BCY CUSTOMER FEEDBACK CATEGORIZATION BY RATING 2025

A C++ Project Report

on

Customer Feedback Categorization by Rating

by

Javeria Ali BCY243018

Kainat Shabbir BCY 243006

A Project Report submitted to the
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
in partial fulfillment of the requirements for the degree of
BACHELORS OF SCIENCE IN CYBER SECURITY

Faculty of Engineering
Capital University of Science & Technology,
Islamabad

January, 2025

Customer Feedback Categorization by Rating

by

Javeria Ali BCY243018
Kainat Shabbir BCY243006

A Project Report submitted to the
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
in partial fulfillment of the requirements for the degree of
BACHELORS OF SCIENCE IN CYBER SECURITY

Faculty of Engineering
Capital University of Science & Technology,
Islamabad

January, 2025

DECLARATION

It is declared that this is an original piece of our own work, except where otherwise acknowledged in text and references. This work has not been submitted in any form for another degree or diploma at any university or other institution for tertiary education and shall not be submitted by us in future for obtaining any degree from this or any other University or Institution.

Javeria Ali
Reg. No. BCY243018

Kainat Shabbir
Reg. No. BCY243006

January,2025

CERTIFICATE OF APPROVAL

It is certified that the project titled "railway track crack detection sytem based on image processing" carried out by Javeria Ali BCY243018, Kainat Shabbir BCY243006 under the supervision of Mr. Waqas Ayub Shah, Capital University of Science & Technology, Islamabad, is fully adequate, in scope and in quality, as a final year project for the degree of BS Electrical Engineering.

Supervisor: ----------

Mr. Waqas Ayub Shah
Lecturer
Department of Electrical and Computer Engineering
Faculty of Engineering
Capital University of Science & Technology, Islamabad

HoD: ----------

Dr. Noor Mohammad Khan
Professor
Department of Electrical and Computer Engineering
Faculty of Engineering
Capital University of Science & Technology, Islamabad

ACKNOWLEDGMENT

We would like to thank “Mr. Waqas Ayub Shah” whose guidance helped to complete our report in the given time. Last but not the least, this project cannot be completed without the effort and co-operation of group members.

ABSTRACT

This program is designed to collect and summarize customer feedback. It uses a user-friendly interface to collect customer ratings and comments, and then provides a categorized summary of the feedback. The program's logic and control flow are clearly represented in a flowchart, making it easy to understand and modify the program. A simple program to collect and summarize customer feedback. This program provides a platform for customers to share their opinions and experiences, which can be invaluable for businesses seeking to improve their products and services.

The program's primary objective is to collect customer ratings and comments, and then generate a categorized summary of the feedback. This summary provides a clear and concise overview of customer opinions, enabling businesses to identify areas for improvement and optimize their products and services.

The program's user-friendly interface and straightforward navigation make it accessible to a wide range of users, from small businesses to large enterprises. The program's flexibility and scalability also enable it to be easily integrated into existing business systems and processes.

Overall, this program is a valuable tool for businesses seeking to collect, summarize, and analyze customer feedback. Its ease of use, flexibility, and scalability make it an ideal solution for businesses of all sizes and industries.

Table of Contents

Chapter 1

INTRODUCTION

Customer feedback is crucial for businesses to understand the customer satisfaction and improve their services. Ratings provided by customers help in categorizing feedback into various levels of satisfaction, such as excellent, good, average/neutral, or poor. This project, Customer Feedback Categorization by Ratings, aims to automate the process of organizing customer ratings using a C++ program. By categorizing feedback, businesses can easily identify patterns, make informed decisions, and enhance customer experience.

1.1 Overview

The project involves creating a program in C++ that takes customer feedback ratings as input and categorizes them into defined levels (e.g., 1 to 5 stars). Using programming concepts such as conditional statements, strings the program provides an efficient way to analyze feedback data. It stores the results ensuring easy accessibility for further analysis.

