

Programming fundamentals

**bank management system**

**Semmester Project**

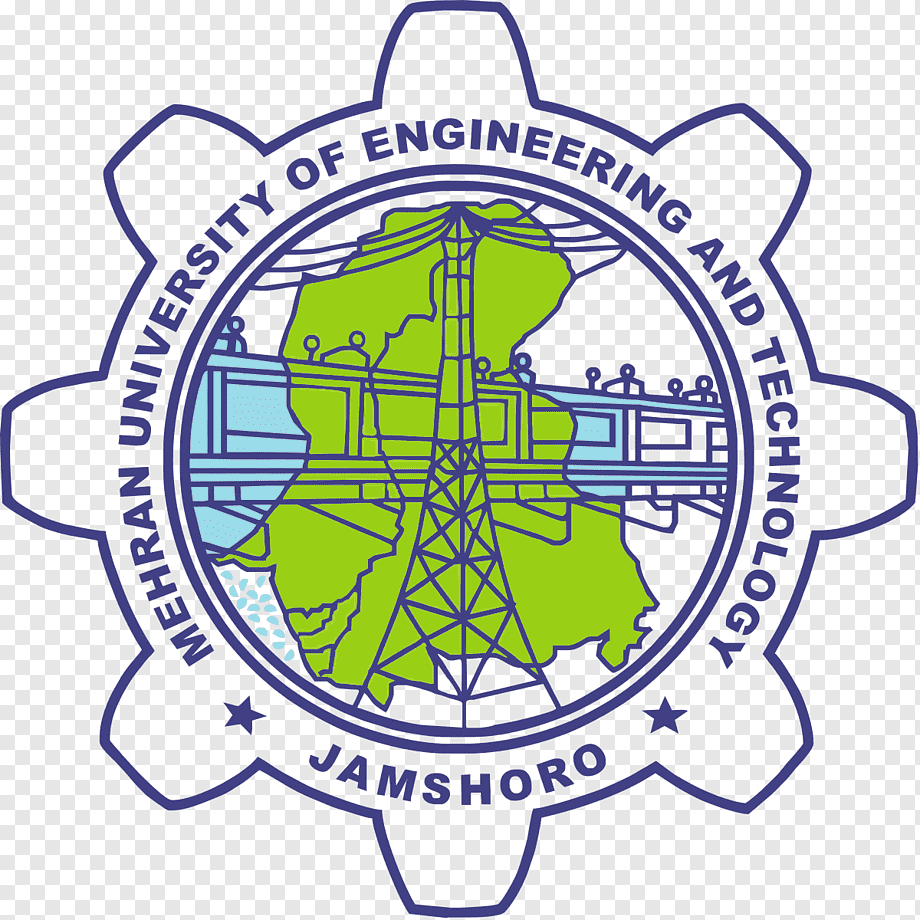


**Submitted to : Sir Sajjad Ali**

**Submitted by : 23SW050**

November 20, 2023

MUET JamshOro

******

***Project Documentation: Bank Management System***

***Problem Statement:***

**The existing manual record-keeping system in banks is time-consuming, error-prone, and lacks efficiency. There is a need for a computerized system to manage customer accounts, transactions, and related activities effectively.**

***Benefits of Implementing Bank Management System***:

**a.** **Efficiency:** Automation reduces manual effort and speeds up processes.

**b. Accuracy:** Minimizes errors associated with manual data entry.

**c. Customer Service:** Enhances customer service with quick access to information.

**d. Security:** Protects sensitive customer data through secure access controls.

**e. Transaction Tracking:** Enables efficient tracking of transactions and account activities.

**f. Record Management:** Facilitates easy retrieval and management of customer records.

**g. Reporting:** Generates accurate reports for better decision-making.

***Purpose:***

**The purpose of the Bank Management System is to provide a computerized platform for efficient management of customer accounts, transactions, and other banking operations. It aims to streamline processes, reduce errors, and enhance overall customer service.**

***Scope:***

**The system covers the creation of new accounts, viewing customer lists, checking details of existing accounts, performing transactions, and deleting accounts. It also includes data storage and retrieval functionalities.**

