Programme Name and Code: CO-3-I **Academic Year: 2020-2021**

Course Name and Code: DSU (22317) **Semester: Third**

MICRO PROJECT REPORT

ON

BANK MANAGEMENT SYSTEM

Submitted in Jan 2021 by the group of (02) students

Sr. No	Roll No (Sem-III)	Full name of Student	Enrollment No	Seat No (Sem-III)
1	33	Kaushal Sunil Khachane	1900210032	
2	49	Abhishek Sunil Patil	1900210325	

Under the Guidance of

Prof. Vijay Bande

In

Three Years Diploma Programme in Computer Engineering of Maharashtra State Board of Technical Education, Mumbai (Autonomous)

ISO 9001:2008 (ISO/IEC-27001:2013)

At

Government Polytechnic Khamgaon

MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION, MUMBAI

Certificate

This is to certify that Mr. Kaushal Sunil Khachane RollNo: 33 of Third **Semester** of Government polytechnic Khamgaon **Diploma Programme in Computer Engineering** at Government polytechnic Khamgaon **has** completed the **Micro Project** satisfactorily in Subject **Data Structure using C (22317)** in the academic year 2020-2021 as per the MSBTE prescribed curriculum of I Scheme.

Place: khamgaon Enrollment No: 1900210032

Date: / / 2021 Exam. Seat No: ____________________

Project Guide **Head of the Department** **Principal**

Prof. Vijay Bande

Seal of Institute

INDEX

1. Annexure II

Micro Project Report:
Bank Management System

1.0 Rationale:

Data Structure is an Important aspect for Computer Engineering and Information Technology Diploma graduates. Data Structure Is Logical & Mathematical model of Storing & Organizing Data in a Particular way in a Computer. The methode and techniques of Data Structure are widely used in industries. After learing the Subject Student will able to identify the proplem, analyse different algorithums to solve the problems and choose most appropriate data structure to represent the data.

3.0 Aim /Benefits of the Micro-Project:

Aim - Bank Management System

Benefits - It manages all the transactions like new account entry, deposit as well as withdraw entry, transaction of money for various processes etc. Thus, above features of this software will save account created time and therefore increase the efficiency of the system.

4.0 Course Outcomes Achieved:

1. Implement basic operation on linked list.
2. Write an algorithum to Search the given node using linear Search.
3. Create relevant Structure to represent node using linked list

5.0 Literature Review:

Bank Management System using c this project is developed using linked list. Linked list is a non-linear data structure. Dynamic memory allocation make linked list operation efficient. We can store customer data upto memory becomes available.

Various operation performed on linked list are insertion, deletion, traversing. We can use such in our project. By using such function we implement bank management system project.

5.0 Actual Methodology Followed:

1. Decide project Topic
2. Design Schema for project
3. Write function for management of user data (create. modify, delete, display, transaction)
4. Develope algorithum and flowchart for each function
5. At last we create report and powerpoint presentation for project.

6.0 Actual Resources Used

Sr. no.	Name of Resources/material	Specification	Qty.	Remarks
1	Computer System	I3 or i5 Processor with 2 gb of Ram	1	
2	I/O devices	Keyboard, Mouse	1	

2. ABSTRACT

The Bank Account Management System is an application for maintaining a person's account in a bank. In this project I tried to show the working of a banking account system and cover the basic functionality of a Bank Account Management System. In this Project user can Create Account, Modify Account, Manage Transaction, Delete Account, Search Existing Account and Display All the Account In Bank.

The main aim of this project is to develop software for Bank Account Management System. This project has been developed to carry out the processes easily and quickly, which is not possible with the manuals systems, which are overcome by this software. This project is developed using Linked list in C Language. Thus, above features of this project will save transaction time and therefore increase the efficiency of the system.

3. INTRODUCTION

Bank Management System

1.1 Project Details

The main objective of the project is to develop banking management system for banks. In banking system all banking work is done manually. User have to visit bank to withdrawal or deposit amount. In present bank system it is also difficult to find account information of account holder. In this bank management system we will automate all the banking process. In our bank management system user can check his balance and he can also deposite and withdrawal money. In this software you can keep record for daily banking transactions. The main purpose of developing bank management system is to design an application, which could store bank data and provide an interface for retrieving customer related details with 100% accuracy.

This bank management system also allow user to add new customer account, delete account and user can also modify existing user account information. Using this system user can also search any individual account in few seconds. Using our bank management system user can also check any translation in your account. The source code of this project is around 470 lines. The source is password protected (password is dsuproject).

1.2 HEADER FILE USED IN THIS MICROPROJECT IMPLEMENTATION:

1. **stdio.h** – This Header file is use to print and accept data from user i.e printf() and scanf()
2. **stdlib.h** – This Header file includes functions involving memory allocation malloc() & free()
3. **windows.h** – This header file contains declarations for all of the functions in the Windows API
   Such as system("cls") – for clear screen.
4. **time.h -** The time.h header file contains definitions of functions to get and manipulate date
   and time information.
   clock_t **-** it represents the date as integer which is part of the calendar time.

1.3 FUNCTION USED IN THIS MICROPROJECT IMPLEMENTATION:

- **menu()** – This function displays the menu or welcome screen to perform different banking activities mentioned below.
- **new_ac()** – This function creates a new customer account. It asks for some personal and banking details of the customer such as name, date of birth, address and phone number.
- **modify_account()** – This function has been used for changing the address and Name of a particular customer account.
- **transaction()** – With this function, you can deposit and withdraw money to and from a particular customer account.
- **del_account()** – This function is for deleting a customer account.
- **Search()** – This function shows account number, name, date of birth, age, address, phone number, type of account, amount deposited and date of deposit.
- **List()** – Display all Record in Bank
- **delay()** – To add Time delay

1.4 STRUCTURE:

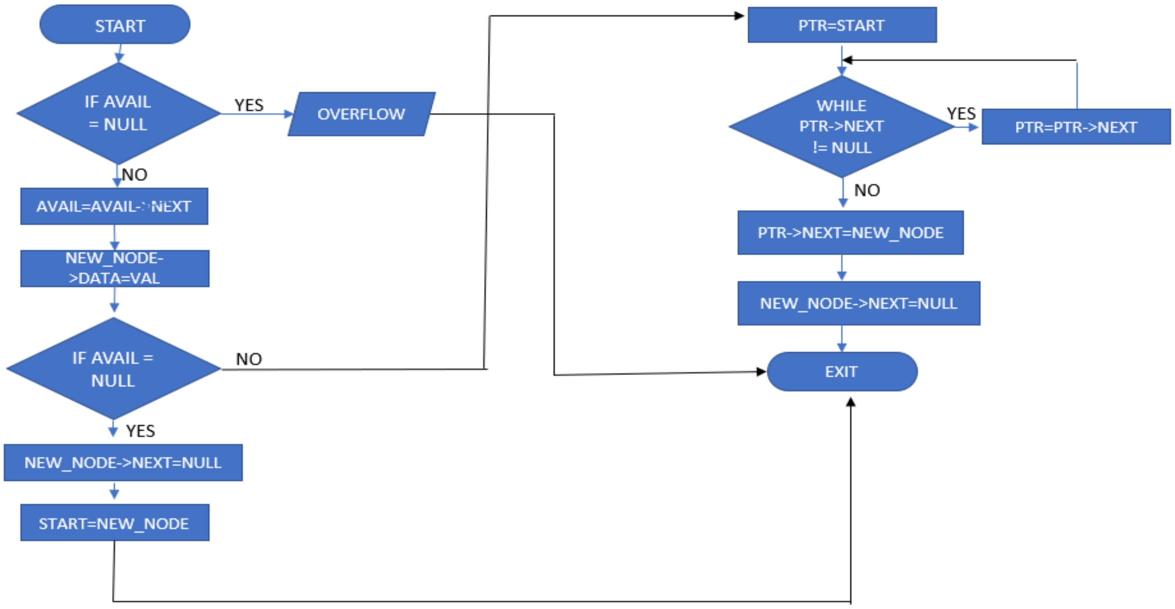
In this Project we use following structure to store data of customers.

```
struct date
{
            int month, day, year;
 };
```

