Alstonair Technologies



**NGO India - System Design Documentation**

**TABLE OF CONTENTS**

1. System Overview
2. System Architecture
3. User Authentication & Role-Based Access
4. Complete User Flow
5. Module Architecture
6. Database Design
7. API Architecture
8. Frontend Architecture
9. Security Implementation
10. Deployment Architecture
11. Performance Optimization
12. Future Enhancements
13. Conclusion

**1. System Overview**

NGO India is a robust, integrated management and collaboration platform architected to address the multifaceted needs of non-governmental organizations (NGOs) and their key stakeholders, including donors, government bodies, employees, and volunteers. The platform serves as a centralized digital backbone, streamlining varied functional areas such as operational workflows, financial management, project lifecycle tracking, regulatory compliance, and human resource administration, all while fostering transparency and accountability.

Designed with scalability and security at its core, NGO India empowers NGOs—ranging from grassroots organizations to large national bodies—to enhance operational efficiency, boost stakeholder engagement, and effectively monitor and communicate their social impact metrics. By bridging disparate processes into one seamless ecosystem, it enables NGOs to focus more on mission-driven activities with reduced administrative burden.

**Key Features**

**Multi-role Dashboard System:** Dynamic, role-specific dashboards crafted for Administrators, Executive Directors, Employees, and Departmental Directors. These dashboards provide real-time insights, prioritized task lists, alerts, and tailored analytical reports designed to optimize decision-making and operational oversight for each user category.

**Comprehensive Donor Management and Donation Tracking:** Facilitates end-to-end donor lifecycle management including onboarding, segmentation, engagement, and retention strategies. The system supports detailed donation recording with versatile payment methods, automatic receipt generation compliant with legal requisites, and fine-grained analytics that empower NGOs to foster donor trust and long-term support.

**Advanced Financial Management and Automated Reporting:** Enables meticulous tracking of income, expenditures, budgeting, tax obligations, and internal audits. Automated, regulation-compliant financial reporting reduces manual overhead while enhancing transparency and facilitating audit readiness, crucial for maintaining donor and public confidence.

**FCRA and Legal Compliance Automation:** Simplifies adherence to the Foreign Contribution Regulation Act (FCRA) and relevant legal frameworks through automated policy management, AI-assisted document validation, seamless regulatory filings, audit preparation kits, and real-time compliance alerts. This reduces risk and administrative workload for NGOs operating within stringent legal mandates.

**Human Resource and Volunteer Management:** Offers complete workforce management capabilities encompassing staff and volunteer databases, attendance and leave tracking, performance appraisal systems, training program scheduling and tracking, payroll processes, and volunteer coordination tools. These features help NGOs maintain motivated, compliant, and well-trained teams.

**Project Impact Tracking and Analytics:** Provides comprehensive tools to initiate, plan, monitor, and evaluate projects. Features include milestone setting, resource allocation, progress dashboards, budget tracking, and impact assessment modules, enabling NGOs to quantify and transparently convey their social contributions to stakeholders and funders.

**Real-Time Dashboards and Data Visualizations:** Utilizes advanced graphical reports, heatmaps, KPIs, and alerting systems to deliver actionable insights to program managers and leadership teams, facilitating agile decision-making and continuous improvement across all organizational domains.

**Government and External API Integration Capability:** Supports seamless integration with government databases, payment gateways, tax systems, and partner platforms. This enables automated receipt and donation processing, compliance report submissions, beneficiary data synchronization, and broader ecosystem interoperability.

