

Brahma Chakra (technical approach)

1. High-Level System Architecture

Technology Stack

Backend

- Django
- Django Rest Framework
- PostgreSQL (hosted on Neon)
- Redis (optional, for caching and live circles)

Frontend

- HTML5
- CSS3 (Tailwind or custom)
- Vanilla JavaScript (Fetch API)

Location & External APIs

- Mapbox or Google Maps Platform
- Panchang API (in home)
- Cloud media storage (Cloudflare)

Auth

- JWT (SimpleJWT)
- Optional OTP via SMS gateway or Twilio API

2. Backend Architecture (Django + DRF)

```
brahma_chakra/
├── core/                      # settings, urls, middleware
├── users/                     # authentication & profiles
├── locations/                 # temples, sacred places
├── checkins/                  # geo check-ins
├── communities/               # circles & mandalas
├── cycles/                    # arambh → seva → phal → visraam
├── karma/                     # karma coins logic
├── stories/                   # darshan stories (24h)
├── governance/                # polls & rotating leadership
├── panchang/                  # calendar & utsav sync
├── analytics/                 # crowd wisdom, insights
└── requirements.txt
```

3. Core Domain Models (Database Design)

User & Identity

```
User
- id
- phone/email
- is_verified
- joined_at

UserProfile
- user (FK)
- dharmic_name
- home_location (lat, lng)
- total_karma
```

Sacred Locations (Mandirs, Ashrams)

```
SacredLocation
- id
- name
- type (temple, ghat, ashram)
- latitude
- longitude
- city
- deity
- description
```

Geo Check-Ins

```
CheckIn
- user (FK)
- sacred_location (FK)
- checked_in_at
- cycle (FK)
- verified (GPS radius check)
```

Cyclical Framework

Cycle

- community (FK)
- phase (ARAMBH, SEVA, PHAL, VISRAAM)
- start_date
- end_date

Karma Coin System

KarmaCoin

- user (FK)
- type (SARASWATI, LAKSHMI, KALI, NARAD)
- source (checkin, seva, post)
- amount
- created_at

Darshan Stories (24-Hour)

Story

- user (FK)
- location (FK)
- media_url
- expires_at

Live Circles (Temporary Groups)

LiveCircle

- location (FK)
- started_at
- ended_at
- active_users_count

4. API Design (DRF)

Authentication APIs

```
POST /api/auth/login/  
POST /api/auth/verify-otp/  
GET /api/profile/
```

Location APIs

```
POST /api/checkins/  
GET /api/checkins/my/
```

Karma APIs

```
GET /api/karma/balance/  
GET /api/karma/history/
```

Stories APIs

```
POST /api/stories/  
GET /api/stories/location/{id}/
```

Governance APIs

```
POST /api/polls/  
GET /api/polls/active/  
POST /api/polls/vote/
```