RNS Institute of Technology

(VTU Affiliated, AICTE Approved, NAAC 'A' Grade Accredited)

(UG Programs- CSE, ECE, ISE, and EIE & EEE have been accredited by NBA for the Academic Year 2018-19, 2019-20, 2020-21 & 2021-22)

DR. VISHNUVARDHAN ROAD, CHANNASANDRA, RR NAGAR POST, BENGALURU - 560098

Department of Artificial Intelligence and Data science

Principles of programming in C
BPOPS203

PROJECT REPORT

ON

BANK MANAGEMENT SYSTEM

Done by:

N.Premkumar(1RN23AD034)

M.Sree Charan(1RN23AD031)

K.V Nidhish reddy(1RN23AD029)

Contents

ACKNOWLEDGEMENT

I take this opportunity to express my Profound gratitude and deep regards to my teacher Mrs. PAVITRA for her exemplary guidance, monitoring and constant encouragement throughout the course of project. The blessing, help and guidance given by her time to time shall carry me along way in the journey of life on which I am about to embark.

I also take this opportunity to express a deep sense of gratitude to our principal Dr. RAMESH BABU H.S for his cordial support, which helped me in completing this task through various stages .I am obliged to the school staff members of RNS Institute Of Technology for the valuable assistance provided by them.

I am grateful for their cooperation during the period of my assignment .Last but not the least. I thank my parents and my friends for their constant encouragement without which this assignment would not be possible.

OBJECTIVE

To create Bank management system to record all the details of the user and also create a login portal and store all the transaction details of the user using c programming language

INTRODUCTION

Welcome to the introduction of our project report on the Bank Management System developed using the C programming language. In this report, we delve into the intricacies of designing and implementing a robust system tailored to the needs of modern banking operations.

The banking sector plays a pivotal role in the economic landscape, serving as a cornerstone for financial transactions, savings, and investments. With the advent of technology, the demand for efficient and secure banking solutions has grown exponentially. Our project aims to address these needs by offering a comprehensive software solution for managing various banking operations.

Utilizing the power and flexibility of the C programming language, we have crafted a system that encompasses a wide range of functionalities, including account management, transaction processing, customer relationship management, and administrative tasks. Our system is designed to

be user-friendly, efficient, and scalable, catering to the diverse requirements of both customers and bank personnel.

Through this project report, we provide an in-depth analysis of the system architecture, design principles, implementation details, and testing procedures. We also discuss the challenges encountered during the development process and the strategies employed to overcome them.

We hope that this project report serves as a valuable resource for understanding the intricacies of bank management systems and inspires further exploration and innovation in this field.

