondary", hex: "#6B7280", rationale: "Subtle gray for descriptions." },
    { role: "Error", hex: "#D32F2F", rationale: "High visibility; for form feedback or network error states." },
    { role: "Success", hex: "#00C49A", rationale: "Reinforces optimism and growth — aligns with routes and connectivity theme." },
  ];

```

```

// Utility to pick white/black text for legibility (approx luminance)
function bestText(hex: string) {
  const c = hex.replace('#','');
  const r = parseInt(c.slice(0,2),16), g = parseInt(c.slice(2,4),16), b = parseInt(c.slice(4,6),16);
  const luminance = 0.299*r + 0.587*g + 0.114*b; // 0..255
  return luminance > 160 ? '#111827' : 'FFFFFF';
}

```

```

function Swatch({ role, hex, rationale }: { role: string; hex: string; rationale: string }) {
  const textColor = bestText(hex);
  return (
    <div className="rounded-2xl overflow-hidden shadow-sm border bg-white">
      <div className="h-24 flex items-end p-3" style={{ backgroundColor: hex, color: textColor }}>
        <div className="text-sm font-semibold tracking-wide">{hex}</div>
      </div>
      <div className="p-4">
        <div className="text-sm font-semibold text-gray-900">{role}</div>
        <div className="text-sm text-gray-600 mt-1">{rationale}</div>
      </div>
    </div>
  );
}

```

```

function ContrastChip({ label, bg, fg }: { label: string; bg: string; fg: string }) {
  return (
    <div className="rounded-lg p-3 text-sm font-medium shadow-sm border" style={{
      backgroundColor: bg, color: fg }}>
      {label}
    </div>
  );
}

```



```
);  
}
```

```
export default function PaletteShowcase() {  
  return (  
    <div className="min-h-screen bg-gray-50 text-gray-800">  
      <header className="px-6 py-5 border-b bg-white">  
        <h1 className="text-2xl font-semibold">DARoutes — Design Palette</h1>  
        <p className="text-sm text-gray-600 mt-1">Material Design 3 inspired, tuned for WCAG  
legibility and the realities of outdoor mobile use.</p>  
      </header>  
  
      { /* Swatches */  
      <section className="px-6 py-8">  
        <h2 className="text-lg font-semibold mb-4">Core Tokens</h2>  
        <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-5">  
          {TOKENS.map((t) => (  
            <Swatch key={t.role} role={t.role} hex={t.hex} rationale={t.rationale} />  
          ))}  
        </div>  
      </section>  
  
      { /* Usage guidance */  
      <section className="px-6 pb-6">  
        <h2 className="text-lg font-semibold mb-3">Usage Guidelines</h2>  
        <div className="grid grid-cols-1 lg:grid-cols-3 gap-5">  
          <div className="rounded-xl border bg-white p-4">  
            <div className="font-semibold mb-2">Primary / Variant</div>  
            <ul className="list-disc list-inside text-sm text-gray-700 space-y-1">  
              <li><span className="font-medium">Header, active nav, primary buttons.</span></li>  
              <li>Hover/pressed states use the variant (Teal 700).</li>  
              <li>Avoid over-using on large areas beyond header/hero.</li>  
            </ul>  
          </div>  
          <div className="rounded-xl border bg-white p-4">  
            <div className="font-semibold mb-2">Accent</div>  
            <ul className="list-disc list-inside text-sm text-gray-700 space-y-1">  
              <li>Use for highlights: chips, pills, small indicators.</li>  
              <li>Prefer subtlety; accent should not compete with content.</li>  
            </ul>  
          </div>  
          <div className="rounded-xl border bg-white p-4">  
            <div className="font-semibold mb-2">Background / Surface / Text</div>  
            <ul className="list-disc list-inside text-sm text-gray-700 space-y-1">
```

- Background keeps the UI airy (#F9FAFB); cards on #FFFFFF.
- Text primary on most content; secondary for metadata.
- Respect minimum sizes: 16px body, 13–14px meta.

</div>
</div>
</section>

```

{ /* Contrast demonstrations */
<section className="px-6 pb-10">
  <h2 className="text-lg font-semibold mb-3">Contrast Demos</h2>
  <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-4">
    <div className="rounded-xl border bg-white p-4 space-y-3">
      <div className="text-sm font-semibold text-gray-800 mb-1">Primary Button</div>
      <div className="flex items-center gap-2">
        <ContrastChip label="Button" bg="#009688" fg="#FFFFFF" />
        <CheckCircle className="text-emerald-600" size={16} />
        <span className="text-xs text-gray-600">Meets contrast on light surfaces</span>
      </div>
    </div>
    <div className="rounded-xl border bg-white p-4 space-y-3">
      <div className="text-sm font-semibold text-gray-800 mb-1">Accent on White</div>
      <div className="flex items-center gap-2">
        <ContrastChip label="Chip" bg="#5CE1E6" fg="#111827" />
        <AlertTriangle className="text-amber-500" size={16} />
        <span className="text-xs text-gray-600">Use for small UI accents; avoid long
text</span>
      </div>
    </div>
    <div className="rounded-xl border bg-white p-4 space-y-3">
      <div className="text-sm font-semibold text-gray-800 mb-1">Text on Background</div>
      <div className="flex items-center gap-2">
        <ContrastChip label="Body" bg="#F9FAFB" fg="#1F2937" />
        <CheckCircle className="text-emerald-600" size={16} />
        <span className="text-xs text-gray-600">Comfortable reading contrast</span>
      </div>
    </div>
  </div>
</section>

<footer className="px-6 py-8 border-t bg-white text-sm text-gray-600">
  <div>
    <span className="font-medium">Design Principles: </span>

```

Contrast ≥ 4.5 for body text; accent sparingly; consistent spacing (8px grid); generous hit targets.

</div>

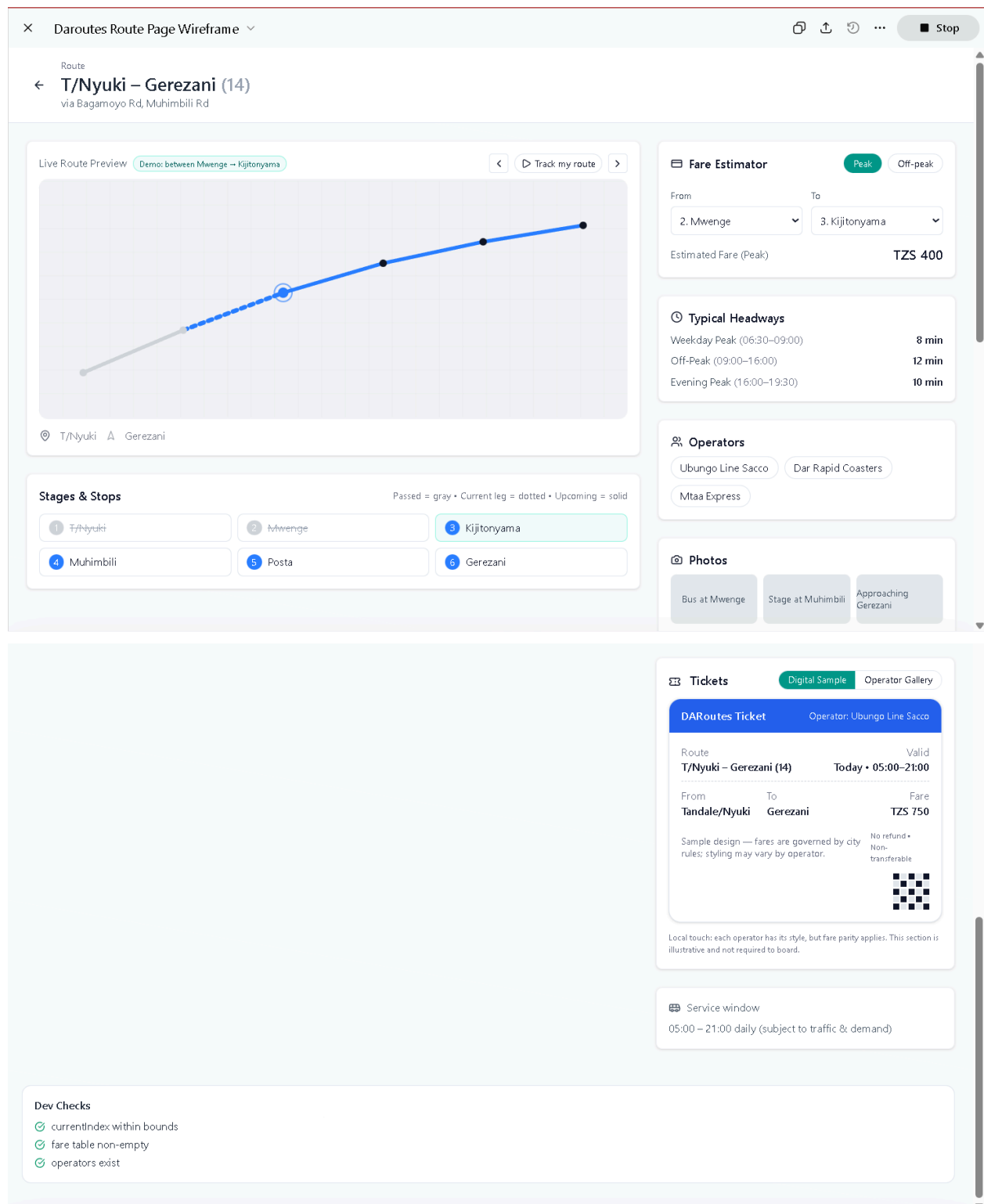
<div className="mt-1">Future: define dark theme tokens; semantic tokens
(primary/secondary/tertiary) map to brand colors.</div>

</footer>

</div>

);

}



```
import React, { useMemo, useState, useEffect } from "react";
import {
  ArrowLeft,
  Clock,

```

```
CreditCard,  
MapPin,  
Navigation2,  
Users,  
Camera,  
Bus,  
Ticket,  
Play,  
Pause,  
ChevronRight,  
ChevronLeft,  
ExternalLink,  
CheckCircle,  
XCircle,  
} from "lucide-react";  
import { Card, CardContent } from "@components/ui/card";
```

```
// -----  
// Palette (kept consistent with homepage)  
// -----  
const PALETTE = {  
  primary: "#009688",  
  primaryDark: "#00796B",  
  accent: "#5CE1E6",  
  bg: "#F9FAFB",  
  surface: "#FFFFFF",  
  text: "#1F2937",  
  textMuted: "#6B7280",  
  passed: "#D1D5DB", // for passed segments  
};
```

```
// -----  
// Demo route data  
// -----  
const ROUTE = {  
  id: "route-14",  
  name: "T/Nyuki – Gerezani",  
  code: "14",  
  color: "#2B7FFF",  
  origin: "Tandale/Nyuki",  
  destination: "Gerezani",  
  via: ["Bagamoyo Rd", "Muhimbili Rd"],  
  time: { start: "05:00", end: "21:00" },  
  headways: [
```

```

    { label: "Weekday Peak", window: "06:30–09:00", headwayMin: 8 },
    { label: "Off-Peak", window: "09:00–16:00", headwayMin: 12 },
    { label: "Evening Peak", window: "16:00–19:30", headwayMin: 10 },
  ],
  operators: [
    { id: "op-1", name: "Ubungo Line Sacco", brandColor: "#2563EB" },
    { id: "op-2", name: "Dar Rapid Coasters", brandColor: "#F59E0B" },
    { id: "op-3", name: "Mtaa Express", brandColor: "#10B981" },
  ],
  stages: [
    { id: 1, name: "T/Nyuki", lat: -6.775, lon: 39.236, busy: "06:30–08:30" },
    { id: 2, name: "Mwenge", lat: -6.765, lon: 39.24, busy: "07:00–09:00" },
    { id: 3, name: "Kijitonyama", lat: -6.761, lon: 39.246, busy: "07:30–09:30" },
    { id: 4, name: "Muhimbili", lat: -6.801, lon: 39.27, busy: "12:00–14:00" },
    { id: 5, name: "Posta", lat: -6.816, lon: 39.287, busy: "16:00–18:00" },
    { id: 6, name: "Gerezani", lat: -6.829, lon: 39.295, busy: "17:30–19:30" },
  ],
  fares: {
    currency: "TZS",
    peakMultiplier: 1.0,
    offpeakMultiplier: 0.9,
    table: [
      { from: 1, to: 2, price: 400 },
      { from: 2, to: 3, price: 400 },
      { from: 3, to: 4, price: 400 },
      { from: 4, to: 5, price: 500 },
      { from: 5, to: 6, price: 600 },
    ],
  },
  photos: [
    { id: "p1", caption: "Bus at Mwenge", url: "" },
    { id: "p2", caption: "Stage at Muhimbili", url: "" },
    { id: "p3", caption: "Approaching Gerezani", url: "" },
  ],
};

```

```

// -----
// Map preview with progress states (passed/current/upcoming)
// -----
function RouteMap({
  color = "#2B7FFF",
  totalStops,
  currentIndex,
}): {

```

```

color?: string;
totalStops: number;
currentIndex: number;
}} {
  const width = 800;
  const height = 320;
  const start = { x: 60, y: 260 };
  const end = { x: 740, y: 60 };

  const pts = Array.from({ length: totalStops }, (_, i) => {
    const t = i / (totalStops - 1);
    const x = start.x + (end.x - start.x) * t;
    const y = start.y + (end.y - start.y) * t - Math.sin(t * Math.PI) * 30;
    return { x, y };
  });

  const segs = pts.slice(0, -1).map((p, i) => ({ a: p, b: pts[i + 1], i }));

  return (
    <svg viewBox={`0 0 ${width} ${height}`} className="w-full h-64 md:h-80 rounded-xl
bg-gray-100">
      <defs>
        <pattern id="grid32" width="32" height="32" patternUnits="userSpaceOnUse">
          <path d="M 32 0 L 0 0 0 32" fill="none" stroke="#e5e7eb" strokeWidth="1" />
        </pattern>
        <filter id="shadow" x="-20%" y="-20%" width="140%" height="140%">
          <feDropShadow dx="0" dy="1" stdDeviation="1" floodOpacity="0.25" />
        </filter>
      </defs>
      <rect x={0} y={0} width={width} height={height} fill="url(#grid32)" />

