FINAL REPORT

COMPUTER PROGRAMMING (CSE101)

TITLE

Online Banking System

Group Members

SR. NO.	NAME	ROLL NO.	REG. NO.
1	KOUSIK MAITY	49	12208142
2	JAYANT KUMAR	50	12208932
3	VIDHI SHARMA	51	12209174
4	ADITYA KUMAR	52	12209197

SUBMITTED BY – KOUSIK MAITY (12208142)

SUBMITTED TO – RICHA JAIN

Acknowledgement

I would like to express my gratitude to all those who have contributed to the successful completion of my CSE101 Computer Programming final project report.

First and foremost, I would like to thank my instructor **Richa Jain Mam**, for providing me with the knowledge and guidance needed to complete this project. Their unwavering support and constructive feedback have been invaluable throughout the process.

I am also grateful to my classmates, for their continuous encouragement, insightful discussions, and collaboration during the project.

Furthermore, I would like to extend my appreciation to the resources provided by the University, such as the computer lab facilities, online libraries, and tutorials, which have enabled me to gain a deeper understanding of the programming concepts and tools.

Lastly, I would like to express my heartfelt thanks to my family and friends for their love, encouragement, and motivation throughout my academic journey. Their unwavering support has been instrumental in helping me achieve my goals.

PROJECT DESCRIPTION

The Online Banking System is a web-based application designed to provide a convenient and efficient way for bank employees and customers to perform banking transactions online. The system will have two panels: one for the admin and one for the user.

The admin panel will allow bank employees to manage customer accounts and perform other administrative tasks. For example, bank employees will be able to view customer account information, update customer information, and perform other tasks related to account management.

The user panel will allow customers to perform various banking transactions, such as checking their account status, opening new accounts, changing their password, and managing payees. Customers will be able to view their current account balance and recent transactions, as well as transfer money between accounts and pay bills online.

In addition to these basic features, the Online Banking System may also include advanced features such as mobile banking, alerts and notifications, and budgeting tools. These features will provide customers with even more convenience and control over their finances.

Overall, the Online Banking System will automate the various activities and functions of a bank through the internet, providing a convenient and efficient way for bank employees and customers to perform banking transactions online.

MODULE EXPLANATION

Account Status: This module will allow customers to view their current account balance and recent transactions. Customers will be able to see a summary of their account information, including their current balance, available funds, and recent transactions. They will also be able to view detailed information about each transaction, such as the date, amount, and description. This module will provide customers with an easy way to keep track of their finances and monitor their account activity.

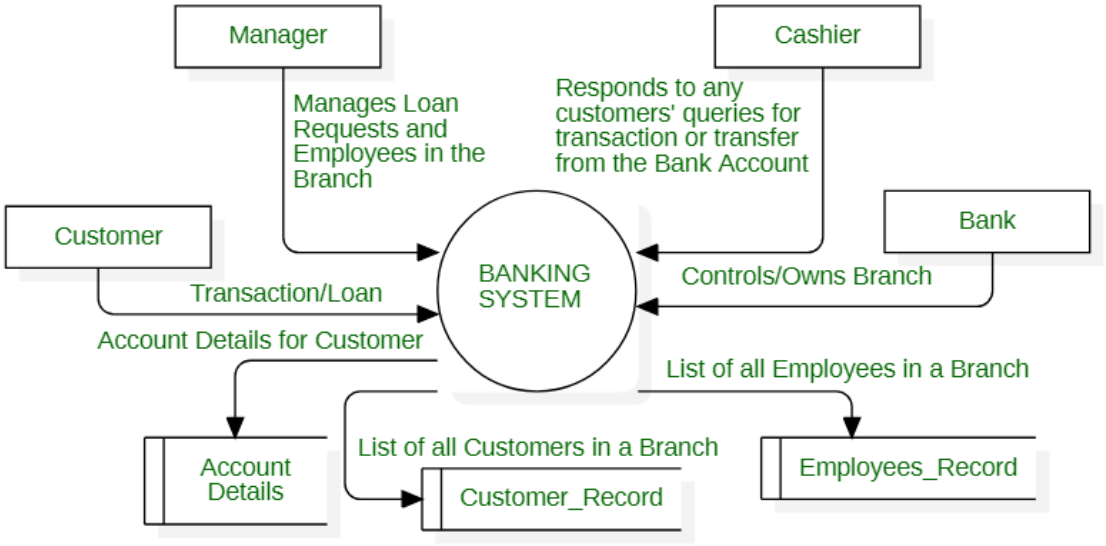
Account Opening: This module will allow customers to open new accounts online. Customers will be able to fill out an online application form and submit it for approval. The application form will include fields for personal information, such as the customer's name, address, and social security number, as well as financial information, such as their income and employment status. Once the application is approved, the customer will be able to access their new account and start using it for banking transactions.

Change Password: This module will allow customers to change their password for added security. Customers will be able to enter their current password and then choose a new password. The system will enforce password complexity requirements to ensure that the new password is strong and secure. This module will help ensure that customers' accounts are secure and protected from unauthorized access.

Payee Management: This module will allow customers to manage their payees for easy and convenient money transfers. Customers will be able to add, edit, and delete payees, as well as view a list of their current payees. When adding a new payee, the customer will need to provide the payee's name and account information. The system will verify the payee's information before adding them to the customer's list of payees. This module will make it easy for customers to transfer money to other people or businesses.

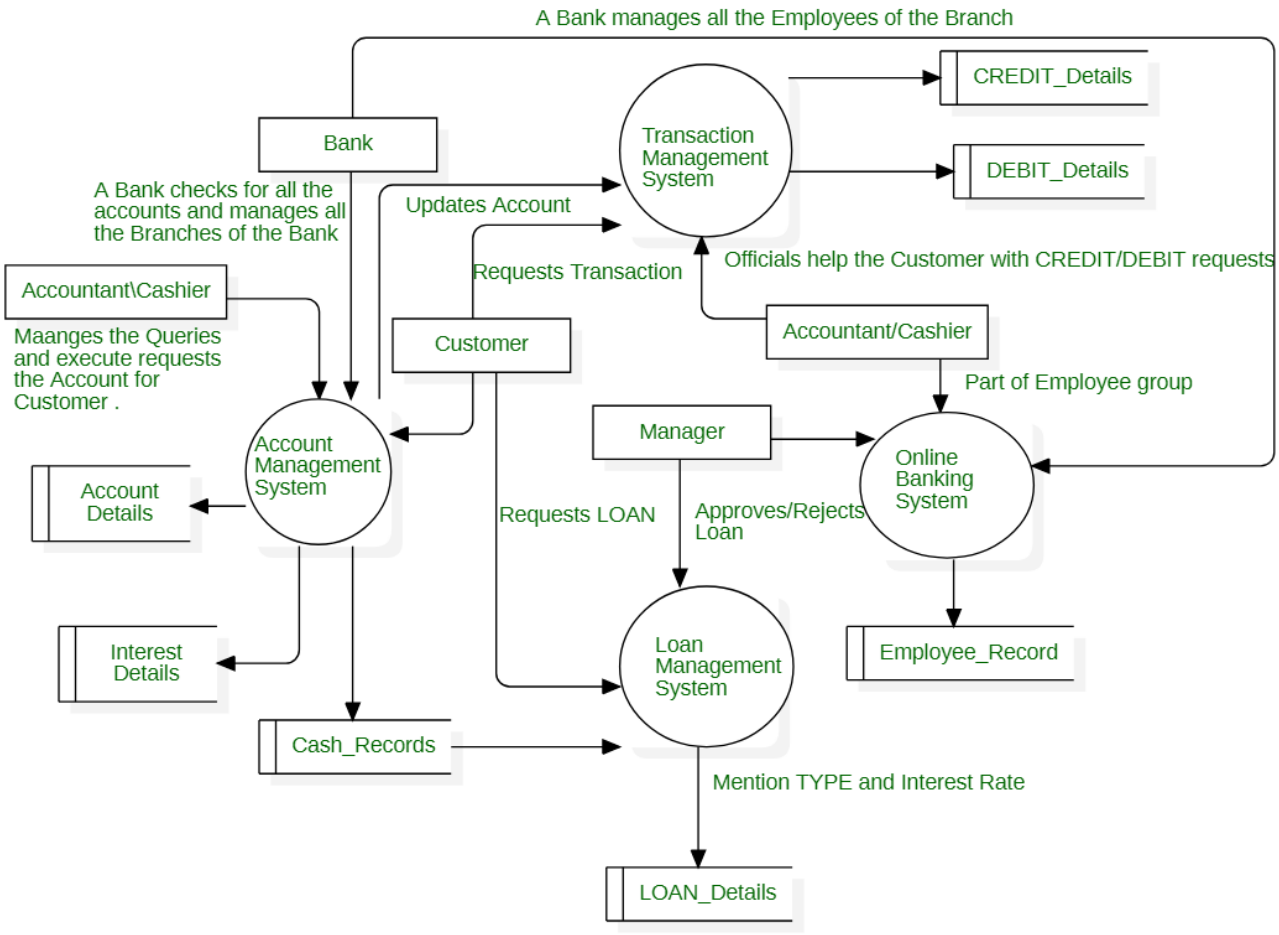
View Statement: This module will allow customers to view and download their account statements. Customers will be able to select a date range and then view a statement of all transactions that occurred during that time period. The statement will include detailed information about each transaction, such as the date, amount, and description. Customers will also be able to download their statements in various formats, such as PDF or CSV, for easy record-keeping.

