

Welcome To

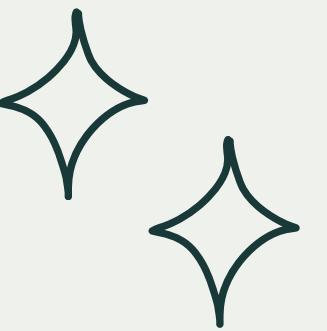
ANNEEX

PAY

Project by TEAM HASH INCLUDE



MEET OUR TEAM



Md. Monirul Hasan

ID: 20255103087



Bithi Rani Debi

ID: 20255103086



Ahana Islam

ID: 20255103118



Arpita Deb Aishee

ID: 20255103083



Hafsa Akter

ID: 20255103119



"OBJECTIVES"

What ANNEX PAY Aims to Achieve :

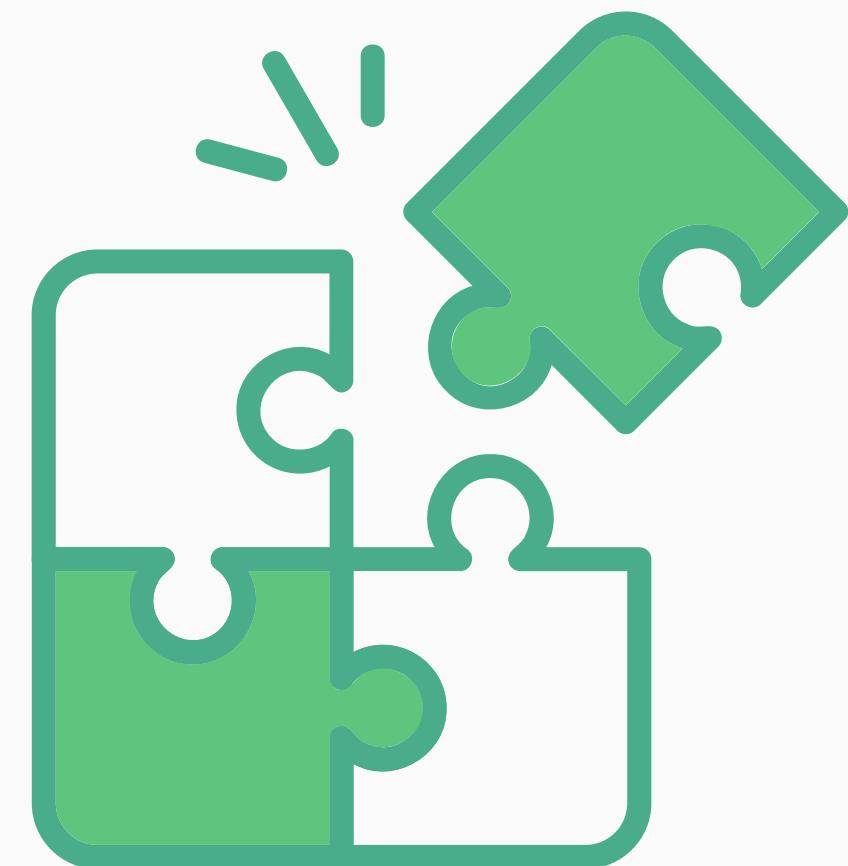
- Build a secure and efficient banking system
- Replace slow manual banking tasks with automation
- Store and manage user data safely using file handling
- Provide smooth user and admin workflows
- Add real-life features to make banking systems easier
- Create a colorful, interactive experience



“PROBLEM STATEMENT”

What Problem We Tackled :

- Real banking operations involve dozens of steps, but academic projects only cover 10% of them.
- Students don't have an easy-to-use tool to visualize how banking logic actually works.
- Manual handling of money in small campus activities often leads to confusion, miscommunication, and missing records.
- There was a need for a secure, modular, and learning-focused banking simulator that behaves like a real system.



