

Project Report: KaushalAI - Bhagavad Gita Chatbot



1. Introduction

1.1 Overview

KaushalAI is an **AI-driven chatbot** that can make the **Bhagavad Gita's wisdom** accessible to readers in an **interactive** and **user-friendly** way. It lets its users **converse with spiritual insights** in an **interactive format**, having their **questions answered on the basis of Bhagavad Gita teachings**. Whether users look for **philosophical guidance, moral clarity, or the practical applications of the Gita's teaching**, KaushalAI provides context-aware responses to their query.

With support for **20 different languages**, KaushalAI ensures that people from diverse linguistic backgrounds can explore the Bhagavad Gita without language barriers. Built with **cutting-edge AI, natural language processing (NLP), and semantic search capabilities**, it delivers **accurate, contextually relevant, and engaging responses** to spiritual seekers.

1.2 Objectives

1. Intelligent Bhagavad Gita-Based Chatbot

- Develop an AI-powered chatbot that provides **contextually relevant responses** based on the teachings of the **Bhagavad Gita**.
- Ensure the chatbot understands **philosophical, ethical, and spiritual inquiries** and delivers **insightful, scripture-backed answers**.

2. User-Friendly and Interactive Frontend

- Design an intuitive **chat interface** similar to modern AI assistants like ChatGPT.
- Provide **real-time, conversational interactions** with **easy navigation and responsiveness** across devices.
- Ensure a **clean, distraction-free UI** to enhance the spiritual learning experience.

3. Customizable API Configuration

- Enable users to **configure API settings** such as:
 - **OpenAI API key** for model customization
 - **Temperature settings** to control response creativity
 - **Token limits** for response length control
- Provide an **easy-to-use settings page** for seamless adjustments.

4. Efficient Semantic Search with Pinecone

- Implement **Pinecone** for **vector-based search**, ensuring accurate and **context-aware retrieval** of Bhagavad Gita verses.

- Use **semantic similarity algorithms** to provide the most **relevant insights** based on user queries.
- Optimize search performance for **fast and accurate** results.

5. Modular and Scalable Architecture

- Ensure **clean code modularization** for **easy maintenance, updates, and scalability**.
- Structure the project with **separate modules for frontend, backend, AI processing, and database handling, research notebook**.
- Support **future integrations** like voice input, additional NLP enhancements, and multi-user capabilities.

6. Flask-Based Backend for API Handling

- Develop a **Flask-based backend** to handle:
 - **User queries and API requests**
 - **Integration with OPENAI models (GPT4, GPT-3-turbo, etc)**
 - **Database and vector search operations**
- Ensure **secure API endpoints** with proper **authentication and rate limiting**.

7. Multilingual Support for a Global Audience

- Provide **real-time translation** for **20+ languages**, allowing non-English speakers to engage with the chatbot.
- Use **Deep translator's Google translate** feature to maintain the **accuracy and integrity** of Bhagavad Gita teachings.
- Allow users to **choose their preferred language** for conversation.

8. Personalized User Experience & Learning Paths

- Allow users to **set learning goals** and receive **daily insights** based on Bhagavad Gita themes via session Ids.

9. Security & Data Privacy

- Ensure **secure API calls** and **encryption** for sensitive data.
- Protect **user queries and preferences** with **privacy-focused policies**.

10. Future Enhancements & AI Model Expansion

- Explore **voice-based interactions** for hands-free engagement.
- Implement **image-based search** for handwritten or scanned verse recognition.
- Continuously refine **AI responses using feedback and machine learning models**.
- Plan for **discussion forums or Q&A sections** where users can share insights.
- Integrate with **spiritual communities and learning groups** for enhanced engagement.

2. System Architecture

2.1 Technologies Used

1. Programming Languages

- **Python** – Backend development, AI/ML processing, and API handling.
- **JavaScript (React.js)** – Frontend development for the chatbot UI.

2. Web Frameworks & APIs

- **Flask** – Backend API development and request handling.
- **Flask-CORS** – Enables Cross-Origin Resource Sharing (CORS) for frontend-backend communication.

