



# IT(161) Project :

## *Telecom Billing System*

**BY-**

Pradyumn Prasad (202451132)

Ishita Garg (202452322)

Suhani Kabra (202451157)

## Acknowledgment

First and foremost, I am sincerely thankful to our professors for continuous support throughout this journey. Their expertise and encouragement have been instrumental in shaping this project and enhancing my learning experience.

I am grateful to IIITV and faculty members for the conducive environment and resources to undertake this project.

Special thanks go to my classmates and peers for their insightful suggestions and collaborative spirit, which encouraged me to approach problems from fresh perspectives.

This project has been a tremendous learning experience, and I am deeply appreciative of the support and inspiration I have received at every step of this endeavor.

## INDEX

SR. NO.	Topic
1.	Description
2.	Methodology
3.	Source code link
4.	Code output

# Project Description:

This C program is a customer record management system for tracking usage and billing.

It allows users to:

Add new customer records, including name, phone number, and usage.

View all records in a formatted list.

Modify an existing record by updating the customer's usage and recalculating the total bill.

View payment details for a specific customer. Delete records by matching a phone number.

Persist records in a file named `customer_records.txt` for data retention across sessions.

The program uses file I/O to load and save records, ensuring data consistency.

It features a menu-driven interface for ease of use.

# METHODOLOGY

The program is a **Customer Records Management System** that uses structured programming and file handling to manage customer data, including name, phone number, usage, and total bill. It features a menu-driven interface (switch case) for adding, viewing, modifying, deleting, and viewing payment records. Customer data is stored in a file and saved upon updates. The system calculates bills based on usage, manages up to 100 records using an array, and ensures error handling for file operations and invalid inputs.

## 1) File Handling:

- Save the customer's records safely and in a structured way.
- Read Mode ("r"): We have used this for loading records in the loadRecords function. If the file won't exist, it will show "No existing data found."
- Write Mode ("w"): We have used this in the saveRecords function to overwrite the file with updated data after adding, modifying, or deleting a record.
- We have used *fprintf* to write each customer's data into the file in a structured format :
- We have used *fscanf* with a format specifier to parse each line and extract: Name, Phone number, Usage, Total bill, And store the data in the customers array and further increment the customer count.

## 2) Structures:

- Providing a clear data model for customer information.
- Simplifying file handling of customer records.
- For the efficient implementation of Create, Read, Update and Delete operation in the project.
- Minimised the use of multiple independent arrays for different fields making the code more readable and easy to maintain.

### 3)Switch case:

- Navigates Menu Option which provides a user-friendly way to select and execute specific functions of the project, such as adding, viewing, modifying, or deleting customer records.
- Controls Program Flow that ensures appropriate action is executed based on the user's choice.
- It simplifies the code's structure making it more organized compared to using multiple if-else statements.

### 4) Creating different functions:

- **void displayMenu():** Displays the main menu with options like Add, View, Modify, Delete, and Exit.
- **void loadrecords():** Reads existing customer data from a file and stores that in a temporary array.
- **void saverecords():** Writes all in-memory customer data back to the file to save updates.
- **void addrecord():** Adds a new customer record to the temporary array. Automatically calculates the total bill based on usage and saves the updated records to the file.
- **void viewrecords():** Displays all customer records in a tabular format.
- **void modifyrecord(char phonenumber[]):** Finds a customer by phone number, updates their usage, and recalculates the bill.
- **void deleterecord(char phonenumber[]):** Removes a customer record by phone number from the temporary array and updates the file.
- **void viewpayment(char phonenumber[]):** Displays payment details for a specific customer based on their phone number.

## 5) String Methodology:

- Strings was used to store customer's names in a structured format.
- The program uses `scanf()` to read strings from the user (`%[^\n]s`, to include spaces, making it suitable for multi-word names)
- `strcmp()` - used to compare two strings to check if they are identical, under header file `#include<string.h>`.
- Fixed-length arrays ensure no memory wastage if the string inputs are properly validated.

