

# Internship Report: Task 1 - C++ Programming Intern

**Name** – Manisha Khadul

---

Intern: C++ Programming Intern

Task: Create a fully functional ATM interface using C++

First report submission –

The task assigned to me as part of my internship at **Brainwave Matrix Solutions** was to create a fully functional ATM interface using C++. This report details the development process, features implemented, and testing conducted during the first phase of the internship.

## Task Objectives

The main objective of this task was to design and implement an ATM system that allows users to:

- Authenticate using account number and PIN.
- Check account balance.
- Deposit and withdraw money.
- View transaction history.
- Calculate interest on the balance.
- Change PIN.
- Handle errors and invalid inputs gracefully.

## Implementation Details

The implementation was done using C++ programming language. The following components were developed as part of the ATM system:

1. Authentication Module: A user is prompted to enter their account number and PIN. If

both match the records, access is granted.

2. Transaction History: The system records each transaction and allows the user to view a history of deposits, withdrawals, and PIN changes.

3. Account Operations: Users can check their balance, deposit money, withdraw money, and calculate interest on their balance.

4. PIN Change: Users are allowed to change their PIN after verifying the old PIN.

5. Error Handling: The system ensures that only valid inputs are accepted and handles incorrect entries gracefully.

## **Features Implemented**

1. Account Authentication: The user enters an account number and PIN. If the details match the stored values, access is granted.

2. Balance Checking: The user can view the current balance after successful authentication.

3. Money Deposit and Withdrawal: The user can deposit money to their account or withdraw a specified amount, provided there are sufficient funds.

4. Transaction History: The system records each deposit, withdrawal, and PIN change, and the user can view this history.

5. Interest Calculation: The system can calculate and display interest based on the current balance.

6. PIN Change Feature: Users can change their PIN after entering the old PIN correctly.

## **Testing and Results**

I performed the following tests to ensure the functionality of the ATM system:

1. Login Test: Entered correct account number and PIN to ensure successful authentication.

2. Balance Test: Checked balance to verify it displays correctly.

3. Deposit and Withdrawal Test: Deposited and withdrew money to verify the operations

were processed correctly.

4. Transaction History Test: Verified the transaction history for correctness.
5. PIN Change Test: Successfully changed the PIN after providing the old PIN.
6. Error Handling Test: Provided incorrect inputs to verify the system displays appropriate error messages.

## **Future Enhancements**

Future enhancements that can be added to the ATM system include:

1. Multi-Account Support: Allow multiple users with different account numbers and PINs.
2. Bank Database Integration: Connect the ATM system to a real bank database for account and transaction management.
3. Security Features: Implement additional security features such as biometric authentication (fingerprint/face recognition).
4. User Interface Improvements: Enhance the user interface to provide a more interactive and user-friendly experience.

## **Conclusion**

In conclusion, I successfully implemented the ATM interface in C++ as part of my internship task. The system provides basic functionalities such as account authentication, balance checking, money deposit and withdrawal, transaction history, and PIN change. I also ensured proper error handling to make the system robust and user-friendly. The next steps will involve refining the user interface and integrating additional security features.