      {segs.map(({ a, b, i }) => {
        const isPassed = i < currentIndex - 1;
        const isCurrent = i === currentIndex - 1;
        const stroke = isPassed ? PALETTE.passed : color;
        const dash = isCurrent ? "6,6" : "0";
        return (
          <line
            key={i}
            x1={a.x}
            y1={a.y}
            x2={b.x}
            y2={b.y}
            stroke={stroke}

```

```

        strokeWidth={isCurrent ? 6 : 5}
        strokeDasharray={dash}
        strokeLinecap="round"
      />
    );
  }}}

  {pts.map((p, i) => {
    const isPassed = i < currentIndex;
    const isCurrent = i === currentIndex;
    const fill = isCurrent ? color : isPassed ? PALETTE.passed : "#111827";
    return (
      <g key={i}>
        <circle cx={p.x} cy={p.y} r={isCurrent ? 7 : 5} fill={fill} />
        {isCurrent && (
          <circle cx={p.x} cy={p.y} r={12} fill="none" stroke={color} strokeWidth={2} opacity={0.6}
        )}
      </g>
    );
  })}
</svg>
);
}

// -----
// Ticket mock components (design only)
// -----
const Perforation = () => (
  <div className="relative w-full h-0.5 border-t border-dashed border-gray-300 my-2" />
);

const QRPlaceholder = () => (
  <div className="grid grid-cols-5 grid-rows-5 gap-0.5">
    {Array.from({ length: 25 }).map((_, i) => (
      <div key={i} className={` ${i % 2 === 0 ? "bg-gray-900" : "bg-gray-200"} w-2 h-2`} />
    ))}
  </div>
);

function DigitalTicket({ operatorName, brandColor }: { operatorName: string; brandColor: string }) {
  return (
    <div className="rounded-2xl border shadow-sm overflow-hidden bg-white">

```



```

<div className="px-4 py-3 flex items-center justify-between" style={{ backgroundColor:
brandColor, color: "#fff" }}>
  <div className="text-sm font-semibold tracking-wide">DARoutes Ticket</div>
  <div className="text-xs opacity-90">Operator: {operatorName}</div>
</div>
<div className="p-4">
  <div className="flex items-center justify-between text-sm">
    <div>
      <div className="text-gray-500">Route</div>
      <div className="font-semibold">
        {ROUTE.name} ({ROUTE.code})
      </div>
    </div>
    <div className="text-right">
      <div className="text-gray-500">Valid</div>
      <div className="font-semibold">Today • {ROUTE.time.start}—{ROUTE.time.end}</div>
    </div>
  </div>
  <Perforation />
  <div className="grid grid-cols-3 gap-3 text-sm">
    <div>
      <div className="text-gray-500">From</div>
      <div className="font-medium">{ROUTE.origin}</div>
    </div>
    <div>
      <div className="text-gray-500">To</div>
      <div className="font-medium">{ROUTE.destination}</div>
    </div>
    <div className="text-right">
      <div className="text-gray-500">Fare</div>
      <div className="font-semibold">{ROUTE.fares.currency} 750</div>
    </div>
  </div>
  <div className="mt-4 flex items-center justify-between">
    <div className="text-xs text-gray-500">
      Sample design — fares are governed by city rules; styling may vary by operator.
    </div>
    <div className="text-[10px] text-gray-500">No refund • Non-transferable</div>
  </div>
  <div className="mt-3 flex justify-end">
    <QRPlaceholder />
  </div>
</div>
</div>

```

```
);
}
```

```
function TicketGallery({ operators }: { operators: { id: string; name: string; brandColor: string }[] })
{
  return (
    <div className="grid grid-cols-2 gap-3">
      {operators.map((op) => (
        <div key={op.id} className="rounded-xl border overflow-hidden bg-white">
          <div className="h-2" style={{ backgroundColor: op.brandColor }} />
          <div className="p-3">
            <div className="text-xs text-gray-500">Operator</div>
            <div className="text-sm font-medium mb-2">{op.name}</div>
            <div className="aspect-video rounded-md bg-gray-100 flex items-center justify-center
text-xs text-gray-500">
              Ticket style placeholder
            </div>
          </div>
        </div>
      ))}
    </div>
  );
}
```

```
// -----
// Stop drawer (click a stop to open)
// -----
```

```
function StopDrawer({ open, onClose, stop }: { open: boolean; onClose: () => void; stop: any }) {
  return (
    <div className={`fixed inset-0 z-50 ${open ? "pointer-events-auto" :
"pointer-events-none"}`}>
      <div
        className={`absolute inset-0 ${open ? "opacity-40" : "opacity-0"} transition-opacity`}
        style={{ background: "#000" }}
        onClick={onClose}
      />
      <div
        className={`absolute left-0 right-0 bottom-0 bg-white rounded-t-2xl shadow-2xl
transition-transform duration-200 ${
          open ? "translate-y-0" : "translate-y-full"
        }}`
      >
        <div className="p-4">
          <div className="h-1 w-10 bg-gray-200 rounded mx-auto mb-3" />

```

```

<div className="flex items-center justify-between mb-2">
  <div>
    <div className="text-xs text-gray-500">Stop</div>
    <div className="text-lg font-semibold">{stop?.name}</div>
  </div>
  <a href="#/stops/placeholder" className="text-sm text-teal-700 inline-flex items-center
gap-1">
    Open stop page <ExternalLink size={14} />
  </a>
</div>
<div className="grid grid-cols-3 gap-3 text-sm">
  <div>
    <div className="text-gray-500">Lat</div>
    <div className="font-medium">{stop?.lat?.toFixed(4)}</div>
  </div>
  <div>
    <div className="text-gray-500">Lon</div>
    <div className="font-medium">{stop?.lon?.toFixed(4)}</div>
  </div>
  <div>
    <div className="text-gray-500">Busy</div>
    <div className="font-medium">{stop?.busy}</div>
  </div>
</div>
<div className="mt-3 text-xs text-gray-600">
  Routes here: <span className="font-medium">14, 15B</span> • Nearby: pharmacy,
cafe
</div>
<div className="mt-4 grid grid-cols-3 gap-2">
  <div className="aspect-video rounded-lg bg-gray-100" />
  <div className="aspect-video rounded-lg bg-gray-100" />
  <div className="aspect-video rounded-lg bg-gray-100" />
</div>
<div className="mt-4 text-right">
  <button onClick={onClose} className="px-3 py-2 text-sm rounded-md border
hover:bg-gray-50">
    Close
  </button>
</div>
</div>
</div>
</div>
);
}

```

```

// -----
// Stage selector + fare calculator
// -----
function StageSelector({
  stages,
  value,
  onChange,
}: {
  stages: { id: number; name: string }[];
  value: { from: number; to: number };
  onChange: (v: { from: number; to: number }) => void;
}) {
  const fromId = "stage-from";
  const told = "stage-to";
  return (
    <div className="grid grid-cols-2 gap-3">
      <div>
        <label className="text-xs text-gray-600" htmlFor={fromId}>
          From
        </label>
        <select
          id={fromId}
          value={value.from}
          onChange={(e) => onChange({ ...value, from: Number(e.target.value) })}
          className="mt-1 w-full rounded-md border px-2 py-2 text-sm"
        >
          {stages.map((s) => (
            <option key={s.id} value={s.id}>
              {s.id}. {s.name}
            </option>
          ))}
        </select>
      </div>
      <div>
        <label className="text-xs text-gray-600" htmlFor={told}>
          To
        </label>
        <select
          id={told}
          value={value.to}
          onChange={(e) => onChange({ ...value, to: Number(e.target.value) })}
          className="mt-1 w-full rounded-md border px-2 py-2 text-sm"
        >

```

```

    {stages.map((s) => (
      <option key={s.id} value={s.id}>
        {s.id}. {s.name}
      </option>
    ))}
  </select>
</div>
</div>
);
}

function FareCalculator({ peak, stages, fares }: { peak: boolean; stages: any[]; fares: any }) {
  const [sel, setSel] = useState<{ from: number; to: number }>({
    from: stages[1].id, // demo defaults Mwenge
    to: stages[2].id, // demo defaults Kijitonyama
  });

  const price = useMemo(() => {
    const [a, b] = [Math.min(sel.from, sel.to), Math.max(sel.from, sel.to)];
    const base = fares.table
      .filter((r: any) => r.from >= a && r.to <= b)
      .reduce((sum: number, r: any) => sum + r.price, 0);
    const mult = peak ? fares.peakMultiplier : fares.offpeakMultiplier;
    return Math.round(base * mult);
  }, [sel, fares, peak]);

  return (
    <div>
      <StageSelector stages={stages} value={sel} onChange={setSel} />
      <div className="mt-3 flex items-center justify-between">
        <div className="text-sm text-gray-600">Estimated Fare ({peak ? "Peak" :
"Off-peak"})</div>
        <div className="text-lg font-semibold">
          {fares.currency} {price.toLocaleString()}
        </div>
      </div>
    </div>
  );
}

// -----
// (Dev) Minimal UI test panel to catch regressions early
// -----
function DevTests({ route, currentIndex }: { route: typeof ROUTE; currentIndex: number }) {

```

```

const tests = [
  {
    name: "currentIndex within bounds",
    pass: currentIndex >= 0 && currentIndex < route.stages.length,
  },
  {
    name: "fare table non-empty",
    pass: Array.isArray(route.fares.table) && route.fares.table.length > 0,
  },
  {
    name: "operators exist",
    pass: route.operators.length >= 1,
  },
];
return (
  <div className="mt-6 p-4 rounded-xl border bg-white">
    <div className="text-sm font-semibold mb-2">Dev Checks</div>
    <ul className="text-sm space-y-1">
      {tests.map((t) => (
        <li key={t.name} className="flex items-center gap-2">
          {t.pass ? <CheckCircle size={14} className="text-emerald-600" /> : <XCircle size={14}
className="text-rose-600" />}
          <span>{t.name}</span>
        </li>
      ))}
    </ul>
  </div>
);
}

```

```
// -----
```

```
// Page
```

```
// -----
```

```

export default function RoutePage() {
  // Demo: show we are between Mwenge (idx 1) and Kijitonyama (idx 2)
  const [currentIndex, setCurrentIndex] = useState(2);
  const [tracking, setTracking] = useState(false);
  const [peak, setPeak] = useState(true);
  const [activeTab, setActiveTab] = useState<"digital" | "gallery">("digital");
  const [drawerOpen, setDrawerOpen] = useState(false);
  const [drawerStop, setDrawerStop] = useState<any>(null);

```

```

  // simple demo tracker: auto-advance every 3s when enabled
  useEffect(() => {

```

```

    if (!tracking) return;
    if (currentIndex >= ROUTE.stages.length - 1) return;
    const t = setInterval(() => setCurrentIndex((i) => Math.min(i + 1, ROUTE.stages.length - 1)),
3000);
    return () => clearInterval(t);
  }, [tracking, currentIndex]);

  const openStop = (idx: number) => {
    setDrawerStop(ROUTE.stages[idx]);
    setDrawerOpen(true);
  };

  return (
    <div className="min-h-screen" style={{ backgroundColor: PALETTE.bg, color:
PALETTE.text }}>
      {/* Top bar */}
      <div className="px-4 md:px-6 py-4 flex items-center gap-3 bg-white border-b">
        <button className="p-2 rounded-md hover:bg-gray-100" aria-label="Back">
          <ArrowLeft size={18} />
        </button>
        <div>
          <div className="text-xs text-gray-500">Route</div>
          <h1 className="text-xl md:text-2xl font-semibold">
            {ROUTE.name} <span className="text-gray-400">({ROUTE.code})</span>
          </h1>
          <div className="text-sm text-gray-600">via {ROUTE.via.join(", ")}</div>
        </div>
      </div>

      {/* Content */}
      <div className="px-4 md:px-6 py-6 grid grid-cols-1 lg:grid-cols-3 gap-6">
        {/* Left column */}
        <div className="lg:col-span-2 space-y-6">
          <Card className="border border-gray-100">
            <CardContent className="p-4">
              <div className="flex items-center justify-between mb-2">
                <div className="text-sm text-gray-600">
                  Live Route Preview
                  <span className="ml-2 inline-flex items-center text-[11px] px-2 py-0.5 rounded-full
bg-teal-50 text-teal-800 border border-teal-200">
                    Demo: between Mwenge → Kijitonyama
                  </span>
                </div>
                <div className="flex items-center gap-2">

```

```

<button
  onClick={() => setCurrentIndex((i) => Math.max(i - 1, 0))}
  className="p-1 rounded border hover:bg-gray-50"
  aria-label="Prev stop"
>
  <ChevronLeft size={16} />
</button>
{tracking ? (
  <button
    onClick={() => setTracking(false)}
    className="px-2 py-1 rounded-full border bg-white inline-flex items-center gap-1
text-xs"
    >
      <Pause size={14} /> Pause
    </button>
  ) : (
    <button
      onClick={() => setTracking(true)}
      className="px-2 py-1 rounded-full border bg-white inline-flex items-center gap-1
text-xs"
      >
        <Play size={14} /> Track my route
      </button>
    )}
<button
  onClick={() => setCurrentIndex((i) => Math.min(i + 1, ROUTE.stages.length - 1))}
  className="p-1 rounded border hover:bg-gray-50"
  aria-label="Next stop"
>
  <ChevronRight size={16} />
</button>
</div>
</div>

```

