**Project Report: Import NoTax JSON to Feature Class**

**Project Overview**

This project shows the process of converting a JSON file containing geometries in Well-Known Text (WKT) format into a shapefile to visualize in ArcGIS Pro. The goal is to provide a tool that automates this workflow, making it easier for users to work with JSON data that cannot be directly converted into shape file using existing tools in ArcGIS Pro.

**Objectives**

1.Develop and test Python code to:

* Read a JSON file containing WKT geometries and metadata.
* Convert JSON data into a feature class stored as a shapefile.

2.Create a Python Toolbox to:

* Provide a user interface for selecting the JSON file, output shapefile name, and workspace.
* Allow execution of the tool directly within ArcGIS Pro.

3.Visualize the Data by:

* Adding the generated shapefile to a map.
* Customizing symbology and creating a map layout.

4.Document the Process in a report and GitHub repository, including the tool’s functionality and final layout design.

**Workflow and Development**

1. Code Development

The Python code for the project was first developed and tested in Jupyter Notebook using Visual Studio Code. This step allowed for interactive exploration of the JSON data, including geometry conversion and feature class creation.

The workflow includes:

Reading the JSON file: Using Python's json module to load and parse the data.

Converting geometries: Using ArcGIS Pro's arcpy.FromWKT method to convert WKT strings into geometries.

Creating a feature class: Employing arcpy.management.CreateFeatureclass to define the spatial dataset.

Populating the feature class: Employing arcpy.da.InsertCursor to add attributes and geometries from the JSON data.

The key function importNoTaxJson encapsulates the logic for these steps.

**2. Designing the Python Toolbox**

The Python Toolbox (.pyt) provides a graphical interface within ArcGIS Pro to run the conversion script. The toolbox was designed with the following parameters:

* Workspace: The folder where the output shapefile will be stored.
* JSON File: The input JSON file containing WKT geometries.
* Output Name: The name of the resulting shapefile.

The execute method in the toolbox calls the importNoTaxJson function to perform the conversion.

**3. Visualizing the Data**

After running the toolbox:

1.The shapefile was added to a new map in ArcGIS Pro.

2.Custom symbology was applied to the map layer to visualize the land value data.

3.A map layout was created, including:

* A title, legend, and scale bar.
* A map frame showing the spatial data.

4.The layout was exported to a PDF.

**Tool Execution**

User Interface

The tool’s user interface includes:

* A text box for specifying the workspace.
* A file browser for selecting the JSON input file.
* A field for defining the output shapefile name.

**Execution Results**

The tool successfully:

1.Read the input JSON file.

2.Converted WKT geometries into feature class geometries.

3.Generated a shapefile with attributes and geometries.

4.The results were verified and visualized in ArcGIS Pro.

**How to Use the Tool**

Step 1: Prepare the Environment

* Install ArcGIS Pro and set up the required Python environment with ArcPy.
* Ensure the JSON file is in the correct format and stored in an accessible directory.

Step 2: Run the Python Toolbox

* Add the Python Toolbox (.pyt) to ArcGIS Pro.
* Open the tool from the geoprocessing pane.

Provide the required parameters:

* Workspace: Folder to store the output shapefile.
* JSON File: Select the input JSON file.
* Output Name: Specify the shapefile name.

Click Run.

Step 3: Visualize and Export

1.Add the shapefile to a map.

2.Customize the symbology to represent the data effectively.

3.Create a layout with additional map elements (e.g., legend, title).

4.Export the layout to a PDF.

**Conclusion**

This project highlights the seamless integration of Python scripting with ArcGIS Pro for spatial data processing and visualization. By following this guided approach, students can effectively manage and analyze JSON-based geospatial data while gaining practical experience in GIS programming and tool development.

The tool and its workflow provide a robust solution for converting JSON data into GIS-compatible formats, making it a valuable resource for handling similar datasets in the future.