smart-city-app
-- የጂኦግራፊያዊ መረጃ ኤክስቴንሽን መጫን
CREATE EXTENSION postgis;

-- የይዞታ (Land Parcel) ሰንጠረዥ
CREATE TABLE land_parcels (
    id SERIAL PRIMARY KEY,
    owner_name VARCHAR(255) NOT NULL,
    parcel_id VARCHAR(50) UNIQUE NOT NULL, -- የካርታ ቁጥር
    area_sqm DECIMAL(10,2),
    land_use VARCHAR(100), -- ለመኖሪያ፣ ለንግድ...
    boundary GEOMETRY(Polygon, 4326), -- የቦታው ወሰን/ካርታ
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
const express = require('express');
const { Pool } = require('pg');
const app = express();
app.use(express.json());

// ዳታቤዝ ግንኙነት
const pool = new Pool({
    user: 'your_user',
    host: 'localhost',
    database: 'smart_city_db',
    password: 'your_password',
    port: 5432,
});

// 1. አዲስ የይዞታ ማረጋገጫ (Title Deed) መመዝገቢያ
app.post('/api/land/register', async (req, res) => {
    const { owner_name, parcel_id, area, coordinates } = req.body;
    try {
        // የPolygon መረጃን ወደ ጂ-አይ-ኤስ ፎርማት መቀየር
        const query = `
            INSERT INTO land_parcels (owner_name, parcel_id, area_sqm, boundary)
            VALUES ($1, $2, $3, ST_GeomFromGeoJSON($4))
            RETURNING *;
        `;
        const values = [owner_name, parcel_id, area, JSON.stringify(coordinates)];
        const result = await pool.query(query, values);
        res.status(201).json({ message: "ይዞታው በተሳካ ሁኔታ ተመዝግቧል", data: result.rows[0] });

```
    } catch (err) {
        res.status(500).json({ error: err.message });
    }
});

// 2. የይዞታ መረጃ በካርታ ቁጥር መፈለጊያ
app.get('/api/land/:parcel_id', async (req, res) => {
    try {
        const query = `SELECT id, owner_name, ST_AsGeoJSON(boundary) as map FROM
land_parcels WHERE parcel_id = $1`;
        const result = await pool.query(query, [req.params.parcel_id]);
        if (result.rows.length === 0) return res.status(404).send("መረጃው አልተገኘም");
        res.json(result.rows[0]);
    } catch (err) {
        res.status(500).json({ error: err.message });
    }
});

const PORT = 3000;
app.listen(PORT, () => console.log(`ሲስተሙ በፖርት ${PORT} ላይ ስራ ጀምሯል`));
const crypto = require('crypto');

function signDocument(data, privateKey) {
    const sign = crypto×createSign('SHA256');
    sign.update(JSON.stringify(data));
    sign.end();
    return sign.sign(privateKey, 'hex'); // ይህ ዲጂታል ፊርማው ነው
}
import React, { useState } from 'react';
import './App.css';

function App() {
  const [parcelId, setParcelId] = useState('');
  const [landData, setLandData] = useState(null);

  const searchLand = async () => {
    const response = await fetch(`/api/land/${parcelId}`);
    const data = await response.json();
    setLandData(data);
  };
```

```
  return (
    <div className="App">
      <header className="header">
        <h1>የከተማ አስተዳደር የይዞታ ማረጋገጫ ፖርታል</h1>
      </header>

      <div className="search-box">
        <input
          type="text"
          placeholder="የካርታ ቁጥር ያስገቡ..."