DFD (LEVEL 0)



LEVEL - 0

DFD (LEVEL 1)



LEVEL-1

Programming Code Snapshots

ProjectFile.c

```c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include <time.h>
#include<conio.h>
// C PROJECT ONLINE BANKING SYSTEM.....TEACHER NAME - RESPECTED RICHA JAIN MAM
// flow of the code -- function definitions, structure initialisation, function calling
//panel selection
void panel_selection();
//admin panel
void admin_panel();
void account_opening();              //MODULE 1
void remove_account();
void display_all_accounts();
//user panel
void user_panel();
void account_status();            //MODULE 2
void change_password();          //MODULE 3
void transactions();            //MODULE 4
void make_deposit();
void make_withdrawal();
void view_statement();        //MODULE 5
// exit and mainmenu
void mainmenu();
void getout();
//initialising the structures
struct account
{
    int acc_number;
    char acc_name[50];
    float balance;
};

struct transaction
{
    int trans_number;
```

ProjectFile.c

```c
    int trans_number;
    char trans_type[50];
    float amount;
};

struct payee
{
    char name[50];
    int acc_number;
};

//PANEL SELECTION FUNCTION
void panel_selection() {
    int panel_choice;
    printf("\t\t\t\t\t  [WELCOME TO ONLINE BANKING SYSTEM]\n\n\n\n\n");
    printf("\t\t\t\t\t.........Choose any panel.........\n\n");
    printf("\t\t\t\t\t\t [1]. User Panel\n");
    printf("\t\t\t\t\t\t [2]. Admin Panel\n");
    printf("\t\t\t\t\t\t [3]. Exit\n");
    printf("\t\t\t\t\t\tEnter your choice: ");
    scanf("%d", &panel_choice);

    switch (panel_choice) {
        case 1:
            user_panel();
            break;
        case 2:
            admin_panel();
            break;
        case 3:
            getout();  // exiting from the program
            break;
        default:
            printf("Invalid choice. Please try again.\n");
            printf("Proceeding to the mainmenu....\n");
            mainmenu();
```

ProjectFile.c

```c
            mainmenu();
            break;
    }
}
// EXITING FUNCTION
void getout(){
    system("cls");
    printf("\t\t  **...THANK YOU RICHA MAM FOR GIVING SO MUCH OF PRECIOUS KNOWLEDGE IN THIS 6 MONTHS...**\n\n\n");
    printf("\t\t\t\t\t ...THANK YOU FOR USING THE SYSTEM...\n\n\n");
    printf("\t\t\t\t\t  PROJECT MADE BY\n\n\n");
    printf("\t\t\t\tKOUSIK MAITY, JAYANT KUMAR, VIDHI SHARMA, ADITYA KUMAR\n\n\n\n\n");
    exit(0);
}
// MAINMENU FUNCTION
void mainmenu(){
        system("cls");   // clearing screen
        panel_selection();}

//ADMIN PANEL FUNCTION CALL
void admin_panel()
{
    int choice;
    char username[20];
    char password[20];
    const char admin_username[] = "admin";
    const char admin_password[] = "admin";
    printf("Enter Your Username: ");
    scanf("%s",username);
    printf("Enter Your Password: ");
    scanf("%s",password);
    system("cls");
    if(strcmp(username,admin_username) == 0 && strcmp(password,admin_password) == 0){
    printf("\t\t\t\t\t\t [WELCOME TO ADMIN PANEL]\n\n\n\n\n");
    printf("\t\t\t\t\t .........Below the Options are.........\n\n");
    printf("\t\t\t\t\t\t [1]. Add Account\n");
    printf("\t\t\t\t\t\t [2]. Remove Account\n");
```

ProjectFile.c

```c
    printf("\t\t\t\t\t\t [3]. Display All Accounts\n");
    printf("\t\t\t\t\t\t [4]. Go to Mainmenu\n");
    printf("\t\t\t\t\t\t [5]. Exit the Program\n");
    printf("\t\t\t\t\t\tEnter your Option: ");
    scanf("%d", &choice);
    switch (choice)
    {
    case 1:
    {
        int c;
        while ((c = getchar()) != '\n' && c != EOF);
        system("cls");
        printf("\t\t\t\t\t\t  [ADDING ACCOUNT]\n\n\n\n\n");
        account_opening();
        printf("Proceeding to Admin Panel...\n\n");
        admin_panel();
        break;
    }
    case 2:
    {
        system("cls");
        printf("\t\t\t\t\t\t  [REMOVING ACCOUNT]\n\n\n\n\n");
        display_all_accounts();
        remove_account();
        break;
    }
    case 3:
    {
        system("cls");
        printf("\t\t\t\t\t\t  [DISPLAYING ALL ACCOUNTS]\n\n\n\n\n");
        display_all_accounts();
        printf("Proceeding to the admin panel\n\n");
        admin_panel();
        break;
    }
    case 4:
```

ProjectFile.c

```c
    case 4:
    {
        mainmenu();
        break;
    }
    case 5:
    {
        printf("Exiting program...\n");
        getout();
        break;
    }
    default:
        printf("Invalid choice.\n");
        printf("Proceeding to the Admin Panel...\n");
        admin_panel();
        break;
    }
}
    else
    {
        printf("Wrong username or password\n");
        printf("Try Again..\n\n");
        admin_panel();
    }
}
//ADMIN PANEL FUNCTIONS

// ACCOUNT OPENING (for both user and admin)
void account_opening() {
    char name[50];
    char address[100];
    char phone[20];
    char account_number[20];
    char password[20];
    float initial_deposit;
    FILE *fp;
```

ProjectFile.c

```c
    FILE *fp;
    printf("Enter your name: ");
    fgets(name, sizeof(name), stdin);
    name[strcspn(name, "\n")] = 0;
    printf("Enter your address: ");
    fgets(address, sizeof(address), stdin);
    address[strcspn(address, "\n")] = 0;
    printf("Enter your phone number: ");
    fgets(phone, sizeof(phone), stdin);
    phone[strcspn(phone, "\n")] = 0;
    printf("Enter your account number: ");
    fgets(account_number, sizeof(account_number), stdin);
    account_number[strcspn(account_number, "\n")] = 0;
    printf("Enter your password: ");
    fgets(password, sizeof(password), stdin);
    password[strcspn(password, "\n")] = 0;
    printf("\nEnter initial deposit amount: ");
    scanf("%f", &initial_deposit);
    fp = fopen("accounts.txt", "a");
    if (fp == NULL) {
        printf("Error opening file.\n");
        return;
    }
    fprintf(fp, "%s,%s,%s,%s,%s,%f\n", name, address, phone, account_number,password, initial_deposit);
    fclose(fp);
    printf("\nAccount created successfully.\n");
}

// REMOVE ACCOUNT
void remove_account() {
    char account_number[20];
    FILE *fp;
    FILE *temp_fp;
    char line[256];
    int found = 0;
    printf("Enter the account number to remove: ");
```

ProjectFile.c

```c
    printf("Enter the account number to remove: ");
    scanf("%s", account_number);
    fp = fopen("accounts.txt", "r");
    if (fp == NULL) {
        printf("Error opening file.\n");
        return;
    }
    temp_fp = fopen("temp.txt", "w");
    if (temp_fp == NULL) {
        printf("Error opening file.\n");
        fclose(fp);
        return;
    }
    while (fgets(line, sizeof(line), fp)) {
        if (strstr(line, account_number)) {
            found = 1;
        } else {
            fputs(line, temp_fp);
        }
    }
    fclose(fp);
    fclose(temp_fp);
    if (found) {
        remove("accounts.txt");
        rename("temp.txt", "accounts.txt");
        printf("Account removed successfully.\n\n\n");
        printf("Proceeding to the Admin Panel...\n\n");
        admin_panel();
    } else {
        remove("temp.txt");
        printf("Account not found.\n\n");
        printf("Proceeding to the Admin Panel...\n\n");
        admin_panel();

    }
}
```

ProjectFile.c

```c
}
// DISPLAYING ALL ACCOUNTS
void display_all_accounts() {
    char line[1024];
    FILE *fp;
    char *token;
    const char delimiter[2] = ",";
    fp = fopen("accounts.txt", "r");
    if (fp == NULL) {
        printf("Error opening file.\n");
        printf("Proceeding to the Admin Panel...\n");
        admin_panel();
        return;
    }
    printf("\nAll Accounts:\n\n");
    printf("%-15s%-15s%-15s%-15s%-15s%-15s\n\n", "Name", "Address", "Phone Number", "Account Number", "Password", "Initial Deposit");
    while (fgets(line, sizeof(line), fp)) {
        token = strtok(line, delimiter);
        while (token != NULL) {
            printf("%-15s", token);
            token = strtok(NULL, delimiter);
        }
        printf("\n");
    }
    fclose(fp);
}

//USER PANEL FUNCTION CALL

void user_panel() {
    int choice;
    char username[20];
    char password[20];
    const char user_username[] = "user";
    const char user_password[] = "user";
    printf("Enter your username: ");
```

