**Tutorial: Web Scraping and Data Management with Python**

**Introduction**

Welcome to this tutorial on web scraping and data management using Python. In this tutorial, we'll walk through a simple mini project that involves extracting data from a website, organizing it in a Pandas DataFrame, and allowing users to perform CRUD operations on the data.

**Prerequisites**

Make sure you have the following installed on your machine:

* Python 3.x
* **requests**
* **beautifulsoup4**
* **pandas**

pip install requests beautifulsoup4 pandas

**Project Overview**

The goal of this mini project is to scrape information about countries' GDP from a website, store the data in a Pandas DataFrame, and provide an interactive interface for users to perform operations like adding new data, updating existing data, deleting data, and displaying specific information.

**Step 1: Understanding the Code**

Let's take a look at the Python script. The script is divided into several sections:

1. **Importing Libraries:**
   * **requests** for making HTTP requests.
   * **BeautifulSoup** for HTML parsing.
   * **pandas** for data manipulation.
2. **Web Scraping:**
   * The script makes a request to a website, parses the HTML content, and extracts data from tables with the class 'ptable2'.
3. **Pandas DataFrame:**
   * The extracted data is organized into a Pandas DataFrame.

# Interactive Operations:

* + Users can interactively perform CRUD operations on the DataFrame, such as adding new rows, updating cell values, deleting rows, and displaying specific information.

python web\_scraping\_project.py

**Step 2: Interacting with the Program**

Follow the on-screen prompts to interact with the program. You can add new rows, update cell values, delete rows, and display specific information.

**Conclusion**

In this mini project, we successfully developed a Python script that combines web scraping and data management using the **requests**, **BeautifulSoup**, and **pandas** libraries. The script extracts information about countries' GDP from a website, organizes the data into a Pandas DataFrame, and provides an interactive interface for users to perform CRUD operations on the data.Top of Form