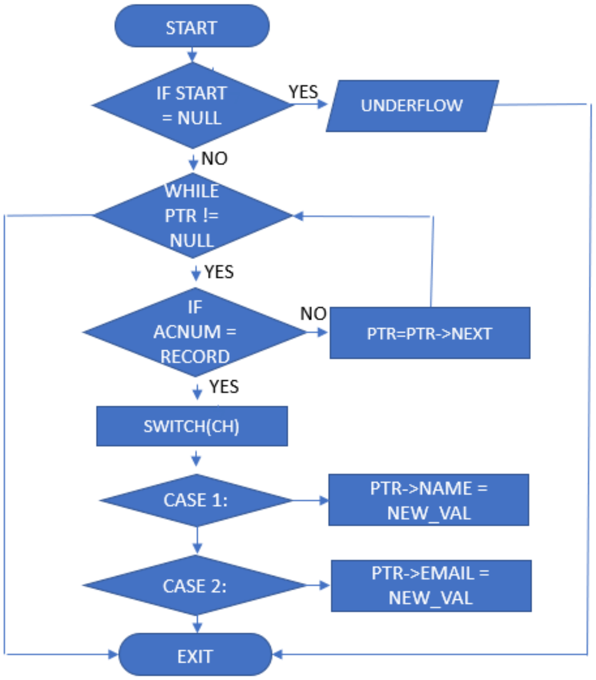
```
struct node{
           int ac_num, age;
           char name[20];
           char address[40];
           char actype[20];
           int phone;
           float amt;
           struct date deposit;
           struct date dob;
           struct node *next;
};
```

4. Flowchart for Micro Project

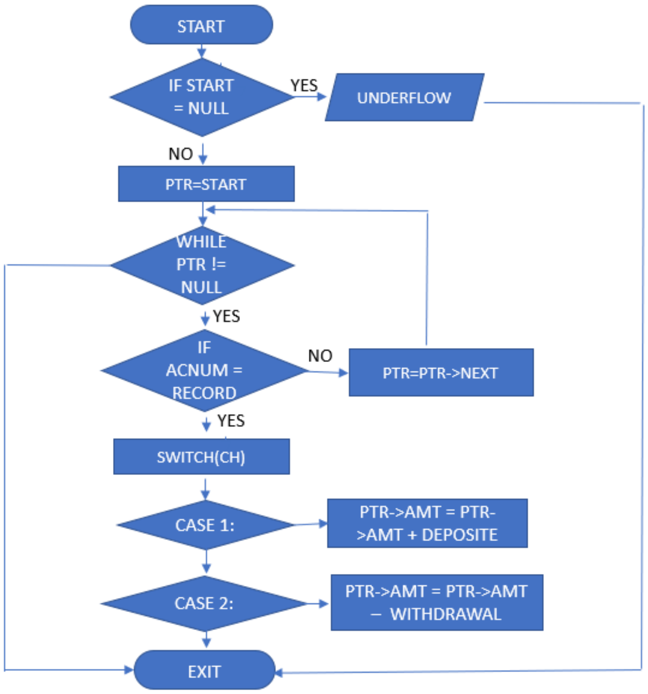
1. Flowchart To Insert Data Into Account



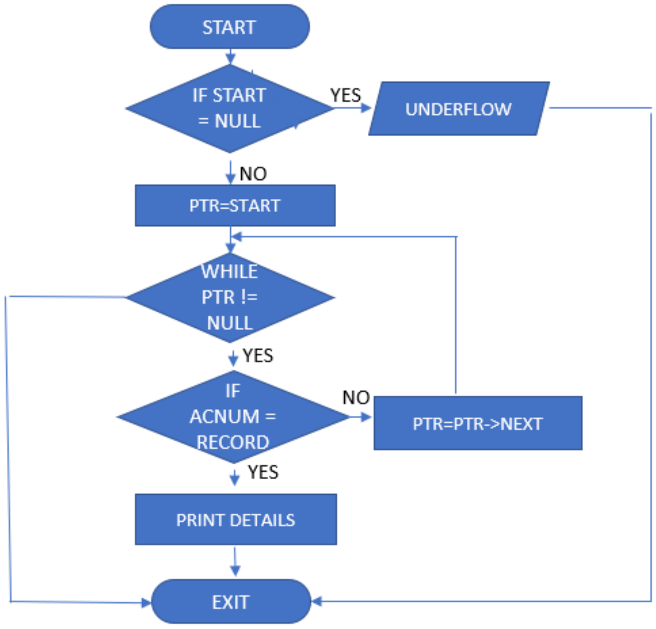
2. Flowchart To Update Details Of Exiting Account