ProjectFile.c

```c
    printf("Enter your username: ");
    scanf("%s", username);
    printf("Enter your password: ");
    scanf("%s", password);
    system("cls");
    if (strcmp(username, user_username) == 0 && strcmp(password, user_password) == 0) {
        while (1) {
            printf("\t\t\t\t\t\t [WELCOME TO USER PANEL]\n\n\n\n\n\n");
            printf("\t\t\t\t\t .........Below the Options are.........\n\n");
            printf("\t\t\t\t\t\t [1]. Account Opening\n");
            printf("\t\t\t\t\t\t [2]. Account Status\n");
            printf("\t\t\t\t\t\t [3]. Change Password\n");
            printf("\t\t\t\t\t\t [4]. Payee Management\n");
            printf("\t\t\t\t\t\t [5]. View Statement\n");
            printf("\t\t\t\t\t\t [6]. Go to MainMenu\n");
            printf("\t\t\t\t\t\t [7]. Exit\n");
            printf("Enter your choice: ");
            scanf("%d", &choice);
            int c;
            while ((c = getchar()) != '\n' && c != EOF);
            switch (choice) {
                case 1:
                    system("cls");
                    printf("\t\t\t\t\t\t  [ACCOUNT OPENING]\n\n\n\n\n\n");
                    account_opening();
                    printf("Account Created Succesfully");
                    break;
                case 2:
                    system("cls");
                    printf("\t\t\t\t\t\t  [ACCOUNT STATUS]\n\n\n\n\n\n");
                    account_status();
                    printf("Proceeding to User Panel..\n");
                    break;
                case 3:
                    system("cls");
                    printf("\t\t\t\t\t\t  [CHANGE PASSWORD]\n\n\n\n\n\n");
```

ProjectFile.c

```c
                    printf("\t\t\t\t\t\t  [CHANGE PASSWORD]\n\n\n\n\n\n");
                    change_password();
                    break;
                case 4:
                    system("cls");
                    printf("\t\t\t\t\t\t  [PAYEE MANAGEMENT]\n\n\n\n\n\n");
                    transactions();
                    break;
                case 5:
                    system("cls");
                    printf("\t\t\t\t\t\t  [VIEW STATEMENT]\n\n\n\n\n\n");
                    view_statement();
                    break;
                case 6:
                    printf("Proceeding to the MainMenu...");
                    mainmenu();
                    break;
                case 7:
                    getout();
                    break;
                default:
                    printf("Invalid choice. Please try again.\n");
                    printf("Proceeding to the User Panel...\n");
                    system("cls");
                    user_panel();
                    break;
            }
        }
    } else {
        printf("Incorrect username or password.\n\n\n");
        printf("Try Again....\n\n\n");
        user_panel();
    }
}
//USER PANEL FUNCTIONS

```

ProjectFile.c

```c
//USER PANEL FUNCTIONS

// ACCOUNT STATUS
void account_status() {
    char account_number[20];
    char line[1024];
    FILE *fp;
    int found = 0;
    printf("Enter your account number: ");
    fgets(account_number, sizeof(account_number), stdin);
    account_number[strcspn(account_number, "\n")] = 0;
    fp = fopen("accounts.txt", "r");
    if (fp == NULL) {
        printf("Error opening file.\n");
        return;
    }
    printf("\n%-15s %-30s %-15s %-15s %-15s %-15s\n", "Name", "Address", "Phone", "Account Number", "Password", "Initial Deposit");
    while (fgets(line, sizeof(line), fp)) {
        char *name = strtok(line, ",");
        char *address = strtok(NULL, ",");
        char *phone = strtok(NULL, ",");
        char *acc_num = strtok(NULL, ",");
        char *password = strtok(NULL, ",");
        char *initial_deposit = strtok(NULL, ",");
        if (name == NULL || address == NULL || phone == NULL || acc_num == NULL || password == NULL || initial_deposit == NULL) {
//            printf("Error parsing line: %s\n", line);
            continue;
        }
        if (strcmp(account_number, acc_num) == 0) {
            printf("%-15s %-30s %-15s %-15s %-15s %s\n", name, address, phone, acc_num,password, initial_deposit);
            found = 1;
            break;
        } else {
//            printf("Debug: acc_num=%s account_number=%s\n", acc_num, account_number);
        }
    }
```

ProjectFile.c

```c
    }

    if (!found) {
        printf("Account not found.\n");
        printf("Proceeding to User Panel..\n");
        user_panel();
    }

    fclose(fp);
}
// CHANGE PASSWORD

void change_password() {
    char account_number[20];
    char new_password[20];
    FILE *fp;
    FILE *temp_fp;
    char line[256];
    int found = 0;
    printf("Enter your account number: ");
    scanf("%s", account_number);
    printf("Enter your new password: ");
    scanf("%s", new_password);
    fp = fopen("accounts.txt", "r");
    if (fp == NULL) {
        printf("Error opening file.\n");
        return;
    }
    temp_fp = fopen("temp.txt", "w");
    if (temp_fp == NULL) {
        printf("Error opening file.\n");
        fclose(fp);
        return;
    }
    while (fgets(line, sizeof(line), fp)) {
        if (strstr(line, account_number)) {
```

ProjectFile.c

```c
        if (strstr(line, account_number)) {
            char *name = strtok(line, ",");
            char *address = strtok(NULL, ",");
            char *phone = strtok(NULL, ",");
            char *acc_num = strtok(NULL, ",");
            char *password = strtok(NULL, ",");
            float initial_deposit = atof(strtok(NULL, ","));

            fprintf(temp_fp, "%s,%s,%s,%s,%s,%f\n", name, address, phone, acc_num,new_password, initial_deposit);
            found = 1;
        } else {
            fputs(line, temp_fp);
        }
    }
    fclose(fp);
    fclose(temp_fp);
    if (found) {
        remove("accounts.txt");
        rename("temp.txt", "accounts.txt");
        printf("Password changed successfully.\n");
    } else {
        remove("temp.txt");
        printf("Account not found.\n");
    }
}
// PAYEE MANAGEMENT
void transactions(){
        printf("Enter your choice:\n1. Deposit\n2. Withdrawal\n");
        int options;
        scanf("%d", &options);
        switch (options)
        {
            case 1:
                printf("Proceeding to Deposit panel..");
                make_deposit();
                break;
```

[*] ProjectFile.c

```c
                break;
            case 2:
                printf("Proceeding to Withdrawal panel..");
                make_withdrawal();
                break;
        }
}
//deposit
void make_deposit() {
    char account_number[20];
    float amount;
    FILE *fp;
    time_t t = time(NULL);
    struct tm tm = *localtime(&t);
    printf("Enter your account number: ");
    scanf("%s", account_number);
    printf("Enter the amount to deposit: ");
    scanf("%f", &amount);
    fp = fopen("transactions.txt", "a");
    if (fp == NULL) {
        printf("Error opening file.\n");
        printf("Proceeding to the User Panel...");
        user_panel();
        return;
    }
    fprintf(fp, "%s %04d-%02d-%02d %02d:%02d:%02d Deposit %.2f\n",
    account_number, tm.tm_year + 1900, tm.tm_mon + 1, tm.tm_mday, tm.tm_hour, tm.tm_min, tm.tm_sec, amount);
    fclose(fp);
    printf("Deposit successful.\n");
}
//withdrawal
void make_withdrawal() {
    char account_number[20];
    float amount;
    FILE *fp;
    time_t t = time(NULL);
```