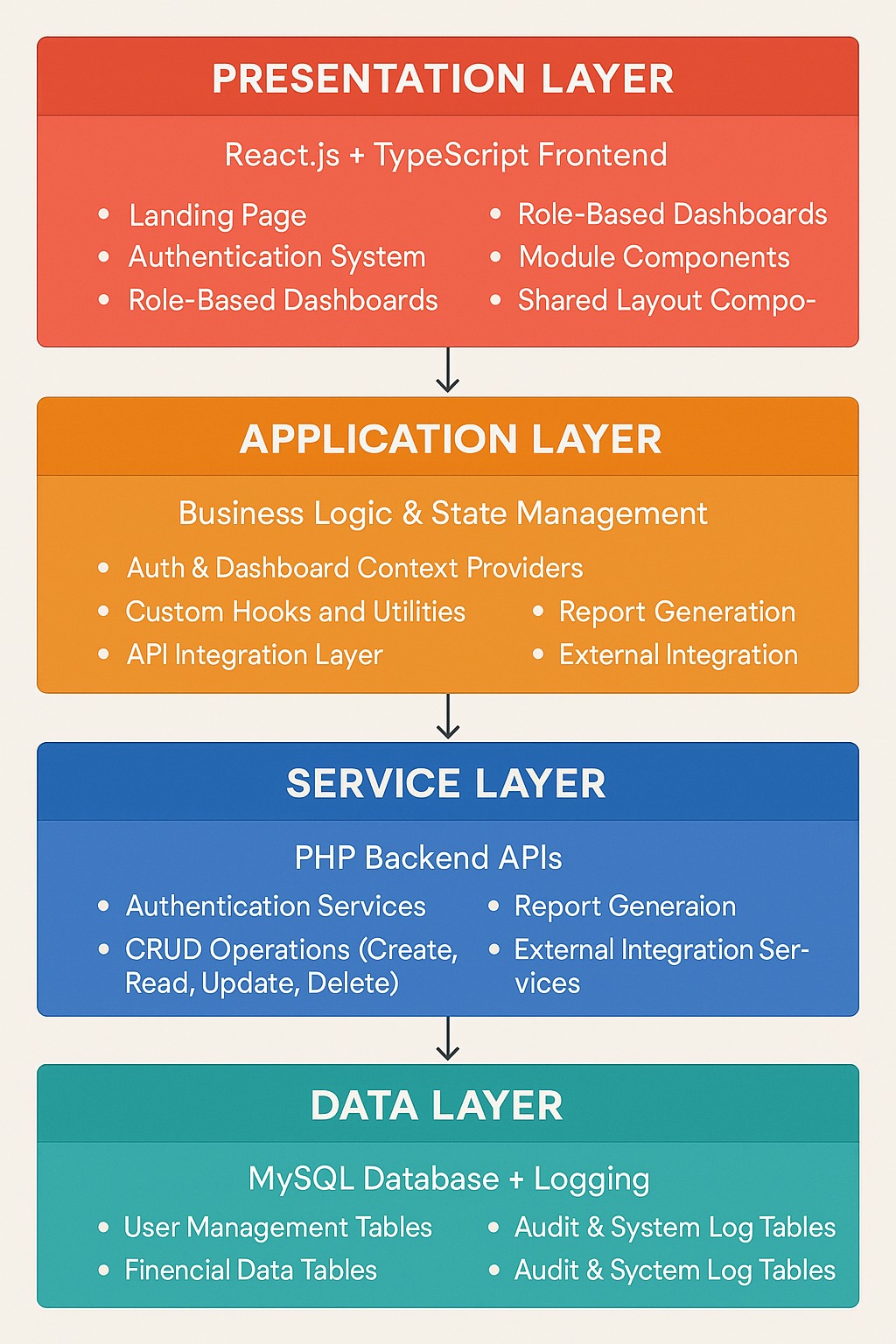
**Collaboration Hub and Networking:** Features in-platform communication channels, document sharing, event coordination, and partner networking designed to strengthen cooperation among NGOs, donors, volunteers, and governmental agencies, driving collective impact and resource optimization.

**Modular and Scalable Architecture:** Built on a flexible, modular foundation that supports NGOs at various scales and stages of digital maturity. Features can be adopted incrementally with support for microservice decomposition to handle growth, multi-language internationalization for broader accessibility, and cloud-friendly deployment options.

NGO India therefore functions as a comprehensive, secure, and user-centric technological foundation that amplifies operational excellence, ensures regulatory compliance, nurtures stakeholder trust, and ultimately accelerates the social impact delivered by NGOs across India and beyond.

**2. System Architecture**

**High-Level Architecture**



**Technology Stack**

**Frontend:**

* React.js 18+ (TypeScript)
* Tailwind CSS
* React Router
* Lucide Icons & Recharts

**Backend:**

* PHP 8.0+ with RESTful services
* MySQL 8.0+ for relational data management
* JSON data exchange

**Developer Tools:**

* Vite for builds
* ESLint for linting
* Git for version management

**3. User Roles and Permissions**

**Administrator**

**Access Level:** Full **Permissions:**

* Manage users, projects, finances, and reports
* Configure system settings
* Approve or reject major actions

**Executive Director**

**Access Level:** Strategic **OversightPermissions:**

* Access executive reports
* Approve initiatives and govern project budgets
* Monitor impact metrics and donor strategies