3. AI & NLP Models

- **OpenAI GPT (via LangChain & OpenAI API)** – AI-based conversational model for chatbot responses.
- **Hugging Face Transformers** – Used for embeddings and potential LLM-based text processing.
- **Deep Translator (Google Translator API)** – Enables multilingual support by translating queries and responses.

4. Vector Search & Database

- **Pinecone** – Vector database for storing and retrieving embeddings efficiently.
- **PineconeVectorStore (LangChain Integration)** – Enhances semantic search using vector storage.

5. LangChain Ecosystem

- **LangChain OpenAI** – Wrapper around OpenAI API for structured AI interactions.
- **LangChain Pinecone** – Enables vector-based search using Pinecone.
- **LangChain Document Loaders (DirectoryLoader, PyPDFLoader)** – Loads and processes document files (PDFs).
- **LangChain Text Splitter (RecursiveCharacterTextSplitter)** – Splits text for efficient processing.
- **LangChain RetrievalQA** – Creates a conversational retrieval system.
- **LangChain Prompts** – Used to create structured prompt templates.

6. Frontend Technologies (React.js)

- **React.js** – Frontend UI framework.
- **Framer Motion** – Animation library for smooth UI interactions.
- **Lucide React** – Icon library for UI components.
- **State Management (useState Hook)** – Used for handling UI state in React.

7. Environment & Configuration Management

- **dotenv (Python & JS)** – Loads environment variables securely.

8. Backend API Architecture

- **Blueprint (Flask Routes)** – Organizes API endpoints into modular components.
- **REST API (JSON-based Communication)** – Manages communication between frontend and backend.

2.2 Component Breakdown

The project consists of a Frontend (React.js) and Backend (Flask) with structured folders for data storage, research, and helper utilities. Below is a detailed breakdown:

1. Frontend (React.js)

The frontend provides a user-friendly chatbot interface.

1.1 ChatApp (ChatApp.jsx)

- Acts as the main wrapper for the chatbot.
- Manages application state and user interactions.

1.2 Sidebar (Sidebar.jsx)

- Contains navigation options for settings like OPENAI API KEY, Temperature, Maximum tokens.
- Allows users to configure API keys and preferences.

1.3 ChatScreen (ChatScreen.jsx)

- Displays the chat interface where users interact with the chatbot.
- Handles input submission and response display.

1.4 ChatMessage (ChatMessage.jsx)

- Renders individual chat messages (User & AI).
- Differentiates between system-generated and user messages.

1.5 Other Basic Files

- index.js – Entry point for the frontend application.
- App.js – Connects components and manages routing.
- styles.css – Manages UI styling and themes.

2. Backend (Flask API)

The backend processes user queries, integrates AI models, and handles API interactions.

2.1 run.py

- Entry point for starting the Flask backend.
- Initializes the app and runs the server.

2.2 app/ (Application Logic)

- `__init__.py` – Sets up Flask with necessary configurations.
- `routes.py` – Defines API routes for chat interactions.

2.3 src/ (Core Functionality)

- `helper.py` – Utility functions for processing PDFs, translations, and embedding models.
- `prompt.py` – Stores structured prompts for AI response generation.

2.4 data/ (Storage)

- Contains PDF files of the Bhagavad Gita used for AI training.

2.5 research/ (Primary Research)

- Jupyter notebooks for initial research and experimentation on AI model performance.
- Logs findings related to AI responses, embeddings, and search functionalities.

3. Implementation

3.1 User Interface

- The UI is designed to resemble **ChatGPT**, providing a familiar and intuitive experience.
- A **sidebar** allows users to:
 - Enter an **API key** for authentication.
 - Adjust **temperature** (response randomness).
 - Set **token limits** (response length).
- The **chat interface** displays messages with alternating styles for **user and bot responses**.
- A **language selector** lets users communicate in **20 different languages**, making the chatbot globally accessible

3.2 Backend API & Database

- **The backend is built with Flask to handle API requests efficiently.**
- **The workflow includes:**
 1. **User sends a query.**
 2. The query is **translated to English** using the **Google Translate API** (if needed).