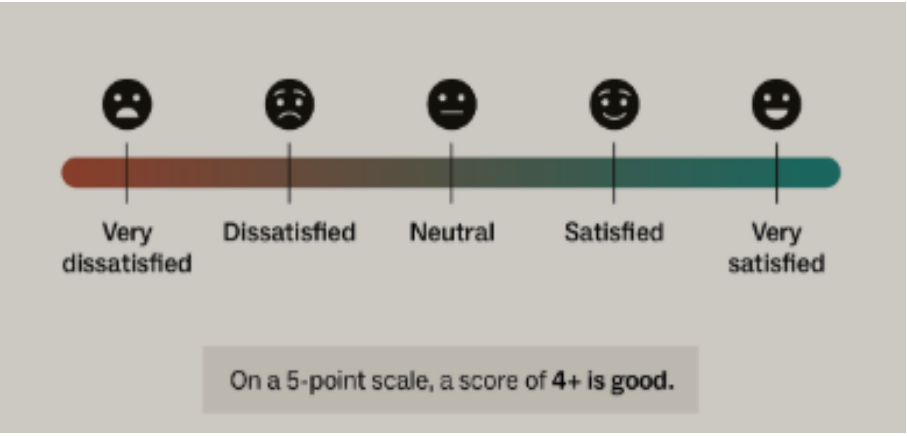


Figure 1-Sample of customer ratings

1.2 Project Idea

The main idea of the project is to develop a program that:

1. Accepts customer ratings as input(customer feedback).
2. Processes the ratings to categorize feedback (e.g., Excellent, Good, etc.).
3. Outputs the categorized data to the console screen.

1.3 Purpose of the Project

The primary purpose of this project is to automate the process of customer feedback analysis. It saves time and resources for businesses by providing quick insights. It also enhances learning and application of C++ programming concepts such as conditional statements. Serve as a practical tool for data categorization and visualization.

1.4 Project Specifications

Project Specifications are as follow:

- Input: Ratings from customers (1 to 5).
- Output: Categorized feedback displayed in the console or stored in file.

Features:

- Categorization of ratings into predefined categories.
- Percentage calculation of feedback in each category.
- File handling to save and retrieve feedback data.
- Support for processing multiple feedback entries.

1.5 Applications of the Project

The following are the application of this project:

E-commerce Websites: To categorize customer reviews for products.

Service Industry: For analyzing customer satisfaction levels.

Feedback Portals: To streamline and process feedback for large-scale surveys.

Educational Institutions: For analyzing student feedback on courses or faculty.

1.6 Project Plan

Design: Create the logic for categorizing ratings and storing data.

Development: Implement the C++ program with functionalities like input taking, categorization, and storing it.

No Invalid Choices: Ensure the program works for various inputs, no invalid choice.

1.7 Report Organization

The Customer Feedback Categorization by Rating program utilizes conditional statements to classify customer ratings into categories such as Excellent, Good, Average, Poor, based on the numerical value of the rating, the customers will enter. The program processes user input and evaluates it using if-else conditions to determine the appropriate category. Additionally, it includes functionality to handle comments provided by customers, storing them as strings for each rating. These comments offer qualitative insights alongside the numerical ratings, enhancing the feedback analysis.

Figure 2-5 star rating

Chapter 2

PROGRAM OBJECTIVES

2.1 Program features.

The Customer Feedback Categorization by Ratings program is designed to simplify and automate the process of analyzing customer feedback.

Key features include:

1.Categorization by Ratings: It classifies feedback into predefined categories (Excellent, Good, Average, Poor) based on numerical ratings.

2.Comment Handling: Customers can provide comments alongside their ratings, which are stored as strings for additional qualitative insights.

3.Dynamic Input: The program accepts multiple ratings and comments, allowing batch processing of feedback data.

4.Error Handling: Basic input validation ensures only valid ratings (1-5) are processed, minimizing inaccuracies.

2.2 Program structure

The program's structure is modular, ensuring clarity and ease of maintenance. It is organized as follows:

1.Input Taking: Captures ratings and comments from customers.

2.Processing: Utilizes conditional statements to categorize ratings and associate comments with each category.

3.Output Displaying: Displays categorized feedback on the console and writes the data to a file for storage.

2.3 Program flow

The program follows below logical flow:

Input Stage: The user enters ratings and optional comments.

Validation Stage: Input is validated to ensure correctness (e.g., ratings between 1 and 5).

Categorization Stage: Conditional statements categorize ratings and associate comments with the appropriate category.

Output Stage: Categorized feedback is displayed for customer review.

This flow ensures seamless data processing, making the program user-friendly and efficient.

2.4 Program Advantages

- **Automation:**
Reduces manual effort in organizing and analyzing feedback.
- **Scalability:**
Can handle large volumes of feedback efficiently.
- **Versatility:**
Supports both numerical and textual feedback.
- **Data Persistence:**
Stores results in files, ensuring accessibility for future use.
- **Error Reduction:**
Input validation minimizes processing errors.

