Project Overview

Project Title: The First MicroFinance Bank – Loan Management System

Goals:

Secure multi-role login for staff

Efficient data entry & tracking of small business loan applicants

Organized customer management by loan status

Staff zone access based on zip code

Printable customer records

View repayment schedule for agreed loans

Tech Stack:

Frontend: HTML, CSS, JavaScript (with print CSS)

Backend: Laravel (PHP)

Database: MySQL

Printing: JS + CSS @media print

💼 Laravel Project Structure

```
/microfinance-bank
├── app/
|   ├── Http/
|   |   ├── Controllers/
|   |   ├── Middleware/
|   |   ├── Requests/
|   ├── Models/
|   ├── Services/
├── database/
|   ├── migrations/
|   ├── seeders/
├── resources/
|   ├── views/
|   ├── css/, js/
├── routes/
|   └── web.php
├── config/
|   └── auth.php
```

📅 Timeline & Detailed Tasks

Week 1

● Set up Laravel project & .env database

● Scaffold authentication & multi-role support

● Create staff and customers tables with migrations + seeders

● Implement login with role-based redirects

Week 2

● Design receptionist form to register customers

● Add loan_amount and repayment_schedule to customers table via migration

● Build reception form validation via FormRequests

Week 3

● Construct Loan Officer dashboard

● Filter customers by zone and status

● Display “Pay Schedule” button for agreed customers

Week 4

● Create Customer detail view with status update and print layout

● Implement status transitions and service logic

● Build route/controller/view for Pay Schedule page

Week 5

● Design schedule.blade.php, dynamically render repayment JSON data

● Add “Pay Schedule” button in officer dashboard

● Adjust print CSS styling to include new fields

Week 6

● Write feature tests for authentication, form submission, schedule access

● Final testing, bug fixes, and documentation

● Prepare deployment and go-live checklist

🔧 *Feature Breakdown*

1. Authentication & Roles

Laravel's auth with staff roles (Receptionist & Loan Officer)

2. Migration & Data Model

customers table fields: basic info + loan_amount (DECIMAL), repayment_schedule (TEXT JSON)

3. Receptionist

Blade form for customer details, including loan fields

Validation via FormRequest

4. Loan Officer Dashboard

Filter by assigned zip and by loan status

Display "Pay Schedule" button next to status = agreed

5. Customer Detail Page

View and update status

Print with clean layout (JS + @media print)

6. Pay Schedule Page (New Feature)

Decode repayment_schedule JSON

Dynamic Blade table rendering schedule

Access only for officer with assigned customer via policy

7. Printing

Include loan and schedule info in printable copy

Hide buttons using print CSS

8. Security & Best Practices

Password hashing, prepared statements (Eloquent)

Role-based middleware and policies

Use FormRequests, Service classes, MVC separation

9. Testing & Deployment

PHPUnit or Pest feature tests

Document routes, processes, usage

Finalize deployment checklist

Optional Enhancements

Add email notifications for status changes

Export to PDF with packages like DomPDF

Integrate search/filtering on schedule page with Livewire or JS

Use Laravel Excel for data exports

Summary

The project is now mapped end-to-end in Laravel:

Multi-role auth and data model with loan fields

Receptionist and Officer workflows

Customer detail, status updates, and print layout

Repayment schedule view with button and protected route

Secure, maintainable architecture and solid testing plan