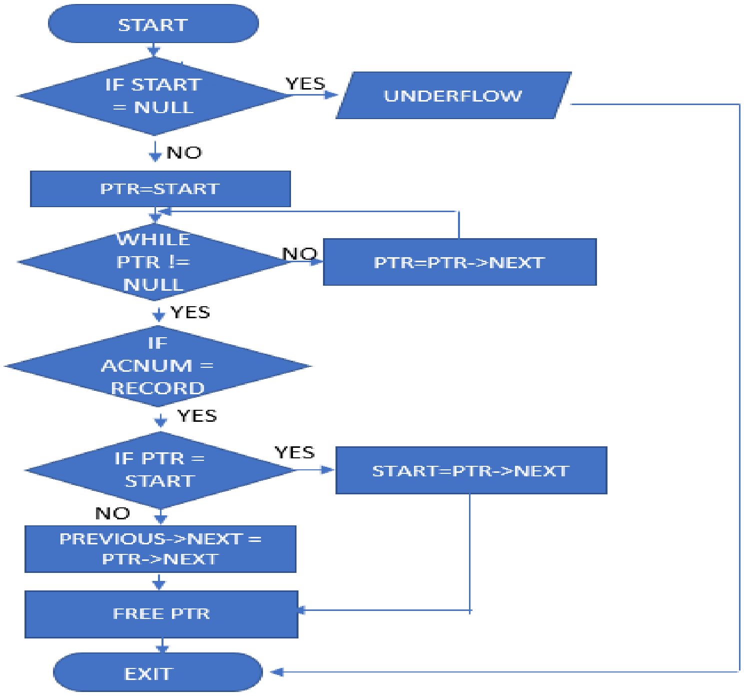
3. Flowchart To Manage Transaction



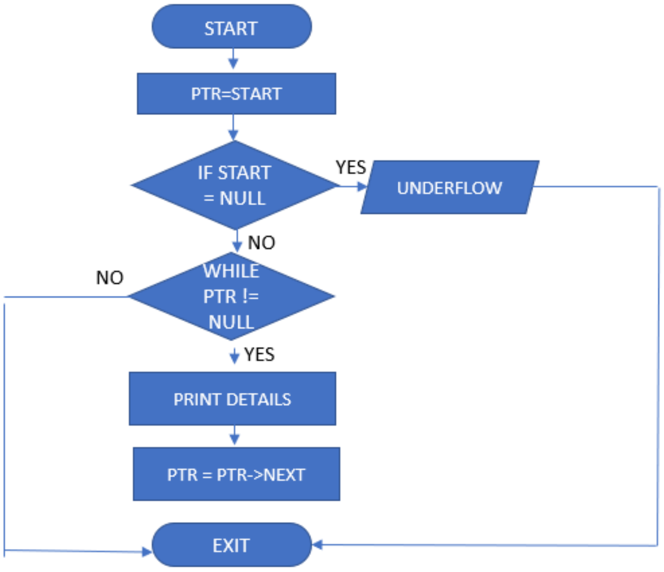
4. Flowchart To Check Details Of Existing Account



5. Flowchart To Remove Existing Account



6. Flowchart To Display All Record



5. Algorithm For Micro Project

1. Algorithm to insert Data into Account

```
STEP 1: START
STEP 2: IF AVAIL =  NULL
                  WRITE OVERFLOW
                  GO TO STEP 15
STEP 3: SET NEW_NODE = AVAIL
STEP 4: SET AVAIL = AVAIL->NEXT
STEP 5: SET NEW_NODE->DATA = VAL
STEP 6: IF START = NULL
STEP 7:       NEW_NODE->NEXT = NULL
STEP 8:       START = NEWNODE
STEP 9: ELSE
STEP 10:      PTR = START
STEP 11:      WHILE PTR->NEXT != NULL
STEP 12:                 PTR = PTR->NEXT
                  [END OF LOOP]
STEP 13: PTR->NEXT = NEW_NODE
STEP 14: NEW_NODE->NEXT = NULL
             [END OF IF]
STEP 15: EXIT
```

2. Algorithm to update record in program

```
STEP 1: START
STEP 2: IF START = NULL
            WRITE UNDERFLOW (THERE IS NO BANK ACCOUNT TO MODIFY)
                GO TO STEP 12
STEP 3: ELSE
STEP 4:     WHILE PTR != NULL
STEP 5:         IF ACNUM = RECORD
STEP 6:             SWITCH(CH)
STEP 7:                 CASE 1: PTR->NAME = NEW_VAL BREAK
STEP 8:                 CASE 2: PTR->EMAIL = NEW_VAL BREAK;
STEP 9:                 DEFAULT:    TRY AGAIN
STEP 10:        ELSE
STEP 11:            PTR = PTR->NEXT
                [END OF IF]
            [END OF LOOP]
        [END OF IF]
 STEP 12: EXIT
```

3. Algorithum For Manage Transaction

```
Step 1: START
Step 2: IF START = NULL
Step 3:      Write UNDERFLOW (THERE IS NO BANK ACCOUNT TO MODIFY)
             Go To Step 13
Step 4: ELSE
Step 5:      PTR = START
Step 6:       WHILE PTR != NULL
Step 7:                IF AcNum = Record
Step 8:                         Switch(ch)
Step 9:                                  Case 1 = PTR->AMT = PTR->AMT + DEPOSITE
Step 10:                                 Case 2 = PTR->AMT = PTR->AMT –  WITHDRAWAL
                                         Default = TRY AGAIN
Step 11:               ELSE
Step 12:               PTR = PTR->NEXT
                       [END OF IF]
              [END OF LOOP]
          [END OF IF]
Step 13: EXIT
```

4. Algorithm to delete Account from Program

```
STEP 1: START
STEP 2: IF START = NULL
                WRITE UNDERFLOW (THERE IS NO BANK ACCOUNT TO MODIFY)
            [END OF IF]
STEP 3: PTR = START
STEP 4:     WHILE PTR != NULL
STEP 5:             IF ACNUM = RECORD
STEP 6:                     IF PTR = START
STEP 7:                             START = PTR->NEXT
STEP 8:                     ELSE
STEP 9:                             PREVIOUS->NEXT = PTR->NEXT
STEP 10:                    FREE PTR
                            [END OF IF]
STEP 10:            ELSE
STEP 11:                    PREVIOUS = PTR
STEP 12:                    PTR = PTR->NEXT
                    [END OF IF]
            [END OF LOOP]
        [END OF IF]
STEP 13: EXIT
```

5. Algorithm to Search Details of existing account

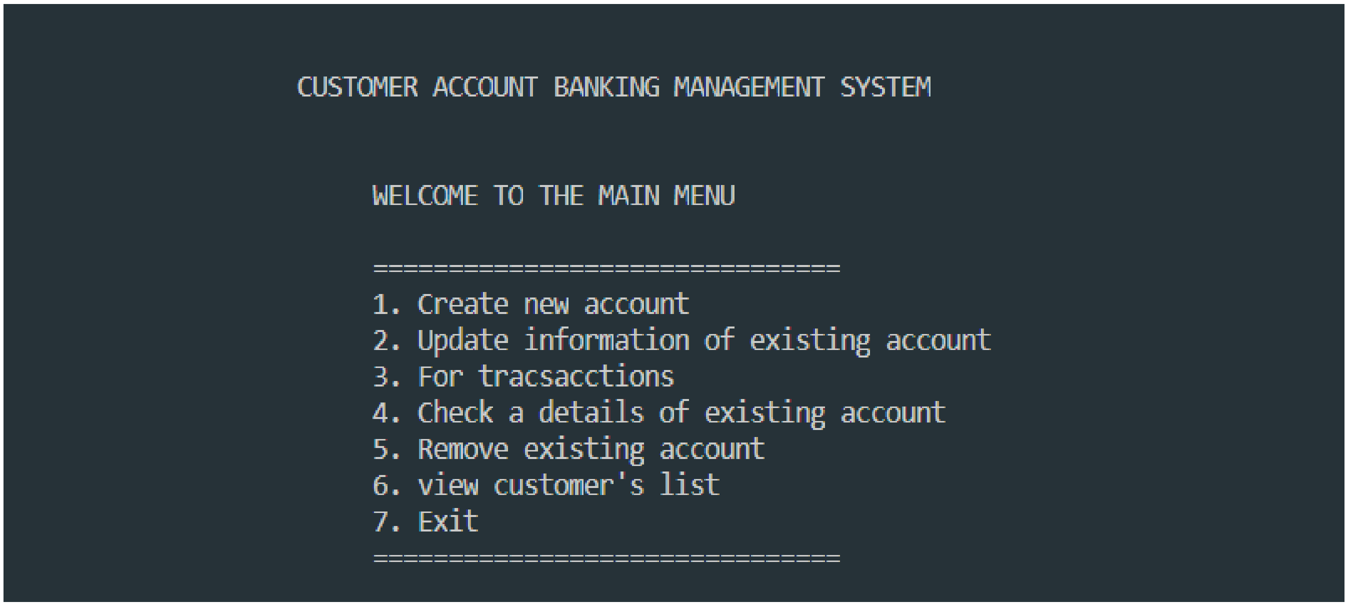
```
STEP 1: START
STEP 2: IF START = NULL
            WRITE UNDERFLOW (THERE IS NO BANK ACCOUNT TO MODIFY)
STEP 3: ELSE
STEP 4:     PTR = START
STEP 5:      WHILE PTR != NULL
STEP 6:             IF ACNUM = RECORD
STEP 7:                     PRINT DETAILS
STEP 8:     ELSE
STEP 9:             PTR = PTR->NEXT
                    [END OF IF]
            [END OF LOOP]
        [END OF IF]
STEP 10: EXIT
```