***Advantages:***

**a. User-Friendly Interface:** The system provides an easy-to-use interface for users.

**b. Data Persistence:** Utilizes file storage to retain customer data between sessions.

**c. Transaction Handling:** Allows users to deposit or withdraw cash efficiently.

**d. Record Deletion:** Provides options for deleting specific records or all records.

**e. Data Security:** Implements basic file-based security for data storage.

***Disadvantages:***

**a. Limited Security:** The current implementation lacks advanced security features.

**b. File Dependency:** The system relies on file storage, which may have limitations.

**c. Scalability:** Limited to a predefined number of customer records (100 in this case).

**d. Console Interface:** The system uses a console interface, which may not be user-friendly for all users.

***Overview:***

**The Bank Management System is a console-based application that allows users to perform various banking operations. It utilizes a simple menu-driven approach for user interaction. The system incorporates basic data validation and error handling.**

***Headers and Functions Used:***

**Header: <iostream>**

**Purpose:** Provides functionality for input and output operations.

**Usage in Code:** Utilized for standard input/output operations, including cout and cin for displaying output and taking user input.

**Header: <fstream>**

**Purpose:** Enables file stream operations for reading from and writing to files.

**Usage in Code:** Used for reading from and writing to a file (bank\_data.txt), facilitating data persistence between program executions.

**Header: <conio.h>**

**Purpose:** Offers functions for console input/output and screen control.

**Usage in Code:** Utilized for getch() to capture a single character from the console without echoing it, enhancing user interaction.

**Header: <windows.h>**

**Purpose:** Provides functions for interacting with the Windows API, including console manipulation.

**Usage in Code:** Utilized for system("color E4") to set the console text and background color for aesthetic purposes.

**Header: <iomanip>**

**Purpose:** Defines manipulators used for formatting output.

**Usage in Code:** Utilized for setiosflags and setfill to format the output with right alignment and asterisk fill for a decorative appearance.

**Function: void choice()**

**Purpose:** Implements the main menu and handles user's choices.

**Usage in Code:** Manages the flow of the program by presenting options to the user and invoking corresponding functions based on the user's selection.

**Function: void perData()**

**Purpose:** Gathers and records personal data for a new bank account.

**Usage in Code:** Captures user input for creating a new account, including name, ID, address, contact, and initial cash balance.

**Function: void show()**

**Purpose:** Displays the details of all customer accounts.

**Usage in Code:** Iterates through the customer data array and prints information for each account.

**Function: void search()**

**Purpose:** Searches for a customer account based on the provided ID.

**Usage in Code:** Takes user input for the ID, searches for a matching account, and displays the details if found.

**Function: void transaction()**

**Purpose:** Handles deposit and withdrawal transactions for a customer account.

**Usage in Code:** Prompts the user for transaction details, updates the account balance, and displays the new balance.

**Function: void del()**

**Purpose:** Manages the deletion of customer records.

**Usage in Code:** Offers options to delete a specific record or all records, updating the total count of customer records accordingly.

**Function: void saveDataToFile()**

**Purpose:** Writes customer data to a file for persistence.

**Usage in Code:** Creates or overwrites the bank\_data.txt file, storing customer details in a structured format for later retrieval.

**Function: void loadDataFromFile()**

**Purpose:** Reads customer data from a file to restore previous records.

**Usage in Code:** Retrieves customer information from the bank\_data.txt file during program initialization, ensuring data continuity between sessions.