SOURCE CODE

```
// C program to implement
// the above approach
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <windows.h>
// Declaring all the functions
void checkbalance(char*);
void transfermoney(void);
void display(char*);
void person(char*);
void login(void);
void loginsu(void);
void account(void);
void accountcreated(void);
void afterlogin(void);
void logout(void);
// Declaring gotoxy
// function for setting
```

```
// cursor position
void gotoxy(int x, int y)
{
                    COORD c;
                    c.X = x;
                    c.Y = y;

                    SetConsoleCursorPosition(
                        GetStdHandle(STD_OUTPUT_HANDLE), c);
}
// Creating  a structure to store
// data of the user
struct pass {
                    char username[50];
                    int date, month, year;
                    char pnumber[15];
                    char adharnum[20];
                    char fname[20];
                    char lname[20];
                    char fathname[20];
                    char mothname[20];
                    char address[50];
```

```
                        char typeaccount[20];
};
// Structure to keep track
// of amount transfer
struct money {
                        char usernameto[50];
                        char userpersonfrom[50];
                        long int money1;
};
struct userpass {
                        char password[50];
};

// Driver Code
int main()
{
                        int i, a, b, choice;
                        int passwordlength;

                        gotoxy(20, 3);

                        // Creating a Main
```

```
                    // menu for the user
                    printf("WELCOME TO BANK ACCOUNT
SYSTEM\n\n");
                    gotoxy(18, 5);

                    printf("**********************************
*");
                    gotoxy(25, 7);

                    printf("DEVELOPER - Prem kumar");

                    gotoxy(20, 10);
                    printf("1.... CREATE A BANK ACCOUNT");

                    gotoxy(20, 12);
                    printf("2.... ALREADY A USER? SIGN IN");
                    gotoxy(20, 14);
                    printf("3.... EXIT\n\n");

                    printf("\n\nENTER YOUR CHOICE..");

                    scanf("%d", &choice);
```

```
                    switch (choice) {
                    case 1:
                        system("cls");
                        printf("\n\n USERNAME 50 CHARACTERS
MAX!!");
                        printf("\n\n PASSWORD 50 CHARACTERS
MAX!!");
                        account();
                        break;

                    case 2:
                        login();
                        break;

                    case 3:
                        exit(0);
                        break;

                        getch();
                    }
}

// Function to create accounts
```

```
// of users
void account(void)
{
                        char password[20];
                        int passwordlength, i, seek = 0;
                        char ch;
                        FILE *fp, *fu;
                        struct pass u1;
                        struct userpass p1;

                        struct userpass u2;

                        // Opening file to
                        // write data of a user
                        fp = fopen("username.txt", "ab");

                        // Inputs
                        system("cls");
                        printf("\n\n!!!!!CREATE ACCOUNT!!!!!");

                        printf("\n\nFIRST NAME..");
                        scanf("%s", &u1.fname);
```

```
printf("\n\n\nLAST NAME..");
scanf("%s", &u1.lname);

printf("\n\nFATHER's NAME..");
scanf("%s", &u1.fathname);

printf("\n\nMOTHER's NAME..");
scanf("%s", &u1.mothname);

printf("\n\nADDRESS..");
scanf("%s", &u1.address);

printf("\n\nACCOUNT TYPE");
scanf("%s", &u1.typeaccount);

printf("\n\nDATE OF BIRTH..");
printf("\nDATE-");
scanf("%d", &u1.date);
printf("\nMONTH-");
scanf("%d", &u1.month);
printf("\nYEAR-");
```

```
scanf("%d", &u1.year);

printf("\n\nADHAR NUMBER");
scanf("%s", u1.adharnum);

printf("\n\nPHONE NUMBER");
scanf("%s", u1.pnumber);

printf("\n\nUSERNAME.. ");
scanf("%s", &u1.username);

printf("\n\nPASSWORD..");

// Taking password in the form of
// stars
for (i = 0; i < 50; i++) {
        ch = getch();
        if (ch != 13) {
                password[i] = ch;
                ch = '*';
                printf("%c", ch);
        }
```

```
                else
                        break;
                }

                // Writing to the file
                fwrite(&u1, sizeof(u1),
                1, fp);

                // Closing file
                fclose(fp);

                // Calling another function
                // after successful creation
                // of account
                accountcreated();
}

// Successful account creation
void accountcreated(void)
{
                int i;
                char ch;
```

```
                    system("cls");
                    printf(
                        "PLEASE WAIT....\n\nYOUR DATA IS
PROCESSING....");
                    for (i = 0; i < 200000000; i++) {
                        i++;
                        i--;
                    }

                    gotoxy(30, 10);

                    printf("ACCOUNT CREATED SUCCESSFULLY....");
                    gotoxy(0, 20);

                    printf("Press enter to login");

                    getch();
                    login();
}

// Login function to check
// the username of the user
```

```
void login(void)
{
                        system("cls");

                        char username[50];
                        char password[50];

                        int i, j, k;
                        char ch;
                        FILE *fp, *fu;
                        struct pass u1;
                        struct userpass u2;

                        // Opening file of
                        // user data
                        fp = fopen("username.txt",
                                   "rb");

                        if (fp == NULL) {
                              printf("ERROR IN OPENING FILE");
                        }
                        gotoxy(34, 2);
```

```
                printf(" ACCOUNT LOGIN ");
                gotoxy(7, 5);
                printf("**********************************
**********"
                    "**********************************");

                gotoxy(35, 10);
                printf("==== LOG IN ====");

                // Take input
                gotoxy(35, 12);
                printf("USERNAME.. ");
                scanf("%s", &username);

                gotoxy(35, 14);
                printf("PASSWORD..");

                // Input the password
                for (i = 0; i < 50; i++) {
                    ch = getch();
                    if (ch != 13) {
                        password[i] = ch;
```

```
            ch = '*';
            printf("%c", ch);
        }

        else
            break;
    }

    // Checking if username
    // exists in the file or not
    while (fread(&u1, sizeof(u1),
                    1, fp)) {
        if (strcmp(username,
                    u1.username)
            == 0) {
            loginsu();
            display(username);
        }
    }

    // Closing the file
    fclose(fp);
```

```
}

// Redirect after
// successful login
void loginsu(void)
{
                        int i;
                        FILE* fp;
                        struct pass u1;
                        system("cls");
                        printf("Fetching account details.....\n");
                        for (i = 0; i < 20000; i++) {
                                i++;
                                i--;
                        }

                        gotoxy(30, 10);