          value={parcelId}
          onChange={(e) => setParcelId(e.target.value)}
        />
        <button onClick={searchLand}>ፈልግ</button>
      </div>

      {landData && (
        <div className="result-card">
          <h2>የባለቤት ስም፡- {landData.owner_name}</h2>
          <p>የይዞታ መታወቂያ፡- {landData.parcel_id}</p>
          <div className="map-placeholder">
            {/* እዚህ ቦታ ላይ የካርታው ምስል (Map View) ይታያል */}
            <p>ካርታው በመጫን ላይ ነው...</p>
          </div>
          <button className="download-btn">ዲጂታል ካርታውን አውርድ (PDF)</button>
        </div>
      )}
    </div>
  );
}

export default App;
const bcrypt = require('bcrypt');
const jwt = require('jsonwebtoken');

app.post('/api/login', async (req, res) => {
    const { username, password } = req.body;
    // ተጠቃሚውን ከዳታቤዝ መፈለግ
    const user = await pool.query('SELECT * FROM staff WHERE username = $1', [username]);

    if (user.rows.length > 0) {
```

```
        const validPass = await bcrypt.compare(password, user.rows[0].password);
        if (validPass) {
            const token = jwt.sign({ id: user.rows[0].id, role: user.rows[0].role }, 'SECRET_KEY');
            res.header('auth-token', token).send({ token, message: "እንኳን ደህና መጡ!" });
        } else {
            res.status(400).send("የይለፍ ቃል ስህተት ነው");
        }
    }
});
const PDFDocument = require('pdfkit');
const fs = require('fs');

app.get('/api/land/generate-pdf/:id', async (req, res) => {
    const doc = new PDFDocument();
    doc.pipe(fs.createWriteStream('Title_Deed.pdf'));

    doc.fontSize(20).text('የከተማ አስተዳደር የይዞታ ማረጋገጫ (Title Deed)', { align: 'center' });
    doc.moveDown();
    doc.fontSize(14).text(`ባለቤት፡ ${landData.owner_name}`);
    doc.text(`የካርታ ቁጥር፡ ${landData.parcel_id}`);
    doc.text(`የቦታ ስፋት፡ ${landData.area_sqm} ካሬ ሜትር`);

    // እዚህ ጋር የዲጂታል ፊርማ (QR Code) መጨመር ይቻላል
    doc.end();
    res.send("ፒ-ዲ-ኤፍ ተዘጋጅቷል");
});
CREATE TABLE audit_logs (
    id SERIAL PRIMARY KEY,
    staff_id INTEGER REFERENCES staff(id),
    action VARCHAR(255), -- "Update", "Delete", "Register"
    parcel_id VARCHAR(50),
    changed_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
# 1. ፕሮጀክቱን ለመጀመር
npm init -y

# 2. አስፈላጊ የሆኑ የኮድ ጥቅሎችን (Packages) ለመጫን
npm install express pg bcrypt jsonwebtoken pdfkit dotenv

CREATE EXTENSION postgis;
```

```
CREATE TABLE land_parcels (
    id SERIAL PRIMARY KEY,
    owner_name VARCHAR(255),
    parcel_id VARCHAR(50) UNIQUE,
    area_sqm DECIMAL,
    boundary GEOMETRY(Polygon, 4326)
);

CREATE TABLE staff (
    id SERIAL PRIMARY KEY,
    username VARCHAR(50) UNIQUE,
    password TEXT,
    role VARCHAR(20) DEFAULT 'officer'
);

const express = require('express');
const { Pool } = require('pg');
const app = express();
app.use(express.json());

const pool = new Pool({
  user: 'postgres', // የእርስም ፖስትግሬስ ዩዘር ስም
  host: 'localhost',
  database: 'city_db',
  password: 'PASSWORD_HERE', // የእርስም የዳታቤዝ ምስጢር ቃል
  port: 5432,
});

app.get('/', (req, res) => res.send('የከተማ አስተዳደር ሲስተም ዝግጁ ነው!'));

app.listen(3000, () => console.log('Server running on http://localhost:3000'));

node server.js

# ሲስተሙን ለማዷስ
pkg update && pkg upgrade

# Node.js ን ለመጫን
pkg install nodejs

# PostgreSQL (ዳታቤዝ) ለመጫን
```

```
pkg install postgresql

<script>
  // 1. ካርታውን መፍጠር
  const map = new ol.Map({
    target: 'map',
    layers: [
      new ol.layer.Tile({ source: new ol.source.OSM() })