2.5 Program Limitations

Basic Analysis: The program does not include advanced sentiment analysis for comments.

Limited Categories: Feedback is restricted to five fixed categories, limiting customization.

Text Input Handling: Comments are stored as raw text without any analysis or filtering.

Single User Focus: The program does not support multi-user access simultaneously.

2.6 Summary

The Customer Feedback Categorization by Ratings program is a robust solution for processing customer feedback, leveraging C++ features like conditional statements,string manipulation. Its modular structure and efficient flow ensure accurate categorization and data persistence, making it a practical tool for businesses. While the program offers significant advantages in terms

of automation and scalability, there is room for improvement in areas like advanced comment analysis and multi-user functionality. Overall, it serves as an excellent foundation for customer feedback management and further development.

Chapter 3

PROJECT STRUCTURE

3.1 Pseudo Code Of Program

Here is pseudo code of program;

i. START
ii. DISPLAY "Customer Feedback Categorization"
iii. ASK for customer rating (1-5).
iv. CHECK if rating is valid (1-5).
v. IF rating is invalid, DISPLAY error message and EXIT.
vi. ASK for customer comment.
vii. DISPLAY "Feedback Summary".
viii. CHECK rating and DISPLAY corresponding feedback (Excellent, Good, Average, Poor).
ix. DISPLAY customer comment.
x. END

3.1.1 Program Structure in parts:

- Program Name: Customer Feedback Categorization
- Purpose: To collect customer feedback and provide a categorized summary
- Display Program Title
DISPLAY "Customer Feedback Categorization"
-Collect Customer Rating
1. ASK for customer rating (1-5)
2. STORE rating in variable "rating"
-Check Customer Rating
1. CHECK if rating is valid (1-5)
2. IF rating is invalid, DISPLAY error message and EXIT
-Collect Customer Comment
1. ASK for customer comment
2. STORE comment in variable "comment"
-Display Feedback Summary
1. DISPLAY "Feedback Summary"
2. CHECK rating and DISPLAY corresponding feedback:

- Excellent (rating = 5)
- Good (rating = 4)
- Average (rating = 3)
- Below Average (rating = 2)
- Poor (rating = 1)

-Display Customer Comment

1. DISPLAY customer comment

-End Program

1. END

Chapter 4

PROJECT CODE AND EXPLANATION

4.1 Project Code

Here is the complete project code:

```
#include <iostream>
#include <string>
using namespace std;
int main () {
   int rating;
   string comment;
   cout << "******* Customer Feedback Categorization ********" <<
endl<<endl;
   cout << "Enter customer rating (1 to 5): ";
   cin >> rating;
   cin.fail();

   if (rating < 1 || rating > 5) {
      cout << "You have entered invalid rating!" << endl;
      return 1;
   }

   cin.ignore();
   cout << "Enter customer comment: ";
   getline(cin, comment);

   cout << "\n******** Feedback Summary ********" << endl;
   if (rating == 5) {
      cout << "Rating feedback: Excellent" << endl;
   }
   else if (rating == 4) {
```

```
        cout << "Rating feedback: Good" << endl;
    }
    else if (rating == 3) {
        cout << "Rating feedback: Average" << endl;
    }
    else if (rating == 2) {
        cout << "Rating feedback: Below Average" << endl;
    }
    else {
        cout << "Rating feedback: Poor" << endl;
    }
    cout << "Customer Comment: " << comment << endl;

    return 0;
}
```

4.1.1 Using of libraries in project:

1. **#include<iostream**

This library tells the compiler to include iostream file, which allows input, output operations in program.

2. **#include<string>**

This library tells the compiler to include string file, which allows for string inputs.

3. **using namespace std;**

This line tells compiler to use standard namespace, which allows for use of standard library functions without prefixing them with std::

3.1.2 Main functions

```
int main() {
    ...
}
```

- The main() function is the entry point of the program, where **execution begins**.
- The int keyword specifies that the main() function returns an integer value.

4.2 Explanation of code

1.Variable Declarations

```
int rating;
string comment;
```

- The int rating; statement declares an integer variable named rating to store the customer's rating.

- The string comment; statement declares a string variable named comment to store the customer's comment.

2. Ask user for rating

```
cout << "Enter customer rating (1 to 5): ";
cin >> rating;
```

- **cout** is used to print a message to the user.
- **cin** is used to read input from the user.
- **rating** is a variable that stores the user's input.

