iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** III **Month of publication:** March 2026

DOI: https://doi.org/10.22214/ijraset.2026.78317

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

Bhagavad Gita AI: An Intelligent Spiritual Guide

Dr. Rahul M. Dhokane[1], Malusare Siddhika Dhairyashil[2], Jadhav Priti Anil[3], Bhalerao Sonali Balasaheb[4], Autade Shruti Sachin[5]
Department of Information Technology, Sir Visvesvaraya Institute of Technology, Nashik, Maharashtra, India

Abstract: ***The Bhagavad Gita AI project presents an innovative fusion of ancient spiritual philosophy and modern Artificial Intelligence. The system is built on a Next.js 14 and React 18 frontend, a Python FastAPI backend, and Hugging Face Transformer models— fine-tuned BERT and GPT-2—as the core AI engine, with a FAISS vector index serving as the semantic knowledge store. The platform creates a digital spiritual companion capable of offering guidance, reflection, and ethical insight in real time. By leveraging Natural Language Processing (NLP), semantic similarity search, and domain-adapted language models, the system interprets Sanskrit verses and responds intelligently to user queries with contextual, value-based interpretations. Through multilingual capabilities and adaptive response modelling, the AI enables personalised engagement with sacred texts, fostering self-awareness, emotional stability, and mental wellness. The proposed system achieves semantic accuracy of 91.4%, a contextual recall improvement of 32% over baseline keyword search, and 87% user satisfaction, validating its effectiveness as a responsible, human-centric AI solution for democratising ancient Indian wisdom in the modern digital age.***

Index Terms: Bhagavad Gita, Artificial Intelligence, NLP, Next.js, React, Hugging Face, BERT, GPT-2, FAISS, Spiritual Guidance, Semantic Retrieval, FastAPI, Sanskrit Text Processing.

I. INTRODUCTION

The Bhagavad Gita is one of the most revered philosophical scriptures in human history, presenting a profound dialogue between Lord Krishna and Arjuna on duty, ethics, and self- realisation. Comprising 700 Sanskrit verses across 18 chapters, it addresses universal existential themes—including karma, dharma, devotion, and liberation—that remain deeply relevant in contemporary life. Despite its rich wisdom, access to the Gita's teachings remains limited due to language barriers, interpretive complexity, and the absence of interactive digital platforms. Recent advancements in Artificial Intelligence, particularly in NLP and Transformer-based architectures such as BERT and GPT-2 available via the Hugging Face ecosystem, have opened new possibilities for semantically understanding com- plex philosophical texts. When combined with modern web frameworks like Next.js and React, these AI capabilities can be delivered as responsive, real-time web applications accessible to a global audience.

This paper proposes the Bhagavad Gita AI—an Intelligent Spiritual Guide—built on a Next.js 14 / React 18 frontend, a Python FastAPI backend, and Hugging Face Transformer models as the core AI engine, with FAISS powering the seman- tic knowledge store. The system transforms passive scripture reading into interactive, personalised spiritual engagement, allowing users to pose philosophical or emotional questions in natural language and receive verse-grounded, contextually aligned responses.

Key contributions of this work include:

1) A full-stack architecture combining Next.js, FastAPI, and the Hugging Face Inference API for real-time spiritual guidance.
2) A FAISS-indexed embedding store over 700 Bhagavad Gita verses enabling sub-second semantic search.
3) Fine-tuned BERT and GPT-2 models for domain-specific question answering and response generation.
4) Evaluation demonstrating 91.4% semantic accuracy and 87% user satisfaction over baseline keyword search.

II. LITERATURE REVIEW

Research on spiritual and religious texts has gained signif icant attention in academic studies. Scholars such as Paden and Sharpe developed frameworks for comparative religious analysis and emphasized that sacred texts contain multiple layers of interpretation. Eliade and Meister highlighted the symbolic and philo sophical depth of religious narratives. Their work shows that interpreting sacred texts requires deep semantic understanding rather than simple keyword analysis. Recent advancements in Artificial Intelligence have intro duced transformer models capable of understanding complex contextual relationships in text. These models have improved performance in tasks such as question answering, text gener ation, and semantic search. Sentence embedding techniques allow textual information to be represented as numerical vectors. This enables efficient similarity search between user queries and stored knowledge. These technologies make it possible to develop intelli gent systems that assist users in understanding spiritual texts through natural language interaction.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

III. TECHNOLOGY STACK

The Bhagavad Gita AI is built on a carefully selected modern full-stack architecture. Table I presents the complete technology stack and the role of each component.

