### **Project Title: Sample Banking System**

### **1. Introduction:**

The **Sample Banking System** project is a simple console-based application developed in C. This system allows users to simulate basic banking operations, including deposits, withdrawals, and balance checks. It provides a streamlined interface for account holders to manage their bank accounts and keep track of their transaction history.

### **2. Objective:**

The primary objective of the project is to demonstrate how a basic banking system can be implemented using C programming. The project aims to:

* Simulate core banking functions such as deposits, withdrawals, and balance checking.
* Provide users with a simple interface to handle transactions.
* Maintain a transaction history for reference.

### **3. Scope:**

The system is designed for individual users who can perform the following operations:

* Deposit money into their account.
* Withdraw money from their account, subject to available balance.
* Check their current balance.
* View a list of past transactions.

The application runs on a command-line interface, ensuring simplicity and ease of use. This project focuses on basic functionality without addressing advanced banking features like account transfers, interest calculations, or multi-user support.

### **4. Key Features:**

* **Deposit**: Users can add a specified amount to their account balance.
* **Withdraw**: Users can withdraw funds if the balance is sufficient; otherwise, they receive an error message.
* **Balance Check**: Users can view their current balance at any time.
* **Transaction History**: The system maintains a record of all transactions (deposits and withdrawals) for the current session.

### **5. Technical Aspects:**

* **Programming Language**: C
* **Data Structures**:
  + Arrays are used to store transaction history.
  + Pointers are used to manage account balance updates.
* **Control Structures**: Loops (do-while) and conditional statements (switch, if-else) control the flow of the application.
* **Functions**: The project uses modular programming with functions for each operation, including deposit, withdraw, check\_balance, and display\_transactions.

### **6. Modules:**

* **Main Menu**: Provides options for deposit, withdrawal, balance check, and exiting the system.
* **Deposit Module**: Allows the user to add funds to their account and updates the transaction history.
* **Withdrawal Module**: Facilitates the withdrawal of funds, ensuring that the balance is sufficient.
* **Balance Check Module**: Displays the current account balance.
* **Transaction History Module**: Logs and displays a list of all transactions performed during the session.

### **7. Conclusion:**

This project serves as an educational example of how to implement basic banking operations in C. It illustrates the use of control flow, functions, and data structures to create a functional, albeit simple, banking system. The modular approach makes it easy to extend the project for additional banking functionalities in the future.