Project Report:
Banking Management System

1. Project Title

Bank-Al-Bahria: A Procedural Banking Management System in C++

2. Objective

To design and implement a file-based banking management system in C++ that allows users to manage bank accounts with features such as account creation, deposits, withdrawals, balance inquiry, transaction history, and inter-account money transfers,

3. System Design

The system uses procedural programming, dynamic memory allocation, and file handling.

Architecture Overview:

- Main Menu
- Account Handling
- Transaction Logic
- File Management

Each module is connected via function calls and data stored in dynamically allocated arrays.

4. Features Implemented

- Create Account with CNIC and PIN
- Deposit and Withdraw money
- Check Balance
- Send money to other accounts

Project and Report by:
Muhammad Muneeb Zahid (01-134251-060)
Awwad Mehdi (01-134251-014)

- Maintain transaction history
- Save/load data using files

5. Code Structure and Key Snippets

5.1 Data Storage

The system uses dynamic arrays to store data for up to 100 accounts:

```
const int MAX_ACCOUNTS = 100;
int* accountNumbers = new int[MAX_ACCOUNTS];
char** names = new char*[MAX_ACCOUNTS];
int* pins = new int[MAX_ACCOUNTS];
double* balances = new double[MAX_ACCOUNTS];
char** cnics = new char*[MAX_ACCOUNTS];
```

- Each index represents one account. Data is persisted via accounts.txt.

5.2 Account Creation

```
void createAccount() {
    ...
    cout << "Enter CNIC (xxxxx-xxxxxxx-x): ";
    cin.ignore();
    cin.getline(tempCNIC, 20);
    ...
    logTransaction(accNo, "Account Created with Deposit", deposit, deposit);
    ...
}
```

- Validates CNIC and PIN
- Logs creation event in transactions_<accNo>.txt

Project and Report by:
Muhammad Muneeb Zahid (01-134251-060)
Awwad Mehdi (01-134251-014)

5.3 Deposit and Withdraw

```
void deposit() {
    ...
    balances[index] += amount;
    logTransaction(accountNumbers[index], "Deposit", amount, balances[index]);
    ...
}
```

```
void withdraw() {
    ...
    if (amount > balances[index]) {
        cout << "Insufficient funds.\n";
        return;
    }
    balances[index] -= amount;
    ...
}
```

- Ensures security using PIN verification
- Updates balance and transaction history

5.4 *Transaction* Logging

```
void logTransaction(int accNo, const char* type, double amount, double balance) {
    ofstream fout(filename, ios::app);
    fout << "[" << getTimestamp() << "] "
         << type << " of " << amount
         << " | Balance: " << balance << "\n";
    fout.close();
}
```

- Every transaction is logged with timestamp and account number.

Project and Report by:
Muhammad Muneeb Zahid (01-134251-060)
Awwad Mehdi (01-134251-014)

5.5 Send Money

```
void sendMoney() {
    ...
    balances[senderIndex] -= amount;
    balances[receiverIndex] += amount;
    logTransaction(senderAcc, "Sent Money", amount, balances[senderIndex]);
    logTransaction(receiverAcc, "Received Money", amount,
balances[receiverIndex]);
    ...
}
```

- Transfers funds securely between two valid accounts

5.6 File Operations

Save Accounts

```
void saveAccounts() {
    ofstream fout("accounts.txt");
    ...
    fout.close();
}
```

Load Accounts

```
void loadAccounts() {
    ifstream fin("accounts.txt");
    ...
    fin.close();
}
```

- Ensures persistence across sessions.

Project and Report by:
Muhammad Muneeb Zahid (01-134251-060)
Awwad Mehdi (01-134251-014)

6. Sample Output Screens

```
==== Wwelcome to Bank-Al-Bahria ====
1. Create Account
2. Deposit
3. Withdraw
4. Check Balance
5. View Transaction History
6. Send Money
7. Exit
Select an option: 1
Enter account number: 234
Enter account holder name: Muneeb Zahid
Set 4-digit PIN: 1234
Enter CNIC (xxxxx-xxxxxxx-x): 34603-5677123-0
Enter initial deposit: 34000
Account created successfully.
```

7. Limitations

- No encryption or hashing of PINs
- Limited to 100 accounts
- No GUI

8. Conclusion

This project successfully demonstrates a basic banking system using procedural programming and file handling. Despite constraints like no OOP, vectors, or advanced DSA, the system achieves essential banking functionalities. It highlights strong fundamentals in logic design, file I/O, and memory management.

Links:
Github: Bank Management System (Github)
LinkedIn: Project Video (LinkdIn)

Project and Report by:
Muhammad Muneeb Zahid (01-134251-060)
Awwad Mehdi (01-134251-014)