

# Bank Management System

## Project report

*Ayush Joshi*

*Institution - UPES*

*Sap id - 590027687*

*Course- BTech CSE*

*Date - December 2025*

# **Project Title**

Bank Management System in C

## **Abstract**

This project implements a simple Bank Management System written in the C programming language. It demonstrates file handling, modular programming, and the use of data structures. The system allows users to create accounts, deposit and withdraw money, check balances, and view all accounts. Account data is stored persistently in a binary file (accounts.dat).

## **Objectives**

- Apply file operations using binary read/write.
- Practice modular programming with header and source files.
- Implement structured data handling using structs.
- Build a user-friendly console-based menu system.

## **System Features**

- Create a new bank account with auto-incrementing account number.
- Deposit money into an account.
- Withdraw money with balance validation.
- Check account balance.
- Display all stored accounts.
- Persistent storage using accounts.dat binary file.

## **Directory Structure**

BankManagementSystem/

    ■■■ docs/

        ■ ■■■ ProjectReport.pdf

    ■■■ include/

        ■ ■■■ banking.h

    ■■■ src/

        ■ ■■■ main.c

        ■ ■■■ banking.c

    ■■■ README.md

## **Data Structure Used**

```
typedef struct {
    int accno; char
    name[50]; float
    balance; }
Account;
```

## **Important Functions**

- next\_account\_number() – finds next available account number.
- create\_account() – creates and stores a new account.
- update\_account() – internal function to modify account data.
- deposit\_money() / withdraw\_money() – update balance.

- `check_balance()` – view specific account details.
- `display_all_accounts()` – list all accounts.

## File Handling Details

The program uses a binary file named `accounts.dat` for storing account records. Functions `fread()` and `fwrite()` ensure efficient and compact data storage.

## Compilation

```
gcc -linclude src/main.c src/banking.c -o bank
```

## Execution

```
./bank
```

## Sample Run

1. Create Account → Enter name → Initial deposit = 1000
2. Deposit → Account 1001 → Amount = 500 → New balance = 1500
3. Withdraw → Account 1001 → Amount = 200 → New balance = 1300
4. Check Balance → Displays 1300

## Limitations

- No authentication system implemented.
- No transaction history stored.
- Does not support deletion/editing of accounts.
- Minimal input validation.

## Future Enhancements

- Add PIN/password authentication.
- Add transaction logs for each account.
- Implement delete/edit account features.
- Improve interface and validation.

*I referred to sir's notes, educational YouTube videos, and existing C programming projects such as Inventory Management System and Student Record Management to understand the concepts and structure necessary for building this Bank Management System.*