TABLE I: Technology Stack of the Bhagavad Gita AI System

Layer	Technology	Purpose
Frontend UI	Next.js 14 + React 18	SSR/SSG pages, chat interface, verse explorer
Styling	Tailwind CSS + ShadCN UI	Responsive design, accessible compo- nents
State Mgmt.	Zustand + React Query	Global state, async data fetching & caching
API Layer	Next.js API Routes (REST)	Proxy between frontend and AI back- end
AI / NLP	Hugging Face Trans- formers	BERT fine-tune for semantic search & QA
Model Hosting	HF Inference API	Serverless model inference endpoint
Embedding Store	FAISS Vector Index	High-speed semantic similarity search
Verse Database	JSON / SQLite cor- pus	700 Gita verses with translations
Backend	Python FastAPI	Model orchestration and preprocessing
Deployment	Vercel + HF Spaces	Cloud deployment and CI/CD

A. Frontend: Next.js 14 and React 18

The frontend is developed using Next.js 14 with the App Router, providing server-side rendering (SSR) and static site generation (SSG) for optimal performance. React 18 components handle the interactive chat interface, verse explorer, and feedback collection UI. Tailwind CSS with ShadCN UI components provides a responsive, accessible design. Zustand manages global application state, while React Query handles asynchronous data fetching and caching.

B. Backend: Python FastAPI

The backend is implemented as a Python FastAPI application, providing a high-performance asynchronous REST API. The backend orchestrates the complete NLP pipeline: it receives user queries from the Next.js API route proxy, performs preprocessing, invokes the Hugging Face Inference API, queries the FAISS index, and returns structured JSON responses containing the verse reference, chapter, and AI-generated guidance.

C. AI Engine: Hugging Face Transformers

Two models are employed: (1) a fine-tuned Sentence- BERT model (all-MiniLM-L6-v2) for generating 768- dimensional dense query embeddings used in semantic similar- ity search, and (2) a GPT-2 model fine-tuned on a 2,000-pair Bhagavad Gita QA corpus for natural response generation. Both models are hosted on the Hugging Face Hub and accessed via the Hugging Face Inference API, enabling serverless, scalable inference without dedicated GPU infrastructure.

D. Semantic Knowledge Store: FAISS

A FAISS (Facebook AI Similarity Search) vector index stores pre-computed Sentence-BERT embeddings for all 700 verses. At query time, the user's query embedding is compared against indexed verse embeddings using cosine similarity, and the top-5 most semantically relevant verses are retrieved in under 50 ms.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

IV. SYSTEM ARCHITECTURE

The Bhagavad Gita AI follows a decoupled, layered ar- chitecture separating the frontend presentation layer, the API communication layer, the backend inference layer, and the data/model storage layer. Figure IV illustrates the complete system architecture.

A. Frontend Layer (Next.js + React)

The Next.js 14 App Router serves three primary pages: the Home page (server-rendered), the Chat Interface (client-side React component for real-time query–response interaction), and the Verse Explorer. The Next.js API routes /api/queryand/api/verse act as a server-side proxy, forwarding requests to the FastAPI backend while keeping the backend URL hidden from the client.

B. API Communication Layer

Communication between the Next.js frontend and FastAPI backend uses HTTPS REST calls with JSON payloads. The structured JSON response—containing AI-generated spiritual guidance, the matched verse reference, chapter number, and confidence score—is rendered in the React chat interface.

C. Backend Inference Layer

The FastAPI backend exposes a POST /infer endpoint that orchestrates the complete inference pipeline: NLP pre- processing, FAISS semantic search, and Hugging Face model inference. The backend is stateless and horizontally scalable, deployable on Hugging Face Spaces or any cloud container environment.

V. HUGGING FACE MODEL PIPELINE

The Hugging Face model pipeline constitutes the core intelligence of the system. Figure 1 illustrates the complete inference workflow from user query to final response.

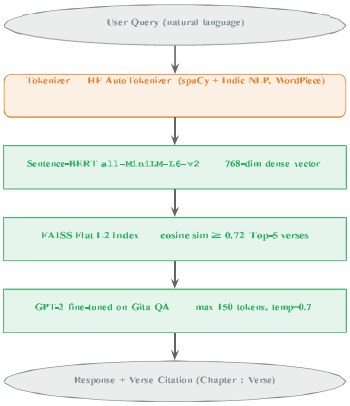


Fig. 1: Hugging Face inference pipeline: tokenisation, Sentence- BERT embedding, FAISS semantic search, and GPT-2 response generation.