3. Pinecone searches for relevant **Bhagavad Gita** verses based on semantic similarity.
4. OpenAI GPT generates a response using the retrieved verses.
5. The response is **translated back to the user's selected language**.
6. The **session history** is optionally stored for future reference.

- **Databases & Caching**

- **Pinecone (Vector Database)**
 - Stores Bhagavad Gita verses as embeddings.
 - Enables fast retrieval of relevant information.

3.3 Pinecone Integration

- **Vectorization Process:**
 - Preprocessing: The Bhagavad Gita verses are cleaned and tokenized.
 - Embedding Generation:
 - The verses are converted into vector embeddings using OpenAI's embedding model.
 - These embeddings are stored in Pinecone for efficient retrieval.
- **Search & Response Generation:**
 - Query Processing:
 - The user query is converted into an embedding.
 - Pinecone retrieves similar stored embeddings (most relevant verses).
 - Response Generation:
 - The retrieved verses are passed to OpenAI GPT.
 - A natural language response is generated.
 - The response is translated back to the user's language (if required).

4. Features

- **Conversational AI**
 - The chatbot engages users in interactive dialogues, making spiritual learning more engaging and intuitive.
- **Thematic responses**
 - Ensures that all replies are aligned with Bhagavad Gita's teachings, maintaining accuracy and authenticity.
- **Customizable API settings**
 - Users can modify API key for authentication, temperature to control randomness in responses, and token limits to adjust the length of responses.

- **Dark/light mode**
 - Provides an adaptive UI that enhances usability based on user preferences.
- **Pinecone-powered search**
 - Retrieves the most relevant Bhagavad Gita verses through efficient vector search.
- **Modular codebase**
 - Designed with scalability and maintainability in mind, making it easy to add new features and updates.
- **Multilingual support**
 - Supports 20 languages, ensuring global accessibility and a seamless user experience.
- **Speech support (upcoming)**
 - Future updates will include voice-to-text interaction, enabling users to speak queries instead of typing.

