1. Project Title:

Banking Information System

2. Abstract:

The **Banking Information System** is a console-based Java application that allows users to perform basic banking operations such as checking balance, depositing funds, and withdrawing money. This system is designed to simulate real-world banking activities in a simple and interactive manner. It ensures that users can manage their virtual accounts efficiently without needing a complex user interface. The project demonstrates the use of conditional statements, loops, and basic input/output handling in Java.

3. Objectives:

- To develop a simple and user-friendly banking management program.
- To allow users to perform key operations like deposit, withdraw, and balance inquiry.
- To implement input validation for secure and accurate transactions.
- To enhance understanding of Java control structures and user interaction.

4. Description:

The Banking Information System works as an interactive console-based application.
When the program starts, it displays a menu with options:

1. Check Balance
2. Deposit
3. Withdraw
4. Exit

The user can select any option and perform the corresponding operation. The balance updates dynamically after every transaction. The program uses loops and conditional statements to manage user input and provide appropriate feedback (like insufficient funds or invalid entries). It continues running until the user chooses to exit.

5. Tools and Technologies Used:

- **Programming Language:** Java

- **IDE:** IntelliJ IDEA / Eclipse / VS Code (any Java-supported IDE)
- **Compiler:** JDK (Java Development Kit)

6. Expected Outcome:

The system successfully performs banking operations such as balance check, deposit, and withdrawal. It provides a clear understanding of how basic financial transactions can be handled using programming logic.

7. Advantages:

- Easy to understand and implement.
- Demonstrates fundamental Java concepts effectively.
- Provides basic functionality similar to a real banking system.
- Can be expanded to include features like multiple accounts or password protection.

8. Limitations:

- Does not store data permanently (no database).
- Only one user can operate at a time.
- No graphical interface (console-based only).

9. Future Enhancements:

- Add user authentication using usernames and passwords.
- Store account data using files or databases (MySQL/SQLite).
- Implement multiple user accounts.
- Develop a graphical interface using Java Swing or JavaFX.

10. Conclusion:

The **Banking Information System** project successfully demonstrates how core Java programming can be used to simulate real-world systems. It helps beginners understand the practical use of loops, conditionals, and user input handling in software applications. This mini project lays the foundation for more advanced financial applications.