|  |
| --- |
| Assignment Report |
| Banking System Code |
| Submitted By:  Shadman Zaman Sajid  Roll: AE-014 |

10/7/2025

Banking System Code

Used language: C and C++

Function Used:

void CreateAccount();

void ViewAccount();

void Deposit();

void Withdraw();

void ListAccounts();

void deleteAccounts();

void take\_Existing\_Accounts\_From\_List(vector<int> &AC\_NOs);

Files Used: Both Binary (.bin) and text (.txt) files.

File handling:

1. Binary Files: Store account balances and transaction histories.
2. Text Files: Store account information and the list of all accounts

**Necessary functions and header files:**

1. Stdio.h -> for input/output purpose
2. String.h -> for operation using string
3. Strcat(string 1, string 2) -> concatenates string 1 and string 2 into string 1
4. Snprintf() -> convert data type into string
5. File related functions (fopen(), fclose(), fscanf(), fgets(), fputs(), fread(), fwrite())
6. Bits/stdc++.h -> to use c++ STL

**take\_Existing\_Accounts\_From\_List(vector<int> &AC\_NOs):**

1. All existing account numbers are stored in a vector
2. Prevents creation of duplicate accounts by iterating through all elements of the vector.

**CreateAccount():**

1. Creates a new bank account for a user.
2. Stores user information (name, account number, email, phone number) in a text file.
3. Stores account balance in a binary file.
4. Prevents duplicate account creation by calling “take\_Existing\_Accounts\_From\_List()”

**ViewAccount():**

1. Displays the current status of a user’s account.
2. To see one’s account number he/she has to enter correct account number.
3. Shows error if the account number does not exist, the function terminates.

**Deposit():**

1. Allows user to deposit money into their account after entering correct account number.
2. Updates account details and stores transactions in a binary file.

**Withdraw():**

1. Enables users to withdraw money from their account after verifying the account number.
2. Checks if the withdrawal amount exceeds the current balance; if so, displays an error.
3. Records the transaction in a binary file.

**ListAccounts():**

1. Maintains a text file named “Accounts\_info.txt” containing all created accounts.
2. Access to this file requires a username and password that is declared globally.

**DeleteAccount():**

1. Allows user to delete their account by entering the correct account number.
2. Removes all data related to the account from both text and binary files.