Society Maintenance- Fees Management

Objective:

Design the database schema and REST API endpoints which will facilitate the end user to input the maintenance fee for each member.

Problem Definition:

Create an application for an Society that will help in maintaining the members fees ledger.

Maintenance fee are collected on **monthly** intervals.

If a member fails to pay the Maintenance fee on or before the 10th of every month. Interest of 1.5% will be added on the pending maintenance fees.

Monthly Maintenance Fees Structure:

1) 1 BHK flat has to pay monthly fee of 500 INR
2) 2 BHK flat has to pay monthly fee of 700 INR
3) 3 BHK flat has to pay monthly fee of 1000 INR
4) Row Villa has to pay monthly fee of 2000 INR
5) Shop has to pay monthly fee of 400 INR

If a premise is given on rent then additional charge of 500 INR will be added to the fees.

If an owner pays the advance rent for 11 months between 1st of April to 30th April, then a discount of 1 month waiver of maintenance fees will be applied.

Note: Each premise (flat/Villa/Shop) should be assigned a unique number, using which it can be easily recognized what type of premise it is.

Example:

F-01 → **F** stands for First Floor and flat no 1 on that floor
S-02 → **S** stands for Second Floor and flat no 2 on that floor
FO-05 → **FO** stands for Fourth Floor and flat no 5 on that floor

Expected API Endpoints:

1) API endpoint to register a new premise.
2) API endpoint to register a new member.
3) API endpoint to link owner to a premise.
4) API endpoint to link a renter to a premise
5) API endpoint to get the list of all members and the type of the premises they own.
6) API endpoint to get the list of all renters in the society
7) API endpoint to get the pending dues for a given member
8) API endpoint to get the pending dues for all members
9) API endpoint to pay dues by a given member

NOTE:

- **Work on Each task individually, don't try to complete all the tasks together.**
- **You are advised to commit your changes once each task is complete and push to GitHub.**
  While committing your changes put proper commit messages.
  Example: If you are working on "Creating the CRUD Endpoints for Maintenance fees " and you have completed writing the code for adding user. Then the commit message should be "TASK-CRUD Maintenance Fees::Wrote end point to Create fees"
  Format is shown below:
  "TASK-TASK-NAME>>::<<COMMIT_MSG>>"

Expected output:

- All the expected API Endpoints should be documented in the Postman Collection.
- Invoking the Postman API endpoint should give the expected results.

Tech Stack:

The following tech stack should be used to implement the task.

- Framework : NodeJS/NestJS
- Database : MongoDB/mySQL/Postgres
- API Documentation : Postman