    ],
    view: new ol.View({
      center: ol.proj.fromLonLat([38.755, 9.015]),
      zoom: 15
    })
  });

  // 2. ከሰርቨር መረጃውን ስቦ በካርታው ላይ መሳል
  fetch('/map-data')
    .then(response => response.json())
    .then(data => {
      data.forEach(item => {
        const feature = new ol.format.GeoJSON().readFeature(item.location, {
          dataProjection: 'EPSG:4326',
          featureProjection: 'EPSG:3857'
        });

        const vectorSource = new ol.source.Vector({ features: [feature] });
        const vectorLayer = new ol.layer.Vector({
          source: vectorSource,
          style: new ol.style.Style({
            stroke: new ol.style.Stroke({ color: 'red', width: 3 }),
            fill: new ol.style.Fill({ color: 'rgba(255, 0, 0, 0.2)' })
          })
        });
        map.addLayer(vectorLayer);
      });
    });
</script>

app.get('/verify/:parcel_id', async (req, res) => {
  const { parcel_id } = req.params;
  const result = await pool.query('SELECT * FROM parcels WHERE parcel_id = $1', [parcel_id]);
```

```
  if (result.rows.length > 0) {
    res.json({ status: "Verified", owner: result.rows[0].owner_name, area:
result.rows[0].area_sqm });
  } else {
    res.status(404).json({ status: "Not Found", message: "ይህ የካርታ ቁጥር በሲስተሙ ውስጥ
የለም" });
  }
});

CREATE TABLE zoning_layers (
    id SERIAL PRIMARY KEY,
    zone_type VARCHAR(50), -- Residential, Commercial, Industrial, Green Area
    color_code VARCHAR(10), -- ለምሳሌ፦ '#FFFF00' ለቢጫ (መኖሪያ)
    geom GEOMETRY(MultiPolygon, 4326)
);

CREATE TABLE roads (
    id SERIAL PRIMARY KEY,
    road_name VARCHAR(100),
    road_type VARCHAR(50), -- Arterial, Collector, Local
    width_meters DECIMAL,
    geom GEOMETRY(LineString, 4326)
);

app.get('/road-buffer/:road_id', async (req, res) => {
  const { road_id } = req.params;
  const query = `
    SELECT ST_AsGeoJSON(ST_Buffer(geom::geography, 15)::geometry) as buffer_zone
    FROM roads
    WHERE id = $1;
  `;
  const result = await pool.query(query, [road_id]);
  res.json(result.rows[0]);
});

// በ OpenLayers ውስጥ ንብርብሮችን መቆጣጠር
const zoningLayer = new ol.layer.Vector({
  source: new ol.source.Vector({ url: '/api/zones', format: new ol.format.GeoJSON() }),
  style: (feature) => new ol.style.Style({
    fill: new ol.style.Fill({ color: feature.get('color_code') })
```

```
  })
});

map.addLayer(zoningLayer); // የዞን ካርታውን ጨምር

app.get('/api/road-buffer/:id', async (req, res) => {
    try {
        const roadId = req.params.id;
        // ST_Buffer ን በመጠቀም 15 ሜትር ክልል ማስላት
        const query = `
            SELECT id, road_name,
            ST_AsGeoJSON(ST_Buffer(geom::geography, 15)::geometry) as buffer_geom
            FROM roads WHERE id = $1;
        `;
        const result = await pool.query(query, [roadId]);
        res.json(result.rows[0]);
    } catch (err) {
        res.status(500).json({ error: err.message });
    }
});

// መንገዱን እና ክልከላውን በካርታው ላይ መሳል
function showRoadBuffer(roadId) {
    fetch(`/api/road-buffer/${roadId}`)
        .then(res => res.json())
        .then(data => {
            const bufferFeature = new ol.format.GeoJSON().readFeature(data.buffer_geom, {
                dataProjection: 'EPSG:4326',
                featureProjection: 'EPSG:3857'
            });

            const bufferLayer = new ol.layer.Vector({
                source: new ol.source.Vector({ features: [bufferFeature] }),