[*] ProjectFile.c

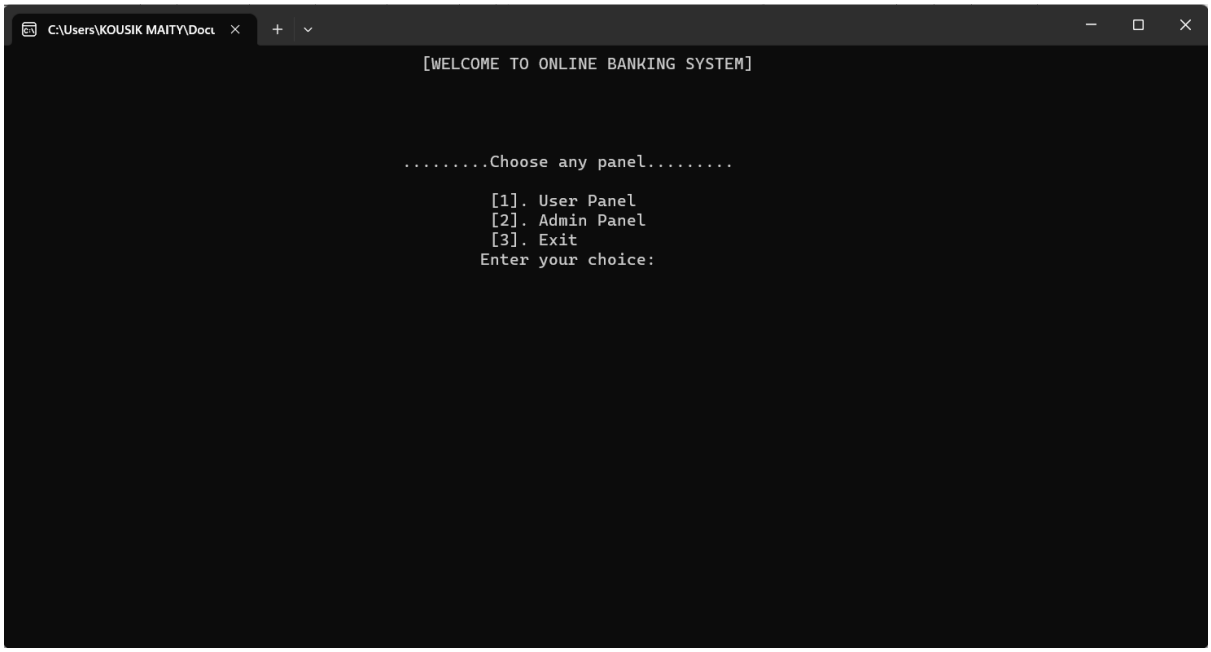
```c
    struct tm tm = *localtime(&t);
    printf("Enter your account number: ");
    scanf("%s", account_number);
    printf("Enter the amount to withdraw: ");
    scanf("%f", &amount);
    fp = fopen("transactions.txt", "a");
    if (fp == NULL) {
        printf("Error opening file.\n");
        return;
    }
    fprintf(fp, "%s %04d-%02d-%02d %02d:%02d:%02d Withdrawal %.2f\n",
    account_number, tm.tm_year + 1900, tm.tm_mon + 1, tm.tm_mday, tm.tm_hour, tm.tm_min, tm.tm_sec, amount);
    fclose(fp);
    printf("Withdrawal successful.\n");
}
// VIEW STATEMENT
void view_statement() {
    char account_number[20];
    FILE *fp;
    char line[256];
    char date[11];
    char time[9];
    char description[50];
    float amount;
    int found = 0;
    printf("Enter your account number: ");
    scanf("%s", account_number);
    fp = fopen("transactions.txt", "r");
    if (fp == NULL) {
        printf("No Transactions to display.\nAfter successfully completing a transaction, the Account Statement will be displayed ");
        return;
    }
    while (fscanf(fp, "%s %s %s %s %f\n", line, date, time, description, &amount) == 5) {
        if (strcmp(line, account_number) == 0) {
            printf("%s %s %s %s %.2f\n", date, time, description, account_number, amount);
            found = 1;
        }
```

[*] ProjectFile.c

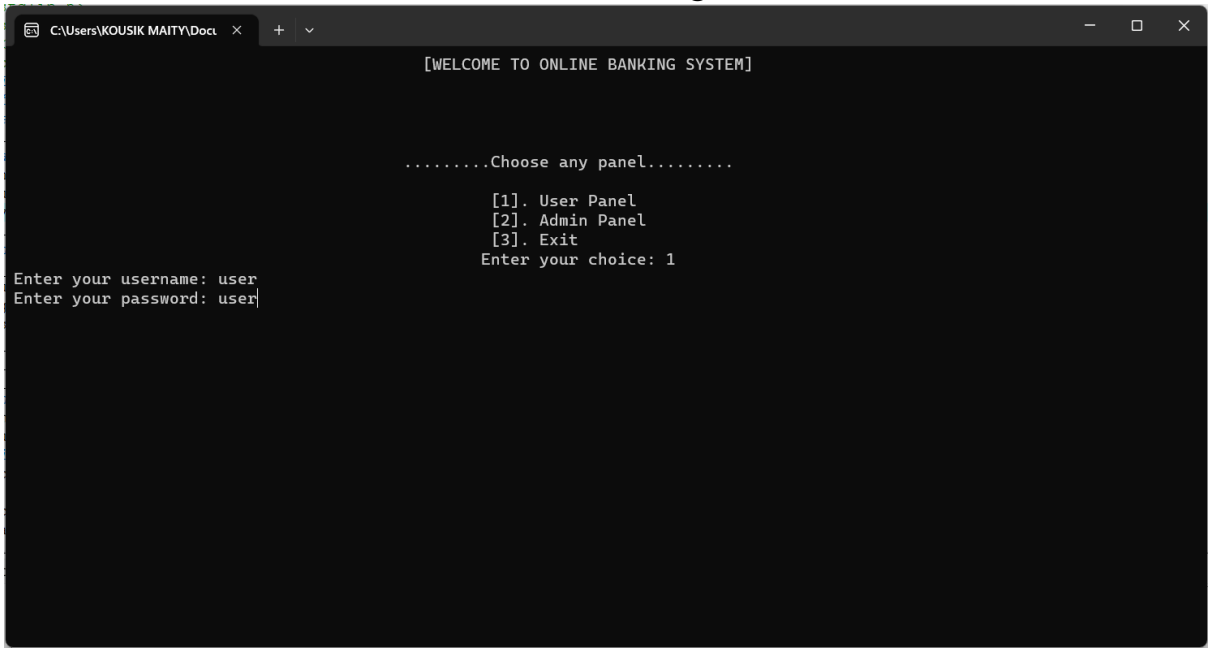
```c
    fprintf(fp, "%s %04d-%02d-%02d %02d:%02d:%02d Withdrawal %.2f\n",
    account_number, tm.tm_year + 1900, tm.tm_mon + 1, tm.tm_mday, tm.tm_hour, tm.tm_min, tm.tm_sec, amount);
    fclose(fp);
    printf("Withdrawal successful.\n");
}
// VIEW STATEMENT
void view_statement() {
    char account_number[20];
    FILE *fp;
    char line[256];
    char date[11];
    char time[9];
    char description[50];
    float amount;
    int found = 0;
    printf("Enter your account number: ");
    scanf("%s", account_number);
    fp = fopen("transactions.txt", "r");
    if (fp == NULL) {
        printf("No Transactions to display.\nAfter successfully completing a transaction, the Account Statement will be displayed ");
        return;
    }
    while (fscanf(fp, "%s %s %s %s %f\n", line, date, time, description, &amount) == 5) {
        if (strcmp(line, account_number) == 0) {
            printf("%s %s %s %s %.2f\n", date, time, description, account_number, amount);
            found = 1;
        }
    }
    if (!found) {
        printf("No transactions found for account number %s.\n", account_number);
    }
    fclose(fp);
}
int main() {
    panel_selection();
return 0;}
// ..........................................................................THANK YOU.....................................................
```

