**SBI Banking System - Code Explanation**

﻿**Introduction**

﻿This document provides a detailed explanation of the SBI Banking System Java code. It covers the code structure, the tools and libraries used, their purposes, and why they were chosen.

﻿Code Structure

﻿This is a Java-based console application simulating a basic banking system for "State Bank of India" (SBI).﻿It has two main user roles:﻿Admin: For managing customers and viewing customer details.﻿Customer: For account creation, login, and banking transactions.﻿Tools and Libraries Used:﻿1. Java Collections (Map, HashMap)﻿Tool: Map<Long, Customer> and Map<String, Customer>﻿Purpose: Efficiently store and manage customer details.﻿Why Used: HashMap offers fast access, insertion, and deletion (O(1) average time).﻿2. Serialization (ObjectInputStream, ObjectOutputStream)﻿Tool: Serializable, ObjectOutputStream, ObjectInputStream﻿Purpose: Persist customer data even after the program closes using file storage (bankdata.ser).﻿Why Used: Directly save/load complete objects, ensuring data integrity.﻿3. File Handling (File, FileOutputStream, FileInputStream)﻿Tool: File, FileOutputStream, FileInputStream﻿Purpose: Manage file storage for customer data.﻿Why Used: Simple, efficient file handling in Java.﻿4. User Input (Scanner)﻿Tool: Scanner class﻿Purpose: Collect user inputs for various operations.﻿Why Used: Provides a flexible way to accept different data types.﻿Class Structure:﻿1. Customer Class﻿Represents a customer in the banking system.﻿Implements Serializable for data persistence.﻿Key Attributes:﻿Account Number, Username, Password﻿Name, Aadhaar, Mobile, Address, Nominee﻿Age, ATM details, Cheque Book details, Balance﻿2. SBIBanking Class﻿Main class with all banking methods.﻿Key Methods:﻿main(): Main program loop.﻿adminMenu(), customerMenu(): Separate menus for Admin and Customer.﻿openAccount(): Creates a new customer account.﻿deposit(), withdraw(), fundTransfer(): Handles transactions.﻿saveData(), loadData(): Manages data persistence.﻿Key Method Explanations:﻿1. main() Method﻿Starts the program and displays the main menu.﻿Options include Admin Login, Customer Login, and Account Creation.﻿2. openAccount() Method﻿Takes user details and creates a new Customer object.﻿Generates a unique account number for each customer.﻿3. adminMenu() Method﻿Allows Admin login using credentials (username: Alok, password: Alok123).﻿Admin can view all customer details.﻿4. customerMenu() Method﻿Allows customers to log in using their username and password.﻿Provides options for transactions (Deposit, Withdraw, Transfer, etc.).﻿Why This Structure is Effective﻿HashMap provides fast access to customer data.﻿Serialization ensures data persistence.﻿Separate Admin and Customer menus improve clarity.﻿File handling secures data between sessions.

﻿Conclusion

﻿This SBI Banking System is a robust console-based application. It uses Java Collections, Serialization, and File Handling effectively, providing a scalable and maintainable code structure.