**10. Source Code  
 10.1 Program Structure Overview**

The upgraded version of the Bank Management System uses a modular approach in C. It separates logic into multiple files and uses file handling to save account data in a .txt file permanently. This makes the program more organized.

**We divided the program into:**

* main.c – The entry point of the program
* bank.c – Contains logic for all banking operations
* bank.h – Header file with structure and function declarations
* account\_data.txt – Text file to save account details

**This update aligns with Section 5: Proposed Methodology where modular programming and file handling were highlighted.**

**10.2 File: bank.h  
// bank.h**

**#ifndef BANK\_H**

**#define BANK\_H**

**struct BankAccount {**

**int accountNumber;**

**char name[50];**

**float balance;**

**};**

**// Function declarations**

**void createAccount(struct BankAccount \*acc);**

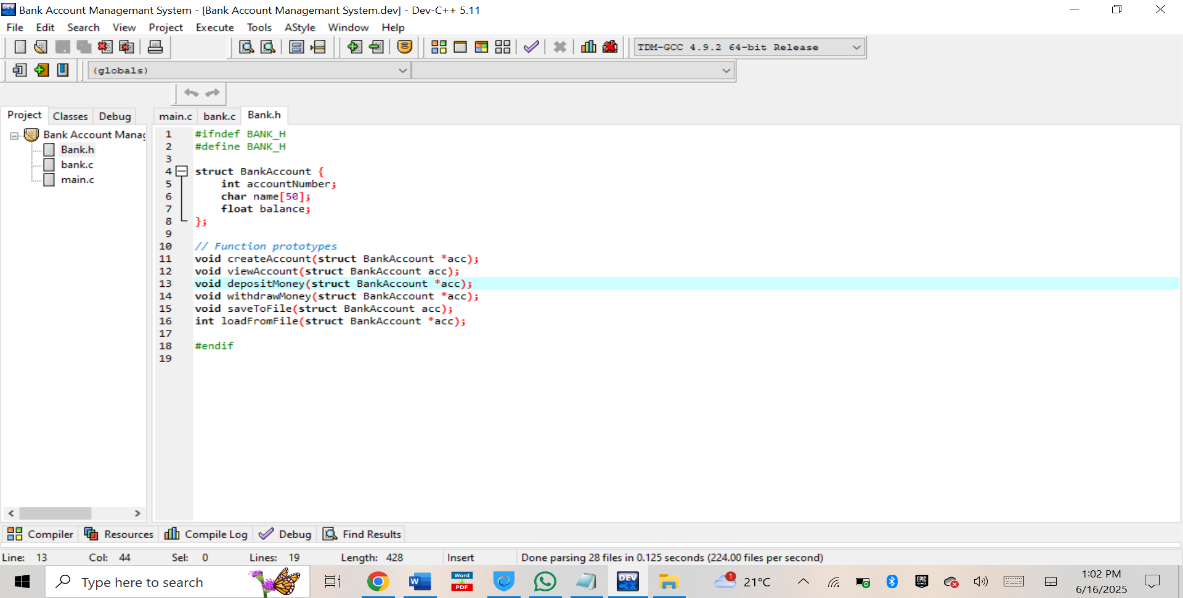
**void viewAccount(struct BankAccount acc);**

**void depositMoney(struct BankAccount \*acc);**

**void withdrawMoney(struct BankAccount \*acc);**

**void saveToFile(struct BankAccount acc);**

**void loadFromFile(struct BankAccount \*acc);**

**#endif  
Explanation:**This header file defines the BankAccount structure and declares all banking functions. It ensures that other files can use these functions via #include "bank.h".  
  