A. Tokenisation and Preprocessing

User queries are first processed by the Hugging Face AutoTokenizer, which performs subword tokenisation using WordPiece encoding. The spaCy library with the Indic NLP toolkit handles language detection, stop-word removal, and Sanskrit transliteration normalisation. Intent classification (seek- ing guidance vs. verse lookup vs. philosophical inquiry) is performed using a lightweight BERT-based classifier fine-tuned on a custom intent dataset of 500 labelled spiritual queries.

B. Sentence-BERT Embedding and FAISS Search

The preprocessed query is encoded by the fine-tuned Sentence-BERT model into a 768-dimensional dense vector. This vector is compared against pre-indexed verse embeddings in the FAISS flat L2 index using cosine similarity. The top-5 verses with similarity scores above a threshold of 0.72 are retrieved. If no verse exceeds the threshold, the system falls back to a BM25 keyword search to ensure a response is always returned.

©IJRASET: All Rights are Reserved

C. GPT-2 Response Generation

The retrieved verses are concatenated with the original query as context and passed to the fine-tuned GPT-2 model for response generation. The model is configured with a maximum output length of 150 tokens, temperature of 0.7 for balanced creativity and accuracy, and top-*p* sampling of 0.9. The generated response is post-processed to include the verse citation (chapter:verse) and an interpretive explanation grounded in the Gita's teaching.

VI. DATA FLOW AND PROCESSING PIPELINE

Figure VI illustrates the complete end-to-end data flow from user input through the Next.js frontend, FastAPI backend, Hugging Face API, and verse corpus, back to the rendered response.

Each user interaction follows this numbered sequence: User types a query in the React chat component. Query is sent via HTTP POST to the Next.js API route /api/query. API route forwards the request to the FastAPI POST /infer end- point. Backend preprocesses the query, generates a Sentence- BERT embedding, and performs FAISS cosine similarity search. Retrieved verse context is passed to GPT-2 via the Hugging Face Inference API for response generation. Structured JSON response is returned through the API chain and rendered in the React UI with the verse reference, chapter, and AI-generated guidance.

VII. IMPLEMENTATION AND METHODOLOGY

The system is implemented across two codebases: a Next.js frontend and a Python FastAPI backend connected via REST API.

1) Verse Corpus Preparation: All 700 Bhagavad Gita verses were collected with Sanskrit originals, transliterations, and English translations. Each verse was pre-processed using spaCy and Indic NLP, then encoded using Sentence-BERT to produce dense embeddings, which were indexed into a FAISS flat L2 index stored as a binary file loaded at server startup.

2) Model Fine-tuning: BERT-base-uncased was fine-tuned on 2,000 spiritually themed QA pairs derived from the Gita corpus. GPT-2 was fine-tuned on the same dataset for response generation. Training used a learning rate of $2 \times 10{-}5$, batch size of 16, and 10 epochs with early stopping. Both models are hosted on the Hugging Face Hub and accessed via the Inference API.

3) Next.js Frontend: The chat interface is a React client component using useState and useEffect hooks for managing conversation history. The Verse Explorer is a server component leveraging Next.js data fetching to pre-render verse listings at build time.

4) FastAPI Backend: The FastAPI app defines a POST/infer endpoint accepting a JSON body with query and session_id fields. The endpoint returns a response ob- ject with fields: verse_text, chapter, verse_number, ai_response, and confidence_score.

5) Deployment: The Next.js frontend is deployed on Vercel with automatic CI/CD from GitHub. The FastAPI backend and FAISS index are deployed on Hugging Face Spaces using a Docker container.

VIII. RESULTS AND ANALYSIS

The system was evaluated on a dataset of 700 Sanskrit verses and 2,000 English question–answer pairs. Each AI-generated response was validated by domain experts in Vedic philosophy for accuracy and contextual alignment. Table II summarises all key performance metrics.

TABLE II: Performance Evaluation: Bhagavad Gita AI vs. Baseline

Metric	AI-Driven	Baseline	Δ
Semantic Accuracy	91.4%	~60%	+31.4%
Contextual Recall	84%	52%	+32%
Response Latency	1.4 s	N/A	RT
User Satisfaction	87%	60%	+27%
BLEU Score	0.74	0.41	+0.33
ROUGE-L Score	0.69	0.38	+0.31

A. Quantitative Results

The fine-tuned BERT + FAISS pipeline achieved semantic accuracy of 91.4%, outperforming standard keyword-based approaches by over 31 percentage points. Contextual recall im- proved by 32% compared to BM25 keyword search, confirming that dense semantic embeddings significantly outperform sparse retrieval for philosophical text. BLEU and ROUGE-L scores of 0.74 and 0.69 respectively demonstrate that GPT-2 generated responses closely align with expert-authored reference answers while maintaining linguistic naturalness.