**Employee**

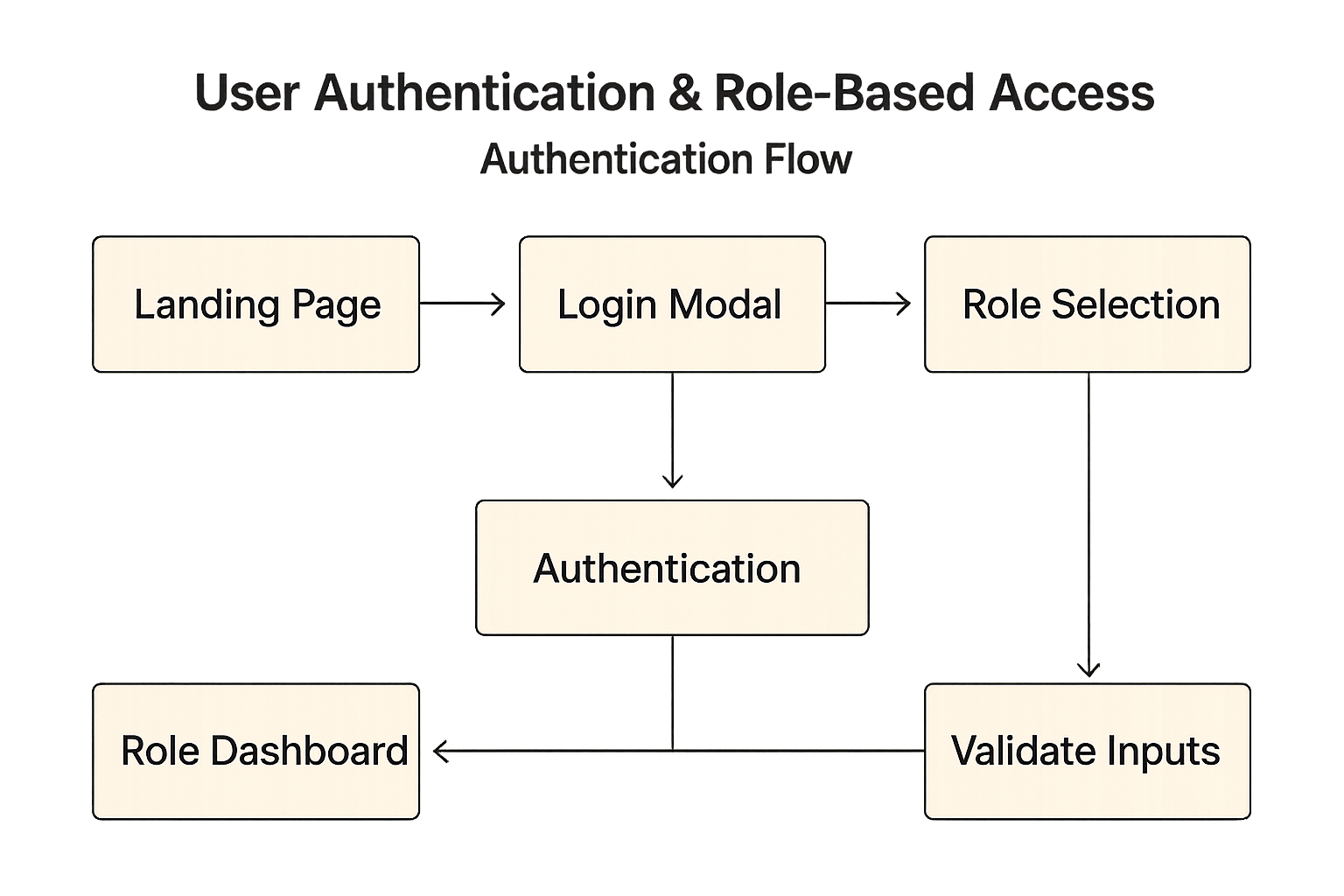
**Access Level:** Restricted **OperationsPermissions:**

* Manage assigned tasks and projects
* Track performance, training, and attendance

**Department Director**

**Access Level:** Department-level Operations **Permissions:**

* Supervise team, manage budgets, create reports

****  
  
  
 **4. Complete User Flow**

**Access and Authentication**

* User visits the Landing Page → Login or Register.
* Upon successful authentication → Redirect to Role-Based Dashboard.
* Each role provides modules and metrics specific to user responsibilities.

