🏛 NSS IITR Portal

NGO Registration & Donation Management System — Technical Report

1. Project Overview

A full-stack web application enabling secure member registration and online donations for the National Service Scheme (NSS) unit at IIT Roorkee. Built with modern technologies following industry best practices.

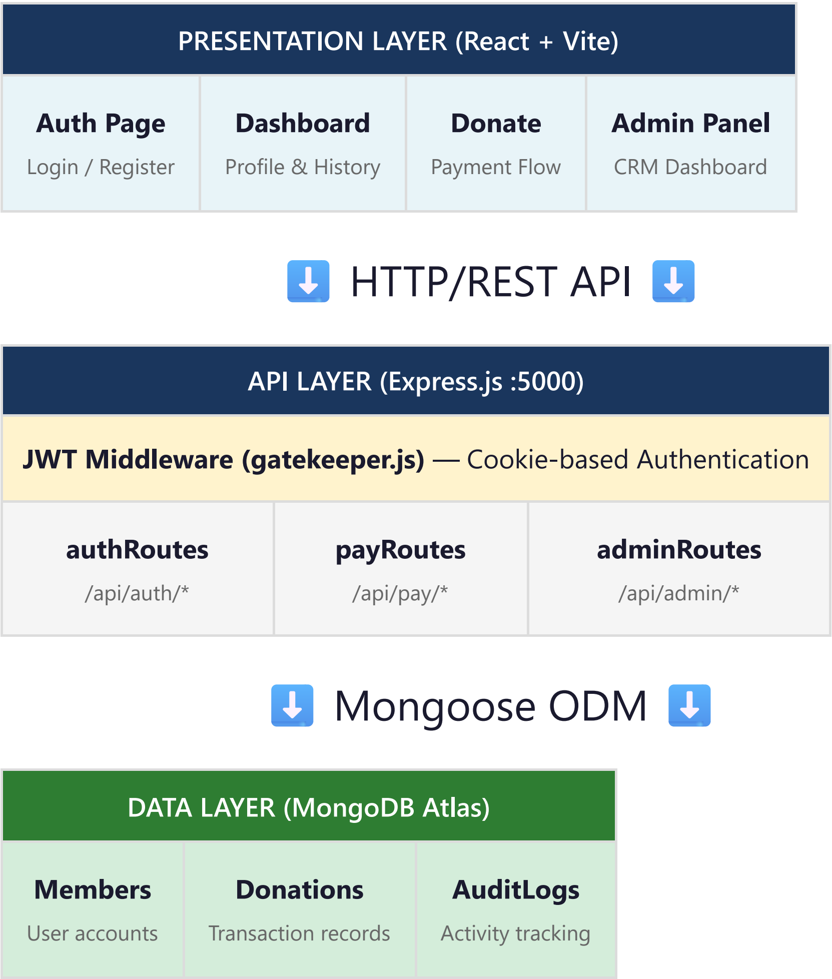
> **Key Features:**
>
> - JWT-based authentication with role separation (admin/supporter)
> - Secure donation processing with transaction tracking
> - Admin CRM dashboard for member and donation management
> - Comprehensive audit logging for compliance

2. Technology Stack

Layer	Technology	Version
Frontend	React + Vite	18.x / 5.x
Routing	React Router	6.x
HTTP Client	Axios	1.x
Charts	Recharts	2.x
Backend	Node.js + Express	20.x / 4.x
Database	MongoDB + Mongoose	7.x / 8.x

Auth	JWT (HttpOnly Cookies)	9.x

3. System Architecture



PRESENTATION LAYER (React + Vite)

Auth Page	Dashboard	Donate	Admin Panel
Login / Register	Profile & History	Payment Flow	CRM Dashboard

⬇️ HTTP/REST API ⬇️

API LAYER (Express.js :5000)

JWT Middleware (gatekeeper.js) — Cookie-based Authentication

authRoutes	payRoutes	adminRoutes
/api/auth/*	/api/pay/*	/api/admin/*

⬇️ Mongoose ODM ⬇️

DATA LAYER (MongoDB Atlas)

Members	Donations	AuditLogs
User accounts	Transaction records	Activity tracking

4. API Endpoints

4.1 Authentication Routes

Method	Endpoint	Description	Auth
POST	`/api/auth/register`	Register new member	Public
POST	`/api/auth/login`	Login & get JWT cookie	Public

POST	`/api/auth/logout`	Clear auth cookie	Public
GET	`/api/auth/verify`	Verify current session	JWT

4.2 Payment Routes

Method	Endpoint	Description	Auth
POST	`/api/pay/initiate`	Create donation record	JWT
POST	`/api/pay/verify`	Verify payment status	JWT
GET	`/api/pay/history`	Get user donation history	JWT

4.3 Admin Routes

Method	Endpoint	Description	Auth
GET	`/api/admin/stats`	Dashboard statistics	Admin
GET	`/api/admin/members`	List all members	Admin
GET	`/api/admin/donations`	List all donations	Admin
PATCH	`/api/admin/members/:id`	Update member role	Admin

5. Database Schema

5.1 Member Model

Field	Type	Constraints
name	String	Required, trim
email	String	Required, unique, lowercase
password	String	Required, hashed (bcrypt)
role	String	Enum: ['admin', 'supporter']

registeredAt	Date	Default: now

5.2 Donation Model

Field	Type	Constraints
memberId	ObjectId	Ref: Member
orderRef	String	Unique, auto-generated
amountPaid	Number	Required, min: 1
paymentStatus	String	Enum: ['success', 'pending', 'failed']
transactionTime	Date	Default: now

6. Security Implementation

🔐 Authentication

- JWT stored in HttpOnly cookies
- bcrypt password hashing (10 rounds)
- Token expiry: 7 days

🛡️ Protection

- CORS with credentials
- Role-based access control
- Input validation & sanitization

7. Frontend Routes

Path	Component	Access
`/`	Auth (Login/Register)	Public
`/dashboard`	User Dashboard	Protected
`/donate`	Donation Form	Protected

`/admin`	Admin Panel	Admin Only

8. Testing Checklist

Test Case	Expected Result	Status
User registration with valid data	Account created, redirect to dashboard	Pass
Duplicate email registration	Error message displayed	Pass
Login with valid credentials	JWT cookie set, dashboard access	Pass
Access protected route without auth	Redirect to login	Pass
Donation with valid amount	Transaction recorded	Pass
Admin panel access by supporter	Access denied, redirect	Pass

9. Environment Variables

Backend Configuration (.env)

Variable	Description	Example Value
`PORT`	Server port number	5000
`MONGO_URI`	MongoDB connection string	mongodb+srv://user:pass@cluster.mongodb.
`JWT_SECRET`	Secret key for JWT signing	your-secret-key

`NODE_ENV`	Environment mode	development / production

Frontend Configuration (.env)

Variable	Description	Example Value
`VITE_API_URL`	Backend API base URL	http://localhost:5000/api

10. Run Instructions

Backend:

```
cd backend npm install npm start
```

Frontend:

```
cd frontend npm install npm run dev
```

NSS IITR - Registration & Donation Management System