

Full Stack Assignment

Your task is to build a mini-loan APP:

It is an app that allows authenticated users to go through a loan application. It doesn't have to contain too many fields, but at least "amount required" and "loan term." All the loans will be assumed to have a "weekly" repayment frequency.

After the loan is approved, the user must be able to submit the weekly loan repayments. It can be a simple repayment functionality, which won't need to check if the dates are correct but will just set the weekly amount to be repaid.

Actions to implement:

1) Customers can create a loan:

Customer submit a loan request defining amount and term

example:

Request **amount** of **10000 \$** with **term 3** on date **7th Feb 2022**

- He will generate 3 scheduled repayments:
 - i. **14th Feb 2023** with amount **3.333,33 \$**
 - ii. **21st Feb 2023** with amount **3.333,33 \$**
 - iii. **28th Feb 2023** with amount **3.333,34 \$**
- The loan and scheduled repayments will have state **PENDING**.

2) Admins can approve the loan:

Admin change the **PENDING** loans to state **APPROVED**.

3) Customers can view loan belonging to him:

Add a policy check to make sure that the customers can view their own loan only.

4) Customers add repayments:

- Customers add a repayment with an amount greater or equal to the scheduled repayment.
- The scheduled repayment will change the status of repayment to **PAID**.
- If all the scheduled repayments connected to a loan are **PAID**, the loan becomes **PAID**

automatically.

Technical Requirements Must have

- You can use any language that you prefer to code the solution.
- Build a fully functional REST API with minimal frontend.

Definition of **Minimal Frontend**: -

- Use HTML5 forms with form validation.
- The forms should handle any error message generated from the backend.
- Use of any loader state to show app loading.
- Different pages/routes for different functionality.

- [README.md](#) should contain all the information that the reviewers need to **run and use** the app.
- Write code with your teammates in mind: apply the standard code style, readable, easy to review & understand.
- Submit any one of the following along with the codebase,
 1. A video showcasing the app's functionalities.
 2. A deployed version of the codebase. So that anyone can try running it.

Nice-to-have:

- Include brief documentation for the project: the choices you made and why.
- Use any frontend framework like React, Angular, Svelte or Vue.
- Use any CSS framework like Material UI, Tailwind, or Bootstrap.
- Unit and feature tests for frontend and backend.
- Application should be responsive. It should work in both App and Mobile.
- Script to install the app in one go (any tool)
- Postman collection/open API document for the API
- Clean application architecture / design patterns
- Setup CI/CD pipeline using GitHub Actions.

Notes:

- We're **not only** taking care a lot of quality **but also** encourage any of your creative ideas to make the project better, so don't hesitate to try anything. (In case, you still have more ideas and need more time, it's okay to ask for delaying sometime, **we'd like to see all your best**)

The complete project should be shared with us as a public **GitHub/Bitbucket/GitLab** repo.

Happy coding!