5. Future enhancements

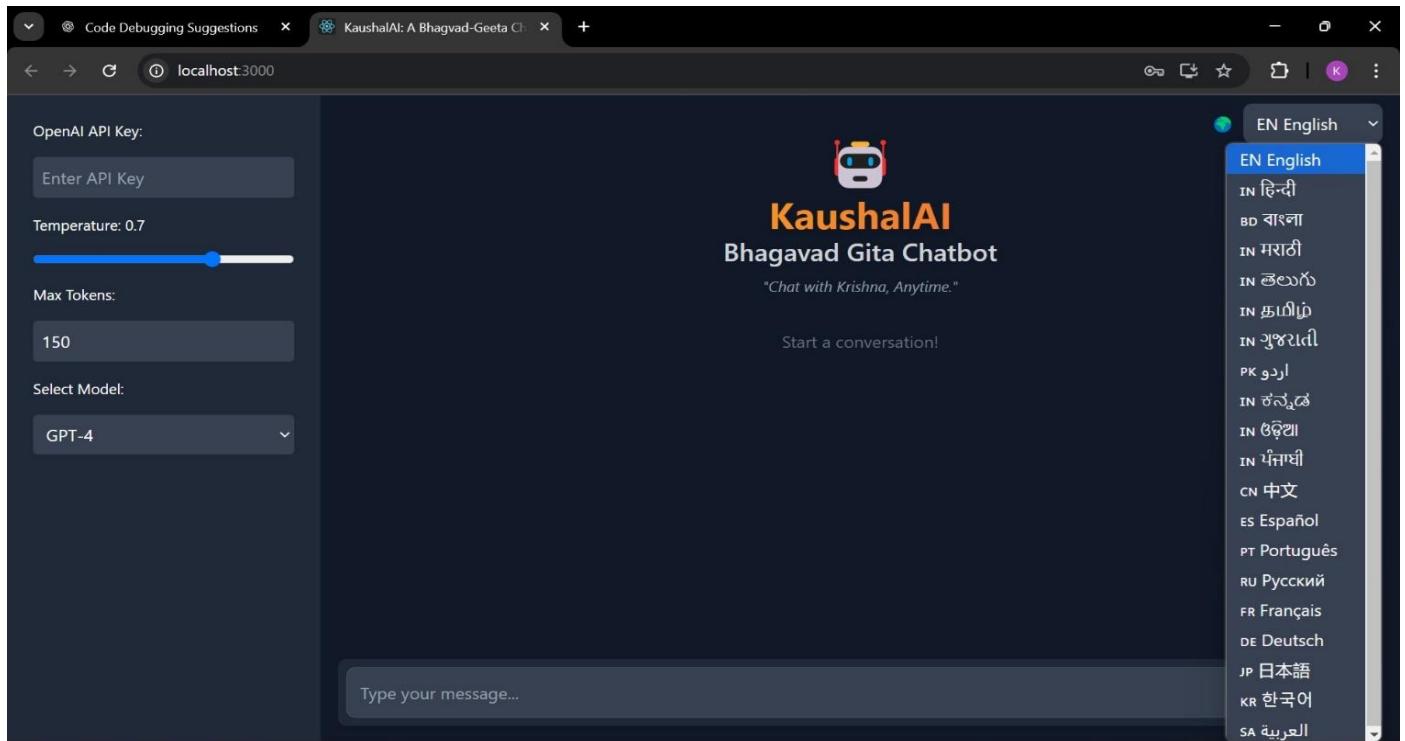
- **Advanced multilingual support**
 - Enhancing translation accuracy and expanding to more languages for a broader audience.
- **Voice interaction**
 - Implementing speech-to-text and text-to-speech features for hands-free interaction.
- **Personalized recommendations**
 - Offering insights based on user history and previous interactions, making responses more relevant.
- **Mobile application**
 - Extending the chatbot's functionality to iOS and Android platforms for on-the-go accessibility.
- **Advanced fine-tuning**
 - Training a specialized GPT model with Bhagavad Gita knowledge, ensuring more context-aware responses.
- **Integration with social media**
 - Enabling chatbot usage on platforms like WhatsApp, Telegram, and Discord, making it more accessible.

6. Conclusion

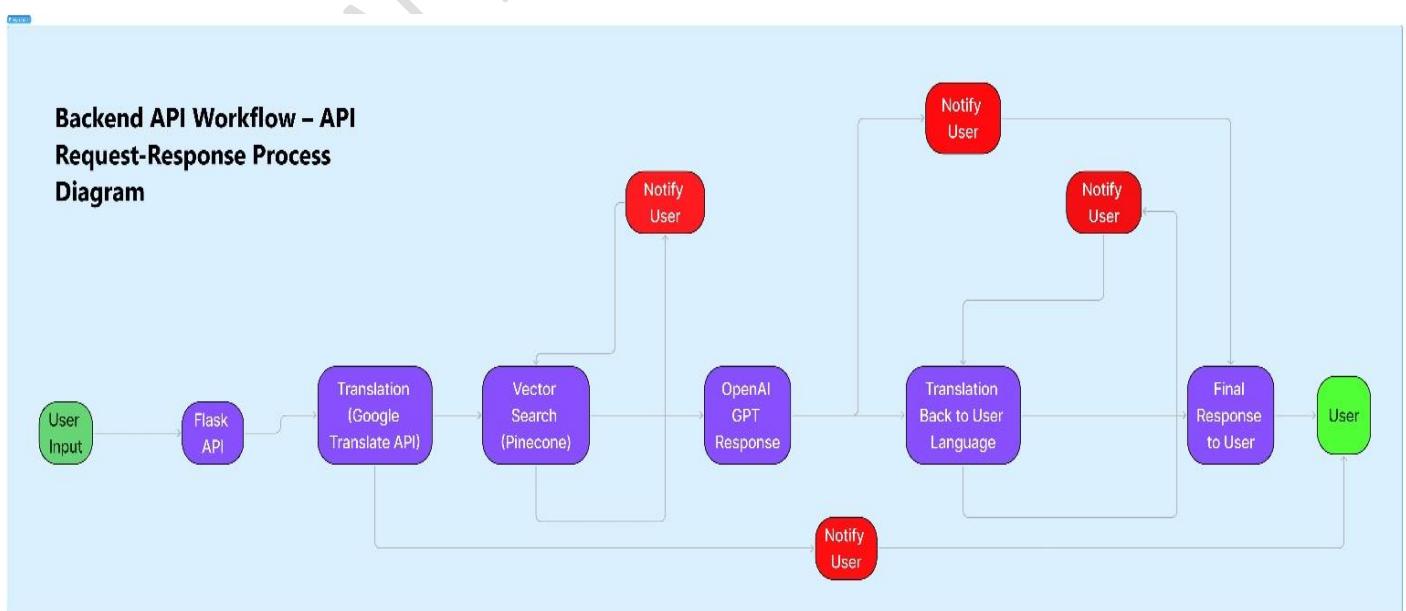
KaushalAI successfully bridges the gap between technology and spirituality by making the teachings of the Bhagavad Gita more accessible. The integration of Pinecone ensures accurate retrieval of verses, while the modular backend and frontend enhance flexibility. With further refinements, the chatbot has the potential to become a valuable spiritual companion for users worldwide.

