Greystone Labs

The test of a strong engineer isn’t how much they know now, but how efficiently they can learn what they need and apply to solve a problem.

At the lab, we look for the problems we have no idea how to solve. That’s part of the thrill. We might be working on a new mobile app, a real-estate underwriting algorithm, financial fraud detection - we may even write an excel prank in VBA.

It’s our job to explore new areas. It’s our job to execute.

This code challenge will involve learning something new, quickly. Use documentation, use stackoverflow, ask ChatGPT - use whatever you need to get the job done.

Just be careful copy and pasting answers. If you don’t understand how something works, you’ll have difficulty improving|fixing it.

Start a new github repo for this project. Part of this assessment includes your usage of git.

In this challenge, you will create a REST API for a Loan Amortization app using the python miniframework FastAPI.

The API should allow a client to,

- Create a user
- Create loan
- Fetch loan schedule
- Fetch loan summary for a specific month

- Fetch all loans for a user
- Share loan with another user

A loan record should at least contain the following fields:

- Amount
- Annual Interest Rate
- Loan Term in months

The loan schedule endpoint should return an array of length loan_term, consisting of:

```
{
  Month: n
  Remaining balance: $xxxx,
  Monthly payment: $xxx
}
```

The loan summary endpoint should accept a month number as a parameter and return:

- Current principal balance at given month
- The aggregate amount of principal already paid
- The aggregate amount of interest already paid

To calculate the above, you will have to code a function that generates an amortization schedule. There are plenty of libraries that can do this for you, but for this challenge, please develop the function yourself. You may use general libraries, such as numpy, if you would like.

We'll be scoring you based on,

- Proper HTTP methods
- Error handling and validation

- API structuring
- Calculation Accuracy
- Git commits

The following items are optional:

- Use pytests or other test framework to test your endpoints and financial calculations
- Use an ORM + SQL DB to save users and loans. We'd recommend using SQLAlchemy or SQLModel, with an in-memory instance of SQLite - but that is up to you.

When you are finished, please share your repo here:

Repo Submission

I appreciate you taking the time to complete this challenge.

Thank you.