

A Micro Project Report

on

Problem Solving using C Language

Submitted by
VENUTHURLA GARUDA SEKHAR REDDY
(24475A0523)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE: NARASARAOPET
(AUTONOMOUS)

Accredited by NAAC with A+ Grade and NBA under Tier-1

**NIRF rank in the band of 201-300 and is an ISO 9001:2015 certified Approved by
AICTE, New Delhi, Permanently affiliated to JNTU Kakinada, Approved by AICTE,
Accredited by NBA and accredited 'A+' grade by NAAC Narasaraopet-522601,
Palnadu(Dt.), Andhra Pradesh, India**

2024-2025

NARASARAOPETA ENGINEERING COLLEGE: NARASARAOPET
(AUTONOMOUS)
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that **VENUTHURLA GARUDA SEKHAR REDDY** , **Roll No: 24475A0523**, a Second Year Student of the Department of Computer Science and Engineering, has completed the Micro Project Satisfactorily in “Problem Solving using C Language” for the Academic Year 2024-2025..

Project Co-Ordinator

Dr. Rama Krishna. E, M.Tech., Ph.D.
Asst. Professor

HEAD OF THE DEPARTMENT

Dr. S. N. Tirumala Rao, M.Tech., Ph.D.
Professor

BANKING SYSTEM

S.No	Description
1.	1. Banking System - Implement account creation, transactions, and balance inquiry with file storage

AIM:

1. Banking System - Implement account creation, transactions, and balance inquiry with file storage

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX_NAME_LENGTH 100
#define FILENAME "accounts.dat"
// Define the Account structure
typedef struct
{
    int accountNumber;
    char name[MAX_NAME_LENGTH];
    float balance;
}
Account;
// Function to create a new account
void createAccount()
{
    FILE *file = fopen(FILENAME, "ab"); // Open the file in append mode

    if (file == NULL)
    {
        printf("Error opening file!\n");
        return;
    }

    Account newAccount;
    printf("Enter account number: ");
    scanf("%d", &newAccount.accountNumber);
    getchar(); // Consume the newline character left by scanf

    printf("Enter account holder's name: ");
    fgets(newAccount.name, MAX_NAME_LENGTH, stdin);
```

```

    newAccount.name[strcspn(newAccount.name, "\n")] = '\0'; //
Remove trailing newline

printf("Enter initial deposit amount: ");

    scanf("%f", &newAccount.balance);

    fwrite(&newAccount, sizeof(Account), 1, file); // Write the account
to the file
    fclose(file);

    printf("Account created successfully!\n");
}

// Function to find an account by account number
Account* findAccount(int accountNumber)
{
    FILE *file = fopen(FILENAME, "rb"); // Open the file in read-binary
mode
    if (file == NULL)
    {
        printf("Error opening file!\n");
        return NULL;
    }

    Account *account = malloc(sizeof(Account));
    while (fread(account, sizeof(Account), 1, file))
    {
        if (account->accountNumber == accountNumber)
        {
            fclose(file);
            return account;
        }
    }

    fclose(file);
    free(account);
    return NULL; // Account not found
}

```

```

// Function to deposit money
void depositMoney(int accountNumber)
{
    Account *account = findAccount(accountNumber);
    if (account == NULL)
    {
        printf("Account not found!\n");
        return;
    }

    float depositAmount;
    printf("Enter deposit amount: ");
    scanf("%f", &depositAmount);

    account->balance += depositAmount;

    // Update the file with the new balance
    FILE *file = fopen(FILENAME, "rb+");
    if (file == NULL)
    {
        printf("Error opening file!\n");
        free(account);
        return;
    }

    Account tempAccount;
    while (fread(&tempAccount, sizeof(Account), 1, file))
    {
        if (tempAccount.accountNumber == account->accountNumber)
        {
            fseek(file, -sizeof(Account), SEEK_CUR); // Move to the position
            of the account
            fwrite(account, sizeof(Account), 1, file); // Update the account
            info
            break;
        }
    }
}

```

```

    }

    fclose(file);
    printf("Deposit successful! New balance: %.2f\n", account->balance);
    free(account);
}

// Function to withdraw money
void withdrawMoney(int accountNumber)
{
    Account *account = findAccount(accountNumber);
    if (account == NULL) {
        printf("Account not found!\n");
        return;
    }

    float withdrawAmount;
    printf("Enter withdrawal amount: ");
    scanf("%f", &withdrawAmount);

    if (withdrawAmount > account->balance)
    {
        printf("Insufficient funds!\n");
        free(account);
        return;
    }

    account->balance -= withdrawAmount;

    // Update the file with the new balance
    FILE *file = fopen(FILENAME, "rb+");
    if (file == NULL) {
        printf("Error opening file!\n");
        free(account);
        return;
    }

```

```

    Account tempAccount;
    while (fread(&tempAccount, sizeof(Account), 1, file))
    {
        if (tempAccount.accountNumber == account->accountNumber)
        {
            fseek(file, -sizeof(Account), SEEK_CUR); // Move to the position
            of the account
            fwrite(account, sizeof(Account), 1, file); // Update the account
            info
            break;
        }
    }

    fclose(file);
    printf("Withdrawal successful! New balance: %.2f\n", account-
    >balance);
    free(account);
}

// Function to check account balance
void checkBalance(int accountNumber)
{
    Account *account = findAccount(accountNumber);
    if (account == NULL) {
        printf("Account not found!\n");
        return;
    }

    printf("Account balance for account number %d: %.2f\n",
    accountNumber, account->balance);
    free(account);
}

// Main menu to interact with the system
void menu()
{
    int choice, accountNumber;

```



```
do
{
    printf("\n----- Bank System ----- \n");
    printf("1. Create Account\n");
    printf("2. Deposit Money\n");
    printf("3. Withdraw Money\n");
    printf("4. Check Balance\n");
    printf("5. Exit\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);

    switch (choice)
    {
        case 1:
            createAccount();
            break;
        case 2:
            printf("Enter account number: ");
            scanf("%d", &accountNumber);
            depositMoney(accountNumber);
            break;
        case 3:
            printf("Enter account number: ");
            scanf("%d", &accountNumber);
            withdrawMoney(accountNumber);
            break;
        case 4:
            printf("Enter account number: ");
            scanf("%d", &accountNumber);
            checkBalance(accountNumber);
            break;
        case 5:
            printf("Exiting...\n");
            break;
        default:
            printf("Invalid choice! Please try again.\n");
    }
}
```

```
        } while (choice != 5);  
    }  
  
    int main()  
    {  
        menu();  
        return 0;  
    }
```

Create Account:

INPUT:

Enter your choice: 1

Enter account number: 1001

Enter account holder's name: John Doe

Enter initial deposit amount: 500.00

OUTPUT:

Account created successfully!

DEPOSIT MONEY:

INPUT:

Enter your choice: 2

Enter account number: 1001

Enter deposit amount: 200.00

OUTPUT:

Deposit successful! New balance: 700.00

WITHDRAWAL MONEY:

INPUT:

Enter your choice: 3

Enter account number: 1001

Enter withdrawal amount: 100.00

OUTPUT:

Withdrawal successful! New balance: 600.00

CHECK BALANCE:

INPUT:

Enter your choice: 4

Enter account number: 1001

OUTPUT:

Account balance for account number 1001: 600.00

EXIT:

INPUT:

Enter your choice: 5

OUTPUT:

Exiting...