                style: new ol.style.Style({
                    fill: new ol.style.Fill({ color: 'rgba(255, 0, 0, 0.3)' }), // ቀይ ጥላ
                    stroke: new ol.style.Stroke({ color: 'red', width: 2 })
                })
            });
            map.addLayer(bufferLayer);
        });
}
```

```
INSERT INTO zoning_layers (zone_type, color_code, geom)
VALUES (
  'Commercial',
  '#FF0000', -- ለንግድ ቀይ ቀለም
  ST_GeomFromText('MULTIPOLYGON(((38.74 9.01, 38.75 9.01, 38.75 9.02, 38.74 9.02, 38.74
9.01)))', 4326)
);

app.get('/api/check-conflict/:parcel_id', async (req, res) => {
  try {
    const { parcel_id } = req.params;

    // ይዞታው ከመንገድ 15 ሜትር ክልከላ ጋር ይነካካል ወይ? የሚል ጥያቄ
    const query = `
      SELECT r.road_name,
          ST_Area(ST_Intersection(p.geom, ST_Buffer(r.geom::geography, 15)::geometry)) as
overlap_area
      FROM parcels p, roads r
      WHERE p.parcel_id = $1
      AND ST_Intersects(p.geom, ST_Buffer(r.geom::geography, 15)::geometry);
    `;

    const result = await pool.query(query, [parcel_id]);

    if (result.rows.length > 0) {
      res.json({
        conflict: true,
        message: `ይህ ይዞታ ከ ${result.rows[0].road_name} የመንገድ ክልከላ ጋር ይጋጫል!`,
        overlap_sqm: result.rows[0].overlap_area
      });
    } else {
      res.json({ conflict: false, message: "ይዞታው ከመንገድ ነፃ ነው!" });
    }
  } catch (err) {
    res.status(500).json({ error: err.message });
  }
});
app.get('/api/check-zoning/:parcel_id', async (req, res) => {
  const { parcel_id } = req.params;
```

```
    const query = `
        SELECT z.zone_type
        FROM parcels p, zoning_layers z
        WHERE p.parcel_id = $1
        AND ST_Within(ST_Centroid(p.geom), z.geom);
    `;

    const result = await pool.query(query, [parcel_id]);

    if (result.rows.length > 0) {
        const zone = result.rows[0].zone_type;
        res.json({
            parcel_id: parcel_id,
            allowed_use: zone,
            message: `ይህ ቦታ በ ${zone} ቀጠና ውስጥ ይገኛል።`
        });
    } else {
        res.status(404).json({ message: "ለዚህ ቦታ የተመደበ ዞን አልተገኘም" });
    }
});
map.on('singleclick', function (evt) {
    const coordinate = evt.coordinate;
    // የነካነውን ቦታ መጋጠሚያ ወደ ሰርቨር ልከን መረጃ መቀበል
    // ... Fetch logic here ...
    alert("የቦታው መረጃ በመፈለግ ላይ ነው...");
});
<div class="dashboard-container" style="display: flex; height: 100vh;">
    <div class="sidebar" style="width: 30%; border-right: 1px solid #ccc; padding: 15px; overflow-
y: auto;">
        <h3>የይዞታዎች ዝርዝር</h3>
        <div id="parcel-list">
            </div>
    </div>

    <div class="main-content" style="width: 70%; position: relative;">
        <div id="map" style="width: 100%; height: 60%;"></div>
        <div id="analysis-panel" style="padding: 20px; background: #f9f9f9; height: 40%;">
            <h3>የቦታ ትንታኔ (Spatial Analysis)</h3>
            <div id="analysis-results">
                <p>እባክዎን ከዝርዝሩ ውስጥ አንድ ይዞታ ይምረጡ...</p>
            </div>
```

```
      </div>
    </div>
</div>
async function inspectParcel(parcelId) {
    // 1. የግጭት ምርመራ API መጥራት
    const conflictRes = await fetch(`/api/check-conflict/${parcelId}`);
    const conflictData = await conflictRes.json();

    // 2. የዞን መረጃ API መጥራት
    const zoningRes = await fetch(`/api/check-zoning/${parcelId}`);
    const zoningData = await zoningRes.json();

    // 3. ውጤቱን በዳሽቦርዱ ላይ ማሳየት
    const resultsDiv = document×getElementById('analysis-results');
