Loan Management System API Documentation

📌 Project Overview

The Loan Management System is a RESTful API built using Django REST Framework (DRF) that allows users to apply for loans, track their repayment schedules, and foreclose loans early. The system also includes role-based access control, enabling admins to manage loans efficiently.

(If you need to see the interface and enter the data using HTML use this link: https://loan-management-api-uujj.onrender.com/api/loan_app/ .)

🔹 Key Features:

✅ User Authentication (OTP-based email verification, JWT-based authentication)

✅ Loan Management (Apply for loans, view loan details, track repayment schedules)

✅ Loan Foreclosure (Users can pay off loans early with recalculated interest)

✅ Admin Panel (Admins can view and delete loans)

✅ Secure API (Proper validation, access control, and security mechanisms implemented)

✅ Deployment Ready (Optimized for Render Free Tier)

📌 Authentication & Security

1. User Registration with OTP Verification

🔹 Endpoint: POST /api/auth/register/

🔹 Description: Users register with an email and receive an OTP for verification.

🔹 Request Body:

```
{
  "email": "user@example.com",
  "password": "securepassword"
}
```

🔹 Response:

```
{
  "success": true,
  "message": "OTP sent to email."
}
```

2 OTP Verification

- Endpoint: POST /api/auth/verify-otp/
- Description: Verifies the OTP and activates the user account.
- Request Body:

```
{
  "email": "user@example.com",
  "otp": "123456"
}
```

- Response:

```
{
  "success": true,
  "message": "User verified successfully."
}
```

3 User Login (JWT Authentication)

- Endpoint: POST /api/auth/login/
- Description: Users log in with their email and password.
- Request Body:

```
{
  "email": "user@example.com",
  "password": "securepassword"
}
```

- Response:

```
{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI..."
}
```

4 Logout

- Endpoint: POST /api/auth/logout/
- Description: Logs out the user and invalidates the token.

📌 Loan Management APIs

1 Apply for a Loan

- 🔹 Endpoint: POST /api/loans/
- 🔹 Description: Users can apply for a loan by providing loan details.
- 🔹 Request Body:

```
{
  "amount": 10000,
  "tenure": 12,
  "interest_rate": 10.5
}
```

- 🔹 Response:

```
{
  "status": "success",
  "data": {
    "loan_id": "LOAN001",
    "monthly_installment": 879.16,
    "total_amount": 11549.92,
    "status": "ACTIVE"
  }
}
```

2 Get All Loans

- 🔹 Endpoint: GET /api/loans/

- Response:

```
{
   "status": "success",
   "data": {
      "loans": [
         {
            "loan_id": "LOAN001",
            "amount": 10000,
            "tenure": 12,
            "monthly_installment": 879.16,
            "total_amount": 11549.92,
            "amount_paid": 1758.32,
            "amount_remaining": 9791.60,
            "next_due_date": "2024-04-24",
            "status": "ACTIVE",
            "created_at": "2024-02-24T10:30:00Z"
         }
      ]
   }
}
```

3 Get Loan by ID

- Endpoint: GET /api/loans/{loan_id}/
- Response:

```
{
   "status": "success",
   "data": {
      "loan_id": "LOAN001",
      "amount": 10000,
      "monthly_installment": 879.16,
      "status": "ACTIVE"
   }
}
```

4 Get Loan Payment Schedule

- Endpoint: GET /api/loans/{loan_id}/schedule/
- Response:

```
{
   "status": "success",
   "schedule": [
      {
         "installment_number": 1,
         "due_date": "2024-04-24",
         "principal_component": 820.83,
         "interest_component": 58.33,
         "remaining_balance": 9179.17
      }
   ]
}
```

5 Loan Foreclosure (Early Loan Closure)

- Endpoint: POST /api/loans/{loan_id}/foreclose/
- Description: Users can close loans early with a foreclosure discount.
- Response:

```
{
   "status": "success",
   "final_settlement_amount": 9600.00,
   "discount_applied": 200.00,
   "new_status": "CLOSED"
}
```

📌 Admin APIs

1 Get All Loans (Admin)

- Endpoint: GET /api/admin/loans/
- Response: Admin can view all loans.

2 Delete Loan (Admin)

- Endpoint: DELETE /api/admin/loans/{loan_id}/
- Response:

```
{
  "success": true,
  "message": "Loan deleted successfully."
}
```

📌 Deployment Guide

1 install Dependencies

```
pip install -r requirements.txt
```

2 Run Migrations

```
python manage.py migrate
```

3 Create Superuser (Admin Access)

```
python manage.py createsuperuser
```

4 Start the Server

```
python manage.py runserver
```

5 Deploy on Render

- Push code to GitHub
- Connect Render to GitHub repository
- Set environment variables (DATABASE_URL, SECRET_KEY, etc.)
- Deploy & test APIs

📌 Security Considerations

✅ JWT-based authentication for secure access

✅ Role-based access control (Admin/User)

✅ Input validation to prevent SQL injection & XSS attacks

✅ Proper error handling & exception management

Conclusion

This API provides a robust loan management system with secure authentication, loan processing, and admin controls.