TerraPlan Implementation Roadmap

# Overview

TerraPlan is envisioned as a web-based smart land-use intelligence platform for Africa, combining satellite imagery, analytics, and user-friendly dashboards to support sustainable agriculture, urban planning, conservation, and development. The implementation will proceed in structured phases, building core mapping and data infrastructure first, then adding advanced analytics, role-specific features, and broad data integration. The stack will use Python/FastAPI for the backend and React/Next.js for the frontend, with a spatial database (PostgreSQL+PostGIS or an Open Data Cube) beneath. Core modules include interactive maps, flood-risk alerting, land suitability analysis, and notification systems. Accessibility and modular design will be emphasized so the open-source system can scale across Africa.

# Development Phases

## Phase 1 – Infrastructure & Mapping MVP

Establish the basic infrastructure. Set up a Python/FastAPI backend with user authentication (JWT/OAuth2) and a spatial database (e.g. PostgreSQL + PostGIS or an Open Data Cube). Create a React/Next.js frontend skeleton. Implement a simple interactive map module using Leaflet or Mapbox GL with open basemaps. This phase delivers user login, role-based authentication logic, and a working map viewer.

## Phase 2 – Core Modules & Data Layers

Develop the key analytics modules and integrate base data. Implement map interaction features: panning/zooming, searching locations, and toggling layers. Load initial Earth Observation layers via tile services or APIs. Begin implementing flood-risk alerts and a basic land suitability tool combining soil, climate, and slope data.

## Phase 3 – Role-Based Dashboards

Introduce user personas and RBAC. Define roles (farmer, urban planner, conservationist, developer) and tailor each dashboard to its needs. Incorporate an admin interface for managing users and settings.

## Phase 4 – Data Integration & Real-Time Feeds

Expand data sources for analysis. Connect to Digital Earth Africa (DE Africa) for calibrated satellite time-series, use Copernicus data, and ingest IoT sensor feeds for local climate. Visualize feeds as overlays or charts.

## Phase 5 – Offline and Mobile Support

Ensure TerraPlan works on mobile devices and under limited connectivity. Implement offline-capable maps using vector tile downloads. Build frontend as a Progressive Web App or with React Native.

## Phase 6 – Accessibility, Scaling & Open Development

Polish the platform with accessibility features and modular architecture. Release the code as open-source, enable multiple language support, and containerize deployment for scaling.

# Technology Stack & Architecture

* \*\*Backend:\*\* Python with FastAPI for RESTful APIs and spatial processing. Use PostgreSQL + PostGIS or Open Data Cube for spatial storage.
* \*\*Frontend:\*\* React with Next.js. Use Leaflet or Mapbox GL for mapping, and component libraries like Material UI.
* \*\*APIs & Integration:\*\* External APIs for satellite, climate, and sensor data. Use tile services, STAC APIs, and data from DE Africa, Copernicus, CHIRPS.
* \*\*Architecture:\*\* Microservices-based. Containerized with Docker/Kubernetes. Use HTTPS, RBAC, and JWT security.

# Key Modules and Features

## Interactive Mapping Module

Toggle layers, draw, measure, and annotate maps. Use vector and raster tile sources.

## Flood Risk Alert Module

Analyze rainfall and terrain data to issue alerts. Display flood zones with notifications.

## Land Suitability Tools

Combine soil, slope, and climate data to rank land for agriculture or urban development.

## User Analytics & Notifications

Role-specific dashboards with visual data. Alert system for weather, flood, or anomalies.

## Customization & Settings

Users can set preferences, alert thresholds, and customize dashboards.

# Role-Based Dashboards and Personas

TerraPlan defines personas like Farmers, Planners, Conservationists, and Developers. Each sees a customized dashboard. Backend enforces RBAC using JWT, and frontend conditionally renders components. Admins can manage roles and view all modules.

# Open Data Sources (EO, Climate, IoT)

* Digital Earth Africa: https://www.digitalearthafrica.org/
* Copernicus Climate Data Store: https://cds.climate.copernicus.eu/
* WEkEO (Copernicus data access): https://wekeo.copernicus.eu/
* EUMETSAT: https://www.eumetsat.int/
* CHIRPS Rainfall Data: https://www.chc.ucsb.edu/data/chirps/
* NASA POWER Climate Data: https://power.larc.nasa.gov/
* WorldClim (Historical/Current Climate Data): https://www.worldclim.org/data/index.html
* OpenStreetMap: https://www.openstreetmap.org/
* FAO GAEZ: https://gaez.fao.org/