

# Charity CrowdFunding Platform

## Project Documentation

### 1. Introduction

The Charity Funding Platform is a web-based application developed using **Django**, **Django REST Framework**, and **PostgreSQL**. It enables users to create campaigns, donate to existing causes, and volunteer for charitable events. The platform provides transparency and efficiency in raising and managing charitable funds.

### 2. Objectives

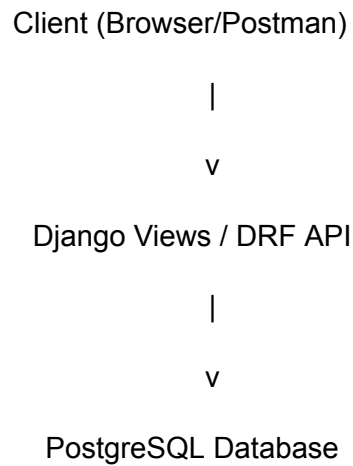
- Provide a secure platform for managing charity campaigns.
- Allow donors to contribute easily through a simple interface.
- Enable volunteers to register their availability.
- Generate QR codes for easy donation tracking.
- Ensure data persistence using **PostgreSQL**.
- Expose APIs for integration and testing with **Postman**.

### 3. Technology Stack

- **Backend Framework:** Django 5.x, Django REST Framework
- **Database:** PostgreSQL (managed via pgAdmin)
- **Frontend:** Django Templates (or React if extended)
- **APIs:** REST APIs tested via Postman
- **Libraries Used:**
  - `qrcode` → for QR code generation
  - `pillow` → image handling

- `djangoRESTframework` → REST APIs
- `psycopg2` → PostgreSQL adapter

## 4. System Architecture



## 5. Database Design

### Tables

#### 1. Campaign

- id (PK)
- title
- description
- goal\_amount
- created\_at (auto timestamp)

#### 2. Donation

- id (PK)
- donor\_name
- amount
- campaign (FK → Campaign)

- verified (Boolean)

### 3. Volunteer

- id (PK)
  - name
  - email
  - phone
  - available (Boolean)
- 

## 6. Features

### Campaign Management

- Create, view, and manage charity campaigns.
- Generate **QR codes** for each campaign.

### Donation Management

- Donors can contribute to campaigns.
- Admins can verify donations.

### Volunteer Management

- Volunteers can register and mark availability.

### REST API Endpoints (examples)

- `POST /api/campaigns/create/` → Create a campaign
- `GET /api/campaigns/` → List campaigns
- `POST /api/donations/create/` → Make a donation

- `POST /api/donations/{id}/verify/` → Verify a donation
- `POST /api/volunteers/create/` → Register volunteer
- `POST /api/volunteers/{id}/avail/` → Update availability

## 7. API Testing with Postman

1. Start Django server: `py manage.py runserver`
2. `py manage.py runserver`

## 8. Security

- CSRF disabled for APIs (handled via DRF).
- Validation on input fields.
- PostgreSQL for secure and reliable data storage.

## 9. Future Enhancements

- Add **payment gateway integration** (Razorpay, Stripe).
- Implement **JWT authentication** for secure API access.
- Build **React/Angular frontend** for better UI.
- Add **admin dashboard** with analytics.

## 10. Conclusion

The Charity Funding Platform provides a transparent and scalable solution for managing charitable contributions. It ensures smooth handling of campaigns, donations, and volunteers while maintaining secure data management with PostgreSQL.

