**189769**

**192217**

**190503**

**190051**

**190500**

**Project Idea Definition**

**Detailed Explanation:**  
Microfinance institutions (MFIs) are crucial for providing financial services (loans, savings, insurance) to underserved communities, especially in developing countries like Kenya. This project aims to develop a database system to improve the management of microfinance services. The database system will facilitate the tracking of client loans, repayments, savings, and eligibility for new loans, ensuring transparency, accountability, and efficient operations.

The database will support various stakeholders, including microfinance clients, staff, management, and regulatory bodies, providing them with reliable data for decision-making, monitoring, and compliance.

**Scope and Objectives:**

* **Scope:**  
  The database will manage clients' personal information, loan issuance, repayment schedules, and savings data. It will also support financial reporting for the microfinance institution’s management.
* **Objectives:**
  1. Centralize client data and transactions for efficient management.
  2. Track loan issuance, repayments, and outstanding balances.
  3. Automate eligibility criteria and loan status monitoring.
  4. Ensure financial transparency with accurate reporting.
  5. Enhance operational efficiency by reducing manual work.

**Stakeholders:**

* **Clients:** Individuals or groups who benefit from the microfinance services.
* **Microfinance Staff:** Personnel who manage loans, repayments, and client interactions.
* **Management:** Decision-makers who need to assess the performance of loans and the financial health of the institution.
* **Regulatory Bodies:** Government agencies overseeing financial institutions to ensure compliance.

**2. Entity-Relationship Diagram (ERD)**

The ERD visually represents the data entities and their relationships. The key entities in the microfinance database include:

1. **Clients**
   * ClientID (PK), Name, DateOfBirth, Gender, PhoneNumber, Address, IDNumber.
2. **Loans**
   * LoanID (PK), LoanAmount, InterestRate, LoanIssueDate, LoanDueDate, ClientID (FK), LoanStatus.
3. **Repayments**
   * RepaymentID (PK), RepaymentDate, RepaymentAmount, LoanID (FK), ClientID (FK).
4. **Savings**
   * SavingsID (PK), ClientID (FK), DepositAmount, DepositDate.
5. **LoanEligibility**
   * EligibilityID (PK), ClientID (FK), LoanLimit, Income, CreditHistory.

The relationships:

* One **Client** can have multiple **Loans** (1

).

* One **Loan** can have multiple **Repayments** (1

).

* One **Client** can have multiple **Savings** (1

).

* Each **Client** has one **LoanEligibility** record (1:1).

**3. Database Schema**

The schema is designed to follow the **Third Normal Form (3NF)** to reduce data redundancy and ensure data integrity. The tables include:

1. **Clients**: Stores client information.
2. **Loans**: Tracks loan details, including amounts, issue dates, and statuses.
3. **Repayments**: Stores repayment records tied to loans and clients.
4. **Savings**: Keeps a record of client savings deposits.
5. **LoanEligibility**: Stores loan eligibility data for each client.

**4. SQL Code Implementation**

* **Database Creation**:

sql

Copy code

CREATE DATABASE MicrofinanceDB;

USE MicrofinanceDB;

CREATE TABLE Clients (

ClientID INT AUTO\_INCREMENT PRIMARY KEY,

Name VARCHAR(100),

DateOfBirth DATE,

Gender CHAR(1),

PhoneNumber VARCHAR(15),

Address TEXT,

IDNumber VARCHAR(20) UNIQUE

);

CREATE TABLE Loans (

LoanID INT AUTO\_INCREMENT PRIMARY KEY,

LoanAmount DECIMAL(10, 2),

InterestRate DECIMAL(5, 2),

LoanIssueDate DATE,

LoanDueDate DATE,

ClientID INT,

LoanStatus ENUM('Pending', 'Active', 'Paid', 'Defaulted'),

FOREIGN KEY (ClientID) REFERENCES Clients(ClientID)

);

CREATE TABLE Repayments (

RepaymentID INT AUTO\_INCREMENT PRIMARY KEY,

RepaymentDate DATE,

RepaymentAmount DECIMAL(10, 2),

LoanID INT,

ClientID INT,

FOREIGN KEY (LoanID) REFERENCES Loans(LoanID),

FOREIGN KEY (ClientID) REFERENCES Clients(ClientID)

);

CREATE TABLE Savings (

SavingsID INT AUTO\_INCREMENT PRIMARY KEY,

ClientID INT,

DepositAmount DECIMAL(10, 2),

DepositDate DATE,

FOREIGN KEY (ClientID) REFERENCES Clients(ClientID)

);

CREATE TABLE LoanEligibility (

EligibilityID INT AUTO\_INCREMENT PRIMARY KEY,

ClientID INT,

LoanLimit DECIMAL(10, 2),

Income DECIMAL(10, 2),

CreditHistory TEXT,

FOREIGN KEY (ClientID) REFERENCES Clients(ClientID)

);

* **CRUD Operations** (Examples):
* Insert new client:

sql

Copy code

INSERT INTO Clients (Name, DateOfBirth, Gender, PhoneNumber, Address, IDNumber)

VALUES ('John Doe', '1985-05-15', 'M', '0700123456', '1234 Nairobi St.', 'A1234567');

* Query all loans for a specific client:

sql

Copy code

SELECT \* FROM Loans WHERE ClientID = 1;

**5. GitHub Collaboration**

* **Repository Setup**:  
  Create a GitHub repository to track the progress of the project, with the following structure:
  + /SQL: For SQL scripts.
  + /ERD: For the Entity-Relationship Diagram.
  + /Documentation: For project reports and instructions.
* **Version Control**:  
  Use Git for version control, ensuring frequent commits with descriptive messages. Branches should be used to manage different features or tasks.