OUTPUT SNAPSHOTS

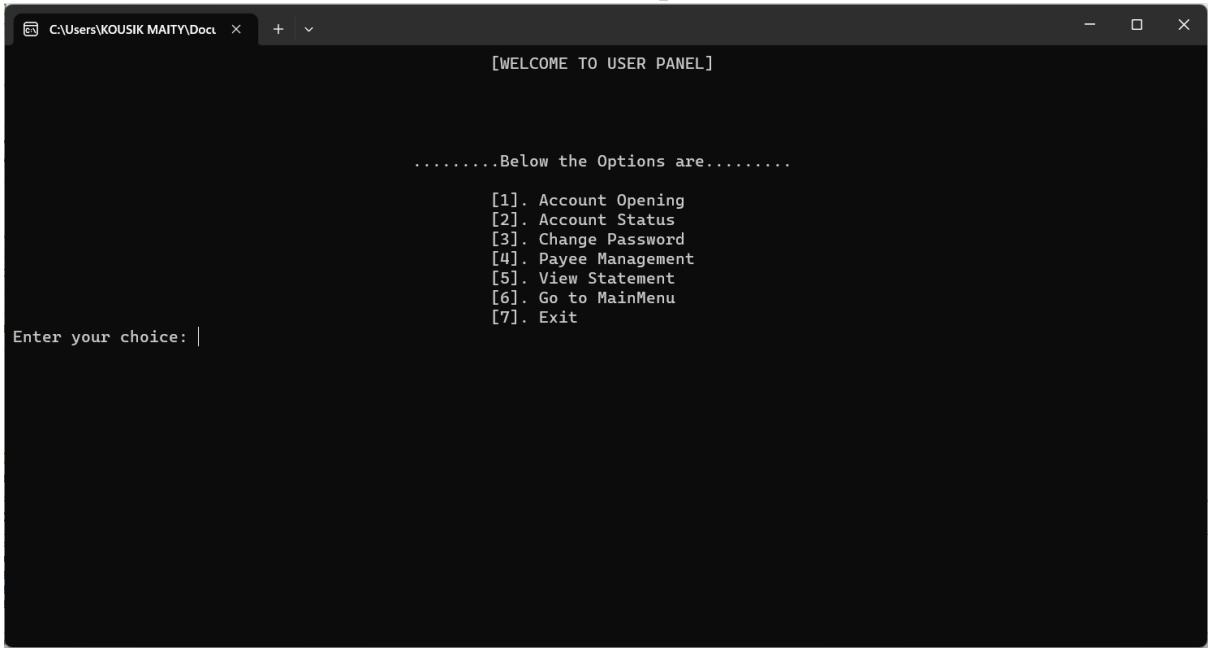
Panel Selection



User Panel Login



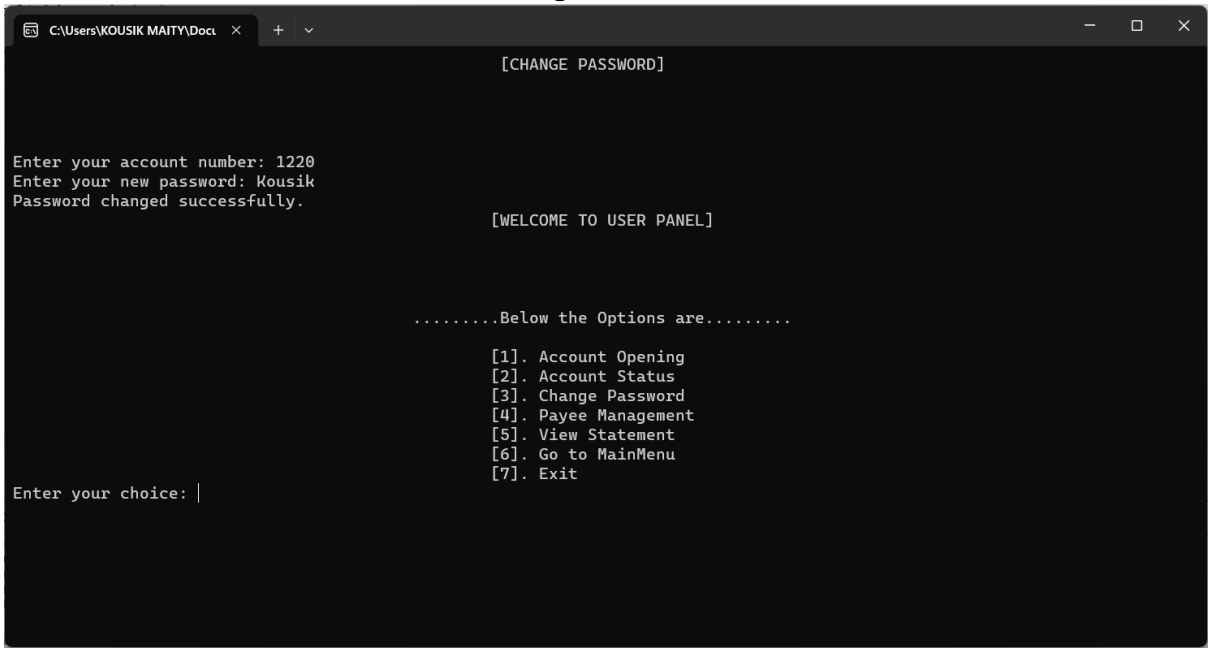
User Panel Options



Account Opening

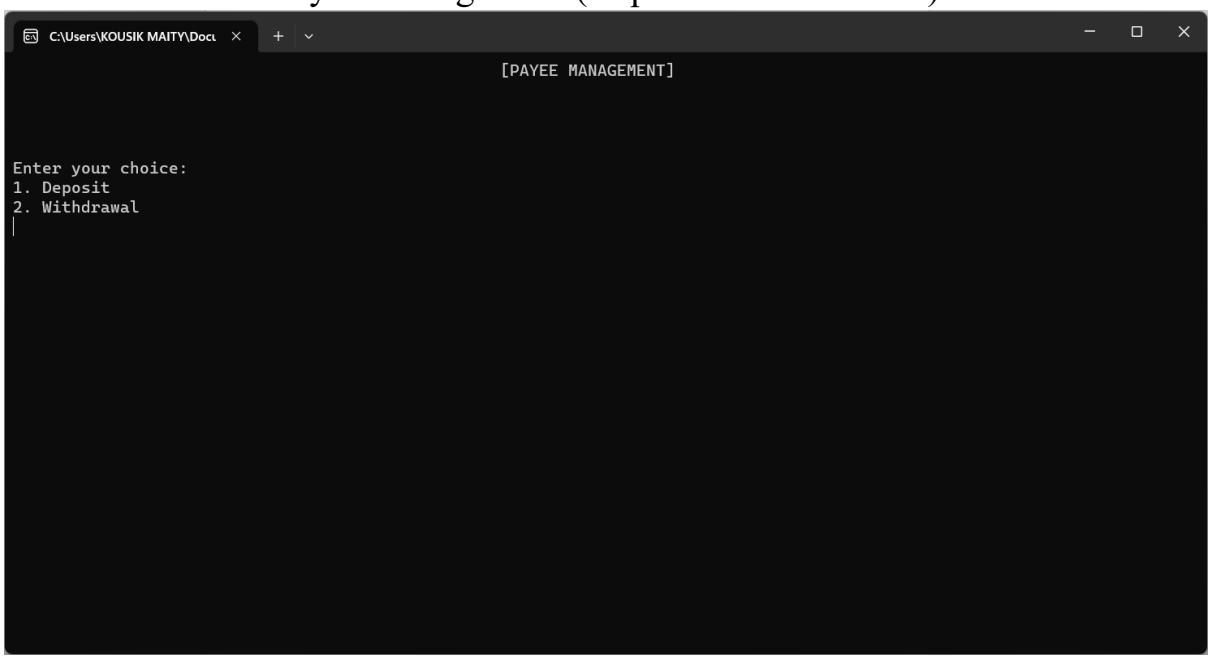
Account Status

Change Password



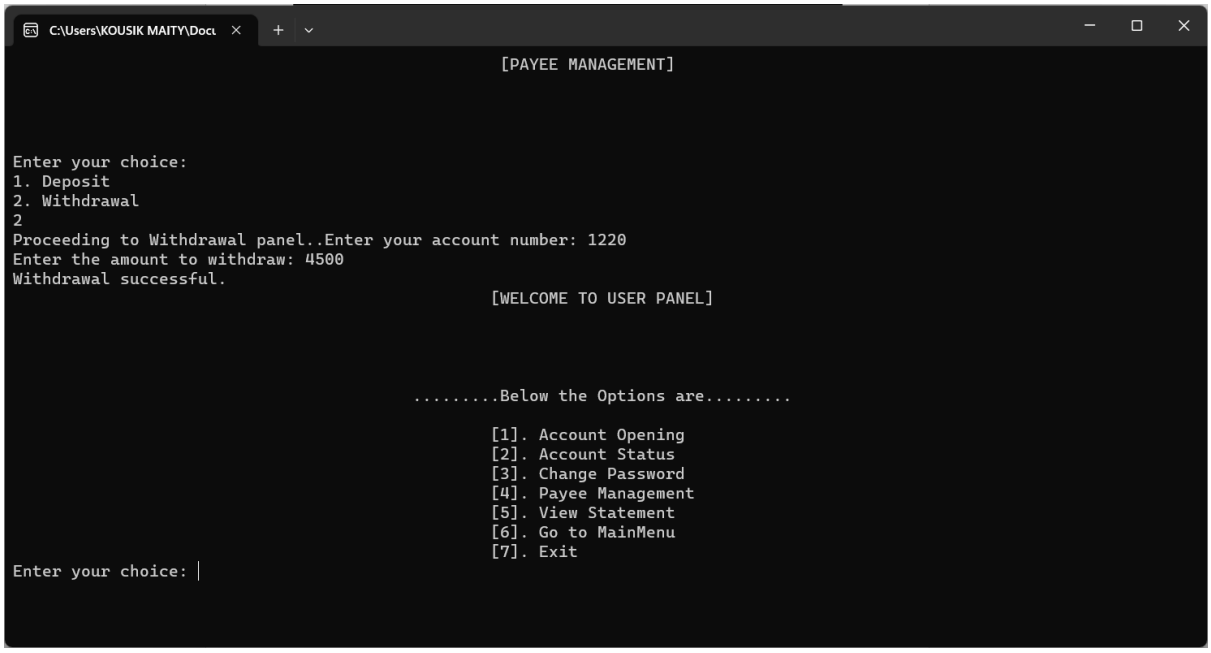
New Password Showing

Payee Management (Deposit or Withdrawal)

