**Backend Specification Document for NatureSpeaks App**

---

# \*\*Technical Specification Document\*\*

\*\*Project:\*\* Nature Speaks App - Backend

\*\*Version:\*\* 1.0

\*\*Technology Stack:\*\* Laravel 10+, MySQL 8, Paystack API

---

## \*\*1. Project Overview\*\*

### \*\*Objective\*\*

Convert existing frontend logic (JavaScript) to a secure Laravel backend while:

- Protecting intellectual property (Odu configurations/meanings)

- Implementing proper authentication/authorization

- Maintaining all existing functionalities

### \*\*Current Architecture Issues\*\*

| Frontend Risk | Backend Solution |

|--------------|------------------|

| Business logic exposed in JS | Move to PHP controllers |

| Client-side payment validation | Server-side verification |

| Local storage for auth/sessions | Sanctum tokens + Redis |

| Unprotected divination algorithms | Encrypted database storage |

---

## \*\*2. Core Functional Requirements\*\*

### \*\*A. Authentication System\*\*

| Feature | Specification |

|---------|--------------|

| User Roles | `user`, `priest`, `admin` |

| Auth Flow | JWT via Laravel Sanctum |

| Special Access | 9-tap admin access (replicate existing pattern) |

| Security | bcrypt hashing + 2FA for priests/admins |

### \*\*B. Divination Module\*\*

| Component | Details |

|-----------|---------|

| Odu Storage | 256 Odu configurations in encrypted JSON (AES-256) |

| Access Control | Tiered access (free/premium/initiated) |

| API Response | Structured JSON matching current frontend expectations |

### \*\*C. Knowledge Base\*\*

| Requirement | Implementation |

|-------------|---------------|

| Ifa Database | Relational structure (see ERD in Appendix A) |

| Search | Full-text search with permission filters |

| Chatbot | Migrate `ifaKnowledgeBase` to MySQL with caching |

### \*\*D. Payment Integration\*\*

```mermaid

sequenceDiagram

Frontend->>Backend: Initiate payment (Odu ID)

Backend->>Paystack: Create transaction

Paystack-->>Frontend: Payment UI

Paystack->>Backend: Webhook verification

Backend->>Database: Grant 24hr access

```

---

## \*\*3. Technical Specifications\*\*

### \*\*Database Schema\*\*

\*\*Key Tables:\*\*

1. `odus` - Encrypted configurations + access levels

2. `odu\_meanings` - Separated by orientation (Ire/Ayewo)

3. `payments` - Paystack reference tracking

4. `access\_tokens` - Time-limited Odu access grants

### \*\*API Endpoints\*\*

| Endpoint | Method | Protected | Description |

|----------|--------|-----------|-------------|

| `/api/odus` | GET | Yes | List accessible Odus |

| `/api/odus/{name}` | GET | Yes | Full Odu data + meanings |

| `/api/pay/initiate` | POST | Yes | Start Paystack transaction |

| `/api/knowledge/search` | GET | Yes | Ifa knowledge base query |

### \*\*Security Requirements\*\*

1. \*\*Content Protection\*\*

- AES-256 encryption for Odu configurations

- Dynamic decryption keys per user session

2. \*\*Rate Limiting\*\*

- 5 requests/minute for divination endpoints

- 20 requests/minute for knowledge base

3. \*\*Data Validation\*\*

- Laravel FormRequest validation for all inputs

- Anti-bot measures for public endpoints

---

## \*\*4. Migration Plan\*\*

### \*\*Phase 1: Core Setup (Week 1)\*\*

1. Implement Laravel Sanctum authentication

2. Create encrypted Odu database structure

3. Set up Paystack webhook handler

### \*\*Phase 2: Feature Parity (Week 2-3)\*\*

1. Rebuild divination algorithm in PHP

2. Migrate numerology/astrology calculations

3. Implement knowledge base search

### \*\*Phase 3: Enhancements (Week 4)\*\*

1. Admin dashboard for content management

2. Usage analytics system

3. Automated backup for Odu data

---

## \*\*5. Acceptance Criteria\*\*

### \*\*Must Have\*\*

✅ All frontend features work via API calls

✅ Zero client-side business logic exposure

✅ 1:1 matching response formats for existing frontend

### \*\*Should Have\*\*

⚠️ 24/7 monitoring setup

⚠️ Automated database backups

### \*\*Nice to Have\*\*

🔹 Yoruba language support toggle

🔹 Offline access sync capability

---

## \*\*Appendix A: Database ERD\*\*

```mermaid

erDiagram

USERS ||--o{ PAYMENTS : makes

USERS ||--o{ ACCESS\_TOKENS : has

ODU ||--|{ ODU\_MEANINGS : has

PAYMENTS }|--|| ODU : grants\_access\_to

USERS {

bigint id PK

string email

string role

datetime email\_verified\_at

}

ODU {

bigint id PK

string name

json configuration

boolean is\_premium

}

```

---

## \*\*Appendix B: Sample API Response\*\*

\*\*Frontend Expects:\*\*

```json

{

"configuration": ["|", "|", "|", "|"],

"message": "Ejiogbe's sacred message...",

"orisha": "Orunmila",

"payment\_required": false

}

```

\*\*Backend Should Return:\*\*

```php

return response()->json([

'configuration' => $encryptedOdu,

'message' => $meaning->message,

'orisha' => $odu->orisha,

'taboos' => $meaning->taboos,

'access\_expires' => $user->access\_expiry

]);

```

---

\*\*Document Approval\*\*

Prepared by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Approved by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

---