```

<RouteMap color={ROUTE.color} totalStops={ROUTE.stages.length}
currentIndex={currentIndex} />
<div className="mt-3 flex items-center gap-3 text-sm" style={{ color:
PALETTE.textMuted }}>
  <MapPin size={16} />
  <span>{ROUTE.stages[0].name}</span>
  <Navigation2 size={14} className="opacity-60" />
  <span>{ROUTE.stages[ROUTE.stages.length - 1].name}</span>
</div>
</CardContent>

```



```

</Card>

<Card className="border border-gray-100">
  <CardContent className="p-4">
    <div className="flex items-center justify-between">
      <div className="font-semibold">Stages & Stops</div>
      <div className="text-xs" style={{ color: PALETTE.textMuted }}>
        Passed = gray • Current leg = dotted • Upcoming = solid
      </div>
    </div>
    <ol className="mt-3 grid grid-cols-2 md:grid-cols-3 gap-2">
      {ROUTE.stages.map((s, idx) => {
        const isPassed = idx < currentIndex;
        const isCurrent = idx === currentIndex;
        return (
          <li
            key={s.id}
            onClick={() => openStop(idx)}
            className={`px-3 py-2 rounded-lg border text-sm flex items-center gap-2
cursor-pointer transition ${
              isCurrent ? "bg-teal-50 border-teal-200" : "bg-white hover:bg-gray-50"
            }`}
            >
            <span
              className={`inline-flex h-5 w-5 items-center justify-center rounded-full text-white
text-xs`}
              style={{ backgroundColor: isPassed ? PALETTE.passed : ROUTE.color }}
            >
              {idx + 1}
            </span>
            <span className={` ${isPassed ? "text-gray-400 line-through" :
""}`}>{s.name}</span>
          </li>
        );
      })}
    </ol>
  </CardContent>
</Card>
</div>

{/* Right column */}
<div className="space-y-6">
  <Card className="border border-gray-100">
    <CardContent className="p-4 space-y-4">

```

```

<div className="flex items-center justify-between">
  <div className="font-semibold flex items-center gap-2">
    <CreditCard size={16} /> Fare Estimator
  </div>
  <div className="flex items-center gap-2 text-xs">
    <button
      onClick={() => setPeak(true)}
      className={`px-3 py-1 rounded-full border ${peak ? "bg-teal-600 text-white
border-transparent" : "bg-white"}`}
    >
      Peak
    </button>
    <button
      onClick={() => setPeak(false)}
      className={`px-3 py-1 rounded-full border ${!peak ? "bg-teal-600 text-white
border-transparent" : "bg-white"}`}
    >
      Off-peak
    </button>
  </div>
</div>
<FareCalculator peak={peak} stages={ROUTE.stages} fares={ROUTE.fares} />
</CardContent>
</Card>

```

```

<Card className="border border-gray-100">
  <CardContent className="p-4">
    <div className="font-semibold flex items-center gap-2">
      <Clock size={16} /> Typical Headways
    </div>
    <ul className="mt-2 space-y-2">
      {ROUTE.headways.map((h) => (
        <li key={h.label} className="flex items-center justify-between text-sm">
          <span className="text-gray-700">
            {h.label} <span className="text-gray-500">({h.window})</span>
          </span>
          <span className="font-medium">{h.headwayMin} min</span>
        </li>
      ))}
    </ul>
  </CardContent>
</Card>

```

```

<Card className="border border-gray-100">

```

```

<CardContent className="p-4">
  <div className="font-semibold flex items-center gap-2">
    <Users size={16} /> Operators
  </div>
  <div className="mt-2 flex flex-wrap gap-2">
    {ROUTE.operators.map((op) => (
      <span key={op.id} className="px-3 py-1 rounded-full border text-sm bg-white
hover:bg-teal-50">
        {op.name}
      </span>
    ))}
  </div>
</CardContent>
</Card>

<Card className="border border-gray-100">
  <CardContent className="p-4">
    <div className="font-semibold flex items-center gap-2">
      <Camera size={16} /> Photos
    </div>
    <div className="mt-2 grid grid-cols-3 gap-2">
      {ROUTE.photos.map((p) => (
        <div key={p.id} className="aspect-video rounded-lg bg-gray-200 flex items-center
justify-center text-xs text-gray-600">
          {p.caption}
        </div>
      ))}
    </div>
  </CardContent>
</Card>

<Card className="border border-gray-100">
  <CardContent className="p-4">
    <div className="flex items-center justify-between">
      <div className="font-semibold">
        <Ticket size={16} className="inline mr-2" /> Tickets
      </div>
      <div className="text-xs flex border rounded-full overflow-hidden">
        <button
          onClick={() => setActiveTab("digital")}
          className={`px-3 py-1 ${activeTab === "digital" ? "bg-teal-600 text-white" :
"bg-white"}}`
        >
          Digital Sample

```

```

        </button>
        <button
            onClick={() => setActiveTab("gallery")}
            className={`px-3 py-1 ${activeTab === "gallery" ? "bg-teal-600 text-white" :
"bg-white"}}`>
            >
            Operator Gallery
        </button>
    </div>
</div>
<div className="mt-3">
    {activeTab === "digital" ? (
        <DigitalTicket operatorName={ROUTE.operators[0].name}
brandColor={ROUTE.operators[0].brandColor} />
    ) : (
        <TicketGallery operators={ROUTE.operators} />
    )}
</div>
<div className="mt-3 text-[11px] text-gray-500">
    Local touch: each operator has its style, but fare parity applies. This section is
illustrative and not required to board.
</div>
</CardContent>
</Card>

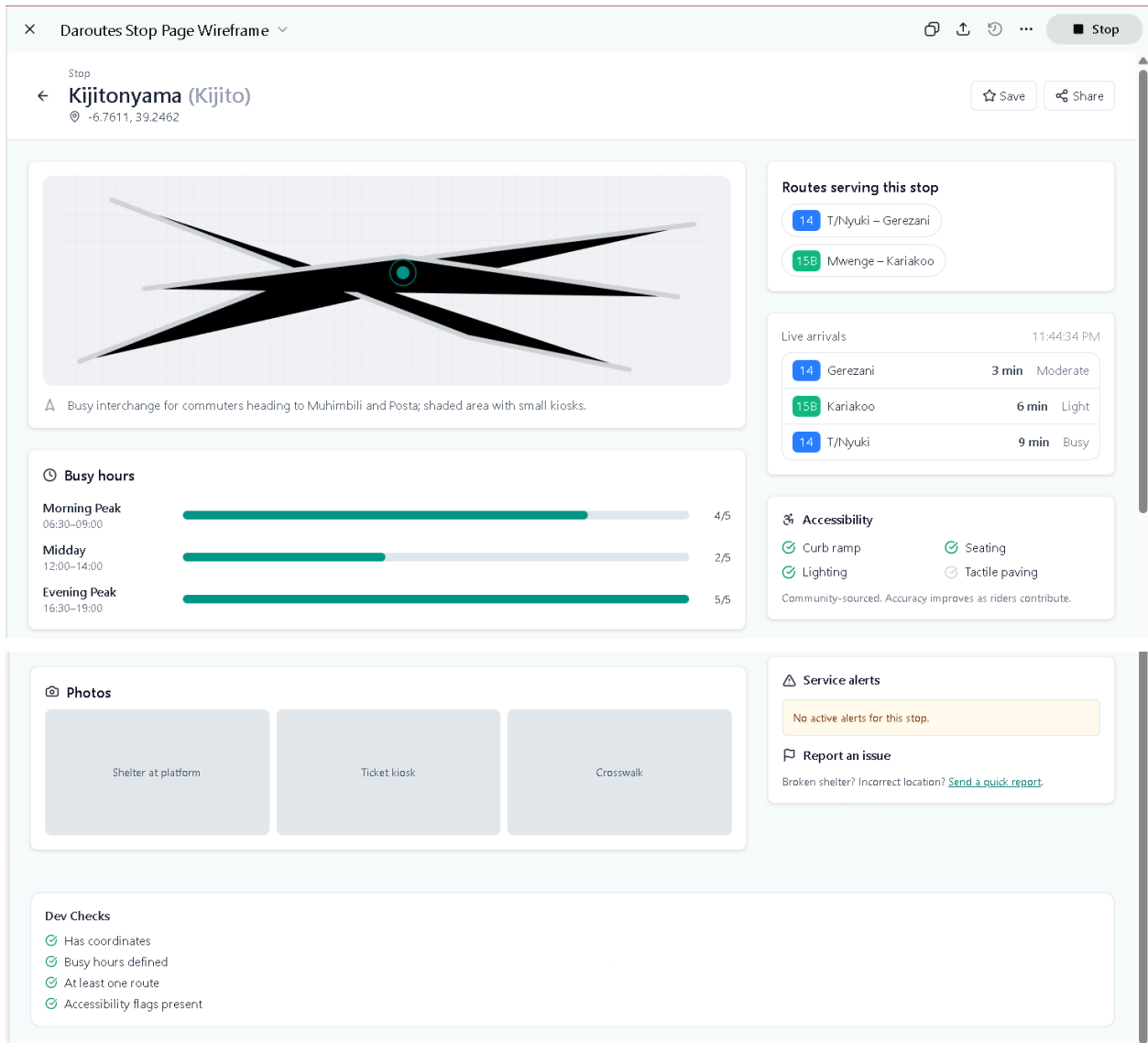
<Card className="border border-gray-100">
    <CardContent className="p-4 text-sm text-gray-600">
        <div className="flex items-center gap-2 mb-2">
            <Bus size={16} /> <span>Service window</span>
        </div>
        <div>
            {ROUTE.time.start} – {ROUTE.time.end} daily (subject to traffic & demand)
        </div>
    </CardContent>
</Card>
</div>
</div>

{/* Dev tests */}
<div className="px-4 md:px-6 pb-10">
    <DevTests route={ROUTE} currentIndex={currentIndex} />
</div>

{/* Stop info drawer */}

```

```
        <StopDrawer open={drawerOpen} onClose={() => setDrawerOpen(false)}  
stop={drawerStop} />  
    </div>  
);  
}
```



```
import React, { useMemo, useState, useEffect } from "react";
import {
  ArrowLeft,
  MapPin,
  Navigation2,
  Clock,
  AlertTriangle,
  Bus,
  Star,
  StarOff,
  Share2,
  Info,
  CheckCircle,
  XCircle,

```

```

    Accessibility,
    Camera,
    Phone,
    Flag,
  } from "lucide-react";
import { Card, CardContent } from "@components/ui/card";

// -----
// Palette tokens (aligned with project)
// -----
const PALETTE = {
  primary: "#009688",
  primaryDark: "#00796B",
  accent: "#5CE1E6",
  bg: "#F9FAFB",
  surface: "#FFFFFF",
  text: "#1F2937",
  textMuted: "#6B7280",
};

// -----
// Demo Stop Data
// -----
const STOP = {
  id: "stop-ki-003",
  name: "Kijitonyama",
  localName: "Kijito",
  lat: -6.7611,
  lon: 39.2462,
  description:
    "Busy interchange for commuters heading to Muhimbili and Posta; shaded area with small kiosks.",
  busyHours: [
    { label: "Morning Peak", window: "06:30–09:00", level: 4 },
    { label: "Midday", window: "12:00–14:00", level: 2 },
    { label: "Evening Peak", window: "16:30–19:00", level: 5 },
  ],
  routes: [
    { id: "14", name: "T/Nyuki – Gerezani", color: "#2B7FFF" },
    { id: "15B", name: "Mwenge – Kariakoo", color: "#10B981" },
  ],
  photos: [
    { id: "p1", caption: "Shelter at platform", url: "" },
    { id: "p2", caption: "Ticket kiosk", url: "" },
  ],
};

```

```

    { id: "p3", caption: "Crosswalk", url: "" },
  ],
  accessibility: {
    curbRamp: true,
    tactilePaving: false,
    seating: true,
    lighting: true,
  },
};

// -----
// Lightweight SVG Map with stop focus & nearby streets
// -----
function StopMap({ lat, lon, color = "#2B7FFF" }: { lat: number; lon: number; color?: string }) {
  const width = 800;
  const height = 260;
  return (
    <svg viewBox={`0 0 ${width} ${height}`} className="w-full h-60 rounded-xl bg-gray-100">
      <defs>
        <pattern id="grid16" width="16" height="16" patternUnits="userSpaceOnUse">
          <path d="M 16 0 L 0 0 0 16" fill="none" stroke="#e5e7eb" strokeWidth="1" />
        </pattern>
        <filter id="shadow" x="-20%" y="-20%" width="140%" height="140%">
          <feDropShadow dx="0" dy="1" stdDeviation="1" floodOpacity="0.25" />
        </filter>
      </defs>
      <rect x={0} y={0} width={width} height={height} fill="url(#grid16)" />
      { /* stylized streets */ }
      <g stroke="#d1d5db" strokeWidth={6} strokeLinecap="round">
        <path d="M20 230 L 300 120 L 780 60" />
        <path d="M60 30 L 500 200 L 700 240" />
        <path d="M100 140 L 420 100 L 760 150" />
      </g>
      { /* stop focus */ }
      <g filter="url(#shadow)">
        <circle cx={420} cy={120} r={8} fill={color} />
        <circle cx={420} cy={120} r={16} fill="none" stroke={color} strokeWidth={2} opacity={0.6} />
      </g>
    </svg>
  );
}

// -----
// Busy-ness bars (0-5)

```



```
// -----
function BusyBar({ label, window, level }: { label: string; window: string; level: number }) {
  const caps = 5;
  return (
    <div className="flex items-center justify-between gap-4">
      <div className="min-w-36">
        <div className="text-sm font-medium text-gray-800">{label}</div>
        <div className="text-xs text-gray-500">{window}</div>
      </div>
      <div className="flex-1 h-2.5 rounded-full bg-gray-200 overflow-hidden">
        <div className="h-full rounded-full" style={{ width: `${(level / caps) * 100}%`,
background-color: PALETTE.primary }} />
      </div>
      <div className="w-8 text-right text-xs text-gray-600">{level}/{caps}</div>
    </div>
  );
}