# Source Code link-

## Github-

<https://github.com/Pradyumn-cloud/IT-161--Project/tree/main>



# Input and Output:

Adding records in File name - "customer record.txt":

```
PS C:\Users\Tuf\Desktop\Project> cd "c:\Users\Tuf\Desktop\Project\" ; if ($?) { gcc First.c -o First } ; if ($?) { .\First }
No existing data found. Starting fresh.

1. Add New Record
2. View List of Records
3. Modify Record
4. View Payment
5. Delete Record
6. Exit
Enter your choice: 1

Enter name: Pradyumn Prasad
Enter phone number: 7096165161
Enter usage (in minutes): 600

Record added successfully!
```

```
1. Add New Record
2. View List of Records
3. Modify Record
4. View Payment
5. Delete Record
6. Exit
Enter your choice: 1

Enter name: Basudev Mahapatro
Enter phone number: 543424254
Enter usage (in minutes): 455

Record added successfully!
```

```
1. Add New Record
2. View List of Records
3. Modify Record
4. View Payment
5. Delete Record
6. Exit
Enter your choice: 1

Enter name: Suhani Kabara
Enter phone number: 1234567891
Enter usage (in minutes): 1

Record added successfully!
```

```
1. Add New Record
2. View List of Records
3. Modify Record
4. View Payment
5. Delete Record
6. Exit
Enter your choice: 1

Enter name: Pradyy
Enter phone number: 23982842
Enter usage (in minutes): 688

Record added successfully!
```

```
1. Add New Record
2. View List of Records
3. Modify Record
4. View Payment
5. Delete Record
6. Exit
Enter your choice: 1

Enter name: Ishita Garg
Enter phone number: 8595569388
Enter usage (in minutes): 100000000

Record added successfully!
```

```
1. Add New Record
2. View List of Records
3. Modify Record
4. View Payment
5. Delete Record
6. Exit
Enter your choice: 1

Enter name: Karan Sir
Enter phone number: 69696969
Enter usage (in minutes): 69

Record added successfully!
```

```
1. Add New Record
2. View List of Records
3. Modify Record
4. View Payment
5. Delete Record
6. Exit
Enter your choice: 1

Enter name: Ram
Enter phone number: 3456765432
Enter usage (in minutes): 3456543
```

```
1. Add New Record
2. View List of Records
3. Modify Record
4. View Payment
5. Delete Record
6. Exit
Enter your choice: 1

Enter name: Jay Hite
Enter phone number: 43992823
Enter usage (in minutes): 556

Record added successfully!
```

```
1. Add New Record
2. View List of Records
3. Modify Record
4. View Payment
5. Delete Record
6. Exit
Enter your choice: 1

Enter name: rohan
Enter phone number: 32434543
Enter usage (in minutes): 34565

Record added successfully!
```

## Viewing records from File name - " customer record.txt ":

Enter phone number to view payment: 7096165161

Pradyumn Prasad 7096165161      6000.00      600.00

1. Add New Record
2. View List of Records
3. Modify Record
4. View Payment
5. Delete Record
6. Exit

Enter your choice:    2

Name	Phone Number	Usage(min)	Total Bill(\$)
Pradyumn Prasad	7096165161	6000.00	600.00
Ishita Garg	8595569388	200.00	20.00
Suhani Kabra	85894837268	50000.00	5000.00
Karan Sir	698689698689	687.00	68.70
garv	34353536353	34535.00	3453.50
yatharth	34383782	454.00	45.40
ayushi 12	12.00	1.20	
ravi 123	123.00	12.30	
swapnil sir	1234	1234.00	123.40
blacksmith	12345	12345.00	1234.50
kallu kaliya	123456	123456.00	12345.60
justin 1234567	1234567.00		123456.70
billie 12345678		12345678.00	1234567.75
dhirendra sir	696969	69.00	6.90
ajay sir	12345	12345.00	1234.50
maths ta	1234566788	12.00	1.20
mahi bhai	123	123.00	12.30

## Modifying records from File name - " customer record.txt ":

```
1. Add New Record
2. View List of Records
3. Modify Record
4. View Payment
5. Delete Record
6. Exit
Enter your choice: 3

Enter phone number to modify record: 8595569388

Enter new usage (in minutes) for Ishita Garg: 1

Record modified successfully!
```

## Viewing Payment records from File name - " customer record.txt ":

```
1. Add New Record
2. View List of Records
3. Modify Record
4. View Payment
5. Delete Record
6. Exit
Enter your choice: 4

Enter phone number to view payment: 7096165161
Pradyumn Prasad 7096165161      6000.00      600.00
```

### Deleting records from File name - "customer record.txt":

```
1. Add New Record
2. View List of Records
3. Modify Record
4. View Payment
5. Delete Record
6. Exit
Enter your choice:
5

Enter phone number to delete record: 7096165161

Record deleted successfully!
```

### Terminating Program:

```
1. Add New Record
2. View List of Records
3. Modify Record
4. View Payment
5. Delete Record
6. Exit
Enter your choice: 6
Exiting the program. Goodbye!
PS C:\Users\Tuf\Desktop\Project> █
```

## **CHALLENGES FACED WHILE WORKING ON THIS PROJECT**

- Time constraints for project completion.
- Difficulty in debugging logical errors.
- Gathering the requirements for a telecom system.
- Distributing tasks between all the members of the group.
- Balancing the project with other academic or personal commitments.