                        printf("LOGIN SUCCESSFUL....");
                        gotoxy(0, 20);
                        printf("Press enter to continue");

                        getch();
```

```
}

// Display function to show the
// data of the user on screen
void display(char username1[])
{
                        system("cls");
                        FILE* fp;
                        int choice, i;
                        fp = fopen("username.txt", "rb");
                        struct pass u1;

                        if (fp == NULL) {
                                printf("error in opening file");
                        }

                        while (fread(&u1, sizeof(u1),
                                                1, fp)) {
                                if (strcmp(username1,
                                                u1.username)
                                        == 0) {
                                        gotoxy(30, 1);
```

```
                printf("WELCOME, %s %s",
                        u1.fname, u1.lname);
                gotoxy(28, 2);
                printf(".....");
                gotoxy(55, 6);
                printf("==== YOUR ACCOUNT INFO
====");
                gotoxy(55, 8);

            printf("**************************");
                gotoxy(55, 10);
                printf("NAME..%s %s", u1.fname,
                        u1.lname);

                gotoxy(55, 12);
                printf("FATHER's NAME..%s %s",
                        u1.fathname,
                        u1.lname);

                gotoxy(55, 14);
                printf("MOTHER's NAME..%s",
                        u1.mothname);
```

```
                gotoxy(55, 16);
                printf("ADHAR CARD NUMBER..%s",
                        u1.adharnum);

                gotoxy(55, 18);
                printf("MOBILE NUMBER..%s",
                        u1.pnumber);

                gotoxy(55, 20);
                printf("DATE OF BIRTH.. %d-%d-%d",
                        u1.date, u1.month, u1.year);

                gotoxy(55, 22);
                printf("ADDRESS..%s", u1.address);

                gotoxy(55, 24);
                printf("ACCOUNT TYPE..%s",
                        u1.typeaccount);
        }
}

fclose(fp);
```

```
gotoxy(0, 6);

// Menu to perform different
// actions by user
printf(" HOME ");
gotoxy(0, 7);
printf("******");
gotoxy(0, 9);
printf(" 1....CHECK BALANCE");
gotoxy(0, 11);
printf(" 2....TRANSFER MONEY");
gotoxy(0, 13);
printf(" 3....LOG OUT\n\n");
gotoxy(0, 15);
printf(" 4....EXIT\n\n");

printf(" ENTER YOUR CHOICES..");
scanf("%d", &choice);

switch (choice) {
case 1:
```

```
                checkbalance(username1);
                break;

            case 2:
                transfermoney();
                break;

            case 3:
                logout();
                login();
                break;

            case 4:
                exit(0);
                break;
            }
}

// Function to transfer
// money from one user to
// another
void transfermoney(void)
```

```
{
        int i, j;
        FILE *fm, *fp;
        struct pass u1;
        struct money m1;
        char usernamet[20];
        char usernamep[20];
        system("cls");

        // Opening file in read mode to
        // read user's username
        fp = fopen("username.txt", "rb");

        // Creating a another file
        // to write amount along with
        // username to which amount
        // is going to be transferred
        fm = fopen("mon.txt", "ab");

        gotoxy(33, 4);
        printf("---- TRANSFER MONEY ----");
        gotoxy(33, 5);
```

```
printf("========================");

gotoxy(33, 11);
printf("FROM (your username).. ");
scanf("%s", &usernamet);

gotoxy(33, 13);
printf(" TO (username of person)..");
scanf("%s", &usernamep);

// Checking for username if it
// is present in file or not
while (fread(&u1, sizeof(u1),
                    1, fp))

{
      if (strcmp(usernamep,
                    u1.username)
            == 0) {
            strcpy(m1.usernameto,
                    u1.username);
            strcpy(m1.userpersonfrom,
```

```
                        usernamet);
            }
        }
        gotoxy(33, 16);

        // Taking amount input
        printf("ENTER THE AMOUNT TO BE
TRANSFERRED..");
        scanf("%d", &m1.money1);

        // Writing to the file
        fwrite(&m1, sizeof(m1),
            1, fm);

        gotoxy(0, 26);
        printf(
            "-----------------------------------------------"
            "-----------------------------------------");

        gotoxy(0, 28);
        printf(
            "-----------------------------------------------"
```

```
                "-----------------------------------------");

                gotoxy(0, 29);
                printf("transferring amount, Please wait..");

                gotoxy(10, 27);
                for (i = 0; i < 70; i++) {
                        for (j = 0; j < 1200000; j++) {
                                j++;
                                j--;
                        }
                        printf("*");
                }

                gotoxy(33, 40);
                printf("AMOUNT SUCCESSFULLY
TRANSFERRED....");
                getch();

                // Close the files
                fclose(fp);
                fclose(fm);
```

```
                        // Function to return
                        // to the home screen
                        display(usernamet);
}

// Function to check balance
// in users account
void checkbalance(char username2[])
{
                        system("cls");
                        FILE* fm;
                        struct money m1;
                        char ch;
                        int i = 1, summoney = 0;

                        // Opening amount file record
                        fm = fopen("mon.txt", "rb");

                        int k = 5, l = 10;
                        int m = 30, n = 10;
                        int u = 60, v = 10;
```

```
gotoxy(30, 2);
printf("==== BALANCE DASHBOARD ====");
gotoxy(30, 3);
printf("****************************");
gotoxy(k, l);
printf("S no.");
gotoxy(m, n);
printf("TRANSACTION ID");
gotoxy(u, v);
printf("AMOUNT");

// Reading username to
// fetch the correct record
while (fread(&m1, sizeof(m1),
                    1, fm)) {
    if (strcmp(username2,
                    m1.usernameto)
            == 0) {
            gotoxy(k, ++l);
            printf("%d", i);
            i++;
```

```
                        gotoxy(m, ++n);
                        printf("%s", m1.userpersonfrom);

                        gotoxy(u, ++v);
                        printf("%d", m1.money1);
                        // Adding and
                        // finding total money
                        summoney = summoney +
m1.money1;
                    }
                }

                gotoxy(80, 10);
                printf("TOTAL AMOUNT");

                gotoxy(80, 12);
                printf("%d", summoney);

                getch();

                // Closing file after
                // reading it
```

```
                fclose(fm);
                display(username2);
}

// Logout function to bring
// user to the login screen
void logout(void)
{
                int i, j;
                system("cls");
                printf("please wait, logging out");

                for (i = 0; i < 10; i++) {
                        for (j = 0; j < 25000000; j++) {
                                i++;
                                i--;
                        }
                        printf(".");
                }

                gotoxy(30, 10);
                printf("Sign out successfully..\n");
```

```
                gotoxy(0, 20);
                printf("press any key to continue..");

                getch();
}
```