*Source code of main.c showing the main program structure and function calls.*

**10.3 File: bank.c  
// bank.c**

**#include <stdio.h>**

**#include <string.h>**

**#include "bank.h"**

**void createAccount(struct BankAccount \*acc) {**

**printf("\n=== Create Account ===\n");**

**printf("Enter Account Number: ");**

**scanf("%d", &acc->accountNumber);**

**printf("Enter Account Holder Name: ");**

**scanf(" %[^\n]", acc->name);**

**acc->balance = 0;**

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

**saveToFile(\*acc);**

**}**

**void viewAccount(struct BankAccount acc) {**

**printf("\n=== Account Details ===\n");**

**printf("Account Number: %d\n", acc.accountNumber);**

**printf("Name: %s\n", acc.name);**

**printf("Current Balance: %.2f\n", acc.balance);**

**}**

**void depositMoney(struct BankAccount \*acc) {**

**float amount;**

**printf("\nEnter amount to deposit: ");**

**scanf("%f", &amount);**

**if (amount > 0) {**

**acc->balance += amount;**

**printf("Deposited successfully!\n");**

**saveToFile(\*acc);**

**} else {**

**printf("Invalid amount!\n");**

**}**

**}**

**void withdrawMoney(struct BankAccount \*acc) {**

**float amount;**

**printf("\nEnter amount to withdraw: ");**

**scanf("%f", &amount);**

**if (amount > 0 && amount <= acc->balance) {**

**acc->balance -= amount;**

**printf("Withdrawal successful!\n");**

**saveToFile(\*acc);**

**} else {**

**printf("Insufficient balance or invalid amount!\n");**

**}**

**}**

**void saveToFile(struct BankAccount acc) {**

**FILE \*file = fopen("account\_data.txt", "w");**

**if (file != NULL) {**

**fprintf(file, "%d\n%s\n%.2f\n", acc.accountNumber, acc.name, acc.balance);**

