**📑 Documentation: Datasets and Processing Workflow**

**1. Tree Census Data of Bengaluru**

* **Description**: Vector dataset of tree locations and attributes (species, age, ward info).
* **Source**: [OpenCity – Bengaluru Tree Census](https://data.opencity.in/dataset/bengaluru-tree-census-data)
* **Steps to Use**:
  1. Download the dataset (kml format).
  2. Convert to **GeoJSON**:

ogr2ogr -f GeoJSON blr\_east\_zone\_tree.geojson blr\_east\_zone\_trees\_11\_2024.kml

ogr2ogr -f GeoJSON blr\_south\_zone\_tree.geojson blr\_south\_zone\_trees\_11\_2024.kml

ogr2ogr -f GeoJSON blr\_west\_zone\_tree.geojson blr\_west\_zone\_trees\_11\_2024.kml

* 1. Store in, **public\data\trees**

**2. Bengaluru Ward Boundaries**

* **Description**: Vector dataset defining administrative ward polygons.
* **Source**: [Datameet – BBMP Wards GeoJSON](https://github.com/datameet/Municipal_Spatial_Data/blob/master/Bangalore/BBMP.geojson)
* **Steps to Use**:
  1. Download the raw GeoJSON directly from GitHub. (BBMP.geojson)
  2. The Geojson has a polygon feature, few features are in invalid geometry.
  3. To resolve run a topology check and quick fix it.
  4. And dissolve the cmd, to create a BBMP\_AOI.geojson

*ogr2ogr -f GeoJSON BBMP\_AOI.geojson BBMP.geojson -dialect sqlite -sql "SELECT ST\_Union(geometry) AS geometry FROM BBMP"*

**3. Bengaluru Schools Data**

* **Description**: Point dataset of schools within Bengaluru.
* **Source**: [Overpass Turbo](https://overpass-turbo.eu/) query
* **Query Template**:

*node*

*[amenity=school]*

*({{bbox}});*

*out;*

* **Steps to Use**:
  1. Run query on Overpass Turbo.
  2. Export as **GeoJSON**. (Overpass\_school.geojson)

**4. Digital Elevation Model (DEM)**

* **Description**: Raster dataset representing elevation values.
* **Source**: **NASADEM** via [Microsoft Planetary Computer](https://planetarycomputer.microsoft.com/dataset/nasadem)
* **Steps to Process**:

**(a) Download and Clip to Bengaluru Boundary**

* + Use planetary\_computer STAC or direct tiles (NASADEM\_HGT\_n13e077.tif and NASADEM\_HGT\_n12e077.tif).
  + Direct downloading permission is not available, so we can use the download\_nasadem\_tiles.py code.
  + Procedure
    - Create a virtual environment
      * *python -m venv venv*
      * *.\venv\Scripts\activate*
      * *pip install --upgrade pip*
      * *pip install pystac-client planetary-computer requests rasterio geopandas shapely*
      * *python download\_nasadem\_tiles.py --items NASADEM\_HGT\_n13e077 NASADEM\_HGT\_n12e077*
  + *gdal\_merge.py -o nasadem\_merge.tif -of GTiff -co COMPRESS=LZW NASADEM\_HGT\_n12e077.tif NASADEM\_HGT\_n13e077.tif*
  + *gdalwarp -cutline BBMP\_AOI.geojson -crop\_to\_cutline nasadem\_merge.tif nasadem\_clipped.tif*

**(b) Inspect Metadata**

*gdalinfo -json nasadem13\_clip.tif > nasadem13\_clip\_bounds.json*

**(c) Apply Color Relief**

* + Create a color\_ramp.txt, refer in the code
  + Apply relief shading:
  + *gdaldem color-relief NASADEM\_HGT\_n13e077.tif color\_ramp.txt nasadem13\_check.png -alpha*
  + *gdaldem color-relief NASADEM\_HGT\_n12e077.tif color\_ramp.txt nasadem\_merge\_check.png -alpha*

**🔗 Integration Workflow**

1. **Collect & Prepare Data**:
   * Trees (blr\_east\_zone\_tree.geojson, blr\_south\_zone\_tree.geojson, blr\_west\_zone\_tree.geojson)
   * Wards (BBMP.geojson)
   * Schools (schools.geojson)
   * DEM (nasadem13\_clip.tif and nasdem12\_clip.tif) & Merged as nasadem\_merge.tif converts to nasadem\_merge.png
2. **Overlay & Visualization**:
   * Load all vector layers (trees, wards, schools).
   * Render DEM with hillshade and 3d terrain.
   * Combine using **Mapbox**.
3. **Outputs**:
   * Map visualizations (interactive web maps / PNG exports).
   * Elevation JSON bounds for analysis.
   * Processed datasets stored in public/data/.
4. **Advanced:**
   * The 3d visualization (Realistic trees based on the few categories added as the glb formats to view in the render in 3d perspective view)
   * Categories are schools and trees
   * Tools added for the Mapbox interface.