Output

Conclusion :

In conclusion, the development and analysis of our Bank Management System project using the C programming language have been both challenging and rewarding. Throughout this project, we have demonstrated the power and versatility of C in crafting efficient and robust software solutions tailored to the complex needs of the banking sector.

Our Bank Management System stands as a testament to the importance of technology in modernizing and optimizing banking operations. By automating routine tasks, streamlining transaction processing, and enhancing data security, our system offers tangible benefits to both banks and their customers.

Through meticulous design, implementation, and testing, we have ensured that our system meets the highest standards of reliability, scalability, and usability. From account management to transaction

processing to administrative tasks, every aspect of the banking process has been carefully considered and addressed in our system.

Looking ahead, there is immense potential for further enhancements and refinements to our Bank Management System. With ongoing advancements in technology and evolving customer expectations, there is a constant need to innovate and adapt to changing market dynamics.

As we conclude this project report, we reflect on the journey we have undertaken and the knowledge we have gained along the way. We are proud of what we have accomplished, and we are confident that our Bank Management System will serve as a valuable asset to the banking industry for years to come.

Bibliography :

www.geeksforgeeks.com

https://www.studytonight.com/c-projects/bank-management-system-project-using-c-language

https://www.geeksforgeeks.org/bank-account-system-in-c-using-file-handling/