Requirements

- Python 3.x
- pandas library

Install the required libraries by running:

```
pip install pandas
```

Code Explanation

Imports

The script uses the `pandas` library to handle data manipulation, and the `os` module to work with file paths.

import pandas as pd
import os

File Paths

These variables define the input and output files:

base_filenames_file = 'base_filenames.csv'
file_paths_file = 'file_paths.csv'
output_file = 'matched_file_paths.csv'

Read CSV Files

`base_filenames_df` reads the list of base filenames.
`file_paths_df` reads the file paths CSV and names the column 'FilePath'.

base_filenames_df = pd.read_csv(base_filenames_file)
file_paths_df = pd.read_csv(file_paths_file, header=None, names=['FilePath'])

Extracting Base Filenames from Paths

The function `extract_base_filename_from_path` extracts the base filename (without extension) from a file path. It uses `os.path.basename` to get the filename and `os.path.splitext` to remove the extension.

```
def extract_base_filename_from_path(file_path):
    filename = os.path.basename(file_path)
    base_filename, ext = os.path.splitext(filename)
    return base_filename
```

Apply the Function

The `apply` method is used to apply the `extract_base_filename_from_path` function to each file path in `file_paths_df`, creating a new column `BaseFilename`.

```
file_paths_df['BaseFilename'] =
file_paths_df['FilePath'].apply(extract_base_filename_from_path)
```

Case-Insensitive Matching

Both base filenames and extracted filenames are converted to lowercase to ensure case-insensitive matching.

```
base_filenames_df['BaseFilename'] = base_filenames_df['BaseFilename'].str.lower()
file_paths_df['BaseFilename'] = file_paths_df['BaseFilename'].str.lower()
```

Merge DataFrames

This performs a left join (merge) between `base_filenames_df` and `file_paths_df` on the `BaseFilename` column. The `how='left'` ensures that all rows from the base filenames are kept, even if no match is found in the file paths.

```
merged_df = pd.merge(base_filenames_df, file_paths_df[['BaseFilename', 'FilePath']],
on='BaseFilename', how='left')
```

Handling Missing Matches

If a base filename does not have a matching file path, the `FilePath` column is filled with 'Not Found'.

```
merged_df['FilePath'].fillna('Not Found', inplace=True)
```

Save the Output

The merged DataFrame is saved to the specified `output_file` in CSV format.

```
merged_df.to_csv(output_file, index=False)
print(f"Matching complete. Output saved to '{output_file}'.")
```

Usage

1. Place the `base_filenames.csv` and `file_paths.csv` in the same directory as the script.
2. Run the script:

```
python match_file_paths.py
```

3. The output will be saved to `matched_file_paths.csv`.

License

This project is licensed under the MIT License.