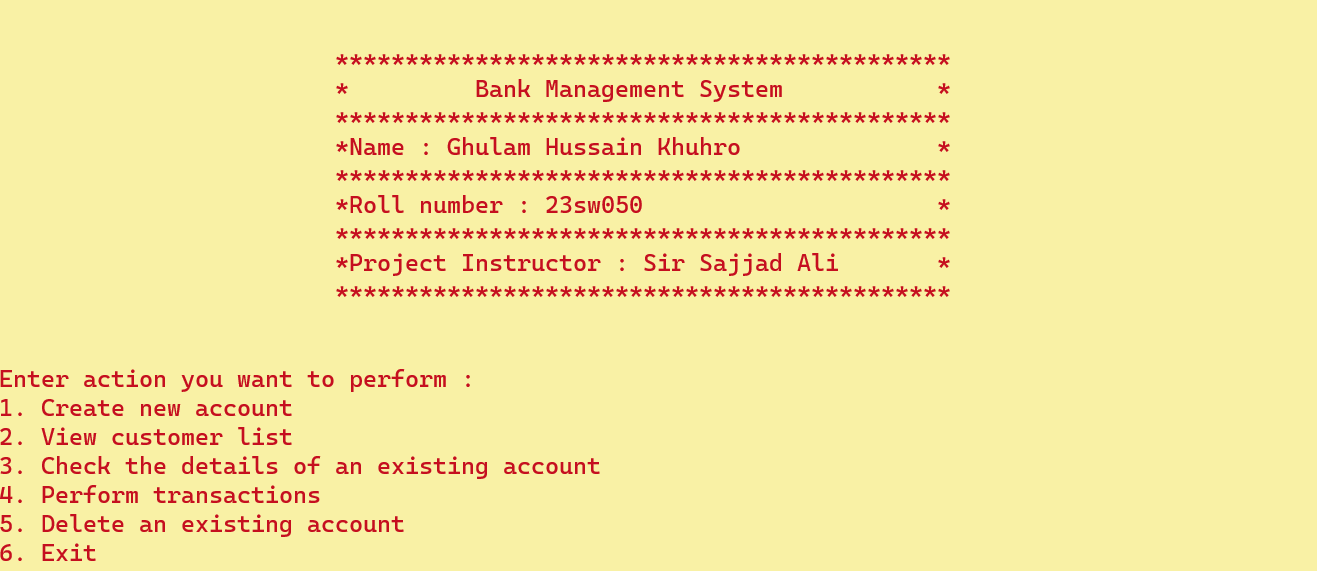
**Function: int main()**

**Purpose:** Entry point of the program.

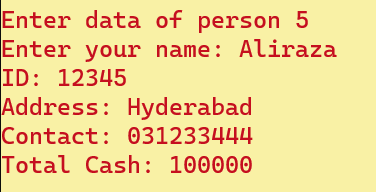
**Usage in Code:** Initializes the Bank Management System, loads existing data from the file, and calls the choice() function to initiate user interaction.

***Note: The system may benefit from further improvements in terms of security, scalability, and user interface design for broader usability and effectiveness.***

