Assignment Overview

Title: Microfinance SACCO Admin Panel – Member & Loan Management System

Context

You are building a minimal **Microfinance SACCO Admin System** for a local Sacco. In this system:

- Microfinance SACCOs are created and managed by an admin.
- Members are registered under a specific microfinance.
- Admins can activate members and apply for loans on their behalf.
- All actions are performed by the admin — **there is no member login or separate portal**.

You are required to build both the backend and a simple frontend for this system.

Core Functional Requirements

1. Microfinance Creation

- **Fields**: You decide.
- Admin can create multiple microfinance SACCOs.
- Each microfinance will have members registered under it.

---

2. Member Registration

- **Fields**: Decide
- Members are created in `Pending` status.

- A member belongs to a microfinance.

---

3. Member Activation

- Admin can change a member's status from `Pending` to `Active`.

---

4. Loan Application (Admin Applies on Behalf of Member)

- A member can only receive a loan if:
    - They are Active
    - They do not already have a **pending** loan application
- **Fields**: Member, Loan Type (e.g., Emergency, Development), Amount, Repayment Period (months)
- Loan status defaults to `Pending` when submitted.

---

5. View Loan Applications

- Admin can view a list of all loan applications.
- Display for each loan:
    - Member Name
    - Microfinance Name
    - Loan Type
    - Amount
    - Repayment Period

- Status (`Pending`, `Approved`, `Rejected`)

UI Requirements

Use any front-end stack (recommended: Angular + Bootstrap).

The system should include these screens:

1. Microfinance Management

- Form to create microfinance SACCOs
- List of existing microfinances

2. Member Management

- Form to register a member under a microfinance
- List of members
    - Include action buttons to change status

3. Loan Management

- Form to apply for a loan
- List of loan applications.

Backend Requirements

Preferred Stack: Laravel (RESTful API)

Use modular code structure, environment variables, validation, and follow Laravel best practices.

API Endpoints

Method	Endpoint	Description
POST	`/microfinances`	Create a new microfinance
GET	`/microfinances`	List all microfinances
POST	`/members`	Register a new member
GET	`/members`	List members
PUT	`/members/activate/{id}`	Activate a member
POST	`/loans`	Apply for a loan
GET	`/loans`	List all loan applications

Database Schema (MySQL)

Deliverables

- **GitHub Repository** with:
    - Clean project structure
    - Setup instructions
- **README.md** including:
    - Overview of the project
    - How to set up and run backend & frontend locally
    - Any decisions or trade-offs made
- **SQL File or Laravel Migrations** to set up the database schema

Evaluation Criteria

- Correctness (does it meet the feature requirements?)
- Code quality (naming, structure, separation of concerns)
- Frontend simplicity and usability
- Database design and relational logic
- Clean commits and documentation

Timeline

- **Assignment duration**: 3 Days
- Submit your GitHub link + any setup instructions in the README

Bonus (Optional)

- Input validation (no empty/invalid data)
- Using our tech stack is a plus(angular, bootstrap, laravel, mysql)
- Pagination on loan/member lists
- UI enhancements using Bootstrap
- Elegant error messages or loading indicators