**Module Operations**

Select Module

↓

Module Loads & Fetches Data

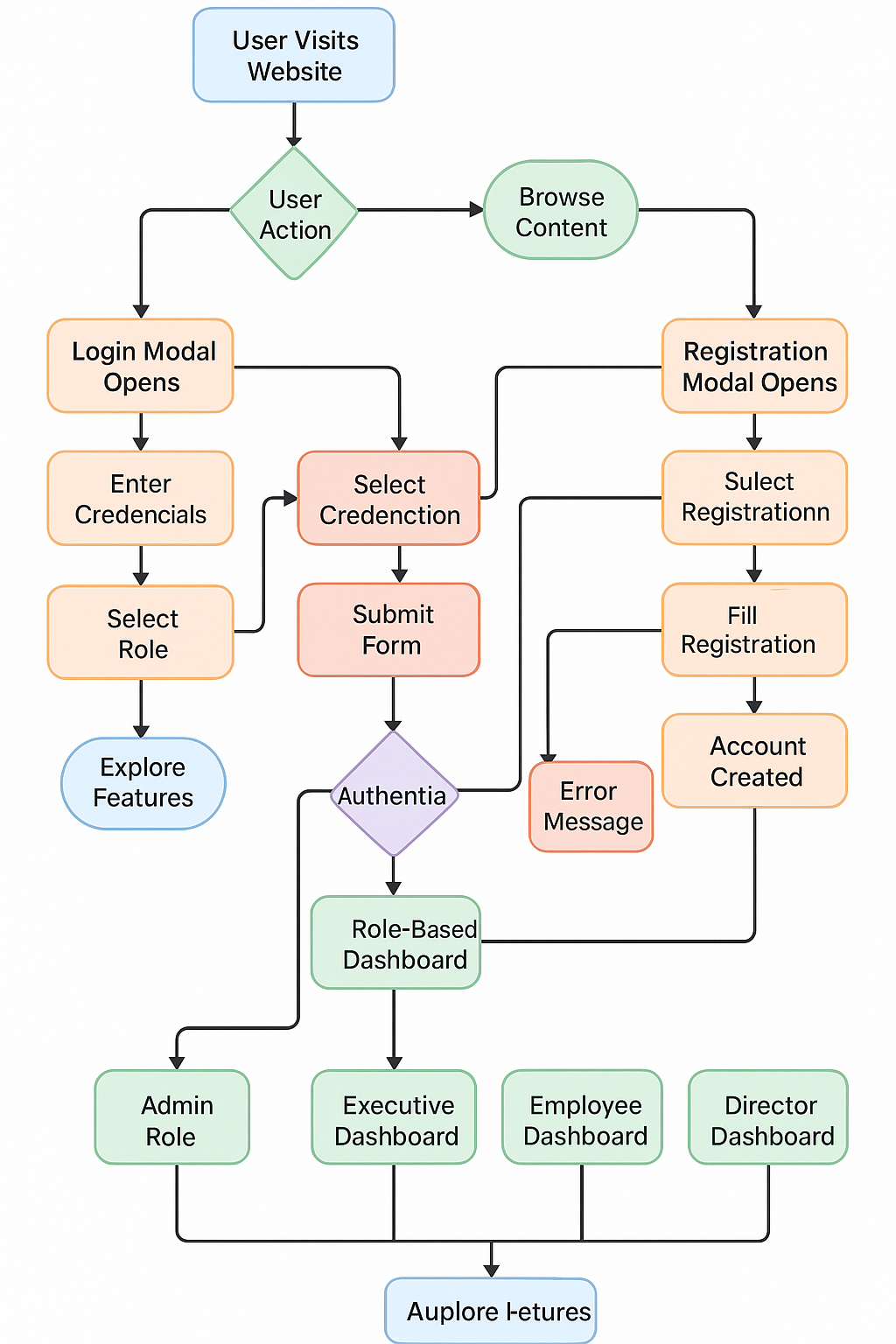
↓

User Performs Actions (Create/Edit/View/Delete)

↓

API Call → Database Transaction

↓

Success/Failure Notification Displayed  
  
  


**5. Module Architecture**

**Donor Management Module**

* Add and update donor records
* Manage donations and generate receipts
* Analyze donor retention and donation trends

**Financial Tracking Module**

* Track income, expenses, and budgets
* Generate financial reports
* Manage tax and audit compliance