// -----
// Live arrivals (mock)
// -----
function LiveArrivals() {
  const [now, setNow] = useState<Date>(new Date());
  useEffect(() => {
    const t = setInterval(() => setNow(new Date()), 1000);
    return () => clearInterval(t);
  }, []);
  const rows = [
    { route: "14", head: "Gerezani", etaMin: 3, crowd: "Moderate" },
    { route: "15B", head: "Kariakoo", etaMin: 6, crowd: "Light" },
    { route: "14", head: "T/Nyuki", etaMin: 9, crowd: "Busy" },
  ];
  return (
    <div>
      <div className="flex items-center justify-between text-sm mb-2">
        <span className="text-gray-600">Live arrivals</span>
        <span className="text-gray-400">{now.toLocaleTimeString()}</span>
      </div>
      <div className="divide-y border rounded-xl overflow-hidden bg-white">
        {rows.map((r, i) => (
          <div key={i} className="flex items-center justify-between px-3 py-2 text-sm">
            <div className="flex items-center gap-2">
              <span className="inline-flex items-center justify-center h-6 w-8 rounded-md
text-white" style={{ background: r.route === "14" ? "#2B7FFF" : "#10B981" }}>{r.route}</span>

```

```

        <span className="text-gray-800">{r.head}</span>
      </div>
      <div className="flex items-center gap-4">
        <span className="text-gray-700 font-medium">{r.etaMin} min</span>
        <span className="text-gray-500">{r.crowd}</span>
      </div>
    </div>
  )}
</div>
</div>
);
}

// -----
// Dev tests (render-only)
// -----
function DevTests() {
  const tests = [
    { name: "Has coordinates", pass: typeof STOP.lat === "number" && typeof STOP.lon ===
"number" },
    { name: "Busy hours defined", pass: STOP.busyHours.length > 0 },
    { name: "At least one route", pass: STOP.routes.length > 0 },
    { name: "Accessibility flags present", pass: STOP.accessibility && typeof
STOP.accessibility.curbRamp === "boolean" },
  ];
  return (
    <div className="mt-6 p-4 rounded-xl border bg-white">
      <div className="text-sm font-semibold mb-2">Dev Checks</div>
      <ul className="text-sm space-y-1">
        {tests.map((t) => (
          <li key={t.name} className="flex items-center gap-2">
            {t.pass ? <CheckCircle size={14} className="text-emerald-600" /> : <XCircle size={14}
className="text-rose-600" />}
            <span>{t.name}</span>
          </li>
        ))}
      </ul>
    </div>
  );
}

// -----
// Page
// -----

```

```

export default function StopPage() {
  const [fav, setFav] = useState(false);

  return (
    <div className="min-h-screen" style={{ backgroundColor: PALETTE.bg, color:
PALETTE.text }}>
      {/* Header */}
      <div className="px-4 md:px-6 py-4 flex items-center gap-3 bg-white border-b">
        <button className="p-2 rounded-md hover:bg-gray-100" aria-label="Back">
          <ArrowLeft size={18} />
        </button>
        <div>
          <div className="text-xs text-gray-500">Stop</div>
          <h1 className="text-xl md:text-2xl font-semibold">
            {STOP.name} <span className="text-gray-400">({STOP.localName})</span>
          </h1>
          <div className="text-sm text-gray-600 flex items-center gap-2">
            <MapPin size={14} /> {STOP.lat.toFixed(4)}, {STOP.lon.toFixed(4)}
          </div>
        </div>
        <div className="ml-auto flex items-center gap-2">
          <button className="px-3 py-1.5 rounded-md border bg-white text-sm inline-flex
items-center gap-1" onClick={() => setFav((v) => !v)}>
            {fav ? <StarOff size={16} /> : <Star size={16} />} {fav ? "Unsave" : "Save"}
          </button>
          <button className="px-3 py-1.5 rounded-md border bg-white text-sm inline-flex
items-center gap-1">
            <Share2 size={16} /> Share
          </button>
        </div>
      </div>

      {/* Content */}
      <div className="px-4 md:px-6 py-6 grid grid-cols-1 lg:grid-cols-3 gap-6">
        {/* Left column */}
        <div className="lg:col-span-2 space-y-6">
          <Card className="border border-gray-100">
            <CardContent className="p-4">
              <StopMap lat={STOP.lat} lon={STOP.lon} color={PALETTE.primary} />
              <div className="mt-3 text-sm text-gray-600 flex items-center gap-3">
                <Navigation2 size={16} className="opacity-60" />
                <span>{STOP.description}</span>
              </div>
            </CardContent>
          </Card>
        </div>
      </div>
    </div>
  );
}

```

```

</Card>

<Card className="border border-gray-100">
  <CardContent className="p-4 space-y-4">
    <div className="font-semibold flex items-center gap-2">
      <Clock size={16} /> Busy hours
    </div>
    <div className="space-y-3">
      {STOP.busyHours.map((b) => (
        <BusyBar key={b.label} label={b.label} window={b.window} level={b.level} />
      ))}
    </div>
  </CardContent>
</Card>

<Card className="border border-gray-100">
  <CardContent className="p-4">
    <div className="font-semibold flex items-center gap-2">
      <Camera size={16} /> Photos
    </div>
    <div className="mt-2 grid grid-cols-3 gap-2">
      {STOP.photos.map((p) => (
        <div key={p.id} className="aspect-video rounded-lg bg-gray-200 flex items-center justify-center text-xs text-gray-600">
          {p.caption}
        </div>
      ))}
    </div>
  </CardContent>
</Card>
</div>

{/* Right column */}
<div className="space-y-6">
  <Card className="border border-gray-100">
    <CardContent className="p-4">
      <div className="font-semibold mb-2">Routes serving this stop</div>
      <div className="flex flex-wrap gap-2">
        {STOP.routes.map((r) => (
          <a key={r.id} href={`#/routes/${r.id}`} className="inline-flex items-center gap-2 px-3 py-1.5 rounded-full border text-sm bg-white hover:bg-teal-50" style={{ borderColor: "#e5e7eb" }}>
            <span className="inline-flex items-center justify-center h-6 w-8 rounded-md text-white" style={{ background: r.color }}>{r.id}</span>
          </a>
        ))}
      </div>
    </CardContent>
  </Card>
</div>

```

```

        <span className="text-gray-800">{r.name}</span>
      </a>
    )))
  </div>
</CardContent>
</Card>

<Card className="border border-gray-100">
  <CardContent className="p-4">
    <LiveArrivals />
  </CardContent>
</Card>

<Card className="border border-gray-100">
  <CardContent className="p-4 space-y-3 text-sm">
    <div className="font-semibold flex items-center gap-2">
      <Accessibility size={16} /> Accessibility
    </div>
    <div className="grid grid-cols-2 gap-2">
      <div className="flex items-center gap-2"><CheckCircle size={16}
className={STOP.accessibility.curbRamp ? "text-emerald-600" : "text-gray-300"}/> Curb
ramp</div>
      <div className="flex items-center gap-2"><CheckCircle size={16}
className={STOP.accessibility.seating ? "text-emerald-600" : "text-gray-300"}/> Seating</div>
      <div className="flex items-center gap-2"><CheckCircle size={16}
className={STOP.accessibility.lighting ? "text-emerald-600" : "text-gray-300"}/> Lighting</div>
      <div className="flex items-center gap-2"><CheckCircle size={16}
className={STOP.accessibility.tactilePaving ? "text-emerald-600" : "text-gray-300"}/> Tactile
paving</div>
    </div>
    <div className="text-xs text-gray-500">Community-sourced. Accuracy improves as
riders contribute.</div>
  </CardContent>
</Card>

<Card className="border border-gray-100">
  <CardContent className="p-4 space-y-3 text-sm">
    <div className="font-semibold flex items-center gap-2"><AlertTriangle size={16}/>
Service alerts</div>
    <div className="rounded-md border p-3 bg-amber-50 text-amber-900 text-xs">No
active alerts for this stop.</div>
    <div className="font-semibold flex items-center gap-2"><Flag size={16}/> Report an
issue</div>
  </CardContent>
</Card>

```

```
        <div className="text-xs text-gray-600">Broken shelter? Incorrect location? <a
href="#/report" className="text-teal-700 underline">Send a quick report</a>.</div>
```

```
        </CardContent>
```

```
    </Card>
```

```
  </div>
```

```
</div>
```

```
{/* Dev tests */}
```

```
<div className="px-4 md:px-6 pb-10">
```

```
  <DevTests />
```

```
</div>
```

```
</div>
```

```
);
```

```
}
```

New Route

Preview Route Page Save Route

Route Number

Route Name

Via Roads

Default Fare
e.g., TSh 750

Upload Route Path

Upload .kml (from Google Earth / OSM) or a .csv containing lat,lon columns.

Choose File No file chosen

Upload a .kml or .csv to see a route path preview

Tips

- Start and end stops define route direction.
- Use accurate coordinates to improve map precision.
- CSV headers can be lat,lon or latitude,longitude.
- Click preview to visualize your route page design.

Stops & Stages + Add Stop

Stop Name	Latitude	Longitude

```
import React, { useState, useMemo } from 'react';
import { Map, Plus, Save, Eye, Bus, X, Upload, Trash2, Info } from 'lucide-react';
import { Card, CardContent } from '@components/ui/card';
```

```
export default function NewRoutePage() {
  const [stops, setStops] = useState([ { name: "", lat: "", lon: "" } ]);
  const [routeName, setRouteName] = useState("");
  const [routeNumber, setRouteNumber] = useState("");
  const [via, setVia] = useState("");
  const [fare, setFare] = useState("");
  const [previewMode, setPreviewMode] = useState(false);
```

```
// NEW: uploaded path (from KML/CSV)
const [routePath, setRoutePath] = useState([]); // [{lat, lon}]
const [uploadError, setUploadError] = useState("");
```

```
const handleAddStop = () => {
  setStops([ ...stops, { name: "", lat: "", lon: "" } ]);
};
```

```
const handleRemoveStop = (index) => {
  const newStops = stops.filter((_, i) => i !== index);
  setStops(newStops);
};
```

```

};

const handleChangeStop = (index, field, value) => {
  const newStops = [...stops];
  newStops[index][field] = value;
  setStops(newStops);
};

// -----
// File parsing
// -----
const parseCSV = (text) => {
  const lines = text.split(/\r?\n/).map(l => l.trim()).filter(Boolean);
  if (!lines.length) return [];
  const header = lines[0].toLowerCase().split(/[;,\t]/).map(h => h.trim());
  const hasHeader = header.some(h => ['lat','latitude','y'].includes(h)) && header.some(h =>
['lon','lng','longitude','x'].includes(h));
  const rows = hasHeader ? lines.slice(1) : lines;
  const idxLat = hasHeader ? header.findIndex(h => ['lat','latitude','y'].includes(h)) : 0;
  const idxLon = hasHeader ? header.findIndex(h => ['lon','lng','longitude','x'].includes(h)) : 1;
  const out = [];
  for (const line of rows) {
    const cols = line.split(/[;,\t]/).map(c => c.trim());
    const lat = parseFloat(cols[idxLat]);
    const lon = parseFloat(cols[idxLon]);
    if (Number.isFinite(lat) && Number.isFinite(lon)) out.push({ lat, lon });
  }
  return out;
};

const parseKML = (text) => {
  // naive parse: collect all <coordinates>...lon,lat(,alt)...</coordinates>
  const coords = [];
  const regex = /<coordinates>([\s\S]*?)<\/coordinates>/gi;
  let m;
  while ((m = regex.exec(text)) !== null) {
    const block = m[1].trim();
    const pairs = block.split(/\s+/);
    for (const p of pairs) {
      const [lonStr, latStr] = p.split(',');
      const lat = parseFloat(latStr);
      const lon = parseFloat(lonStr);
      if (Number.isFinite(lat) && Number.isFinite(lon)) coords.push({ lat, lon });
    }
  }
}

```



```

    }
    return coords;
};

```

```

const onUpload = async (file) => {
  setUploadError("");
  if (!file) return;
  try {
    const text = await file.text();
    const ext = (file.name.split('.').pop() || "").toLowerCase();
    let pts = [];
    if (ext === 'csv') pts = parseCSV(text);
    else if (ext === 'kml') pts = parseKML(text);
    else throw new Error('Unsupported file type. Please upload .kml or .csv');
    if (!pts.length) throw new Error('No coordinates found. Ensure your file has lat/lon pairs. ');
    setRoutePath(pts);
  } catch (e) {
    setRoutePath([]);
    setUploadError(e.message || 'Failed to parse file');
  }
};

```

```

const clearPath = () => { setRoutePath([]); setUploadError(""); };

```

```

// Simple SVG preview for uploaded path
const PathPreview = ({ points, height = 256 }) => {
  const pad = 16;
  const width = 480;
  if (!points || points.length < 2) return (
    <div className="h-64 w-full rounded-lg bg-gray-100 flex items-center justify-center
text-gray-400 text-sm">
      Upload a .kml or .csv to see a route path preview
    </div>
  );
  const lats = points.map(p => p.lat);
  const lons = points.map(p => p.lon);
  const minLat = Math.min(...lats), maxLat = Math.max(...lats);
  const minLon = Math.min(...lons), maxLon = Math.max(...lons);
  const spanLat = Math.max(maxLat - minLat, 1e-6);
  const spanLon = Math.max(maxLon - minLon, 1e-6);
  const project = (pt) => {
    const x = pad + (pt.lon - minLon) / spanLon * (width - 2*pad);
    const y = pad + (1 - (pt.lat - minLat) / spanLat) * (height - 2*pad);
    return [x, y];
  }

```

