Product Requirements Document (PRD)

Project Name

Bhagavad Gita Learning Application (Phase-1)

Vision

To make the timeless wisdom of the Bhagavad Gita accessible, simple, and practical for students and children through a modern, attractive, and distraction-free digital application built entirely using open-source technologies.

The application should feel like a calm mentor, not a religious textbook.

Goals & Objectives

Primary Goals

- Enable users to learn Bhagavad Gita chapter-by-chapter and shloka-by-shloka
- Provide simple, student-friendly explanations
- Offer practical life guidance using Gita verses
- Ensure correctness by grounding explanations strictly in the Gita text

Secondary Goals

- Create a future-ready platform for gamification and animations
- Support AI-powered explanations using RAG
- Maintain offline-friendly and low-cost infrastructure

Target Users

Primary Users

- Students (12–22 years)
- Beginners interested in Gita philosophy

Secondary Users

- Teachers
- Parents introducing children to Gita

Scope Definition

In Scope (Phase-1)

- Bhagavad Gita content consumption
- RAG-based explanations
- Daily wisdom
- Life guidance feature
- Audio support (optional)

Out of Scope (Phase-1)

- Animated stories
- Gamification (points, badges, streaks)
- Social/community features

Functional Requirements

1. Authentication

- User login via email or Google OAuth
- Anonymous read-only access (optional)

2. Content Browsing

- Chapter list (18 chapters)
- Shloka list per chapter
- Shloka detail view

3. Shloka Detail View

- Sanskrit shloka
- Transliteration
- Meaning
- Simple explanation (AI-powered via RAG)
- Audio playback
- Bookmark shloka

4. Daily Gita Wisdom

- One shloka per day
- Short explanation
- Optional notification

5. Life Guidance

- User selects a life problem (stress, fear, confusion, etc.)

- System retrieves relevant shlokas
- AI generates practical explanation using RAG

6. Search

- Search by keyword (chapter, theme, verse)

7. User Profile

- View bookmarks
- Manage notification settings

Non-Functional Requirements

- Entire stack must be open-source
- Low latency responses (<2s for RAG)
- Mobile-first UI
- Scalable to millions of users
- Secure and privacy-respecting

High-Level Architecture

1. Frontend Layer

Technology: - React Native (open-source)

Responsibilities: - UI rendering - Navigation - API communication - Local caching

---

2. Backend API Layer

Technology: - Python FastAPI

Responsibilities: - User authentication - Content delivery - RAG orchestration - Business logic

---

3. RAG Engine (Core Intelligence)

Components:

a. PDF Ingestion Module

- Extract text from Bhagavad Gita PDF
- Clean and normalize text

- Split into chunks (one shloka per chunk)

b. Embedding Generator

- Sentence-Transformers (open-source)
- Generate vector embeddings

c. Vector Database

- FAISS
- Stores embeddings + metadata (chapter, shloka, theme)

d. Retriever

- Similarity search based on user query

e. Generator

- Open-source LLM (e.g., LLaMA / Mistral via Ollama)
- Prompt constrained to retrieved context only

---

4. Data Storage Layer

a. Relational / NoSQL DB

- PostgreSQL or MongoDB

Stores: - Users - Bookmarks - Reading progress

b. Static Content Storage

- Local file system / MinIO

---

5. Audio Service

- Pre-recorded chanting files
- Linked via metadata

---

6. Notification Service

- Open-source scheduler (Celery / Cron)
- Push notifications via Firebase (optional)

End-to-End Data Flow

1. User opens app
2. Frontend requests chapter list
3. User selects shloka
4. Backend fetches shloka text
5. RAG engine retrieves relevant context
6. LLM generates simple explanation
7. Response sent to frontend

Open-Source Technology Stack Summary

Frontend

- React Native
- Expo (optional)

Backend

- FastAPI
- Uvicorn

RAG & AI

- pdfplumber
- Sentence-Transformers
- FAISS
- Ollama (local LLM)

Database

- PostgreSQL / MongoDB

DevOps

- Docker
- GitHub Actions

Security & Ethics

- No hallucinated content (RAG enforced)
- Respect religious authenticity
- No dark patterns or addictive mechanics

Future Roadmap

Phase-2

- Gamification
- Progress tracking
- AI mentor chat

Phase-3

- Animated stories
- Multilingual support

Success Metrics

- Daily active users
- Average session time
- Daily wisdom open rate
- Retention after 7 days

Risks & Mitigation

Risk	Mitigation
AI hallucination	Strict RAG constraints
Heavy infra cost	Open-source & local models
User drop-off	Clean UX & daily wisdom

Conclusion

This project is designed as a scalable, ethical, and modern learning platform rooted in ancient wisdom and powered by open-source technology.