OUR SOLUTION – ANNEX PAY ↗

YOUR COMPLETE BANKING ECOSYSTEM

- Console-based interface with colorful visuals & smooth navigation
- 19+ banking operations: deposits, transfers, bill payments, loans & more
- Secure login & authentication with PIN & recovery options
- Two modules: Admin and User, each with full control
- Cross-platform compatible: Works on Windows & Linux
- Data persistence: All transactions & accounts saved safely





KEY FEATURES:



- Account Creation & Management
 - Deposits, Withdrawals, Transfers
 - Bill Payments & Mobile Recharge
 - Loan Processing System
- 
- 





ADVANCED FEATURES:



Fund Request

User can request Fund
between users.



Account Recovery

User can set a Security
Questions for PIN Recovery

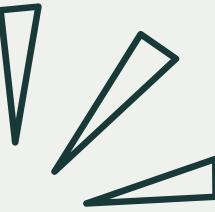


Apply for Checkbook

User Can Sent Checkbook
Requests to Admin



ADVANCED FEATURES:



Messaging System

If user face any problems, they can sent direct message to the admin.



Referral Program

If a User refer to another, they both will get bonuses.

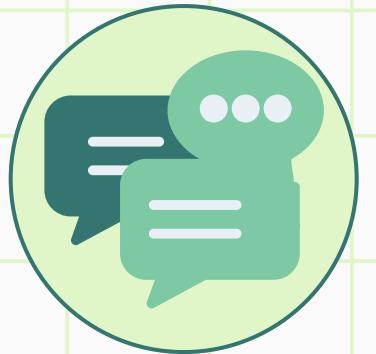
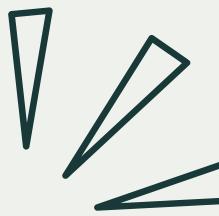


Transaction Search & History

User can search Transaction History by date and amount



PROGRAM SCENARIO:



Messaging System

If user face any problems, they can sent direct message to the admin.



Referral Program

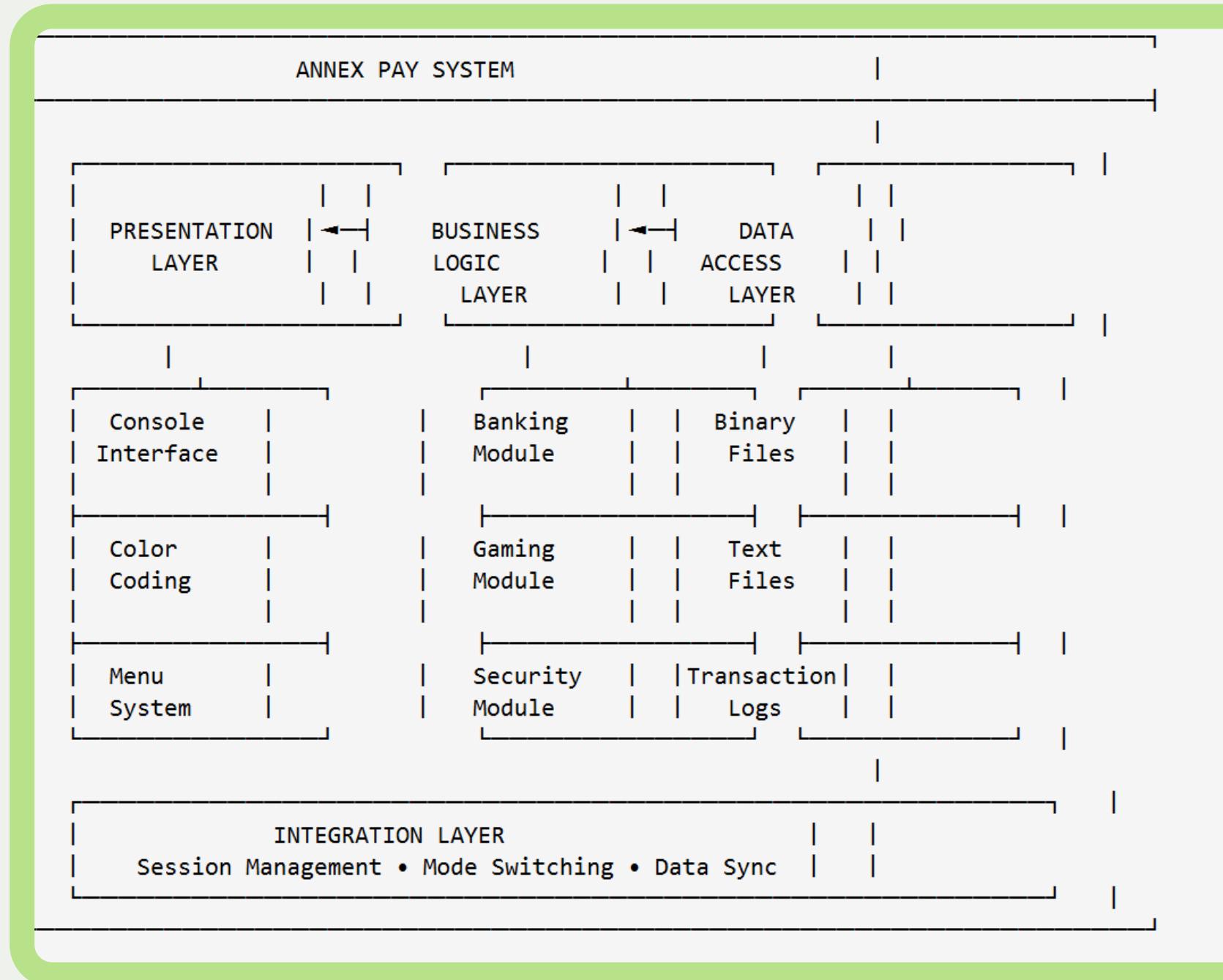
If a User refer to another, they both will get bonuses.