```

};
const d = points.map(project).map(([x,y]) => `${x},${y}`).join(' ');
return (
  <svg viewBox={`0 0 ${width} ${height}`} className="w-full h-64 rounded-lg bg-gray-100">
    <defs>
      <pattern id="grid16" width="16" height="16" patternUnits="userSpaceOnUse">
        <path d="M 16 0 L 0 0 0 16" fill="none" stroke="#e5e7eb" strokeWidth="1" />
      </pattern>
    </defs>
    <rect x={0} y={0} width={width} height={height} fill="url(#grid16)" />
    <polyline points={d} fill="none" stroke="#2B7FFF" strokeWidth={3} strokeLinecap="round"
strokeLinejoin="round" />
    {points.map((p, i) => {
      const [x,y] = project(p);
      return <circle key={i} cx={x} cy={y} r={3} fill="#111827" />;
    })}
  </svg>
);
};

```

```

const pathMeta = useMemo(() => ({ count: routePath.length }), [routePath]);

```

```

return (
  <div className="min-h-screen bg-gray-50 text-gray-800">
    {/* Header */}
    <header className="flex items-center justify-between bg-white px-6 py-4 border-b">
      <h1 className="text-2xl font-semibold flex items-center gap-2">
        <Bus size={22} className="text-teal-600" />
        New Route
      </h1>
      <div className="flex items-center gap-3">
        <button className="px-4 py-2 border rounded-md hover:bg-teal-50" onClick={() =>
setPreviewMode(!previewMode)}>
          <Eye size={16} className="inline mr-1" /> {previewMode ? 'Hide Route Preview' :
'Preview Route Page'}
        </button>
        <button className="px-4 py-2 bg-teal-600 text-white rounded-md hover:bg-teal-700 flex
items-center gap-1">
          <Save size={16} /> Save Route
        </button>
      </div>
    </header>

    {previewMode ? (

```

```

<div className="p-6">
  { /* Preview Simulation of the Route Page */ }
  <Card className="border border-gray-200">
    <CardContent className="p-6 space-y-4">
      <h2 className="text-xl font-semibold text-teal-700">{routeNumber || '00'} -
{routeName || 'Route Name'}</h2>
      <p className="text-sm text-gray-600">Via {via || 'main roads'}</p>
      <p className="text-sm text-gray-700">Default Fare: {fare || 'TSh 0'}</p>
      <div className="mt-4">
        <h3 className="text-md font-semibold mb-2">Stops on this route:</h3>
        <ul className="list-disc pl-5 space-y-1 text-sm text-gray-700">
          {stops.map((s, i) => (
            <li key={i}>{s.name || `Stop ${i + 1}`} {s.lat || 'lat'}, {s.lon || 'lon'}</li>
          ))}
        </ul>
      </div>
      <div>
        <div className="flex items-center gap-2 text-sm text-gray-600 mb-2">
          <Map size={16}/> Route path preview {pathMeta.count ? <span className="ml-1">•
{pathMeta.count} points</span> : null}
        </div>
        <PathPreview points={routePath} />
        {uploadError ? <div className="mt-2 text-xs text-rose-600">{uploadError}</div> :
null}
      </div>
    </CardContent>
  </Card>
</div>
) : (
  <main className="grid grid-cols-1 lg:grid-cols-3 gap-6 px-6 py-6">
    { /* Left - Form */ }
    <section className="lg:col-span-2 space-y-6">
      <Card className="border border-gray-200">
        <CardContent className="p-6 space-y-4">
          <div>
            <label className="block text-sm font-medium text-gray-700">Route
Number</label>
            <input type="text" className="mt-1 p-2 w-full border rounded-md"
value={routeNumber} onChange={(e) => setRouteNumber(e.target.value)} />
          </div>
          <div>
            <label className="block text-sm font-medium text-gray-700">Route Name</label>
            <input type="text" className="mt-1 p-2 w-full border rounded-md"
value={routeName} onChange={(e) => setRouteName(e.target.value)} />

```

```

    </div>
    <div>
      <label className="block text-sm font-medium text-gray-700">Via Roads</label>
      <textarea className="mt-1 p-2 w-full border rounded-md" rows={2} value={via}
onChange={(e) => setVia(e.target.value)} />
    </div>
    <div>
      <label className="block text-sm font-medium text-gray-700">Default Fare</label>
      <input type="text" className="mt-1 p-2 w-full border rounded-md" value={fare}
onChange={(e) => setFare(e.target.value)} placeholder="e.g., TSh 750" />
    </div>
  </CardContent>
</Card>

<Card className="border border-gray-200">
  <CardContent className="p-6">
    <div className="flex items-center justify-between mb-4">
      <h2 className="text-lg font-semibold">Stops & Stages</h2>
      <button className="flex items-center gap-1 px-2 py-1 border rounded
hover:bg-teal-50" onClick={handleAddStop}>
        <Plus size={14} /> Add Stop
      </button>
    </div>

    <div className="space-y-3">
      {stops.map((stop, index) => (
        <div key={index} className="grid grid-cols-6 gap-2 items-center">
          <input type="text" placeholder="Stop Name" value={stop.name} onChange={(e)
=> handleChangeStop(index, 'name', e.target.value)} className="col-span-2 p-2 border
rounded-md" />
          <input type="text" placeholder="Latitude" value={stop.lat} onChange={(e) =>
handleChangeStop(index, 'lat', e.target.value)} className="col-span-2 p-2 border rounded-md"
/>
          <input type="text" placeholder="Longitude" value={stop.lon} onChange={(e) =>
handleChangeStop(index, 'lon', e.target.value)} className="col-span-2 p-2 border
rounded-md" />
          <button onClick={() => handleRemoveStop(index)} className="text-red-500
hover:text-red-700">
            <X size={16} />
          </button>
        </div>
      )))
    </div>
  </CardContent>

```

```

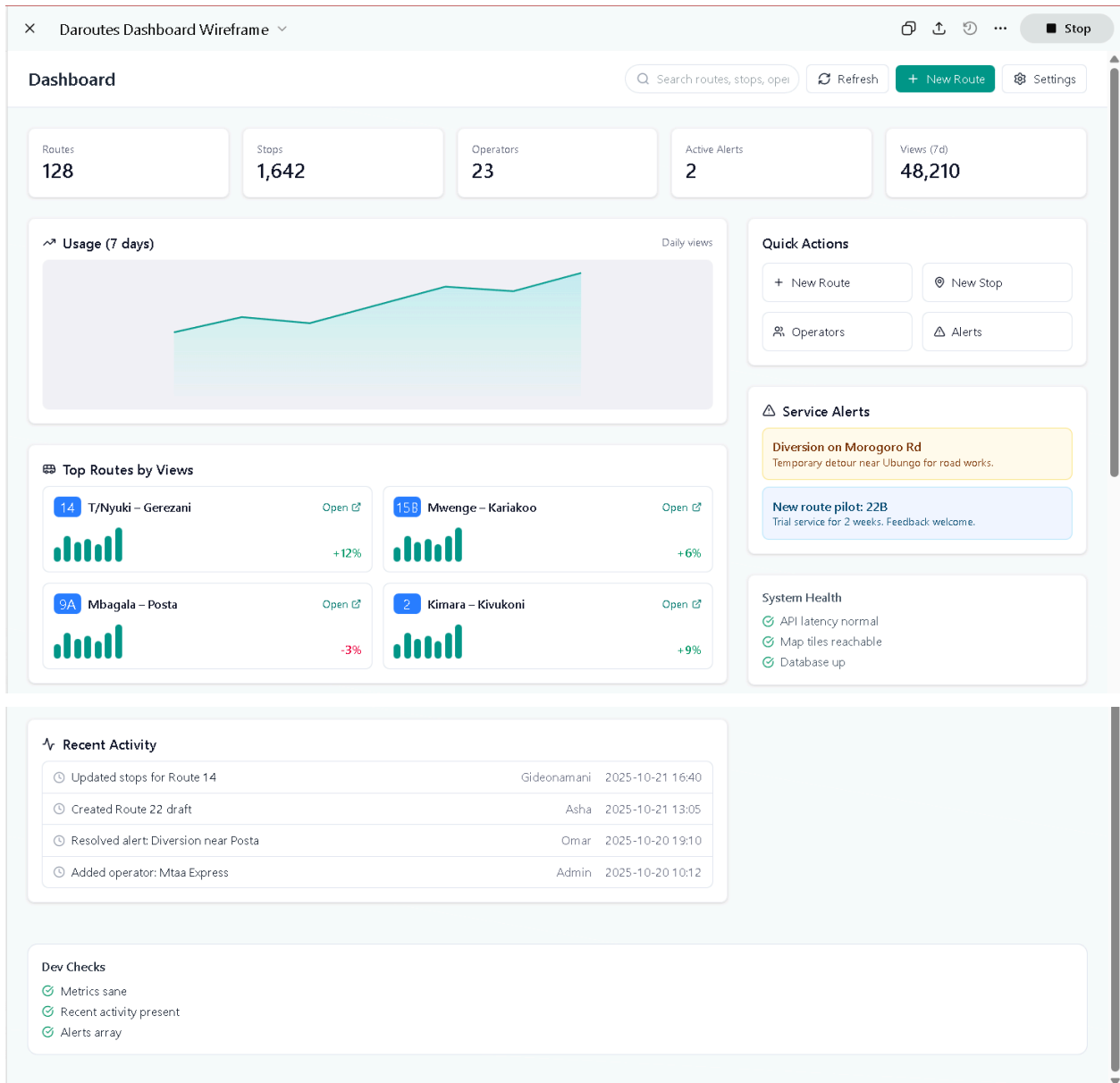
</Card>
</section>

{/* Right - Uploads, Map & Tips */}
<section className="space-y-6">
  {/* Upload card */}
  <Card className="border border-gray-200">
    <CardContent className="p-6 space-y-3">
      <div className="flex items-center gap-2 mb-1">
        <Upload size={18} className="text-gray-600" />
        <h2 className="text-lg font-semibold">Upload Route Path</h2>
      </div>
      <p className="text-xs text-gray-600 flex items-start gap-2"><Info size={14}/> Upload
a <strong>.kml</strong> (from Google Earth / OSM) or a <strong>.csv</strong> containing
<em>lat,lon</em> columns.</p>
      <input
        type="file"
        accept=".kml,.csv"
        onChange={(e) => onUpload(e.target.files && e.target.files[0])}
        className="block w-full text-sm text-gray-600 file:mr-3 file:py-2 file:px-3
file:rounded-md file:border file:border-gray-300 file:bg-white hover:file:bg-gray-50"
      />
      {routePath.length > 0 && (
        <div className="flex items-center justify-between text-sm mt-1">
          <span className="text-gray-700">Loaded <b>{routePath.length}</b>
points</span>
          <button onClick={clearPath} className="inline-flex items-center gap-1 px-2 py-1
text-rose-600 hover:text-rose-700"><Trash2 size={14}/> Clear</button>
        </div>
      )}
      {uploadError ? <div className="text-xs text-rose-600">{uploadError}</div> : null}
      <div className="mt-2">
        <PathPreview points={routePath} />
      </div>
    </CardContent>
  </Card>

  {/* Tips card */}
  <Card className="border border-gray-200">
    <CardContent className="p-6 space-y-3 text-sm text-gray-600">
      <div className="text-gray-800 font-semibold">Tips</div>
      <ul className="list-disc pl-4 space-y-1">
        <li>Start and end stops define route direction.</li>
        <li>Use accurate coordinates to improve map precision.</li>

```

```
        <li>CSV headers can be <code>lat,lon</code> or
<code>latitude,longitude</code>.</li>
        <li>Click preview to visualize your route page design.</li>
    </ul>
</CardContent>
</Card>
</section>
</main>
    )}
</div>
);
}
```



```
import React, { useMemo, useState, useEffect } from "react";
import {
  Plus,
  Bus,
  MapPin,
  Users,
  AlertTriangle,
  TrendingUp,
  Activity,
  Clock,
  RefreshCw,
  Settings,
  Search,
```

```

    ExternalLink,
    CheckCircle,
    XCircle,
  } from "lucide-react";
import { Card, CardContent } from "@components/ui/card";

// Palette tokens (aligned with project)
const PALETTE = {
  primary: "#009688",
  primaryDark: "#00796B",
  accent: "#5CE1E6",
  bg: "#F9FAFB",
  surface: "#FFFFFF",
  text: "#1F2937",
  textMuted: "#6B7280",
};

// Mock data
const METRICS = {
  routes: 128,
  stops: 1642,
  operators: 23,
  activeAlerts: 2,
  views7d: 48210,
};

const TOP_ROUTES = [
  { id: "14", name: "T/Nyuki – Gerezani", views: 8210, trend: +12 },
  { id: "15B", name: "Mwenge – Kariakoo", views: 6780, trend: +6 },
  { id: "9A", name: "Mbagala – Posta", views: 5421, trend: -3 },
  { id: "2", name: "Kimara – Kivukoni", views: 4980, trend: +9 },
];

const RECENT_ACTIVITY = [
  { ts: "2025-10-21 16:40", who: "Gideonamani", what: "Updated stops for Route 14", type: "edit" },
  { ts: "2025-10-21 13:05", who: "Asha", what: "Created Route 22 draft", type: "create" },
  { ts: "2025-10-20 19:10", who: "Omar", what: "Resolved alert: Diversion near Posta", type: "resolve" },
  { ts: "2025-10-20 10:12", who: "Admin", what: "Added operator: Mtaa Express", type: "create" },
];

const ALERTS = [

```



```

    { id: 1, level: "warning", title: "Diversion on Morogoro Rd", detail: "Temporary detour near Ubungu for road works." },
    { id: 2, level: "info", title: "New route pilot: 22B", detail: "Trial service for 2 weeks. Feedback welcome." },
  ];

```

// Mini bar sparkline placeholder

