ATM Simulator in C - Project Report

1. Introduction

This project simulates basic ATM operations using the C programming language. The system allows a user to interact with a virtual bank account through a command-line menu. Operations such as checking balance, depositing, and withdrawing funds are supported.

2. Objective

To develop a simple, menu-driven ATM simulator in C that performs the following operations:
- Check account balance
- Deposit money
- Withdraw money
- Exit the application

3. Tools and Technologies Used

- Programming Language: C
- Compiler: GCC (GNU Compiler Collection)
- Development Environment: Any C IDE or terminal-based compilation

4. Program Structure

The ATM simulator is structured using a C struct to represent a bank account. It also uses modular functions to separate logic for each operation.

4.1. Structure Definition

A structure named BankAccount is defined to represent a bank account:

```
struct BankAccount {
    int accountNumber;
    char holderName[50];
```

```
    float balance;
};
```

4.2. Functions Used

- showMenu() - Displays the options to the user.
- checkBalance() - Prints account details and current balance.
- deposit() - Adds money to the account balance.
- withdraw() - Deducts money from the account if sufficient balance is available.

4.3. Main Function

Initializes a sample account with fixed values.

Displays the menu in a loop and takes user input for operations.

Uses a switch statement to call appropriate functions based on the user's choice.

5. Sample Output

```
===== ATM Simulator =====
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
Enter your choice: 1
Account Holder: AA8
Account Number: 9840254735
Current Balance: Rs. 1000000.00
```

6. Features

- Validates deposit and withdrawal amounts.

- Checks for sufficient funds before withdrawal.
- Uses pointers for updating balance directly.
- Infinite loop with an option to exit.

7. Limitations

- Only one user account is hardcoded.
- No security (e.g., PIN verification).
- Data is not stored persistently.
- Basic input validation only.

8. Suggestions for Improvement

- Implement multiple account handling.
- Add user authentication via PIN.
- Include transaction history.
- Enhance UI formatting.
- Use file I/O for data storage.

9. Conclusion

This ATM Simulator project effectively demonstrates basic banking transactions in C using structures and functions. It serves as a strong foundational exercise in procedural programming, structure manipulation, and menu-based input handling.