**fclose(file);**

**}**

**}**

**void loadFromFile(struct BankAccount \*acc) {**

**FILE \*file = fopen("account\_data.txt", "r");**

**if (file != NULL) {**

**fscanf(file, "%d\n", &acc->accountNumber);**

**fgets(acc->name, 50, file);**

**acc->name[strcspn(acc->name, "\n")] = 0;**

**fscanf(file, "%f\n", &acc->balance);**

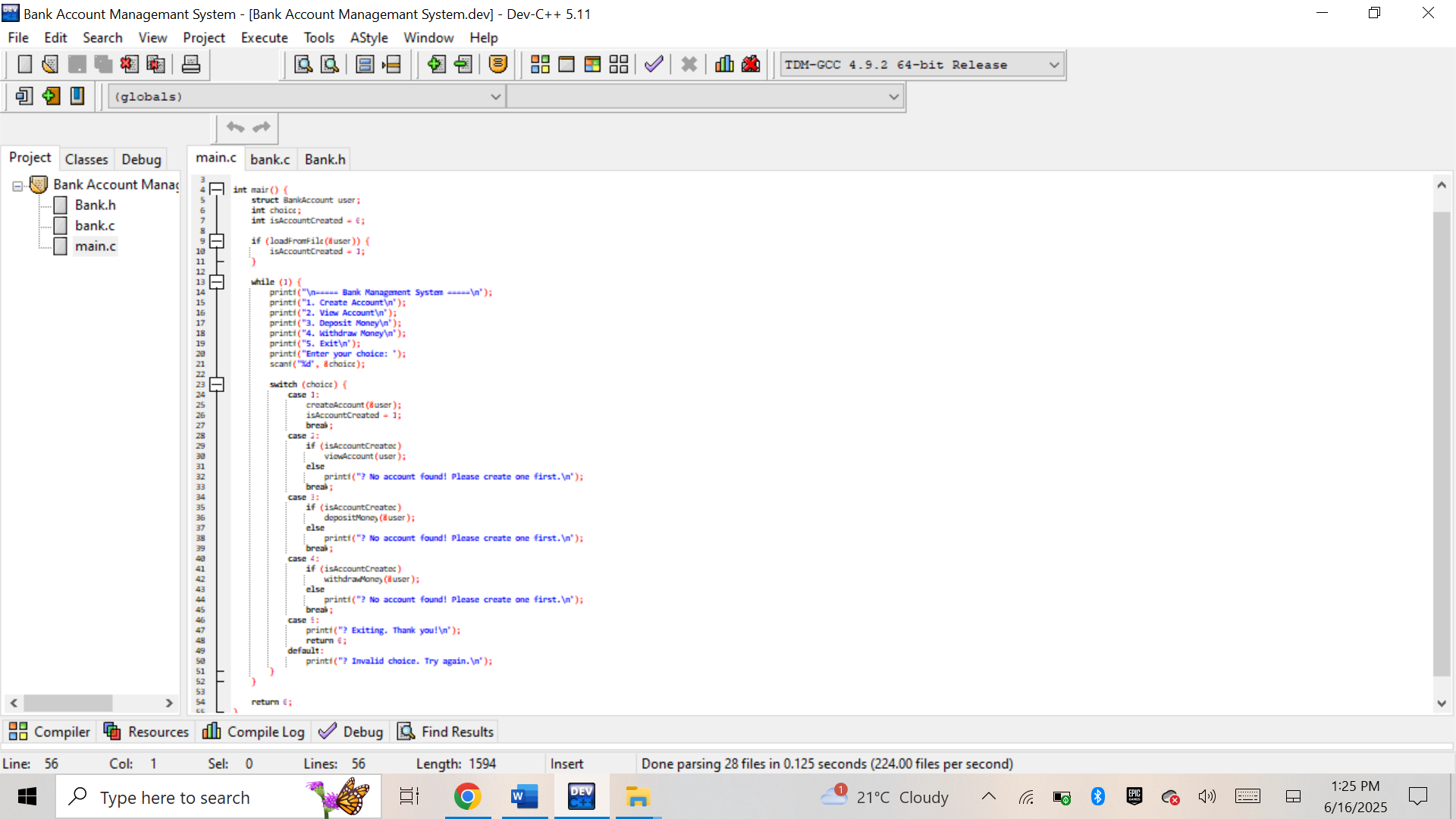
**fclose(file);**

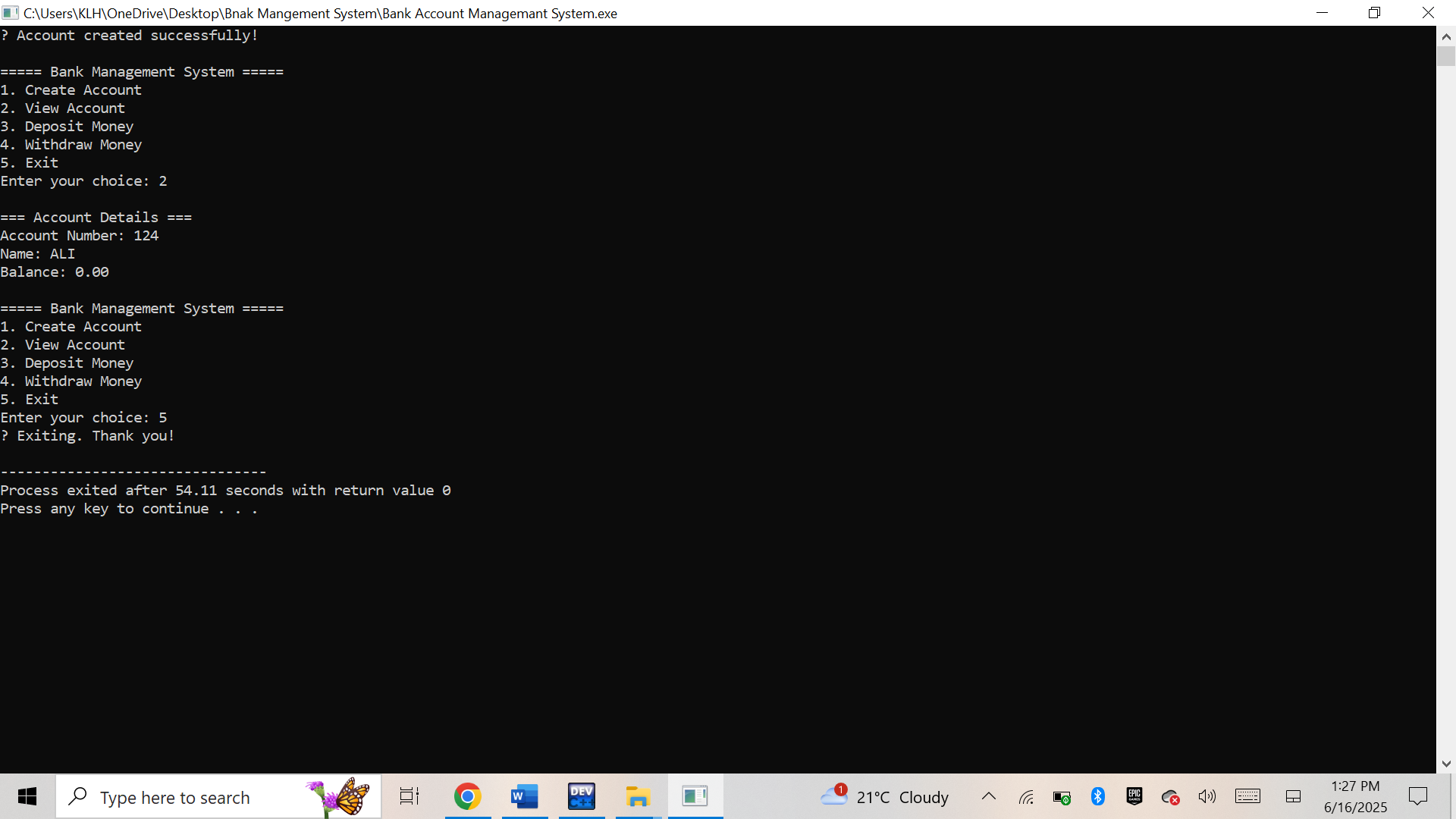
**}**

**}**

**Explanation:**

* saveToFile() saves account data permanently
* loadFromFile() loads existing data when the program starts
* All transactions like deposit/withdraw also update the file.

  
*Implementation of core banking operations including account management and file handling in bank.c.*

  
*Bank Management System running in console showing menu options and user interaction.*