```

function SparkBars({ values }: { values: number[] }) {
  const max = Math.max(...values, 1);
  return (
    <div className="flex items-end gap-1 h-10">
      {values.map((v, i) => (
        <div key={i} className="w-2 rounded-sm bg-teal-600" style={{ height: `${(v / max) * 100}%` }} />
      ))}
    </div>
  );
}

```

// Simple Area-like chart placeholder

```

function AreaChart({ series }: { series: number[] }) {
  const width = 520, height = 180, pad = 16;
  const max = Math.max(...series, 1), min = Math.min(...series, 0);
  const span = Math.max(max - min, 1);
  const pts = series.map((v, i) => {
    const x = pad + (i / (series.length - 1)) * (width - 2 * pad);
    const y = pad + (1 - (v - min) / span) * (height - 2 * pad);
    return `${x},${y}`;
  });
  const path = `M ${pts.join(" L ")}`;
  return (
    <svg viewBox={`0 0 ${width} ${height}`} className="w-full h-44 rounded-lg bg-gray-100">
      <defs>
        <linearGradient id="g" x1="0" y1="0" x2="0" y2="1">
          <stop offset="0%" stopColor="#5CE1E6" stopOpacity="0.5" />
          <stop offset="100%" stopColor="#5CE1E6" stopOpacity="0.05" />
        </linearGradient>
      </defs>
      <polyline points={pts.join(" ")} fill="none" stroke="#009688" strokeWidth={2} />
      <path d={` ${path} L ${width - pad},${height - pad} L ${pad},${height - pad} Z`} fill="url(#g)"
        opacity={0.6} />
    </svg>
  );
}

```

```

// Dev checks
function DevTests() {
  const tests = [
    { name: "Metrics sane", pass: METRICS.routes > 0 && METRICS.stops > 0 },
    { name: "Recent activity present", pass: RECENT_ACTIVITY.length > 0 },
    { name: "Alerts array", pass: Array.isArray(ALERTS) },
  ];
  return (
    <div className="mt-6 p-4 rounded-xl border bg-white">
      <div className="text-sm font-semibold mb-2">Dev Checks</div>
      <ul className="text-sm space-y-1">
        {tests.map((t) => (
          <li key={t.name} className="flex items-center gap-2">
            {t.pass ? <CheckCircle size={14} className="text-emerald-600" /> : <XCircle size={14}
className="text-rose-600" />}
            <span>{t.name}</span>
          </li>
        ))}
      </ul>
    </div>
  );
}

```

```

export default function Dashboard() {
  const [loading, setLoading] = useState(false);
  const [query, setQuery] = useState("");
  const viewsSeries = useMemo(() => [4200, 5200, 4800, 6000, 7200, 6900, 8100], []);

  useEffect(() => {
    // simulate refresh
    let t: any;
    if (loading) t = setTimeout(() => setLoading(false), 800);
    return () => clearTimeout(t);
  }, [loading]);

  return (
    <div className="min-h-screen" style={{ backgroundColor: PALETTE.bg, color:
PALETTE.text }}>
      </* Header */>
      <div className="px-4 md:px-6 py-4 flex items-center gap-3 bg-white border-b">
        <div className="text-xl font-semibold">Dashboard</div>
        <div className="ml-auto flex items-center gap-2">
          <div className="hidden md:flex items-center bg-white border rounded-full px-3 py-1.5">

```

```

        <Search size={16} className="text-gray-400 mr-2" />
        <input value={query} onChange={(e) => setQuery(e.target.value)} placeholder="Search
routes, stops, operators..." className="outline-none text-sm text-gray-700" />
      </div>
      <button onClick={() => setLoading(true)} className="inline-flex items-center gap-2 px-3
py-1.5 rounded-md border bg-white hover:bg-gray-50 text-sm">
        <RefreshCw size={16} className={loading ? "animate-spin" : ""} /> Refresh
      </button>
      <button className="inline-flex items-center gap-2 px-3 py-1.5 rounded-md bg-teal-600
text-white text-sm hover:bg-teal-700">
        <Plus size={16} /> New Route
      </button>
      <button className="inline-flex items-center gap-2 px-3 py-1.5 rounded-md border
bg-white hover:bg-gray-50 text-sm">
        <Settings size={16} /> Settings
      </button>
    </div>
  </div>

```

```

  {/* Content */}
  <div className="px-4 md:px-6 py-6 grid grid-cols-1 xl:grid-cols-3 gap-6">
    {/* KPIs */}
    <div className="xl:col-span-3 grid grid-cols-2 md:grid-cols-4 lg:grid-cols-5 gap-4">
      <Card className="border border-gray-100"><CardContent className="p-4"><div
className="text-xs text-gray-500">Routes</div><div className="text-2xl
font-semibold">{METRICS.routes}</div></CardContent></Card>
      <Card className="border border-gray-100"><CardContent className="p-4"><div
className="text-xs text-gray-500">Stops</div><div className="text-2xl
font-semibold">{METRICS.stops.toLocaleString()}</div></CardContent></Card>
      <Card className="border border-gray-100"><CardContent className="p-4"><div
className="text-xs text-gray-500">Operators</div><div className="text-2xl
font-semibold">{METRICS.operators}</div></CardContent></Card>
      <Card className="border border-gray-100"><CardContent className="p-4"><div
className="text-xs text-gray-500">Active Alerts</div><div className="text-2xl
font-semibold">{METRICS.activeAlerts}</div></CardContent></Card>
      <Card className="border border-gray-100"><CardContent className="p-4"><div
className="text-xs text-gray-500">Views (7d)</div><div className="text-2xl
font-semibold">{METRICS.views7d.toLocaleString()}</div></CardContent></Card>
    </div>

```

```

  {/* Left column: charts & tables */}
  <div className="xl:col-span-2 space-y-6">
    <Card className="border border-gray-100">
      <CardContent className="p-4">

```

```

    <div className="flex items-center justify-between mb-2">
      <div className="font-semibold flex items-center gap-2"><TrendingUp size={16}/>
Usage (7 days)</div>
      <div className="text-xs text-gray-500">Daily views</div>
    </div>
    <AreaChart series={viewsSeries} />
  </CardContent>
</Card>

```

```

<Card className="border border-gray-100">
  <CardContent className="p-4">
    <div className="font-semibold mb-2 flex items-center gap-2"><Bus size={16}/> Top
Routes by Views</div>
    <div className="grid grid-cols-1 md:grid-cols-2 gap-3">
      {TOP_ROUTES.map(r => (
        <div key={r.id} className="rounded-lg border p-3 bg-white">
          <div className="flex items-center justify-between">
            <div className="flex items-center gap-2">
              <span className="inline-flex items-center justify-center h-6 w-8 rounded-md
text-white" style={{ background: '#2B7FFF' }}>{r.id}</span>
              <div className="text-sm font-medium">{r.name}</div>
            </div>
            <a href={`#/routes/${r.id}`} className="text-xs text-teal-700 inline-flex
items-center gap-1">Open <ExternalLink size={12}/></a>
          </div>
          <div className="mt-3 flex items-end justify-between">
            <SparkBars values={[5,9,7,8,6,9,12]} />
            <div className={`text-sm font-semibold ${r.trend}>=0?
'text-emerald-600':'text-rose-600'}`>{r.trend}>=0? '+'+r.trend: r.trend}%</div>
          </div>
        </div>
      )))
    </div>
  </CardContent>
</Card>

```

```

<Card className="border border-gray-100">
  <CardContent className="p-4">
    <div className="font-semibold mb-2 flex items-center gap-2"><Activity size={16}/>
Recent Activity</div>
    <div className="divide-y rounded-lg border bg-white overflow-hidden">
      {RECENT_ACTIVITY.map((a, i) => (
        <div key={i} className="flex items-center justify-between px-3 py-2 text-sm">
          <div className="flex items-center gap-2">

```

```

        <Clock size={14} className="text-gray-400" />
        <span className="text-gray-800">{a.what}</span>
      </div>
      <div className="flex items-center gap-4 text-gray-500">
        <span>{a.who}</span>
        <span>{a.ts}</span>
      </div>
    </div>
  )}}
</div>
</CardContent>
</Card>
</div>

{ /* Right column: actions & alerts */}
<div className="space-y-6">
  <Card className="border border-gray-100">
    <CardContent className="p-4">
      <div className="font-semibold mb-3">Quick Actions</div>
      <div className="grid grid-cols-2 gap-3 text-sm">
        <a href="#/new-route" className="rounded-lg border p-3 bg-white hover:bg-teal-50
inline-flex items-center gap-2"><Plus size={14}/> New Route</a>
        <a href="#/new-stop" className="rounded-lg border p-3 bg-white hover:bg-teal-50
inline-flex items-center gap-2"><MapPin size={14}/> New Stop</a>
        <a href="#/operators" className="rounded-lg border p-3 bg-white hover:bg-teal-50
inline-flex items-center gap-2"><Users size={14}/> Operators</a>
        <a href="#/alerts" className="rounded-lg border p-3 bg-white hover:bg-teal-50
inline-flex items-center gap-2"><AlertTriangle size={14}/> Alerts</a>
      </div>
    </CardContent>
  </Card>

  <Card className="border border-gray-100">
    <CardContent className="p-4">
      <div className="font-semibold mb-2 flex items-center gap-2"><AlertTriangle
size={16}/> Service Alerts</div>
      <div className="space-y-2">
        {ALERTS.map(al => (
          <div key={al.id} className={`rounded-lg border p-3 text-sm ${al.level==='warning'?
'bg-amber-50 border-amber-200 text-amber-900':'bg-sky-50 border-sky-200 text-sky-900'}`}>
            <div className="font-medium">{al.title}</div>
            <div className="text-xs opacity-90">{al.detail}</div>
          </div>
        ))}
      </div>
    </CardContent>
  </Card>

```

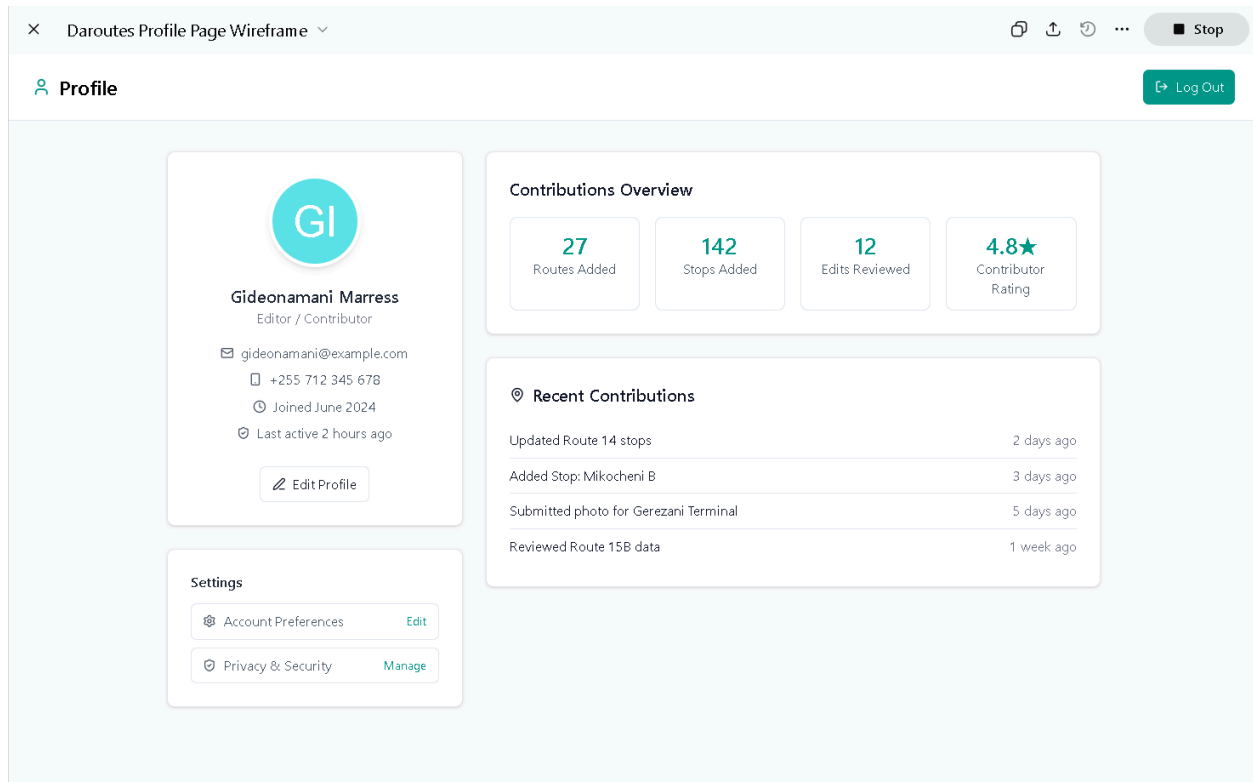
```

    </div>
  </CardContent>
</Card>

<Card className="border border-gray-100">
  <CardContent className="p-4 text-sm text-gray-600">
    <div className="font-semibold mb-2">System Health</div>
    <ul className="space-y-1">
      <li className="flex items-center gap-2"><CheckCircle size={14}
className="text-emerald-600"/> API latency normal</li>
      <li className="flex items-center gap-2"><CheckCircle size={14}
className="text-emerald-600"/> Map tiles reachable</li>
      <li className="flex items-center gap-2"><CheckCircle size={14}
className="text-emerald-600"/> Database up</li>
    </ul>
  </CardContent>
</Card>
</div>
</div>

