Full Stack Engineer Coding Challenge

Context

Landbay is a peer-to-peer #FinTech company that invests lender's money into mortgages.
In order to lower the risks, lender's money is spread across several mortgages.

Goal

The purpose of the exercise is to develop a small application that creates a mortgage and diversify the lenders' investment. The data model is rather simple and is composed of a mortgage containing basic mortgage information, and mortgage parts representing the diversification of lenders' investments and allowing to calculate interests.
Please provide any assumptions that you have made as part of developing this.

Step 1 - Loan management

We would like to manage loans in a REST-based service. No UI is required. A loan (or mortgage) can be set up. A lender can subscribe to part of a loan.

Desired REST operations:

- Create a loan
- Get a loan and its investments
- Delete a loan
- Lender investment into a loan (create)

For now, no need to support removing investments, neither are updates required. A investment is perpetual.

Step 2 - Interest calculation

Assuming all investments are valid for 1 month. Run interest calculation on the investment. The interest rate can be stored against a loan. We want a service that outputs interest owed to each lender. Interest does not need to be re-invested.

Step 3 - Investments over multiple time periods

All investments now have a different start and end date, granular to the day. The end service will accept a date range and calculate the interest owed over that period.

Step 4 - Package the application

Package the application in a Docker image.

Other information

The candidate should go as far as he can, depending on the time he can dedicate to this exercise. It is also an opportunity to demonstrate what good production-ready code looks like.
Making further assumptions and/or not implementing all the possible cases is perfectly acceptable, but this should be documented.

As a result of the exercise, we expect the following:

- The code is written in Java using the Spring framework
- You are free to use any other libraries or technologies that you like, so long as we are able to run the system in a container.
- Usage of Maven

- Code is made available in GitHub
- Development best practices are followed
- A release note explaining how to deploy and run the application
- No data needs to be populated on start-up. (i.e. a blank database is fine).