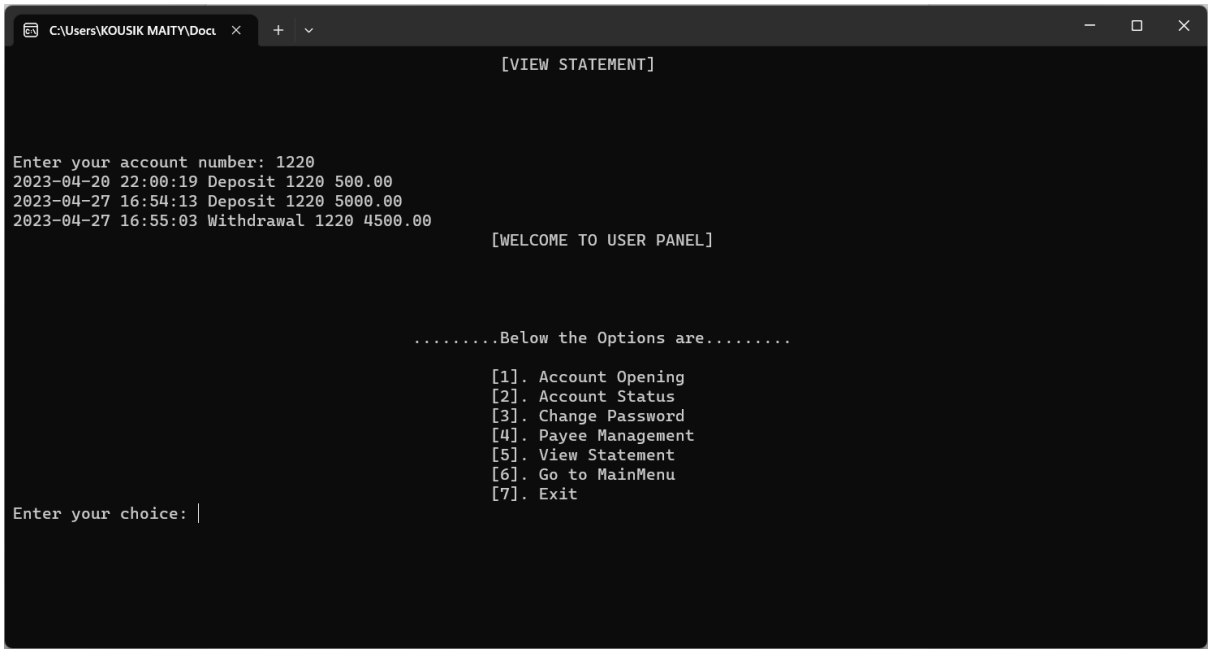


Deposit

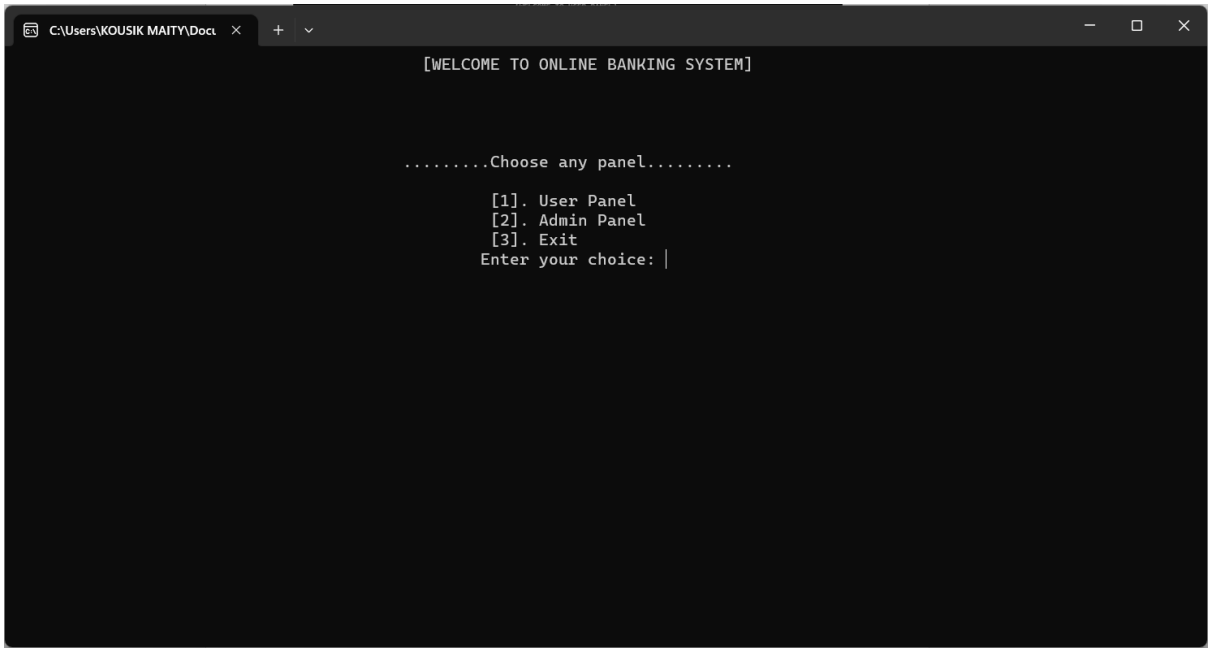
Withdrawal



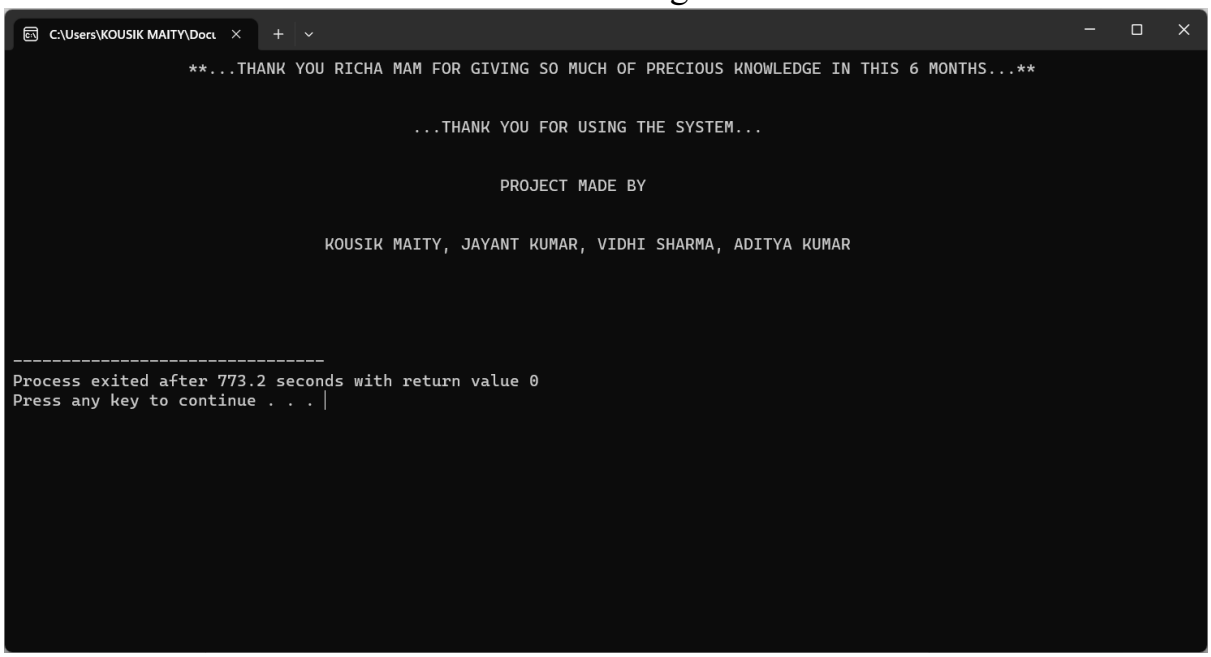
View Statement



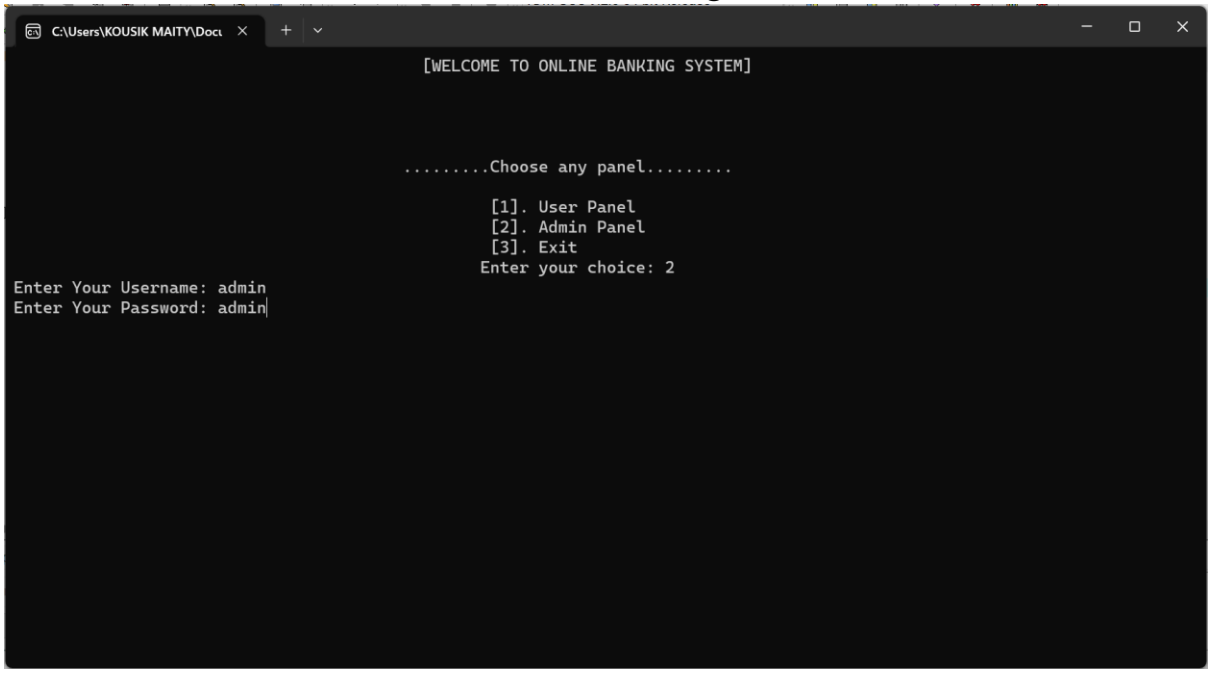
Main Menu



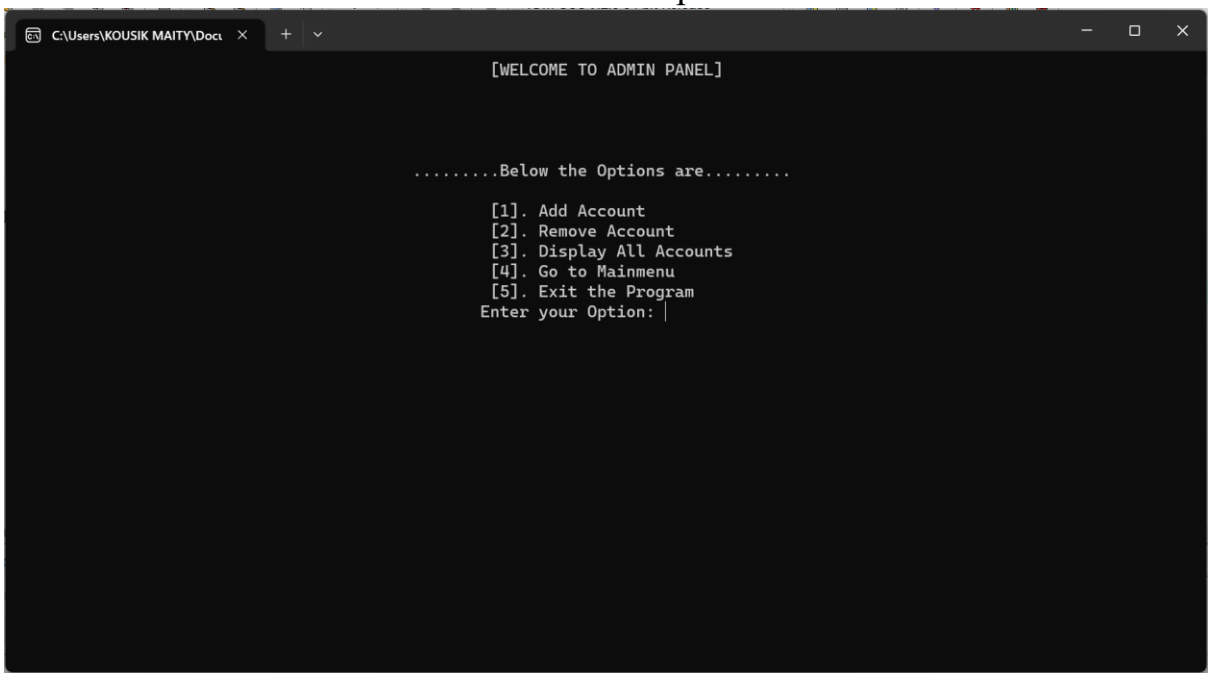
Exit from Program



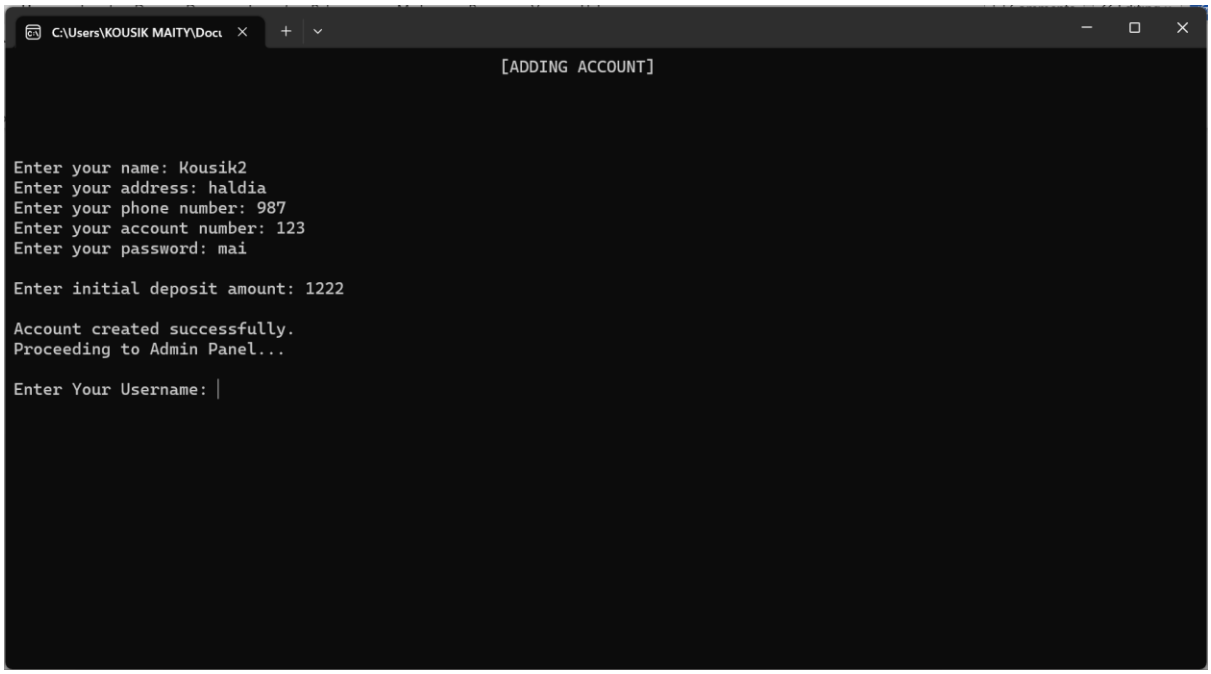
Admin Panel Login