**Project Monitoring Module**

* Create and monitor projects
* Manage budgets, milestones, timelines
* Generate project and progress reports  
   **Compliance Module**
* Handle FCRA and legal policies
* AI-based validation for documents
* Generate compliance and audit reports

**HR Management Module**

* Manage employee information and payroll
* Track attendance and training sessions
* Monitor performance and volunteer contributions

**6. Database Design**

**Example Schemas**

**Users**

sql

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255),

email VARCHAR(255) UNIQUE,

password VARCHAR(255),

role ENUM('admin','executive','employee','director'),

department VARCHAR(100),

position VARCHAR(100),

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

**Projects**

sql

CREATE TABLE projects (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255),

description TEXT,

budget DECIMAL(15,2),

start\_date DATE,

end\_date DATE,

status ENUM('planning','active','completed','on-hold'),

manager\_id INT,

FOREIGN KEY (manager\_id) REFERENCES users(id)

);

**7. API Architecture**

**REST Endpoints**

**Authentication**

* POST /backend/simple\_login.php – User login
* POST /backend/register\_user.php – Registration
* POST /backend/logout.php – Logout

**Projects**

* GET /backend/get\_projects\_api.php – Project listing
* POST /backend/add\_project\_api.php – New project
* PUT /backend/update\_project\_api.php – Update project

**Donations**

* GET /backend/get\_donors\_api.php – Fetch donors
* POST /backend/add\_donor\_api.php – Add donor
* POST /backend/donate\_api.php – Record donation

**Finance**

* POST /backend/add\_expense\_api.php – Record expense
* GET /backend/get\_expenses\_api.php – List expenses
* GET /backend/get\_financial\_reports.php – Summaries

**API Response Structure**

json

{

"success": true,

"data": {...},

"message": "Operation successful",

"timestamp": "2025-11-07T10:30:00Z"

}

**8. Frontend Architecture**

**Component Tree**

App.tsx

│

├── AuthProvider

├── DashboardProvider

├── Router

│ ├── LandingPage

│ ├── LoginForm

│ ├── Dashboard

│ │ ├── AdminDashboard

│ │ ├── ExecutiveDashboard

│ │ ├── EmployeeDashboard

│ │ └── Modules (Donor, Finance, Project, HR, Compliance)

│ └── Reports

└── Chatbot

**State Management**

**Contexts:**

* AuthContext: Manages authentication
* DashboardContext: Manages module-wide data (donations, projects, etc.)

**9. Security Implementation**

**Core Security Practices**

* Password hashing with bcrypt
* JWT-based authentication
* Role-based middleware access
* SQL injection and XSS protection
* CORS limited to trusted origins
* File validation during uploads

**Example fragment:**

php

header('Access-Control-Allow-Origin: http://localhost:5173');

function validateInput($data) {

return htmlspecialchars(strip\_tags(trim($data)));

}

**10. Deployment Architecture**

**Development Environment**

Frontend → Vite Server (Port 5173)

Backend → XAMPP Apache (Port 80)

Database → MySQL (Port 3306)

**Production Environment**

Frontend → Nginx (Static)

Backend → Apache + PHP-FPM

Database → MySQL Cluster

CDN → Cloudflare

SSL → Let’s Encrypt

**Deployment Pipeline**

Development → Testing → Staging → Production

via Git CI/CD + Docker Compose → AWS/VPS

**11. Performance Optimization**

**Frontend**

* Tree-shaking and code splitting
* WebP images and lazy loading
* Service Worker caching

**Backend**

* MySQL query optimization
* Redis caching for sessions
* Gzip compression on responses

**Monitoring**

* Google Analytics
* Sentry for error tracking
* Prometheus for server dashboards

**12. Future Enhancements**

* Mobile app via React Native
* AI-powered insights for donors and projects
* Blockchain donation tracking
* Multilingual UI with i18n support
* Advanced analytics dashboards

**13. Conclusion**

The NGO India System is designed as a performance-oriented, scalable, and modular platform that supports the functional, financial, and compliance needs of Indian non-government organizations. Its layered architecture ensures high maintainability, while strong security and data integrity measures make it ready for growth and future government collaboration.

**CONTACT INFORMATION**

**Development Team**

**Email:**grants@ngoindia.org

**Phone:**+91 8068447416

**Address:** REGISTERED OFFICE | #28 Third floor MCHS Layout KV Jayaram Road, Jakkur Bangalore 560064