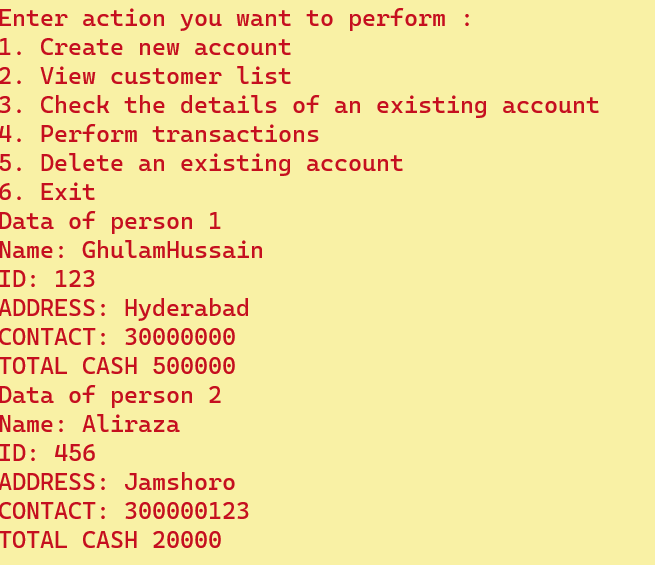
**Interface :**

******

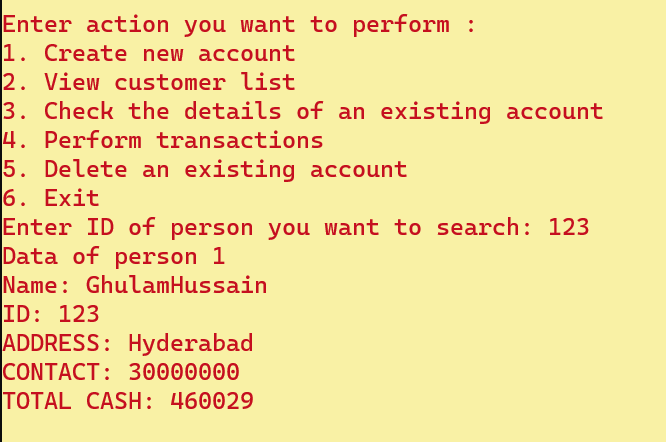
**Creating an account : Press 1**



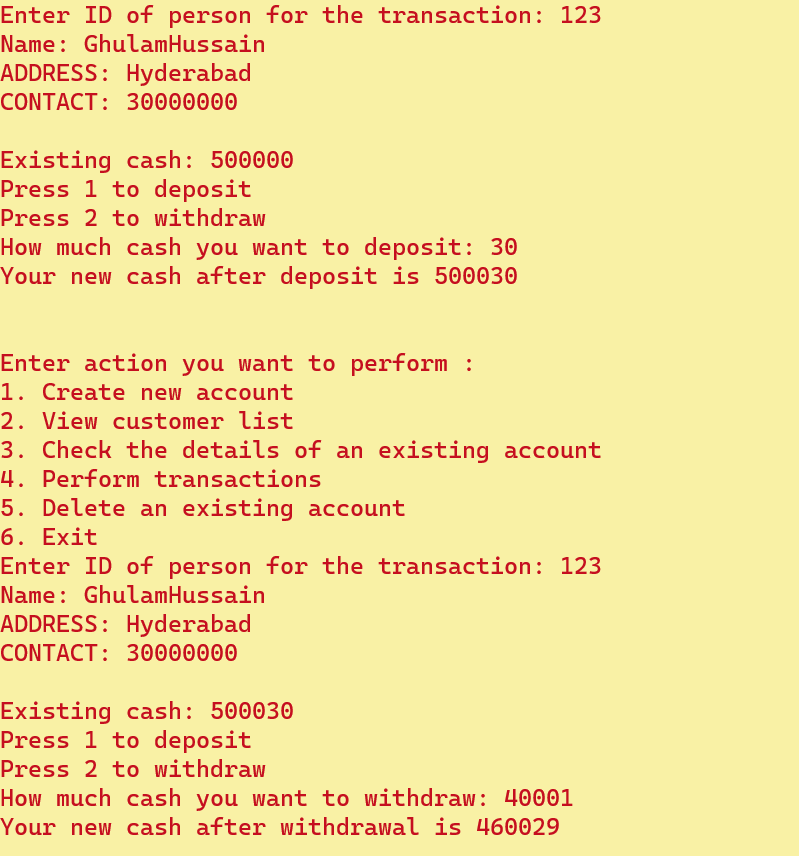
**View customer list : Press 2**



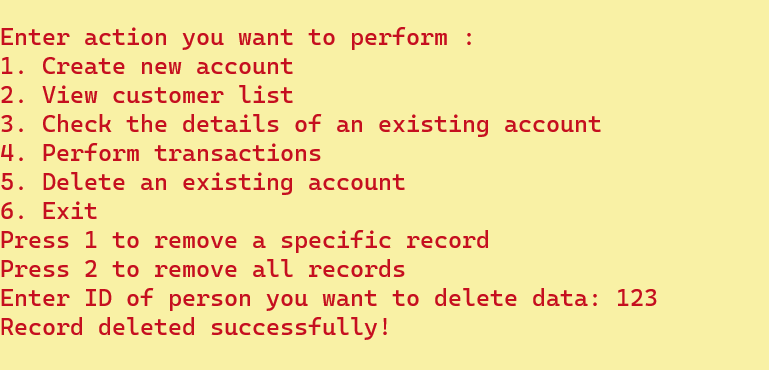
**Check the details of an existing account : Press 3**



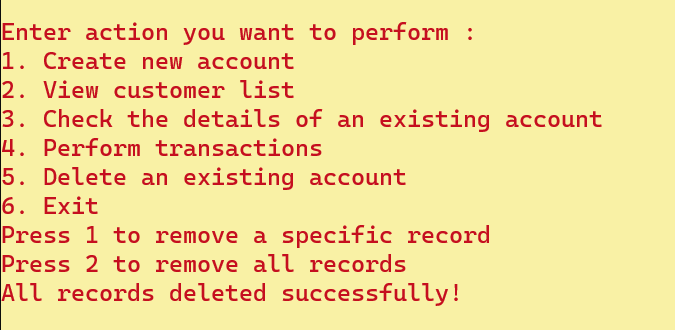
**Perform transactions : Press 4 then🡪1 to deposit and 2 to withdraw amount.**



**Deleting record : Press 5 then🡪1 to remove specific record.**



**2 to clear all records.**



**Exiting program : Press 6 to exit program**