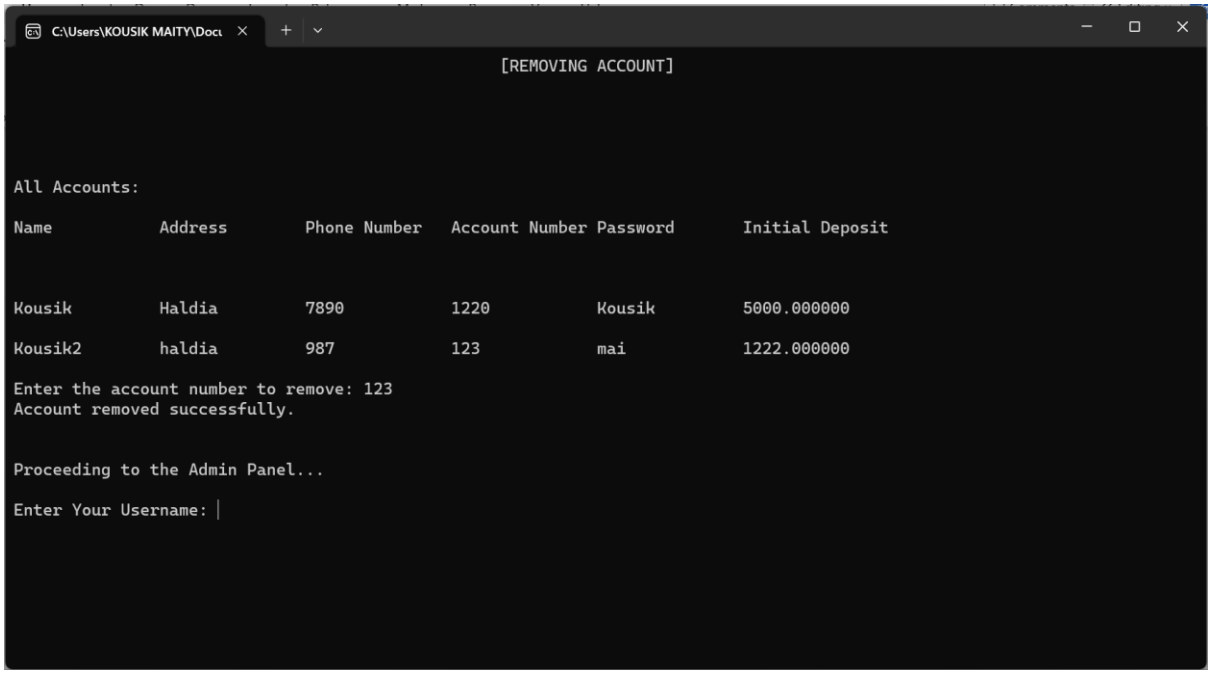
Admin Panel Options



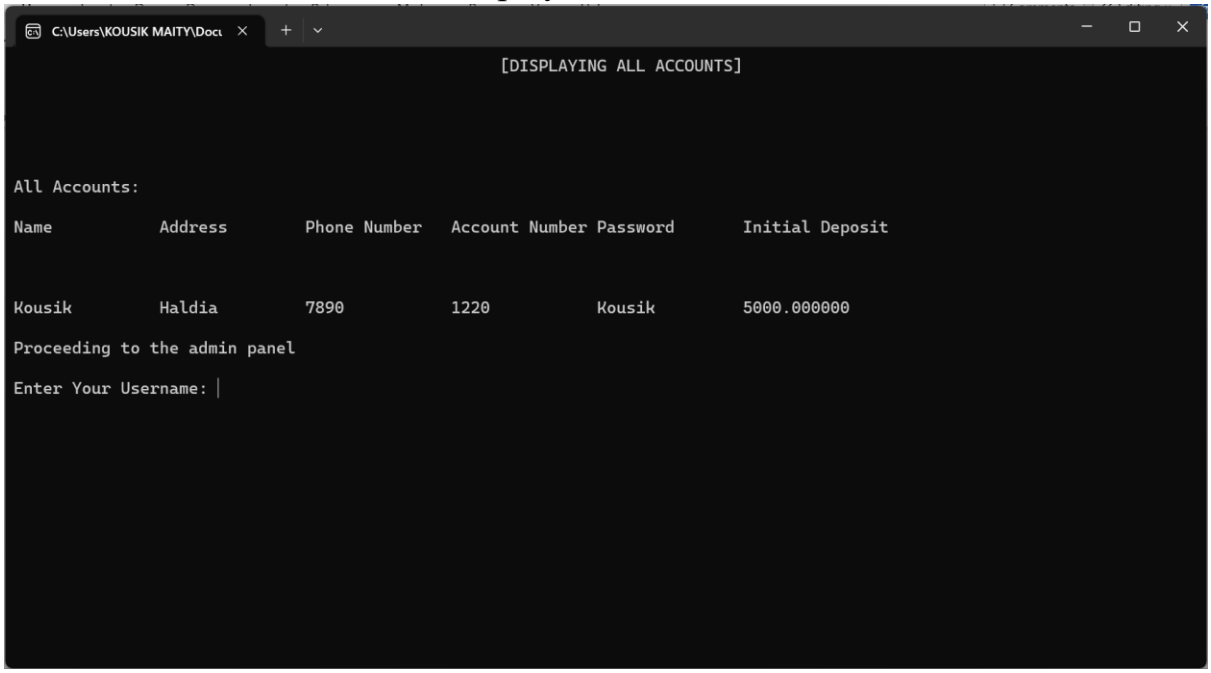
Add Account



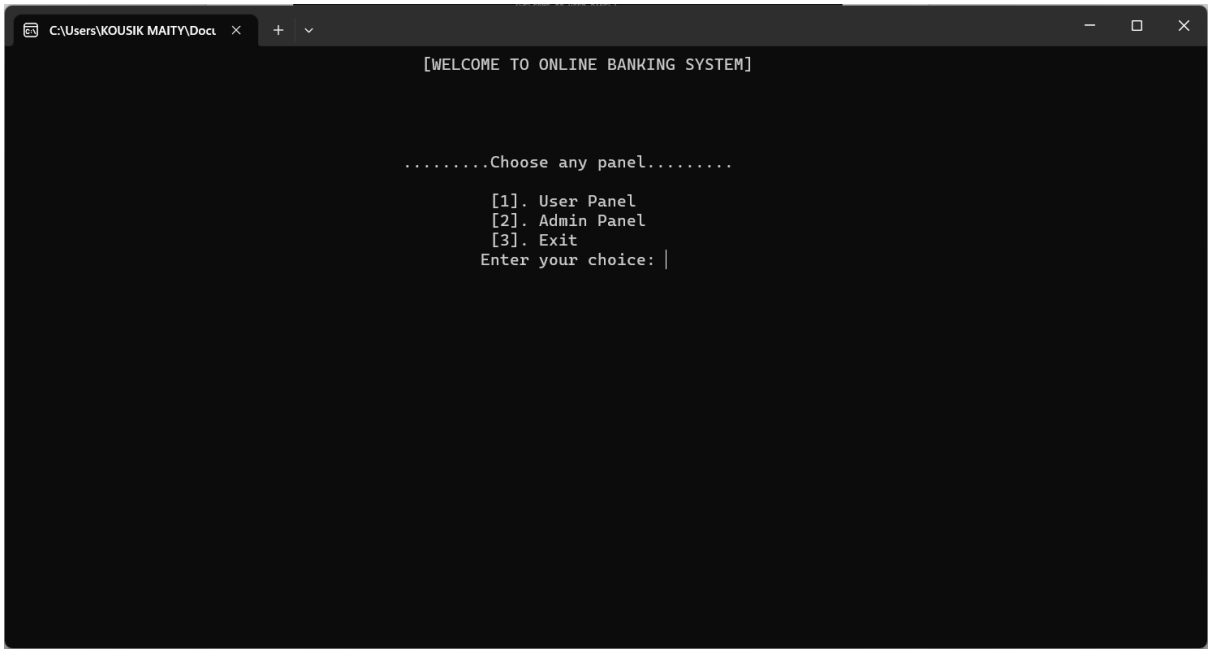
Remove Account



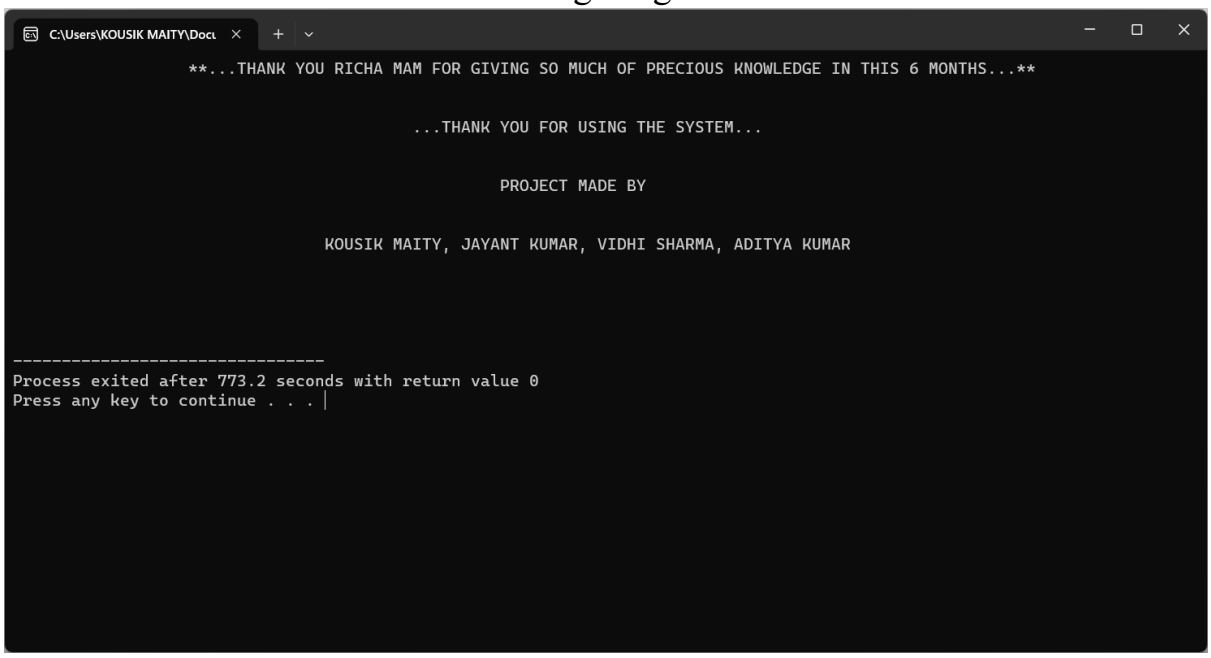
Display All Accounts



Main menu



Exiting Program



CONCLUSION:

In conclusion, the Online Banking System project is a comprehensive and user-friendly online platform that allows customers to perform banking transactions and manage their accounts from the comfort of their homes or offices. With its secure features and easy-to-use interface, this project is a valuable asset to any bank that is looking to improve its online banking services.

.....THANK YOU.....