6. Algorithm To Print all details in Program

```
STEP 1: START
STEP 2: PTR = START
STEP 3: IF START = NULL
            WRITE UNDERFLOW (THERE IS NO BANK ACCOUNT TO MODIFY)
      [END OF IF]
STEP 4: ELSE
STEP 5:     DO WHILE PTR !=NULL
STEP 6:             PRINT DETAILS
STEP 7:             PTR = PTR->NEXT
            [END OF DO WHILE]
      [END OF IF]
STEP 8: EXIT
```

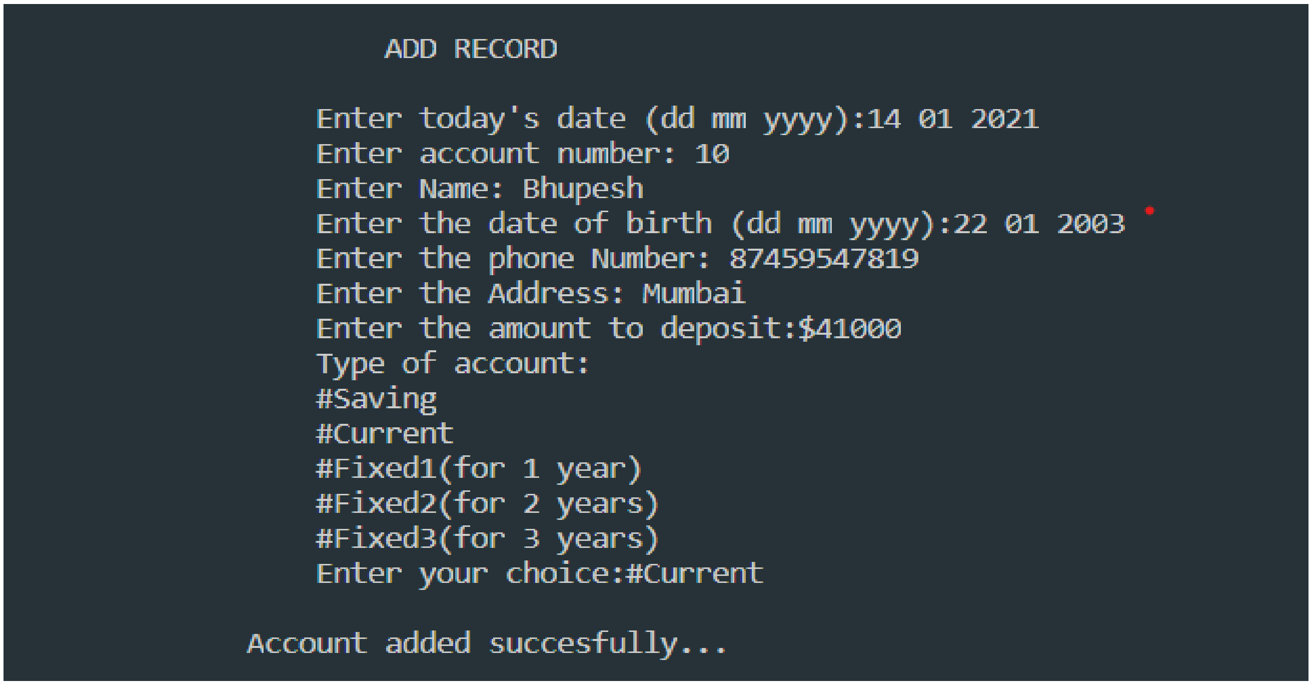
6. Micro Project Outputs

1. Menu



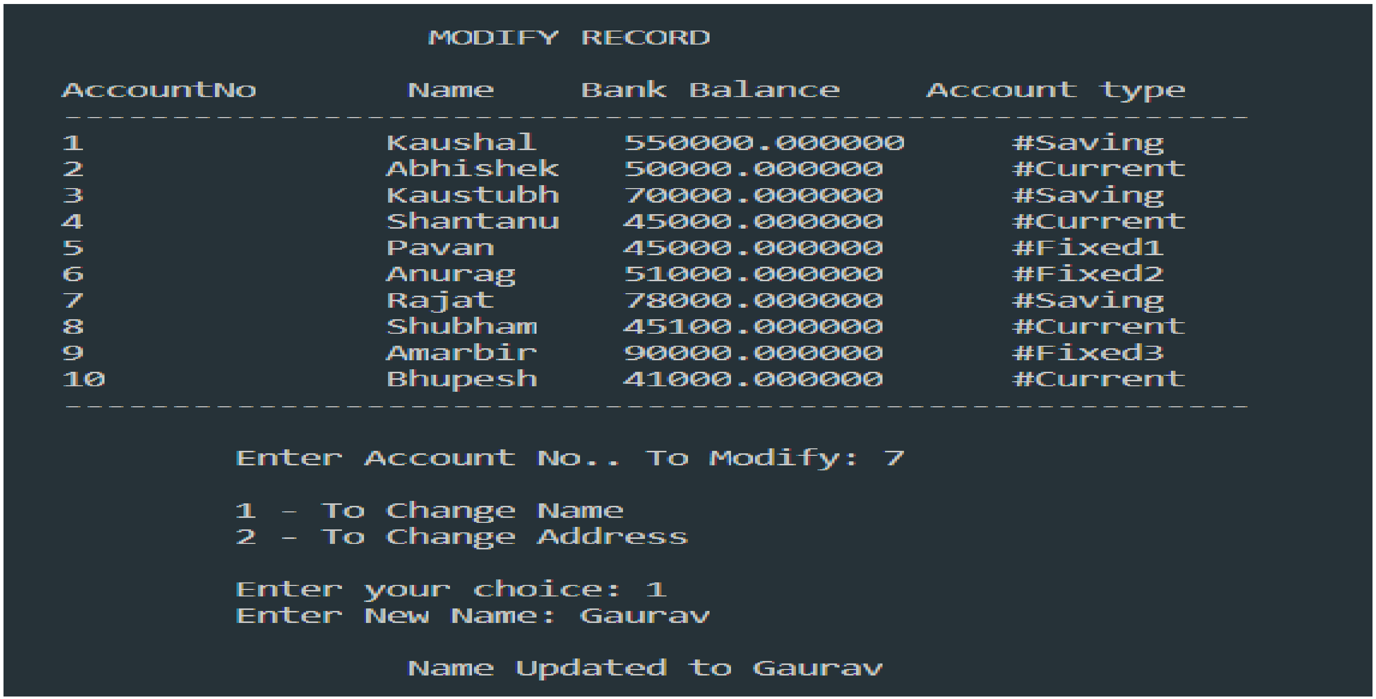
```
          CUSTOMER ACCOUNT BANKING MANAGEMENT SYSTEM


                WELCOME TO THE MAIN MENU

                ================================
                1. Create new account
                2. Update information of existing account
                3. For tracsacctions
                4. Check a details of existing account
                5. Remove existing account
                6. view customer's list
                7. Exit
                ================================
```

