## ****Project Planning & Management****

This section outlines the overall strategy, project scope, tools, and risk management.

### ****Project Scope****

The project focuses on developing a **deep learning model** for **land classification** using **Sentinel-2 satellite data**. The goal is to extract meaningful insights from multispectral imagery to categorize land types (e.g., urban, vegetation, water bodies, barren land). The model will leverage **NDPI (Normalized Difference Plant Index)** and **multispectral analysis** to improve classification accuracy.

### ****Methodology & Approach****

**Data Collection**:

* 1. Obtain **Sentinel-2 imagery** from sources like Copernicus Open Access Hub or Google Earth Engine.
  2. Preprocess data using **GDAL, Rasterio, and Fiona** to handle raster and vector geospatial formats.

**Data Preprocessing**:

* 1. Apply **GeoPandas** for handling **vector-based geospatial data** like shapefiles containing land type boundaries.
  2. Use **Rasterio** to process and manipulate **GeoTIFF multispectral images**.
  3. Normalize spectral bands and apply **NDPI calculations** for vegetation classification.

**Model Development & Training**:

* 1. Implement a **deep learning model** (e.g., CNN, U-Net, or a transformer-based architecture) using **TensorFlow/PyTorch**.
  2. Train the model on labeled Sentinel-2 images with land type annotations.

**Model Evaluation**:

* 1. Use accuracy metrics like **Precision, Recall, F1-score**, and **IoU (Intersection over Union)**.

**Visualization & Deployment**:

* 1. Use **Folium** to create an **interactive map** visualizing classification results.
  2. Deploy the model using a **web-based GIS application or API**.

### ****Tools & Technologies****

* **Programming Language**: Python
* **Libraries**: TensorFlow, PyTorch, GDAL, Rasterio, Fiona, GeoPandas, Folium
* **Geospatial Data Formats**: GeoTIFF (raster), Shapefiles (vector)
* **Cloud/Storage**: Google Earth Engine, AWS S3 for large datasets

### ****Risk Management****

* **Challenge**: High computational cost for training deep learning models.
  + **Solution**: Use cloud computing (Google Colab, AWS, or a GPU-powered system).
* **Challenge**: Inaccurate land classification due to poor-quality images.
  + **Solution**: Apply NDPI and multispectral filtering to improve feature extraction.