**CSV Comparison Report Generator**

**Overview**

The "CSV Comparison Report Generator" is a Python script that allows you to compare data between two sets of CSV files and create a comparison report in an Excel (XLSX) format. This tool is particularly useful for data migration or data integrity checks where you want to ensure that data from two sources align properly.

**Prerequisites**

Before using the script, make sure you have the following:

1. Python installed on your system.
2. Required Python packages installed:
   * pandas
   * openpyxl

**Usage**

1. **Folder Setup**:
   * Place the script in a directory that contains the CSV files you want to compare.
   * Set the **csv\_folder** variable to the path of the folder containing your CSV files.
   * Set the **Output\_file\_path** variable to the path of the folder where output report should be placed.
2. **Run the Script**:
   * Execute the Python script. It will perform the following actions:
     + Check for CSV files starting with "OLD\_" and corresponding "NEW\_" files in the specified folder.
     + Load data from these files.
     + Sort data frames by the primary key (first column) to ensure data alignment.
     + Compare data between the two data frames cell by cell.
     + Generate a comparison report in XLSX format.
3. **Output**:
   * The script will generate a comparison report in the same directory as the script with a timestamp in the filename.

**Report Columns**

The generated comparison report contains the following columns:

1. S.No: Serial number.
2. Table name: The name of the data table.
3. OLD file name: The name of the OLD data file.
4. NEW file name: The name of the NEW data file.
5. Column Count from OLD file: Number of columns in the OLD data file.
6. Column Count from NEW file: Number of columns in the NEW data file.
7. Column count matches: "Yes" if the column counts match, "No" if they don't.
8. Row Count from OLD file: Number of rows in the OLD data file.
9. Row Count from NEW file: Number of rows in the NEW data file.
10. Row count matches: "Yes" if the row counts match, "No" if they don't.
11. Row count with data mismatch: Number of rows with data mismatches.
12. Cells with mismatch: Comma-separated list of cell labels with data mismatches.

**Conditional Formatting**

The script applies conditional formatting to the following columns:

* **Row count with data mismatch**: Cells with a value greater than zero are highlighted in red.
* **Column count matches**, **Row count matches**: Cells with "No" are highlighted in red.

**Troubleshooting**

* If you encounter any issues or errors, check the prerequisites, and ensure that your data files are in the correct format.