2. Create Account



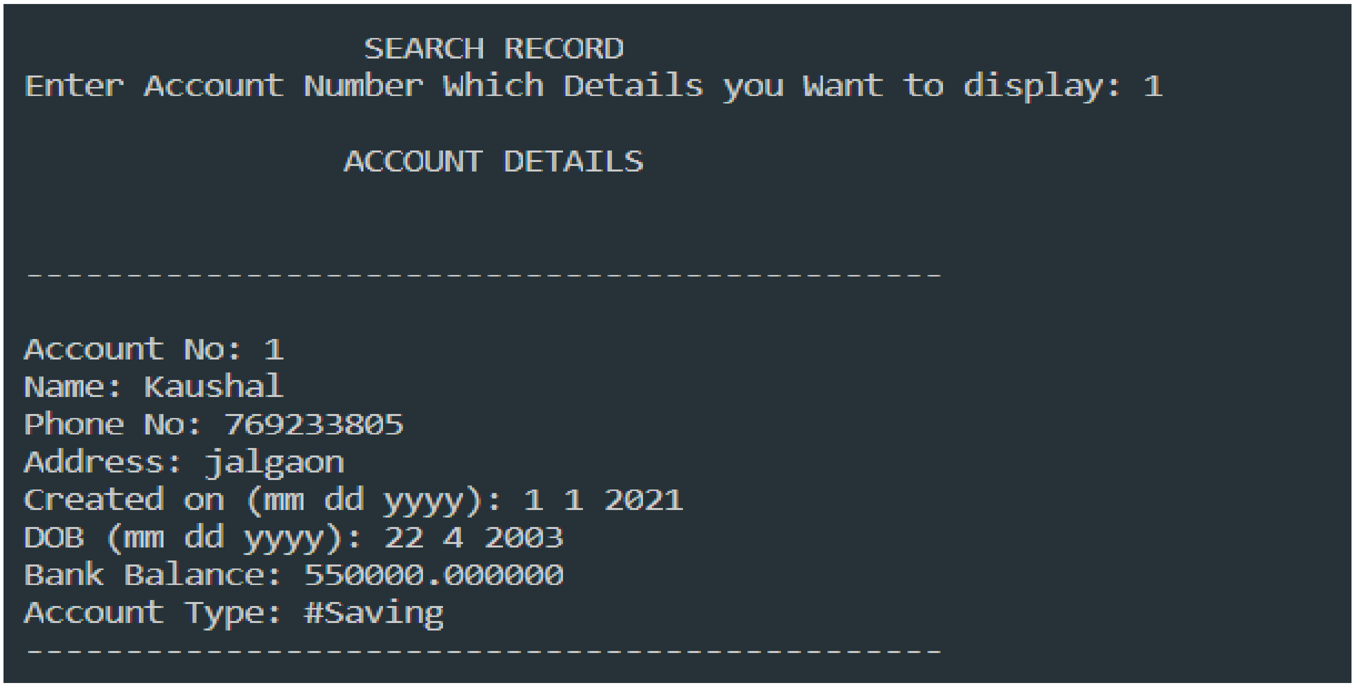
```
                ADD RECORD

            Enter today's date (dd mm yyyy):14 01 2021
            Enter account number: 10
            Enter Name: Bhupesh
            Enter the date of birth (dd mm yyyy):22 01 2003
            Enter the phone Number: 87459547819
            Enter the Address: Mumbai
            Enter the amount to deposit:$41000
            Type of account:
            #Saving
            #Current
            #Fixed1(for 1 year)
            #Fixed2(for 2 years)
            #Fixed3(for 3 years)
            Enter your choice:#Current

        Account added succesfully...
```

3. Modify Account Record



```
                       MODIFY RECORD

   AccountNo          Name     Bank Balance     Account type
   -------------------------------------------------------
   1                Kaushal     550000.000000      #Saving
   2                Abhishek    50000.000000       #Current
   3                Kaustubh    70000.000000       #Saving
   4                Shantanu    45000.000000       #Current
   5                Pavan       45000.000000       #Fixed1
   6                Anurag      51000.000000       #Fixed2
   7                Rajat       78000.000000       #Saving
   8                Shubham     45100.000000       #Current
   9                Amarbir     90000.000000       #Fixed3
   10               Bhupesh     41000.000000       #Current
   -------------------------------------------------------

           Enter Account No.. To Modify: 7

           1 - To Change Name
           2 - To Change Address

           Enter your choice: 1
           Enter New Name: Gaurav

                   Name Updated to Gaurav
```

