

# Data Structures-Based SACCO Management System

## Project Title:

### Design and Implementation of a Data Structures-Based SACCO Management System for Rural Farmers

## Project Description:

A farmers' SACCO (*Savings and Credit Cooperative Organization*) provides financial services such as **deposits, withdrawals, and account statement retrieval**. Farmers rely on these services to manage their savings and expenditures efficiently. Given the need for efficient data management, you are required to design a **SACCO management system** that utilizes appropriate **data structures** to handle financial transactions effectively.

---

## Problem Statement:

Traditional SACCO management systems rely on databases, but an optimized in-memory approach using **data structures** can improve transaction speed and efficiency. The challenge is to determine the most suitable **data structure(s)** to store and process farmers' transactions, ensuring fast deposits, withdrawals, and statement retrievals while maintaining data integrity and security.

---

## Project Requirements and Focus on Data Structures

Students must develop a **SACCO management system** that performs the following operations using appropriate **data structures**:

### 1. Deposit Money (Efficient Storage and Retrieval)

- Farmers should be able to deposit money into their SACCO accounts.
- Each deposit should store **farmer ID, amount, and timestamp**.
- The data structure should allow **efficient insertion and retrieval of deposits**.
- **Recommended Data Structures:**
  - **Linked List** (for maintaining an ordered list of deposits)
  - **Hash Map** (for quick lookup of a farmer's deposit history)

### 2. Withdraw Money (Efficient Balance Management)

- Farmers can withdraw money if they have sufficient balance.
- The system should check **account balance before withdrawal**.
- Each withdrawal should store **farmer ID, amount, and timestamp**.
- **Recommended Data Structures:**
  - **Hash Maps** (for storing and updating farmer balances in constant time)
  - **Stack** (for retrieving the most recent transactions first)

### 3. Retrieve Account Statement (Reverse Order Retrieval)

- Farmers should be able to request a **statement of their last N transactions**.
- The system should return transactions in **reverse chronological order** (most recent first).
- Farmers can filter statements by **date range**.
- **Recommended Data Structures:**
  - **Stack** (Last-In-First-Out for quick retrieval of recent transactions)
  - **Queue** (First-In-First-Out if statements need to be processed in order)

### 4. Check Balance

- Retrieve the farmer's balance from the hash map and display it.

---

### Project Objectives:

1. **Analyze different data structures** (linked lists, stacks, queues, hash maps, trees) to determine their efficiency in managing SACCO transactions.
2. Implement a **data structure-based transaction system** that supports deposits, withdrawals, and statement retrieval efficiently.
3. Optimize retrieval speed for **recent transactions** using appropriate **in-memory data structures**.
4. Compare and contrast the performance of different data structures in storing and retrieving SACCO transactions.

---

### Expected Deliverables:

- A **functional system** that efficiently manages SACCO deposits, withdrawals, and statements using **well-chosen data structures**.
- A **technical report** analyzing the efficiency of different data structures in handling financial transactions.
- **Performance benchmarks** comparing the response times of different data structures in retrieving SACCO transactions.

- A **graphical user interface (GUI)** for user interaction.
- 

#### **Evaluation Criteria:**

1. **Correctness:** Does the system correctly process deposits, withdrawals, and statements?
  2. **Efficiency:** Are the chosen data structures optimal for each operation?
  3. **Scalability:** Can the system handle thousands of farmers and transactions efficiently?
  4. **Code Quality:** Is the implementation structured, well-documented, and maintainable?
  5. **Performance Analysis:** Is there a clear comparison of different data structures in handling transactions?
- 

#### **Question:**

Design and implement a SACCO Management System for farmers that utilizes efficient data structures to handle deposits, withdrawals, and statement retrieval. Your system should analyze and compare the efficiency of different data structures (linked lists, stacks, queues, hash maps etc) in managing financial transactions. Provide a detailed performance evaluation and justify the choice of data structures used.