NGO Registration and Donation Management System

1. Introduction

Non-Governmental Organizations (NGOs) play a vital role in addressing social, environmental, and humanitarian issues. With the increasing use of digital platforms, NGOs often conduct online campaigns to register supporters and collect donations. However, many existing systems tightly couple the registration and payment process. As a result, if a user fails to complete a donation, their registration data is often lost. Additionally, administrators lack transparent mechanisms to track registrations, donation attempts, and payment outcomes.

This project focuses on the development of a **backend-driven NGO Registration and Donation Management System** that clearly separates user registration from the donation flow. The system ensures **data integrity, transparency, and ethical handling of payments**, allowing both users and administrators to interact with the platform reliably and secure.

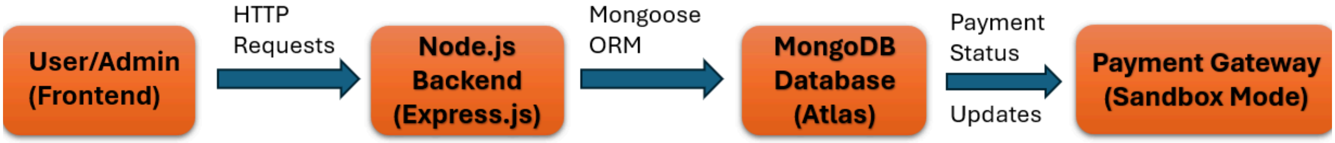
2. Objective

The main objectives of this project are:

- To design a secure system where users can **register independently of donations**
- To ensure **user data is preserved**, even if a donation attempt fails or remains pending.
- To allow users to optionally donate any amount
- To track **all donation attempts with proper status handling**
- To provide administrators with accurate insights into registrations and donations
- To maintain ethical payment practices by avoiding any fake or forced payment success logic.

3. System Overview

The system follows a **client–server architecture** where the backend handles authentication, business logic, database operations, and payment status tracking. The frontend communicates with the backend using RESTful APIs.



4. Functional Requirements

4.1 Authentication

A common **login and registration system** for both users and administrators. Role-based access control using JWT (JSON Web Tokens). Automatic redirection based on role:

- User → User Dashboard
- Admin → Admin Dashboard

4.2 User Side Requirements

Donation Flow:-

- Users can donate any amount

- Each donation attempt is recorded in the database
- Donation status is maintained as:
  - **Pending**
  - **Success**
  - **Failed**

User Access:-

- Users can view their registration details
- Users can view donation history
- Users can track the current status of each donation

4.3 Admin Side Requirements

Admin Dashboard

- View total registered users
- View total donations received
- View aggregated donation amounts

Registration Management

- View all registered users
- Filter users based on basic parameters
- Export registration data (future enhancement)

Donation Management

- View all donation records
- Track payment status and timestamps
- Monitor pending, successful, and failed transactions

5. Database Design

The system uses **MongoDB** as the database, with **Mongoose** as the ODM.

User Schema

Field Name	Description
Name	User’s full name
Email	Unique email address
Password	Hashed password
Role	USER or ADMIN
CreatedAt	Registration timestamp

6. Data and Payment Handling Rules

The system strictly follows ethical and transparent data handling principles:

- User registration data is stored **independently** of donation completion
- Donation success is marked **only after genuine payment confirmation**

- Failed and pending payments are clearly recorded
- No fake or forced payment success logic is implemented
- All donation attempts remain auditable for administrative review

7. Technology Stack

Layer	Technology
Backend	Node.js, Express.js
Database	Mongo DB Atlas
Authentication	JWT, bcrypt
ORM	Mongoose
Frontend	HTML, JavaScript
Payment Gateway	Sandbox/Test Mode
Version Control	GitHub

8. Payment Gateway Integration

The system is designed to support payment gateway integration in sandbox/test mode. While real payment APIs are not mandatory, the backend logic ensures that:

- Donation records are created with `PENDING` status
- Payment success updates are applied only after verification
- Failed or incomplete transactions are properly recorded

This approach satisfies the evaluation criteria while maintaining ethical payment practices.

9. Key Design Decisions and Assumptions

Design Decisions

- Separation of registration and donation flows
- Role-based access control using JWT
- Use of MongoDB Atlas for scalability and reliability
- Express.js (v4) for stable backend routing

Assumptions

- Users have internet access to complete registrations
- Admin accounts are created manually or seeded
- Sandbox payment gateway responses simulate real-world behavior

10. Conclusion

The NGO Registration and Donation Management System successfully addresses the limitations of traditional donation platforms by separating registration from payment processing. The system ensures data integrity, transparency, and ethical payment handling while providing administrators with clear visibility into registrations and donation activities. This project demonstrates a robust backend-driven approach suitable for real-world NGO applications.