<div className="px-4 md:px-6 pb-10">
  <DevTests />
</div>
</div>
);
}

```



```
import React from 'react';
import { User, Edit3, LogOut, Settings, Mail, Smartphone, ShieldCheck, Clock, MapPin } from
' lucide-react';
import { Card, CardContent } from '@components/ui/card';

export default function ProfilePage() {
  const user = {
    name: 'Gideonamani Marress',
    role: 'Editor / Contributor',
    email: 'gideonamani@example.com',
    phone: '+255 712 345 678',
    joined: 'June 2024',
    routesContributed: 27,
    stopsAdded: 142,
    lastActive: '2 hours ago',
    avatar:
'https://api.dicebear.com/7.x/initials/svg?seed=Gideonamani&backgroundColor=5CE1E6',
  };

  return (
    <div className="min-h-screen bg-gray-50">
      { /* Header */ }
      <header className="bg-white border-b px-6 py-4 flex items-center justify-between">
        <h1 className="text-xl font-semibold flex items-center gap-2">
```

```

    <User className="text-teal-600" size={20}/> Profile
  </h1>
  <button className="flex items-center gap-2 text-sm px-3 py-2 rounded-md bg-teal-600
text-white hover:bg-teal-700">
    <Logout size={14}/> Log Out
  </button>
</header>

{/* Main Content */}
<main className="max-w-5xl mx-auto px-6 py-8 grid grid-cols-1 lg:grid-cols-3 gap-6">
  {/* Left: User Info */}
  <section className="space-y-6">
    <Card className="border border-gray-200">
      <CardContent className="p-6 flex flex-col items-center text-center">
        <img src={user.avatar} alt="avatar" className="w-24 h-24 rounded-full border-4
border-white shadow-md mb-4" />
        <h2 className="text-lg font-semibold text-gray-800">{user.name}</h2>
        <p className="text-sm text-gray-500">{user.role}</p>

        <div className="mt-4 flex flex-col items-center gap-2 text-sm text-gray-600">
          <div className="flex items-center gap-2"><Mail size={14}/> {user.email}</div>
          <div className="flex items-center gap-2"><Smartphone size={14}/>
{user.phone}</div>
          <div className="flex items-center gap-2"><Clock size={14}/> Joined
{user.joined}</div>
          <div className="flex items-center gap-2"><ShieldCheck size={14}/> Last active
{user.lastActive}</div>
        </div>

        <button className="mt-6 flex items-center gap-2 text-sm px-3 py-2 border
rounded-md hover:bg-gray-50">
          <Edit3 size={14}/> Edit Profile
        </button>
      </CardContent>
    </Card>

    <Card className="border border-gray-200">
      <CardContent className="p-6 space-y-3 text-sm text-gray-600">
        <div className="font-semibold text-gray-800">Settings</div>
        <div className="space-y-2">
          <button className="w-full flex items-center justify-between border rounded-md px-3
py-2 hover:bg-teal-50">
            <span className="flex items-center gap-2"><Settings size={14}/> Account
Preferences</span>

```



```

        <span className="text-teal-600 text-xs">Edit</span>
      </button>
      <button className="w-full flex items-center justify-between border rounded-md px-3
py-2 hover:bg-teal-50">
        <span className="flex items-center gap-2"><ShieldCheck size={14}/> Privacy &
Security</span>
        <span className="text-teal-600 text-xs">Manage</span>
      </button>
    </div>
  </CardContent>
</Card>
</section>

```

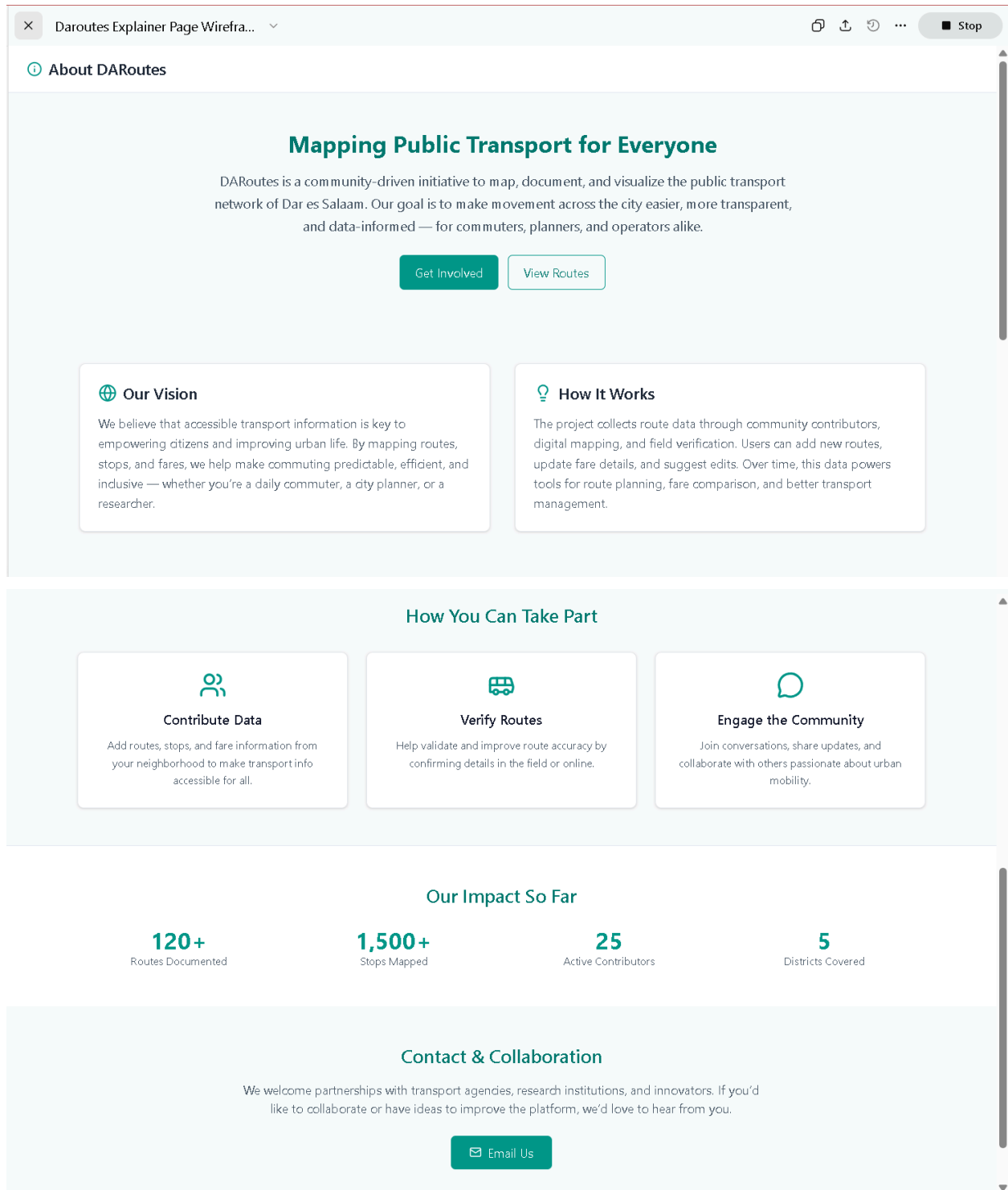
```

{ /* Right: Contributions Overview */ }
<section className="lg:col-span-2 space-y-6">
  <Card className="border border-gray-200">
    <CardContent className="p-6">
      <h2 className="text-lg font-semibold mb-4">Contributions Overview</h2>
      <div className="grid grid-cols-2 md:grid-cols-4 gap-4 text-center">
        <div className="p-3 bg-white rounded-lg border">
          <div className="text-2xl font-semibold
text-teal-600">{user.routesContributed}</div>
          <div className="text-sm text-gray-600">Routes Added</div>
        </div>
        <div className="p-3 bg-white rounded-lg border">
          <div className="text-2xl font-semibold text-teal-600">{user.stopsAdded}</div>
          <div className="text-sm text-gray-600">Stops Added</div>
        </div>
        <div className="p-3 bg-white rounded-lg border">
          <div className="text-2xl font-semibold text-teal-600">12</div>
          <div className="text-sm text-gray-600">Edits Reviewed</div>
        </div>
        <div className="p-3 bg-white rounded-lg border">
          <div className="text-2xl font-semibold text-teal-600">4.8★</div>
          <div className="text-sm text-gray-600">Contributor Rating</div>
        </div>
      </div>
    </CardContent>
  </Card>

  <Card className="border border-gray-200">
    <CardContent className="p-6">
      <h2 className="text-lg font-semibold mb-4 flex items-center gap-2"><MapPin
size={16}/> Recent Contributions</h2>

```

```
        <ul className="divide-y text-sm">
          <li className="py-2 flex items-center justify-between"><span>Updated Route 14
stops</span><span className="text-gray-500">2 days ago</span></li>
          <li className="py-2 flex items-center justify-between"><span>Added Stop:
Mikocheni B</span><span className="text-gray-500">3 days ago</span></li>
          <li className="py-2 flex items-center justify-between"><span>Submitted photo for
Gerezani Terminal</span><span className="text-gray-500">5 days ago</span></li>
          <li className="py-2 flex items-center justify-between"><span>Reviewed Route 15B
data</span><span className="text-gray-500">1 week ago</span></li>
        </ul>
      </CardContent>
    </Card>
  </section>
</main>
</div>
);
}
```



```
import React from 'react';
import { Info, Globe, Users, Bus, Map, Heart, Lightbulb, MessageCircle, Mail } from
' lucide-react';
import { Card, CardContent } from '@components/ui/card';

export default function ExplainerPage() {
```

```

return (
  <div className="min-h-screen bg-gray-50 text-gray-800">
    { /* Header */ }
    <header className="bg-white border-b px-6 py-4 flex items-center justify-between">
      <h1 className="text-xl font-semibold flex items-center gap-2">
        <Info className="text-teal-600" size={20}/> About DARoutes
      </h1>
    </header>

    { /* Hero Section */ }
    <section className="max-w-6xl mx-auto px-6 py-12 text-center">
      <h2 className="text-3xl font-bold text-teal-700 mb-4">Mapping Public Transport for
Everyone</h2>
      <p className="text-gray-600 max-w-3xl mx-auto text-lg leading-relaxed">
        DARoutes is a community-driven initiative to map, document, and visualize the public
transport network of Dar es Salaam.
        Our goal is to make movement across the city easier, more transparent, and
data-informed — for commuters, planners, and operators alike.
      </p>
      <div className="mt-6 flex justify-center gap-3">
        <button className="px-5 py-2.5 rounded-md bg-teal-600 text-white
hover:bg-teal-700">Get Involved</button>
        <button className="px-5 py-2.5 rounded-md border border-teal-600 text-teal-700
hover:bg-teal-50">View Routes</button>
      </div>
    </section>

    { /* Vision Section */ }
    <section className="max-w-6xl mx-auto px-6 py-12 grid md:grid-cols-2 gap-8">
      <Card className="border border-gray-200">
        <CardContent className="p-6">
          <h3 className="flex items-center gap-2 text-xl font-semibold text-gray-800
mb-3"><Globe className="text-teal-600"/> Our Vision</h3>
          <p className="text-gray-600 leading-relaxed">
            We believe that accessible transport information is key to empowering citizens and
improving urban life. By mapping routes, stops, and fares,
            we help make commuting predictable, efficient, and inclusive — whether you're a daily
commuter, a city planner, or a researcher.
          </p>
        </CardContent>
      </Card>

      <Card className="border border-gray-200">
        <CardContent className="p-6">

```

```
<h3 className="flex items-center gap-2 text-xl font-semibold text-gray-800 mb-3"><Lightbulb className="text-teal-600"/> How It Works</h3>
```

```
<p className="text-gray-600 leading-relaxed">
```

The project collects route data through community contributors, digital mapping, and field verification. Users can add new routes, update fare details, and suggest edits.

Over time, this data powers tools for route planning, fare comparison, and better transport management.

```
</p>
```

```
</CardContent>
```

```
</Card>
```

```
</section>
```

```
{/* Participation Section */}
```

```
<section className="max-w-6xl mx-auto px-6 py-12">
```

```
<h3 className="text-2xl font-semibold text-center text-teal-700 mb-8">How You Can Take Part</h3>
```

```
<div className="grid md:grid-cols-3 gap-6">
```

```
<Card className="border border-gray-200 text-center">
```

```
<CardContent className="p-6 flex flex-col items-center">
```

```
<Users size={36} className="text-teal-600 mb-3" />
```

```
<h4 className="text-lg font-semibold mb-2">Contribute Data</h4>
```

<p className="text-gray-600 text-sm leading-relaxed">Add routes, stops, and fare information from your neighborhood to make transport info accessible for all.</p>

```
</CardContent>
```

```
</Card>
```

```
<Card className="border border-gray-200 text-center">
```

```
<CardContent className="p-6 flex flex-col items-center">
```

```
<Bus size={36} className="text-teal-600 mb-3" />
```

```
<h4 className="text-lg font-semibold mb-2">Verify Routes</h4>
```

<p className="text-gray-600 text-sm leading-relaxed">Help validate and improve route accuracy by confirming details in the field or online.</p>

```
</CardContent>
```

```
</Card>
```

```
<Card className="border border-gray-200 text-center">
```

```
<CardContent className="p-6 flex flex-col items-center">
```

```
<MessageCircle size={36} className="text-teal-600 mb-3" />
```

```
<h4 className="text-lg font-semibold mb-2">Engage the Community</h4>
```

<p className="text-gray-600 text-sm leading-relaxed">Join conversations, share updates, and collaborate with others passionate about urban mobility.</p>