    resultsDiv×innerHTML = `
        <div style="color: ${conflictData.conflict ? 'red' : 'green'}">
            <strong>የመንገድ ግጭት፡</strong> ${conflictData.message}
        </div>
        <div>
            <strong>የተፈቀደ አገልግሎት (Zoning)፡</strong> ${zoningData.allowed_use}
        </div>
        <button onclick="generatePDF('${parcelId}')" style="margin-top:10px;">የይዞታ ማረጋገጫ
አውርድ (PDF)</button>
    `;
}
const PDFDocument = require('pdfkit');

app.get('/api/generate-certificate/:id', async (req, res) => {
    const doc = new PDFDocument();
    res.setHeader('Content-Type', 'application/pdf');
    doc.pipe(res);

    doc.fontSize(20).text('የከተማ አስተዳደር የይዞታ ማረጋገጫ', { align: 'center' });
    doc.moveDown();

    // ከዳታቤዝ የመጣ መረጃ
    doc.fontSize(12).text(`የይዞታ ቁጥር: ${parcelId}`);
    doc.text(`ባለቤት: ${ownerName}`);
    doc.text(`የቦታ ስፋት: ${area} ካሬ ሜትር`);

    // እዚህ ጋር የካርታ ምስል ወይም QR code መጨመር ይቻላል
```

```
    doc.end();
});
function checkCompliance(parcelData) {
    const statusBox = document×getElementById('compliance-status');

    if (parcelData.is_overlapping_road) {
        statusBox×innerHTML = "<div style='background: #ffcccc; color: red; padding: 10px;
border-radius: 5px;'>" +
                        "⚠️ ቀይ መብራት፡ ይዞታው የመንገድ ክልከላን ይጥሳል!</div>";
        document.getElementById('approve-btn').disabled = true; // የማጽደቂያ ቁልፉን መዝጋት
    } else if (parcelData.wrong_zone) {
        statusBox×innerHTML = "<div style='background: #fff3cd; color: #856404; padding: 10px;
border-radius: 5px;'>" +
                        "⚠️ ቢጫ መብራት፡ የአገልግሎት አይነት ከዞኑ ጋር አይጣጣምም!</div>";
    } else {
        statusBox×innerHTML = "<div style='background: #d4edda; color: green; padding: 10px;
border-radius: 5px;'>" +
                        "✅ አረንጓዴ መብራት፡ ይዞታው ከማስተር ፕላኑ ጋር ሙሉ በሙሉ ይጣጣማል።</div>";
        document.getElementById('approve-btn').disabled = false;
    }
}
const PDFDocument = require('pdfkit');
const QRCode = require('qrcode');

app.get('/api/generate-certificate/:id', async (req, res) => {
    const doc = new PDFDocument({ size: 'A4', margin: 50 });
    const parcelId = req.params.id;

    // የ QR Code ማመንጫ (ወደ ሲስተሙ ሊንክ የሚያደርግ)
    const qrData = `https://city-admin.gov.et/verify/${parcelId}`;
    const qrImage = await QRCode.toDataURL(qrData);

    res.setHeader('Content-Type', 'application/pdf');
    doc.pipe(res);

    // የሰነዱ ራስጌ
    doc.fontSize(20).text('የከተማ አስተዳደር የይዞታ ማረጋገጫ ካርታ', { align: 'center', underline:
true });
    doc.moveDown();

    // የባለቤት መረጃ
```

```
    doc.fontSize(12).text(`የይዞታ መለያ ቁጥር: ${parcelId}`);
    doc.text(`የባለቤት ስም: አቶ አበበ በላይ`);
    doc.text(`የቦታ ስፋት: 200.50 ካሬ ሜትር`);
    doc.text(`የተሰጠበት ቀን: ${new Date().toLocaleDateString('et-ET')}`);

    // የ QR Code ምስል መጨመር
    doc.image(qrImage, 400, 100, { width: 100 });

    // የካርታ ምስል (Static Map Image) እዚህ ጋር ይገባል
    doc.rect(50, 250, 500, 300).stroke(); // ለካርታው ቦታ ማስቀመጫ
    doc.text('የይዞታው ካርታ እዚህ ጋር ይታተማል', 200, 400);

    doc.end();
});
function captureLocation() {
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition((position) => {
            const lat = position.coords.latitude;
            const lng = position.coords.longitude;
            alert(`መጋጠሚያው ተመዝግቧል: Lat: ${lat}, Lng: ${lng}`);
            // ይህ መረጃ በቀጥታ ወደ Backend ይላካል
        });
    } else {
        alert("ስልክዎ GPS መጠቀም አይችልም!");
    }
}
```