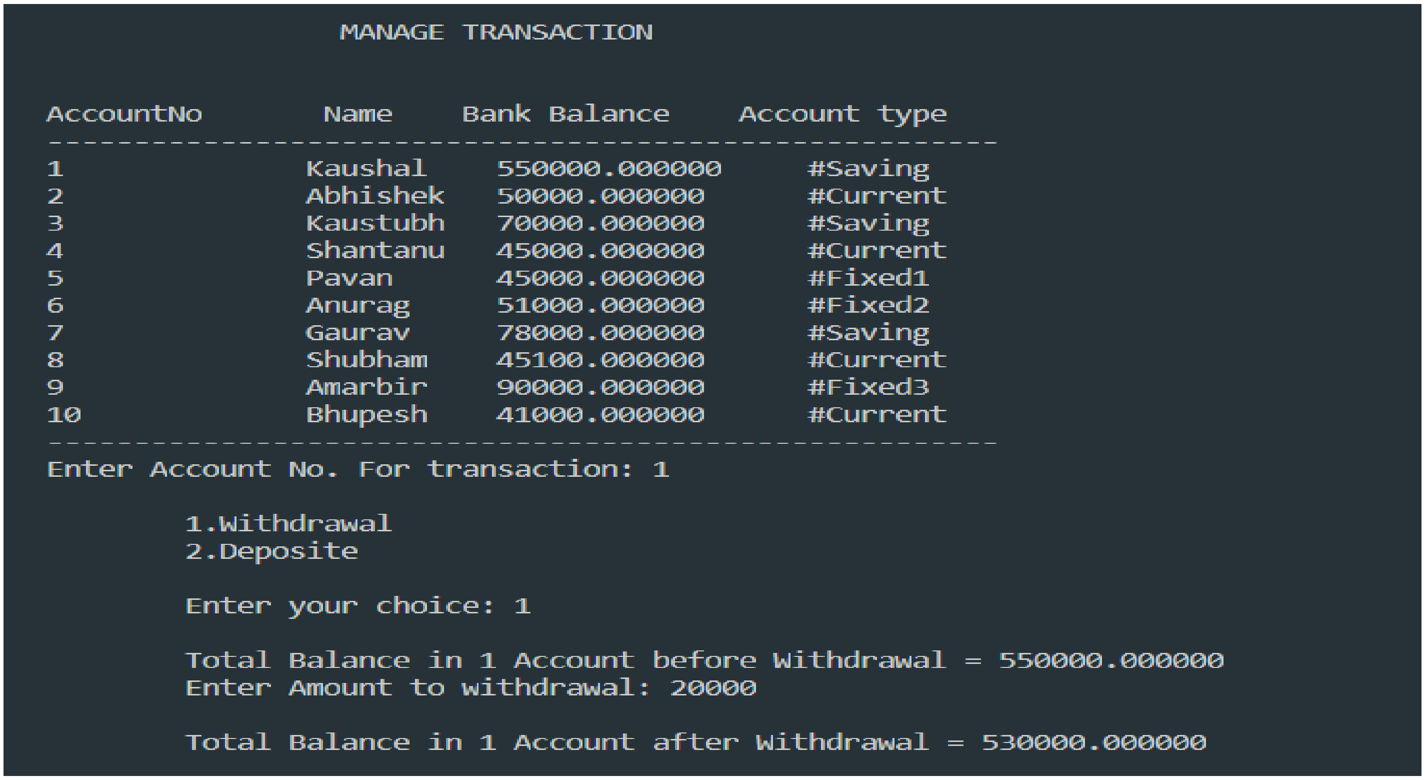
4. Check Details Of Existing Account



```
                 SEARCH RECORD
Enter Account Number Which Details you Want to display: 1

                ACCOUNT DETAILS



----------------------------------------------


Account No: 1
Name: Kaushal
Phone No: 769233805
Address: jalgaon
Created on (mm dd yyyy): 1 1 2021
DOB (mm dd yyyy): 22 4 2003
Bank Balance: 550000.000000
Account Type: #Saving
----------------------------------------------
```