2. Check if input is valid

```
if (cin.fail()) {   }
```

- **cin.fail()** checks if the input operation failed.
- If the input is not a valid integer, cin.fail() returns true.

3.Clean the buffer

```
cin.ignore();
cout << "Enter customer comment: ";
getline(cin, comment);
```

- The cin.ignore(); statement ignores the newline character left in the input buffer after reading the rating.

4. Print error message

```
cout << "You have entered invalid rating!" << endl;
```

- If the input is invalid, print an error message to the user.

5. Exit program with error code

```
return 1;
```

- If the input is invalid, exit the program with an error code of 1.

6. Print rating feedback

```
 cout << "Rating feedback: ";
 if (rating == 5) {
   cout << "Excellent" << endl;
```

```
  } else if (rating == 4) {
    cout << "Good" << endl;
  } else if (rating == 3) {
    cout << "Average" << endl;
  } else if (rating == 2) {
    cout << "Below Average" << endl;
  } else if (rating == 1) {
    cout << "Poor" << endl;
  } else {
    cout << "Invalid rating!" << endl;
  }
  return 0;
}
```

The use of conditional statements makes the code very easy to use. Like if customer has entered the rating of 5, it is considered as excellent, else-if rating is 4, it is considered as good or nice and at rating 2,1 it is considered poor, all of these are categorized by using of if-else statements.