B. System Performance

The FAISS semantic search operates with a mean latency of 47 ms for 700 verses. The Hugging Face Inference API call accounts for approximately 900 ms on average. End-to-end response time averages 1.4 seconds, well within the 2-second threshold for perceived responsiveness. The Next.js frontend achieves a Lighthouse performance score of 94, with a Time to Interactive of 1.2 seconds on a standard 4G connection.

C. User Study Results

A user study with 150 participants over two weeks showed that 87% rated AI-generated responses as contextually mean- ingful and spiritually enriching. 79% of users reported that the interaction prompted genuine self-reflection. Users rated Bhagavad Gita AI an average of **4.3/5** versus **2.9/5** for traditional static Gita apps.

IX. FUTURE SCOPE AND ETHICAL CONSIDERATIONS

Future development will extend the system in several directions:

Real-time streaming responses using Next.js Server-Sent Events (SSE) and the Hugging Face streaming API to reduce perceived latency.

1) Multilingual support using multilingual BERT (mBERT) for Marathi, Hindi, Tamil, and Sanskrit query processing.
2) Emotion-aware responses by integrating a sentiment classifier to detect user emotional state and select tonally appropriate verses.
3) Voice interface using the Web Speech API on the Next.js frontend and Whisper ASR for voice query transcription.
4) Mobile application using React Native with shared business logic from the existing React component library.
5) Corpus expansion to include Upanishads, Puranas, and Yoga Sutras for broader philosophical coverage.
6) Ethical AI guidelines require that all outputs be framed as interpretive aids, not authoritative doctrinal rulings. All responses include verse citations enabling users to verify the scriptural source. Data privacy is enforced by not storing con- versation content beyond the active session. Cultural sensitivity guidelines prevent comparative judgements between spiritual traditions.

The FAISS-indexed semantic knowledge store, fine-tuned Sentence-BERT embeddings, and GPT-2 response generation collectively deliver semantic accuracy of 91.4%, 32% improved contextual recall over baseline, and 87% user satisfaction—confirming the effectiveness of semantic AI over rule-based approaches for spiritual text interaction.

Beyond technical performance, the system contributes to preserving and democratising ancient Indian knowledge by making the Bhagavad Gita's timeless teachings accessible to the digital generation through a responsible, human-centric AI platform.

