

## Assignment No. 1

**Problem Statement:** Reading and writing different types of datasets.

**Objective:** The objective of this assignment is to familiarize ourselves with reading and writing different types of datasets including `.txt`, `.csv`, and `.xml` from the web and local disk storage. We will explore how to load these datasets into memory, process them, and save them to a specific location on the disk.

**Prerequisite :**

1. A Python environment set up with libraries like pandas, xml.etree.ElementTree, and requests (for web access).
2. Internet connection (for reading datasets from the web).
3. Text editor and basic knowledge of file handling in Python.

**Theory :**

In data analysis and machine learning, datasets are essential as they hold the information needed for building models, making predictions, and extracting insights. Different file formats are used to store datasets, each with unique characteristics suited to different use cases. The most common formats include CSV, Excel, JSON, and SQL databases. Understanding how to read from and write to these formats is crucial in data handling and processing.

**1. CSV (Comma-Separated Values):**

CSV is a simple text format where each line represents a row, and columns are separated by commas. It is widely used due to its simplicity and compatibility with many tools.

**2. Excel Files (.xlsx):**

Excel files are widely used in business environments for data storage and manipulation. They allow multiple sheets and complex formatting.

**3. JSON (JavaScript Object Notation):**

JSON is a lightweight data-interchange format, ideal for APIs and web applications. It represents data in a hierarchical, human-readable form.

**5. Text Files (.txt):**

Plain text files can store data with simple structure and are easy to read. They are useful for reading raw data or logs

**References :**

**Database :** <https://www.kaggle.com/datasets>

**Conclusion**

Each format has its advantages depending on the application, and tools like Python's **pandas** library provide robust functions to handle a variety of formats with ease.