4.3 Output Samples

4.3.1 Taking output from user:

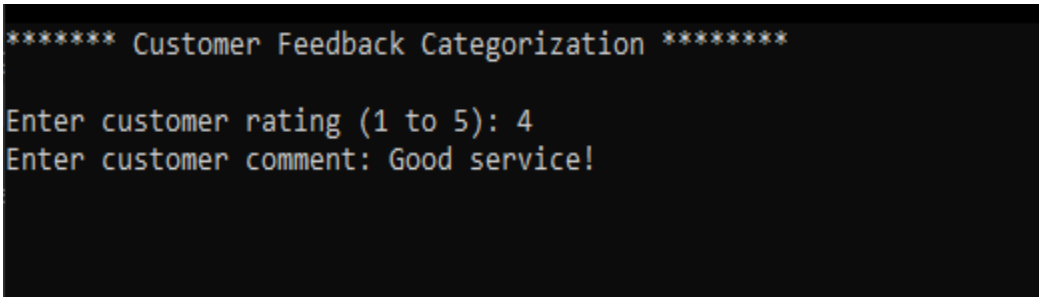


Figure 3-Taking input from user

4.3.2. Storing the comment and rating of customer:

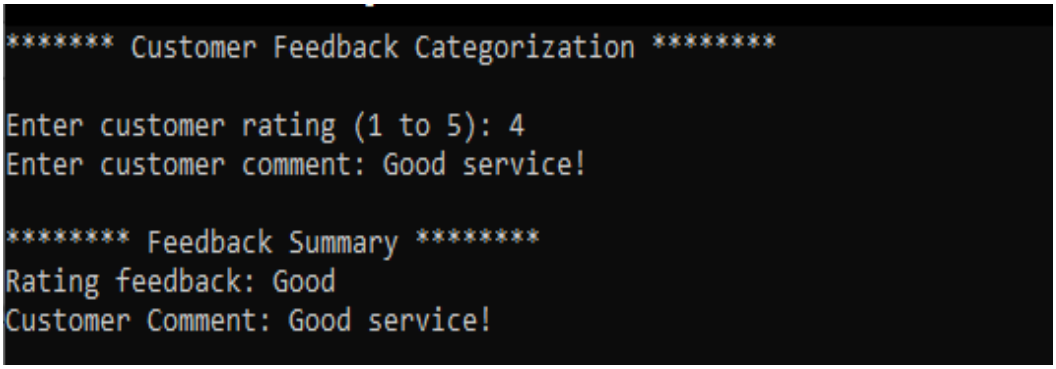


Figure 4-Storing customer comment

4.3.3. Invalid Case

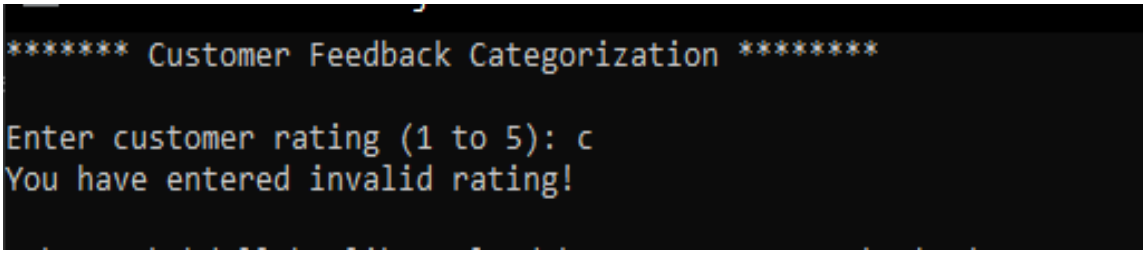


Figure 5-Invalid rating (Error Case)

4.3.4 Poor Feedback

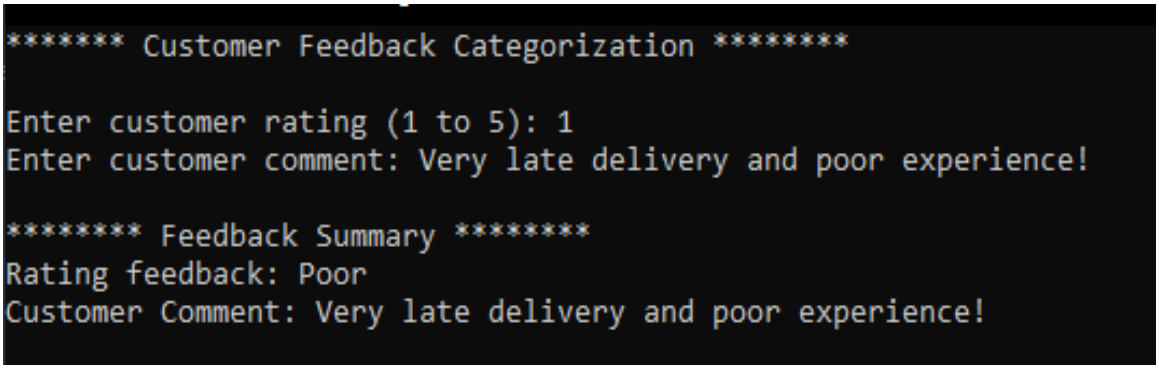
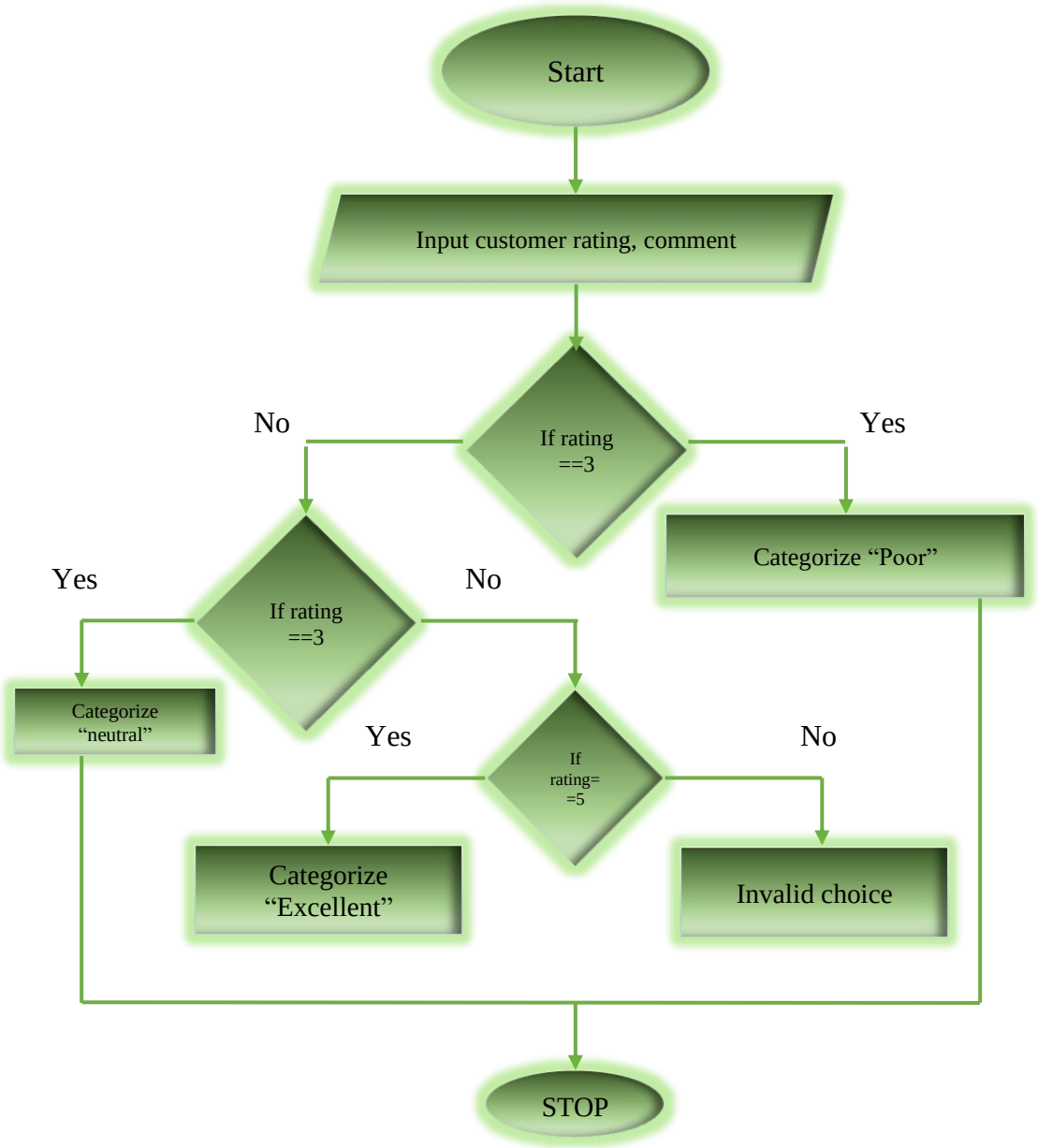


