**ASPIRE-API**

**Introduction**: This project contains APIs for performing some basic loan scheduling functionality. We can create a loan with scheduling, view the loans and payments, approve the loan request by admin and Process the payment of scheduled loan.

**Technologies**: I have used the following technologies to perform the mentioned functionality

* PHP as the backend programming language.
* Laravel as the backend Framework
* MySql as the Database.
* I have also used Laravel Sanctum library for API authentication

**Database Tables:** These are the main tables used(Sample db already added).

* Users => This table is used to maintain the user list
* loans => This table is store and maintain the record of user loan request and its status
* loan\_payments => This table relates to the “Loans” table and maintain the scheduled payments for loan
* personal\_access\_tokens => This stores the token for authenticated Users

**APIS:**

1. **Login:** This API is used to obtain the access token to be passed for subsequent request. We provide the credentials in the Request after successfully authenticating, the application issues an access token with expiration for every particular user. The user gets authenticated using the token to be passed in the header of next request as bearer token. The Laravel assess the token in middle of request [Middleware] and do the needful with request if token matches with any record.
2. **CreateRequest:** This API is used to insert the loan request with amount and term. Also the loan breakage the done in this API and scheduled payments are created.
3. **ApproveRequest:** This API is used to approve the loan by Admin.
4. **GetLoans:** This API is used to get the list all the Loans for a particular user.
5. **PayLoan:** This API is used pay the amount by user. The amount should be greater than the scheduled payment amount.

**Notes:**

* Middleware are used to handle authorization and authentication for user and admin.
* Method description are added to each method
* Error handling is applied to handle the exceptions.