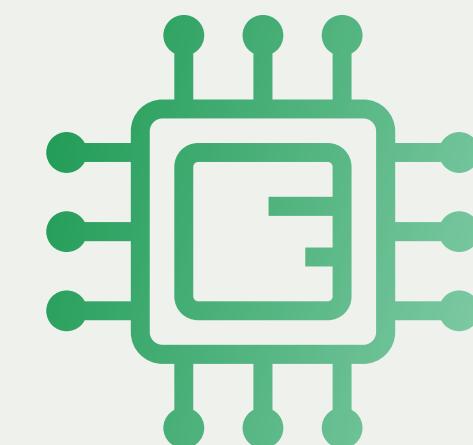
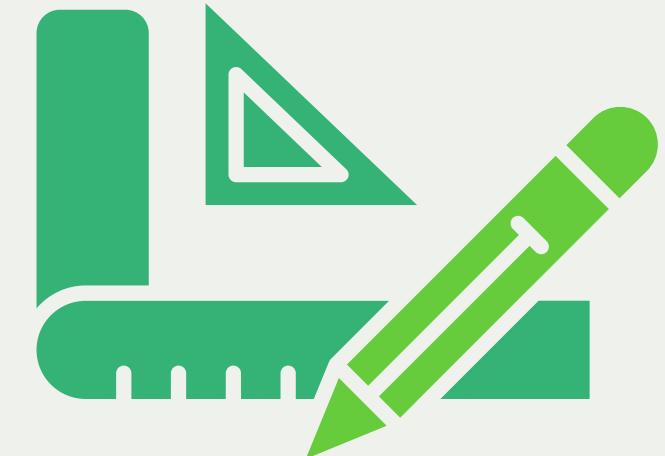
Transaction Search & History

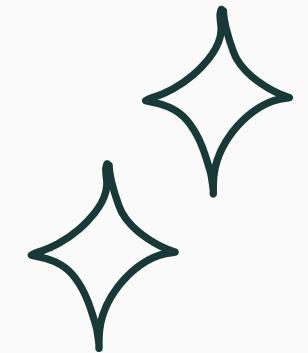
User can search Transaction History by date and amount

TECHNICAL ARCHITECTURE :



- File-based data storage (binary + text)
- Modular programming approach
- Platform-independent code
- Secure input handling with PIN masking
- Error handling & validation





SECURITY FEATURES MULTI-LAYER PROTECTION:

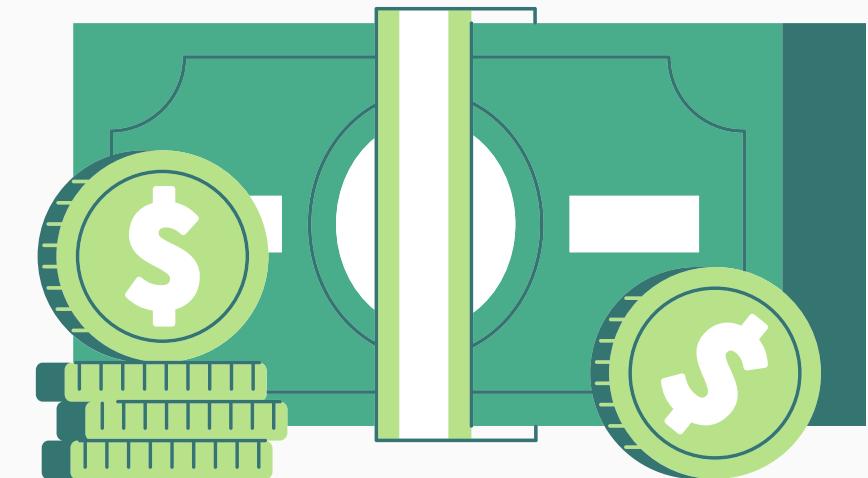


4-digit PIN authentication

Account locking after 3 failed attempts

Security question-based recovery

Daily withdrawal limits
(20,000 BDT)





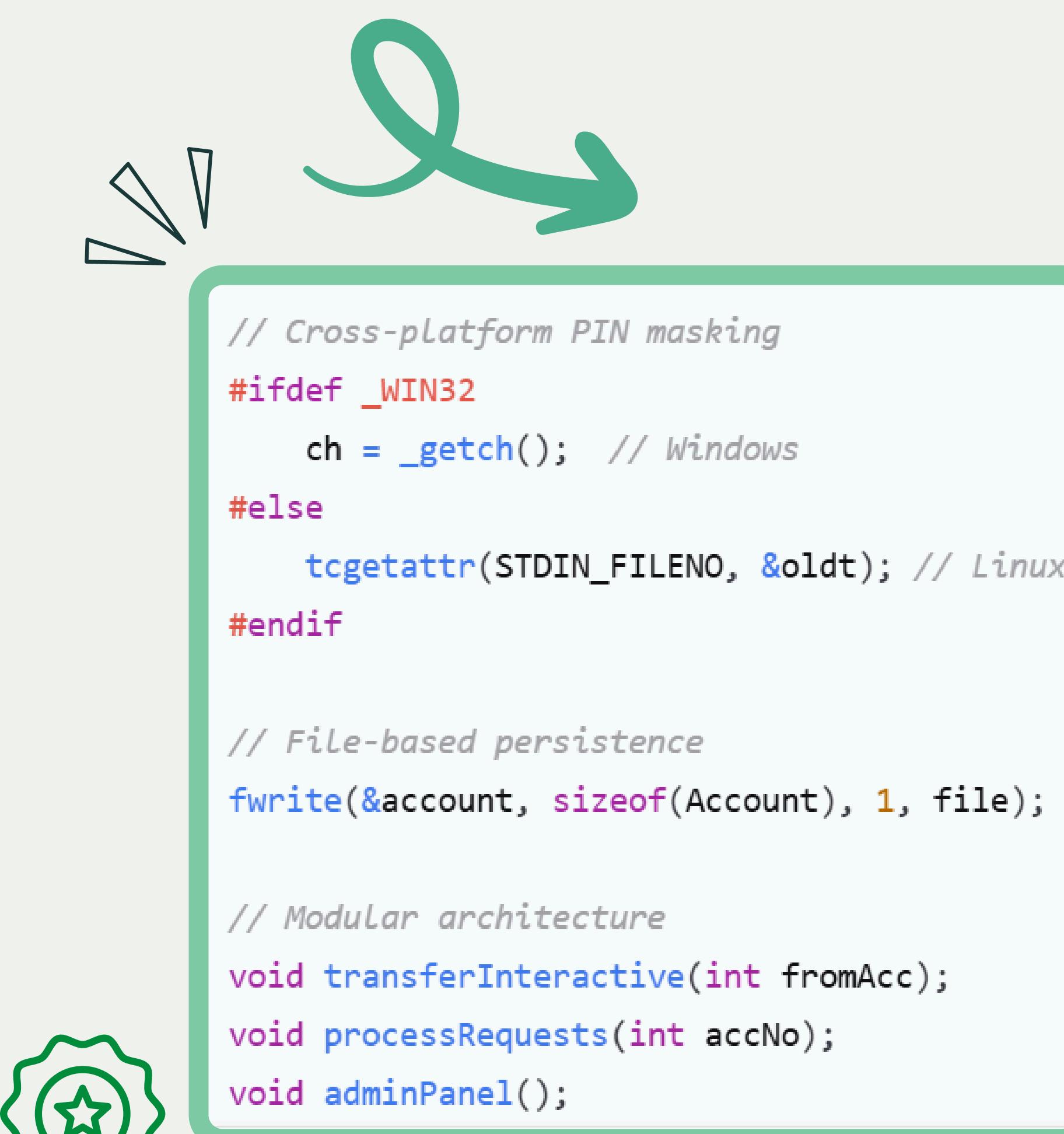
WHAT MAKES ANNEX PAY SPECIAL:

- User Experience - Colorful interface with loading animations

- Referral System - Earn bonuses by inviting friends

- Comprehensive Features - Beyond basic deposit/withdraw

- Admin Control - Complete system management



CODE HIGHLIGHTS TECHNICAL EXCELLENCE:

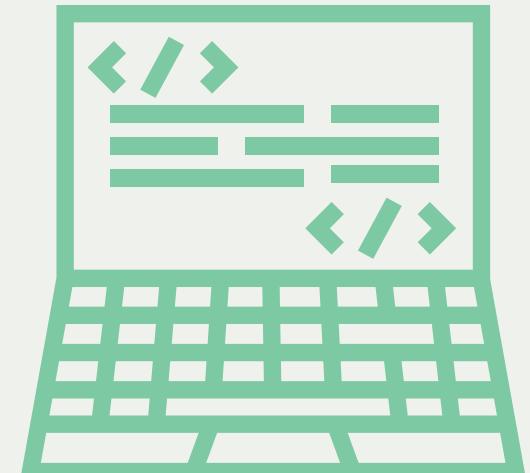
```
// Cross-platform PIN masking
#ifndef _WIN32
    ch = _getch(); // Windows
#else
    tcgetattr(STDIN_FILENO, &oldt); // Linux
#endif

// File-based persistence
fwrite(&account, sizeof(Account), 1,
file);

// Modular architecture
void transferInteractive(int fromAcc);
void processRequests(int accNo);
void adminPanel();
```

```
// File-based persistence
fwrite(&account, sizeof(Account), 1,
file);

// Modular architecture
void transferInteractive(int fromAcc);
void processRequests(int accNo);
void adminPanel();
```





CHALLENGES & SOLUTIONS

TECHNICAL CHALLENGES

Data Persistence → Binary file handling

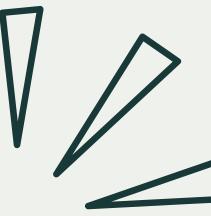
Security → PIN masking & account locking

Cross-platform → Conditional compilation

UI/UX → ANSI colors & loading animations



FUTURE ENHANCEMENTS



WHAT'S NEXT FOR ANNEX PAY:

1. GRAPHICAL USER INTERFACE (GUI)
2. NETWORK CONNECTIVITY FOR MULTI-USER
3. DATABASE INTEGRATION
4. MOBILE APPLICATION
5. ENHANCED SECURITY WITH ENCRYPTION
6. INTEREST CALCULATION & ADVANCED REPORTING



CONCLUSIONS

ANNEX PAY - More Than Just Code

A complete banking solution

Foundation for real financial systems

Demonstration of software engineering principles

$$\frac{x}{a} + \frac{y}{b} = 1$$





THANK YOU!

ANY QUESTIONS?

contact : Team Hash Include