Figure 6-Poor feedback (1-star rating)

Chapter 5

PROGRAM FLOWCHART

The flowchart uses decision symbols to represent conditional statements, such as checking if the customer rating is valid. It also uses input/output symbols to represent user interactions, such as collecting customer ratings and comments.
The flowchart of the program is following:


Start
Input customer rating, comment
If rating ==3
No
Yes
Categorize “Poor”
If rating ==3
Yes
No
Categorize “neutral”
If rating= =5
Yes
No
Categorize “Excellent”
Invalid choice
STOP

Chapter 6

CONCLUSION

The Customer Feedback Categorization program is a simple yet effective tool for collecting and summarizing customer feedback. The program uses a user-friendly interface to collect customer ratings and comments, and then provides a categorized summary of the feedback. The program's logic and control flow are clearly represented in the flowchart, making it easy to understand and modify the program.
The Customer Feedback Categorization program is a comprehensive tool designed to collect and summarize customer feedback in a structured and organized manner. The program's primary objective is to provide a platform for customers to share their opinions and experiences, which can be invaluable for businesses seeking to improve their products and services.
Throughout the program, customers are guided through a simple and intuitive interface that enables them to provide ratings and comments. The program then processes this feedback and generates a categorized summary, which provides a clear and concise overview of customer opinions.
One of the key benefits of the Customer Feedback Categorization program is its ability to provide actionable insights for businesses. By analyzing customer feedback, businesses can identify areas for improvement, optimize their products and services, and ultimately enhance customer satisfaction.
Furthermore, the program's user-friendly interface and straightforward navigation make it accessible to a wide range of users, from small businesses to large enterprises. The program's flexibility and scalability also enable it to be easily integrated into existing business systems and processes.
In conclusion, the Customer Feedback Categorization program is a valuable tool for businesses seeking to collect, summarize, and analyze customer feedback. Its ease of use, flexibility, and scalability make it an ideal solution for businesses of all sizes and industries.

References

- C++ Standard Library
- https://en.cppreference.com/w/
Standard Library
- http://www.cplusplus.com/reference/
- Strings
- https://en.cppreference.com/w/cpp/string/basic_string
- If-Else Statements
- https://en.cppreference.com/w/cpp/language/if - documentation for the if statement
- http://www.cplusplus.com/doc/tutorial/control/ - documentation for the if statement
- Cin and Input Validation
- https://en.cppreference.com/w/cpp/io/basic_istream

- Error Handling
- https://en.cppreference.com/w/cpp/error/exception - documentation for the std::exception class
- http://www.cplusplus.com/reference/exception/exception/ - documentation for the std::exception class