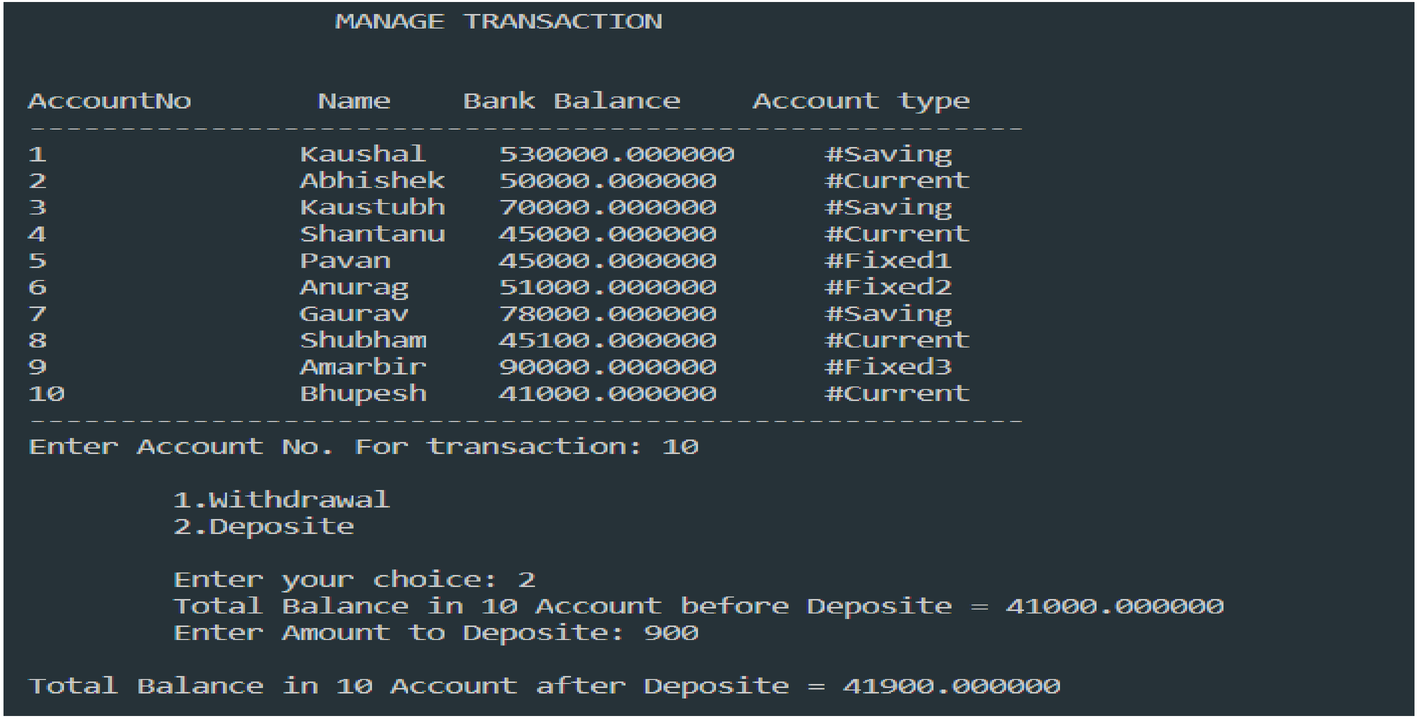
5. Manage Transaction

Withdrawal



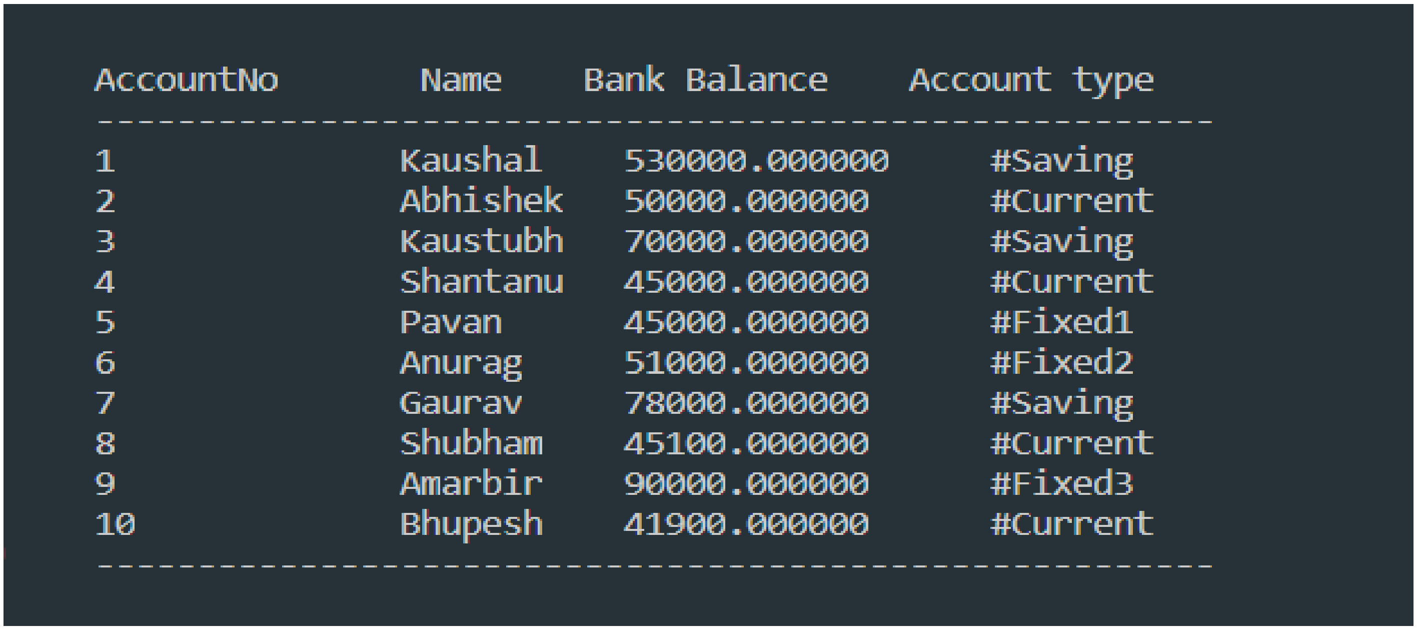
```
                   MANAGE TRANSACTION


AccountNo        Name     Bank Balance    Account type
-------------------------------------------------------
1               Kaushal    550000.000000     #Saving
2               Abhishek   50000.000000      #Current
3               Kaustubh   70000.000000      #Saving
4               Shantanu   45000.000000      #Current
5               Pavan      45000.000000      #Fixed1
6               Anurag     51000.000000      #Fixed2
7               Gaurav     78000.000000      #Saving
8               Shubham    45100.000000      #Current
9               Amarbir    90000.000000      #Fixed3
10              Bhupesh    41000.000000      #Current
-------------------------------------------------------
Enter Account No. For transaction: 1

        1.Withdrawal
        2.Deposite

        Enter your choice: 1

        Total Balance in 1 Account before Withdrawal = 550000.000000
        Enter Amount to withdrawal: 20000

        Total Balance in 1 Account after Withdrawal = 530000.000000
```

Deposite



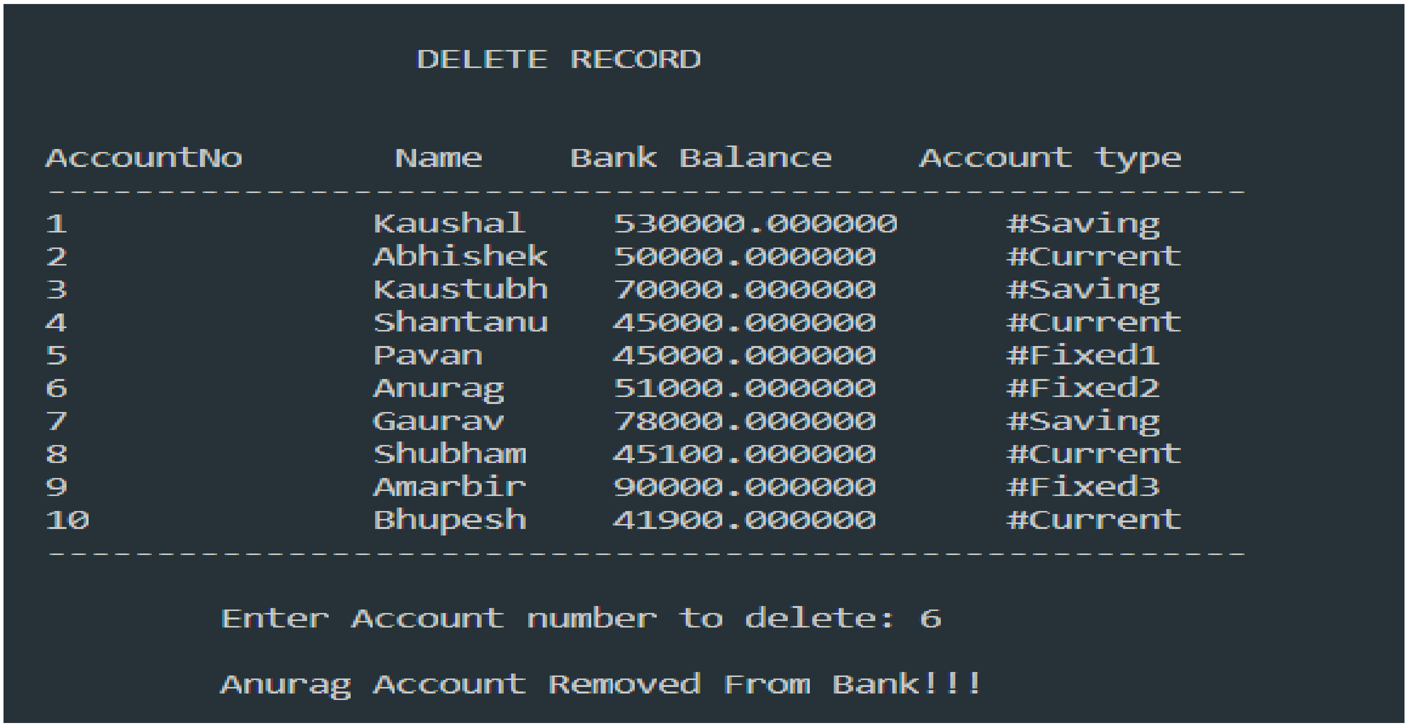
```
                  MANAGE TRANSACTION


AccountNo        Name     Bank Balance    Account type
-------------------------------------------------------
1               Kaushal    530000.000000     #Saving
2               Abhishek   50000.000000      #Current
3               Kaustubh   70000.000000      #Saving
4               Shantanu   45000.000000      #Current
5               Pavan      45000.000000      #Fixed1
6               Anurag     51000.000000      #Fixed2
7               Gaurav     78000.000000      #Saving
8               Shubham    45100.000000      #Current
9               Amarbir    90000.000000      #Fixed3
10              Bhupesh    41000.000000      #Current
-------------------------------------------------------
Enter Account No. For transaction: 10

        1.Withdrawal
        2.Deposite

        Enter your choice: 2
        Total Balance in 10 Account before Deposite = 41000.000000
        Enter Amount to Deposite: 900

Total Balance in 10 Account after Deposite = 41900.000000
```