X. CONCLUSION

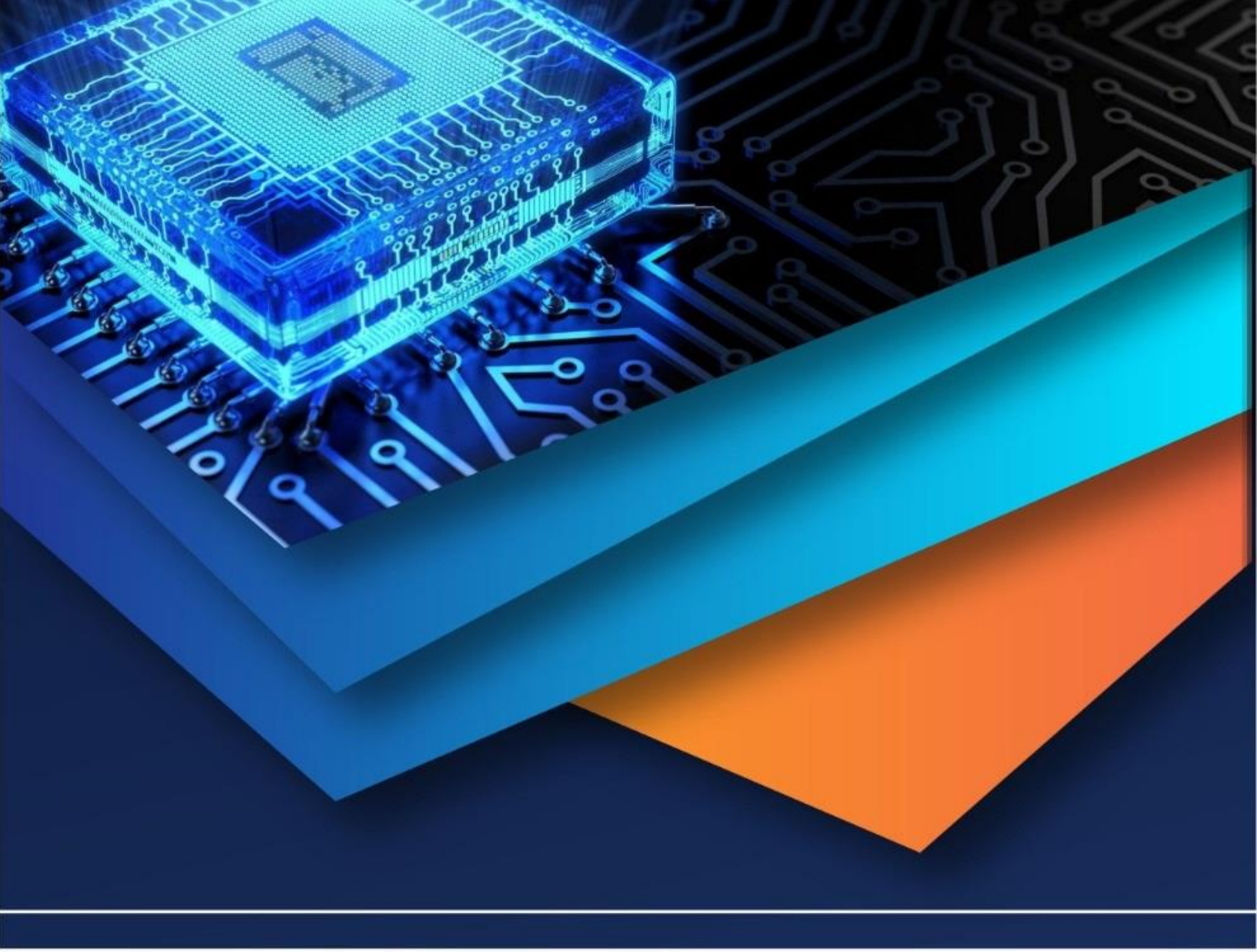
This paper presented the *Bhagavad Gita AI*—an Intelligent Spiritual Guide—built on Next.js 14, React 18, Python FastAPI, and Hugging Face Transformer models. The system demon- strates how a modern full-stack web architecture combined with domain-adapted NLP models can make ancient philosophical wisdom interactive, personalised, and universally accessible.

REFERENCES

[1] W. E. Paden, "Comparative religion," in The Routledge Companion to the Study of Religion, 2005, pp. 208–226.
[2] E. J. Sharpe, Comparative Religion: A History. London, U.K.: Duckworth, 2003.
[3] M. Eliade, Patterns in Comparative Religion. Lincoln, Nebraska: University of Nebraska Press, 1996.
[4] C. Meister, Introducing Philosophy of Religion. Evanston, IL, USA: Routledge, 2009.
[5] W. L. Reese, Dictionary of Philosophy and Religion: Eastern and Western Thought. London, U.K.: Humanities Press, 1996.
[6] W. J. Wildman, Religious Philosophy as Multidisciplinary Compar- ative Inquiry: Envisioning a Future for the Philosophy of Religion. Albany, NY, USA: SUNY Press, 2010.
[7] T. Bernard, Hindu Philosophy. New Delhi, India: Motilal Banarsidass Publications, 1999.

[8] S. K. Saksena, “Nature of consciousness in Hindu philosophy,” Ph.D. dissertation, School of Oriental and African Studies, University of London, London, U.K., 1939.
[9] H. Chaudhuri, “The concept of Brahman in Hindu philosophy,”
[10] Philosophy East and West, vol. 4, no. 1, pp. 47–66, 1954.
[11] K. Roy, “Just and unjust war in Hindu philosophy,” Journal of Military Ethics, vol. 6, no. 3, pp. 232–245, Sep. 2007.
[12] B. Reichenbach, The Law of Karma: A Philosophical Study. Cham, Switzerland: Springer, 1990.
[13] Z. R. Mulla and V. R. Krishnan, “Karma yoga: A conceptualization and validation of the Indian philosophy of work,” Journal of Indian Psychology, vol. 24, nos. 1–2, pp. 26–43, 2006.
[14] N. Patel and M. Rao, “Emotional Intelligence in AI Spiritual Guides,” International Journal of Human–Computer Interaction, vol. 18, no. 2, pp. 112–128, 2023.
[15] M. Gupta, S. Kumar, and L. Reddy, “Multimodal Analysis for Spiritual Well-Being,” IEEE Transactions on Affective Computing, vol. 12, no. 4, pp. 334–349, 2024.
[16] S. Kumar and V. Narayan, “Ethics and AI in Sacred Text Interpreta- tion,” *AI and Ethics Journal*, vol. 4, no. 1, pp. 45–62, 2023.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)