# Documentation Of Feedback Processing

**Overview**

This script processes customer feedback by reading entries from a file, counting the total number of feedback entries, extracting occurrences of specific keywords, generating a summary report, and saving the report to a specified file.

**Requirements**

- Python 3.x

- A text file named `customer_feedback.txt` containing the feedback entries.

**Step-by-Step Guide**

**1. Reading the File**

- Function: `read_file(file_path)`

- Description: Reads the feedback entries from the specified file path.

- Input: `file_path` (str) - Path to the input file.

- Output: List of feedback entries.

- Code:

```python
def read_file(file_path):
    with open(file_path, 'r') as file:
        lines = file.readlines()
    return lines
```

**2. Counting Feedback Entries**

- Description: Counts the total number of feedback entries.
- Output: Prints the total number of feedback entries.
- Code:

```python
total_feedback = len(feedback_entries)
print(f"Total feedback entries: {total_feedback}")
```

**3. Extracting Keywords**

- Function: `extract_keywords(feedback)`
- Description: Extracts occurrences of specific keywords from the feedback entries.
- Input: `feedback` (list) - List of feedback entries.
- Output:** Dictionary with keyword counts.
- Code:

```python
keywords = ["good", "bad", "excellent", "poor"]

def extract_keywords(feedback):
    keyword_counts = {key: 0 for key in keywords}
    for entry in feedback:
        for keyword in keywords:
            if keyword in entry.lower():
                keyword_counts[keyword] += 1
    return keyword_counts
```

**4. Generating a Summary Report**

- Function: `generate_summary(total, keyword_counts)`

- Description: Generates a summary report containing the total number of feedback entries and the occurrences of each keyword.

- Input: `total` (int) - Total number of feedback entries, `keyword_counts` (dict) - Dictionary with keyword counts.

- Output: Summary report as a string.

- Code:

```python
def generate_summary(total, keyword_counts):
    summary = f"Total number of feedback entries: {total}\n"
    summary += "Keyword occurrences:\n"
    for keyword, count in keyword_counts.items():
        summary += f"{keyword}: {count}\n"
    return summary
```

**5. Saving the Report**

- Function: `save_report(report, file_path)`

- Description: Saves the summary report to the specified file path.

- Input: `report` (str) - Summary report, `file_path` (str) - Path to the output file.

- Code:

```python
def save_report(report, file_path):
    # Ensure the directory exists
    os.makedirs(os.path.dirname(file_path), exist_ok=True)
    with open(file_path, 'w') as file:
        file.write(report)
```

```

**6. Main Script Execution**

- Description: Executes the main processing steps.
- Code:

```python
import os

# Use an absolute path for the input file
feedback_file_path = r"C:\Users\User\Python Course SMIT\CustomerFeedbackProcessing\data\customer_feedback.txt"

# Ensure the file exists
if not os.path.isfile(feedback_file_path):
    raise FileNotFoundError(f"The file {feedback_file_path} does not exist.")

feedback_entries = read_file(feedback_file_path)

# Step 2: Counting Feedback Entries
total_feedback = len(feedback_entries)
print(f"Total feedback entries: {total_feedback}")

# Step 3: Extracting Keywords
keyword_counts = extract_keywords(feedback_entries)

# Step 4: Generating a Summary Report
summary_report = generate_summary(total_feedback, keyword_counts)
```
```

```
print(summary_report)


# Step 5: Saving the Report

# Use an absolute path for the output file

report_file_path = r"C:\Users\User\Python Course SMIT\CustomerFeedbackProcessing\summary_report.txt"


# Save the report

save_report(summary_report, report_file_path)

print(f"Report saved successfully at {report_file_path}")
```

**Running the Script**

1. Ensure the `customer_feedback.txt` file is located in the `data` folder.

2. Open a terminal or command prompt.

3. Navigate to the `CustomerFeedbackProcessing` directory.

4. Run the script using Python:

" python feedback_processing.py"

The script will read the feedback entries, count them, extract keyword occurrences, generate a summary report, and save the report to `summary_report.txt` in the specified directory. If everything is set up correctly, the terminal will display the total feedback entries and keyword occurrences, and confirm that the report has been saved successfully.

## CONTRIBUTOR:

**ID Number: PYDEVINT-240324-XT0036**
**Abdul Samad Khan ( https://www.linkedin.com/in/abdul-samad khan-15086a252/) –**