6. Display All Record



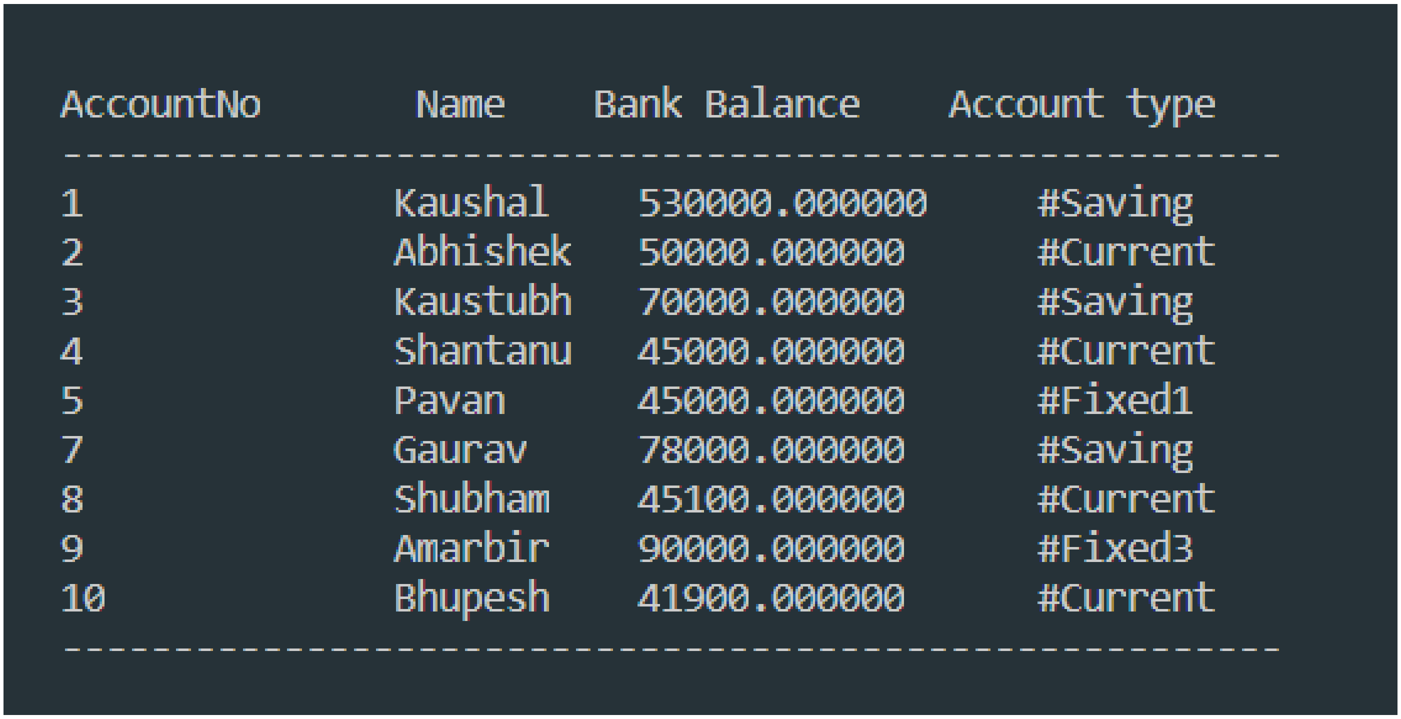
```
AccountNo        Name     Bank Balance     Account type
-------------------------------------------------------
1               Kaushal    530000.000000     #Saving
2               Abhishek   50000.000000      #Current
3               Kaustubh   70000.000000      #Saving
4               Shantanu   45000.000000      #Current
5               Pavan      45000.000000      #Fixed1
6               Anurag     51000.000000      #Fixed2
7               Gaurav     78000.000000      #Saving
8               Shubham    45100.000000      #Current
9               Amarbir    90000.000000      #Fixed3
10              Bhupesh    41900.000000      #Current
-------------------------------------------------------
```

7. Delete Account From Bank



```
                    DELETE RECORD


AccountNo        Name     Bank Balance     Account type
-------------------------------------------------------
1               Kaushal    530000.000000     #Saving
2               Abhishek   50000.000000      #Current
3               Kaustubh   70000.000000      #Saving
4               Shantanu   45000.000000      #Current
5               Pavan      45000.000000      #Fixed1
6               Anurag     51000.000000      #Fixed2
7               Gaurav     78000.000000      #Saving
8               Shubham    45100.000000      #Current
9               Amarbir    90000.000000      #Fixed3
10              Bhupesh    41900.000000      #Current
-------------------------------------------------------

        Enter Account number to delete: 6

        Anurag Account Removed From Bank!!!
```

8. Dislplay All Record At Last



```
AccountNo       Name    Bank Balance    Account type
-------------------------------------------------------
1              Kaushal    530000.000000     #Saving
2              Abhishek   50000.000000      #Current
3              Kaustubh   70000.000000      #Saving
4              Shantanu   45000.000000      #Current
5              Pavan      45000.000000      #Fixed1
7              Gaurav     78000.000000      #Saving
8              Shubham    45100.000000      #Current
9              Amarbir    90000.000000      #Fixed3
10             Bhupesh    41900.000000      #Current
-------------------------------------------------------
```

7. CONCLUSION

This course covered the basics of data structures. With this we have only scratched the surface. Although we have built a good foundation to move ahead. Data structures allow information storage, it provides the means for management of large data like databases, work together and are necessary for efficient algorithms, safe storage of data, allows easier processing of data

Data Structures is not just limited to Stack, Queues, and Linked Lists but is quite a vast area. Each data structure has its own advantages and disadvantages and must be used according to the needs of the application. We use linked list in this project it provide dyanamic approach which is efficient for memory management.

Future Scope:

Project scope is the part of project planning that involves determining a list of specific project goals, features, functions, tasks,. In other words, it is what needs to be achieved and the work that must be done to deliver a project.

1. Using File Handling
2. By Using Concept of Object Oriented Programming reliability and reusability increase.
3. By connecting project to Oracle Database.

8. REFERENCES

1. https://www.geeksforgeeks.org/time-h-header-file-in-c-with-examples/ - Use for adding time delay to the project
2. http://www.cplusplus.com/forum/windows/166491/ - use for formatted the output
3. https://www.sanfoundry.com/c-program-illustrate-user-authentication/ - use for authenticated code
4. https://www.tutorialspoint.com/c_standard_library/c_function_malloc.htm - use for dynamic memory allocation
5. https://www.upgrad.com/blog/data-structure-project-ideas-beginners/ - use for project idea