**10.4 File: main.c  
// main.c**

**#include <stdio.h>**

**#include "bank.h"**

**int main() {**

**struct BankAccount user;**

**int choice;**

**int isAccountCreated = 0;**

**loadFromFile(&user);**

**isAccountCreated = 1;**

**while (1) {**

**printf("\n===== Bank Management System =====\n");**

**printf("1. Create Account\n");**

**printf("2. View Account\n");**

**printf("3. Deposit Money\n");**

**printf("4. Withdraw Money\n");**

**printf("5. Exit\n");**

**printf("Enter your choice (1-5): ");**

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

**switch (choice) {**

**case 1:**

**createAccount(&user);**

**isAccountCreated = 1;**

**break;**

**case 2:**

**if (isAccountCreated) viewAccount(user);**

**else printf("No account found. Create one first.\n");**

**break;**

**case 3:**

**if (isAccountCreated) depositMoney(&user);**

**else printf("No account found. Create one first.\n");**

**break;**

**case 4:**

**if (isAccountCreated) withdrawMoney(&user);**

**else printf("No account found. Create one first.\n");**

**break;**

**case 5:**

**printf("Thank you for using Bank Management System. Goodbye!\n");**

**return 0;**

**default:**

**printf("Invalid choice. Please try again.\n");**

**}**

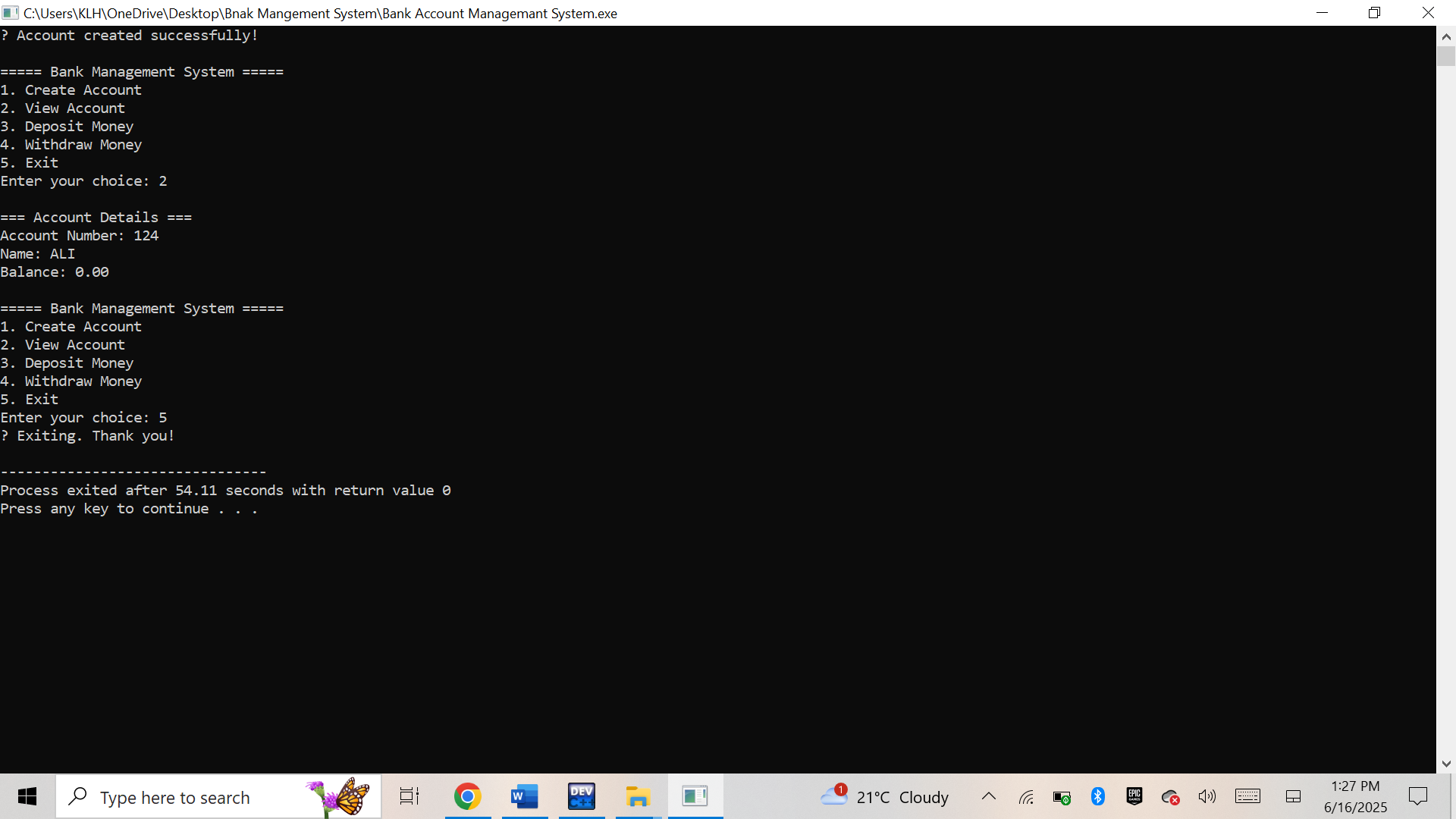
**}**

**return 0;**

**}**

**Explanation:**

* Loads data automatically if available
* Calls functions based on user’s menu choice
* All logic is kept clean by separating functions into different files

  
*Screenshot displaying the Bank Management System program running with sample input and output*