```
</CardContent>
```

```
</Card>
```

```
</div>
```

</section>

{/* Impact Section */}

<section className="bg-white border-t py-12">

<div className="max-w-6xl mx-auto px-6 text-center">

<h3 className="text-2xl font-semibold text-teal-700 mb-6">Our Impact So Far</h3>

<div className="grid grid-cols-2 md:grid-cols-4 gap-4">

<div>

<div className="text-3xl font-bold text-teal-600">120+</div>

<p className="text-sm text-gray-600">Routes Documented</p>

</div>

<div>

<div className="text-3xl font-bold text-teal-600">1,500+</div>

<p className="text-sm text-gray-600">Stops Mapped</p>

</div>

<div>

<div className="text-3xl font-bold text-teal-600">25</div>

<p className="text-sm text-gray-600">Active Contributors</p>

</div>

<div>

<div className="text-3xl font-bold text-teal-600">5</div>

<p className="text-sm text-gray-600">Districts Covered</p>

</div>

</div>

</div>

</section>

{/* Contact Section */}

<section className="max-w-4xl mx-auto px-6 py-12 text-center">

<h3 className="text-2xl font-semibold text-teal-700 mb-4">Contact & Collaboration</h3>

<p className="text-gray-600 max-w-2xl mx-auto mb-6">We welcome partnerships with transport agencies, research institutions, and innovators.

If you'd like to collaborate or have ideas to improve the platform, we'd love to hear from you.</p>

<button className="px-6 py-2.5 rounded-md bg-teal-600 text-white hover:bg-teal-700 inline-flex items-center gap-2">

<Mail size={16}/> Email Us

</button>

</section>

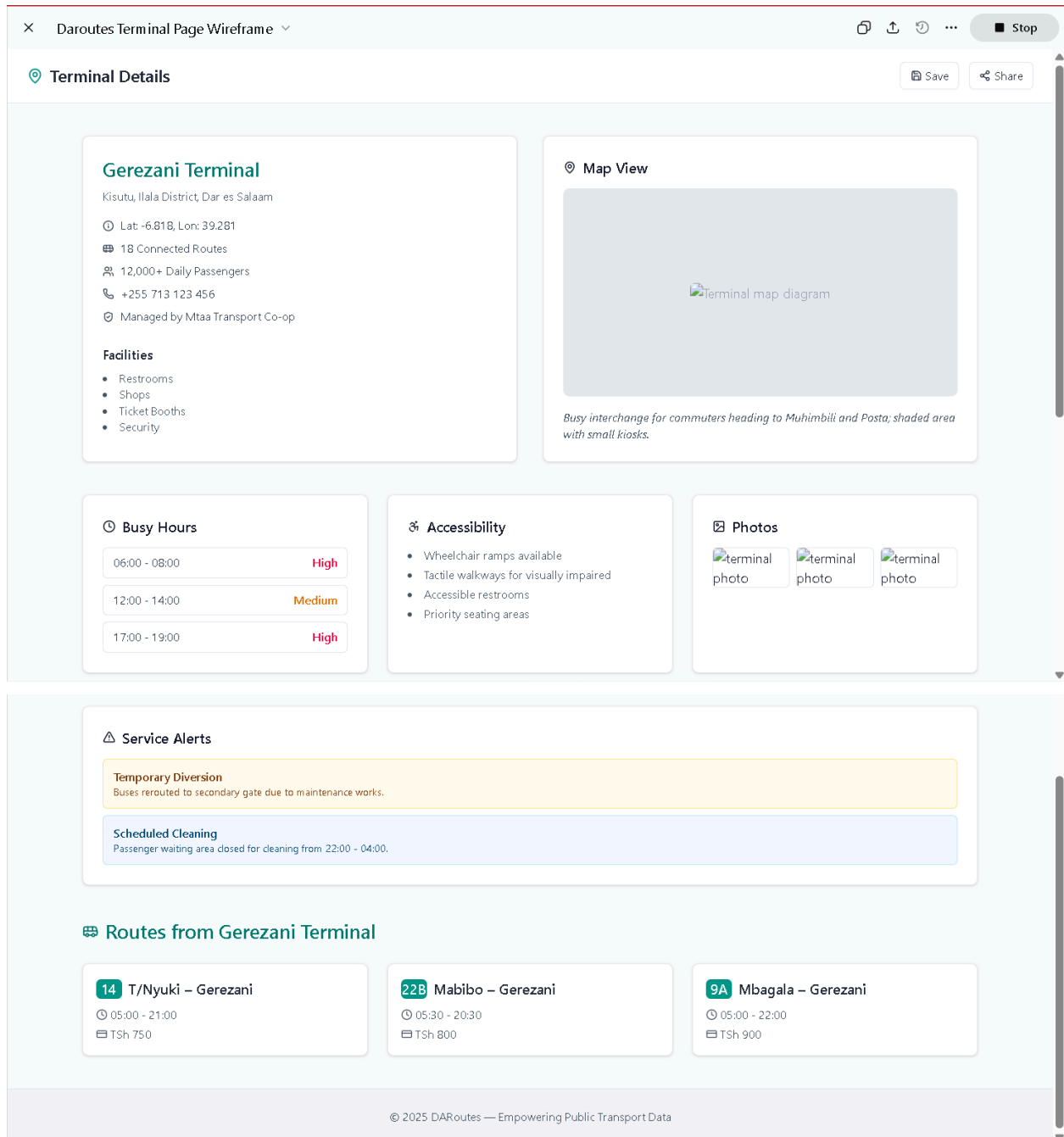
{/* Footer */}

<footer className="bg-gray-100 border-t text-center py-6 text-sm text-gray-600">

<p>© {new Date().getFullYear()} DARoutes • A Community Mobility Initiative</p>

</footer>

```
</div>  
);  
{
```



```
import React from 'react';
import { MapPin, Bus, Users, Clock, CreditCard, Info, Image, Phone, ShieldCheck, Share2,
Save, AlertTriangle, Accessibility } from 'lucide-react';
import { Card, CardContent } from '@components/ui/card';

export default function TerminalPage() {
  const terminal = {
    name: 'Gerezani Terminal',
    location: 'Kisutu, Ilala District, Dar es Salaam',
```



```
coordinates: 'Lat: -6.818, Lon: 39.281',
totalRoutes: 18,
dailyPassengers: '12,000+',
facilities: ['Restrooms', 'Shops', 'Ticket Booths', 'Security'],
contact: '+255 713 123 456',
supervisor: 'Mtaa Transport Co-op',
description: 'Busy interchange for commuters heading to Muhimbili and Posta; shaded area
with small kiosks.'
};
```

```
const connectedRoutes = [
  { id: '14', name: 'T/Nyuki – Gerezani', fare: 'TSh 750', time: '05:00 - 21:00' },
  { id: '22B', name: 'Mabibo – Gerezani', fare: 'TSh 800', time: '05:30 - 20:30' },
  { id: '9A', name: 'Mbagala – Gerezani', fare: 'TSh 900', time: '05:00 - 22:00' },
];
```

```
const gallery = [
  'https://source.unsplash.com/400x300/?bus-terminal',
  'https://source.unsplash.com/400x300/?bus-dar',
  'https://source.unsplash.com/400x300/?bus-stop',
];
```

```
const busyHours = [
  { time: '06:00 - 08:00', level: 'High' },
  { time: '12:00 - 14:00', level: 'Medium' },
  { time: '17:00 - 19:00', level: 'High' },
];
```

```
const accessibility = [
  'Wheelchair ramps available',
  'Tactile walkways for visually impaired',
  'Accessible restrooms',
  'Priority seating areas',
];
```

```
const alerts = [
  { id: 1, title: 'Temporary Diversion', detail: 'Buses rerouted to secondary gate due to
maintenance works.', level: 'warning' },
  { id: 2, title: 'Scheduled Cleaning', detail: 'Passenger waiting area closed for cleaning from
22:00 - 04:00.', level: 'info' },
];
```

```
return (
  <div className="min-h-screen bg-gray-50 text-gray-800">
```

```

    { /* Header */ }
    <header className="bg-white border-b px-6 py-4 flex items-center justify-between">
      <h1 className="text-xl font-semibold flex items-center gap-2">
        <MapPin className="text-teal-600" size={20}/> Terminal Details
      </h1>
      <div className="flex items-center gap-3">
        <button className="px-3 py-1.5 border rounded-md text-sm hover:bg-gray-50 flex
items-center gap-1"><Save size={14}/> Save</button>
        <button className="px-3 py-1.5 border rounded-md text-sm hover:bg-gray-50 flex
items-center gap-1"><Share2 size={14}/> Share</button>
      </div>
    </header>

    { /* Overview Section */ }
    <section className="max-w-6xl mx-auto px-6 py-10 grid md:grid-cols-2 gap-8">
      <Card className="border border-gray-200">
        <CardContent className="p-6">
          <h2 className="text-2xl font-semibold text-teal-700 mb-2">{terminal.name}</h2>
          <p className="text-sm text-gray-600 mb-4">{terminal.location}</p>
          <div className="text-sm text-gray-700 space-y-2">
            <div className="flex items-center gap-2"><Info size={14}/>
{terminal.coordinates}</div>
            <div className="flex items-center gap-2"><Bus size={14}/> {terminal.totalRoutes}
Connected Routes</div>
            <div className="flex items-center gap-2"><Users size={14}/>
{terminal.dailyPassengers} Daily Passengers</div>
            <div className="flex items-center gap-2"><Phone size={14}/> {terminal.contact}</div>
            <div className="flex items-center gap-2"><ShieldCheck size={14}/> Managed by
{terminal.supervisor}</div>
          </div>

          <h3 className="mt-6 mb-2 font-semibold text-gray-800">Facilities</h3>
          <ul className="list-disc list-inside text-sm text-gray-600">
            {terminal.facilities.map((f, i) => (<li key={i}>{f}</li>))}
          </ul>
        </CardContent>
      </Card>

      { /* Map Card */ }
      <Card className="border border-gray-200">
        <CardContent className="p-6">
          <h3 className="text-lg font-semibold flex items-center gap-2 mb-3"><MapPin
size={16}/> Map View</h3>

```

```

        <div className="h-64 w-full bg-gray-200 rounded-lg flex items-center justify-center
text-gray-400 mb-4">
            
        </div>
        <p className="text-sm text-gray-600 italic">{terminal.description}</p>
    </CardContent>
</Card>
</section>

```

```

    {/* Additional Info Cards */}
    <section className="max-w-6xl mx-auto px-6 grid md:grid-cols-2 lg:grid-cols-3 gap-6">
        {/* Busy Hours */}
        <Card className="border border-gray-200">
            <CardContent className="p-6">
                <h3 className="text-lg font-semibold flex items-center gap-2 mb-3"><Clock size={16}/>
Busy Hours</h3>
                <ul className="space-y-2 text-sm text-gray-700">
                    {busyHours.map((b, i) => (
                        <li key={i} className="flex items-center justify-between border rounded-md px-3 py-2
bg-white">
                            <span>{b.time}</span>
                            <span className={`font-medium ${b.level === 'High' ? 'text-rose-600' :
'text-amber-600'}`}>{b.level}</span>
                        </li>
                    ))}
                </ul>
            </CardContent>
        </Card>

```

```

        {/* Accessibility */}
        <Card className="border border-gray-200">
            <CardContent className="p-6">
                <h3 className="text-lg font-semibold flex items-center gap-2 mb-3"><Accessibility
size={16}/> Accessibility</h3>
                <ul className="list-disc list-inside text-sm text-gray-600 space-y-1">
                    {accessibility.map((a, i) => (<li key={i}>{a}</li>))}
                </ul>
            </CardContent>
        </Card>

```

```

    {/* Photos */}
    <Card className="border border-gray-200">
        <CardContent className="p-6">

```

```

        <h3 className="text-lg font-semibold flex items-center gap-2 mb-3"><Image
size={16}/> Photos</h3>
        <div className="grid grid-cols-3 gap-2">
            {gallery.map((g, i) => (
                <img key={i} src={g} alt="terminal photo" className="rounded-md border
hover:opacity-90 cursor-pointer" />
            ))}
        </div>
    </CardContent>
</Card>

    {/* Service Alerts */}
    <Card className="border border-gray-200 lg:col-span-3">
        <CardContent className="p-6">
            <h3 className="text-lg font-semibold flex items-center gap-2 mb-3"><AlertTriangle
size={16}/> Service Alerts</h3>
            <div className="space-y-2">
                {alerts.map((al) => (
                    <div key={al.id} className={`rounded-md border p-3 text-sm ${al.level==='warning'?
'bg-amber-50 border-amber-200 text-amber-900':'bg-sky-50 border-sky-200 text-sky-900'}`}>
                        <div className="font-medium">{al.title}</div>
                        <div className="text-xs opacity-90">{al.detail}</div>
                    </div>
                ))}
            </div>
        </CardContent>
    </Card>
</section>

    {/* Connected Routes Section */}
    <section className="max-w-6xl mx-auto px-6 py-10">
        <h2 className="text-2xl font-semibold text-teal-700 mb-6 flex items-center gap-2"><Bus
size={20}/> Routes from {terminal.name}</h2>
        <div className="grid md:grid-cols-2 lg:grid-cols-3 gap-6">
            {connectedRoutes.map((r) => (
                <Card key={r.id} className="border border-gray-200 hover:shadow-sm transition">
                    <CardContent className="p-4">
                        <div className="flex items-center justify-between mb-2">
                            <div className="flex items-center gap-2 text-lg font-semibold text-gray-800">
                                <span className="inline-flex items-center justify-center h-6 w-8 rounded-md
text-white bg-teal-600">{r.id}</span>
                                {r.name}
                            </div>
                        </div>
                    </div>
                </div>
            ))}
        </div>
    </section>

```

```
<div className="text-sm text-gray-600 flex flex-col gap-1">
  <div className="flex items-center gap-1"><Clock size={14}/> {r.time}</div>
  <div className="flex items-center gap-1"><CreditCard size={14}/> {r.fare}</div>
</div>
</CardContent>
</Card>
  )}
</div>
</section>

{/* Footer */}
<footer className="bg-gray-100 border-t py-6 text-center text-sm text-gray-600">
  <p>© {new Date().getFullYear()} DARoutes — Empowering Public Transport Data</p>
</footer>
</div>
);
}
```