7. Screenshots & Visuals

1. Homepage & Chat Screen



2. API workflow in Backend



3. Some Example questions and their answers

- Who are you?

The screenshot shows a web browser window with the URL localhost:3000. The page title is "KaushalAI: A Bhagavad-Geeta Chatbot". On the left, there are configuration options: "OpenAI API Key" (redacted), "Temperature: 0.7" (sliding bar), "Max Tokens: 150", and "Select Model: GPT-4". The main area features the KaushalAI logo and the text "Bhagavad Gita Chatbot" with the tagline "Chat with Krishna, Anytime.". A message bubble shows "You 🙏 : Who are you?". Below it, a response from "Krishna 🕉️" reads: "I am Krishna, the Supreme Soul, the eternal witness. I am the light of the sun, the wisdom of sages, the silence of the wise. I dwell in the hearts of all." At the bottom is a text input field with "Type your message..." and a green send button.

- What should I do in Human Life?

The screenshot shows the same web browser window with the URL localhost:3000. Configuration options remain the same. The main area shows the KaushalAI logo and "Bhagavad Gita Chatbot" text. A message bubble shows "You 🙏 : What I should do in human life so that I will not have to rebirth?". Below it, a response from "Krishna 🕉️" reads: "According to the teachings of Bhagavad-gita, one should strive to attain liberation from the cycle of birth and death by understanding the nature of the material world, the Supersoul, and the individual soul. This can be achieved through personal effort, association with saintly persons and a spiritual master, and understanding the instructions of the Bhagavad-gita. By following these principles, one can attain spiritual consciousness and be free from the need to take birth again in this material world." At the bottom is a text input field with "Type your message..." and a green send button.

- What is the responsibility of being human?

The screenshot shows a web browser window with multiple tabs open. The active tab is 'localhost:3000' and features the KaushalAI logo and the text 'Bhagavad Gita Chatbot'. A message from the bot reads: "Krishna: The responsibility of being human is to understand your true identity as a spiritual being and to strive for self-realization and spiritual growth. It is to act with compassion, kindness, and righteousness towards all living beings, and to ultimately connect with the Supreme Soul, Krishna, through devotion and surrender. It is to fulfill your duty and prescribed responsibilities while remaining detached from the fruits of your actions. As a human, you have the ability to elevate your consciousness and transcend the material world. Embrace this responsibility and strive towards your ultimate goal of liberation." On the left side of the interface, there are input fields for 'OpenAI API Key', 'Temperature' (set to 0.7), 'Max Tokens' (set to 1000), and 'Select Model' (set to GPT-4). A message input field at the bottom contains the question "What is the responsibility of being human?".

- Summary of Shrimad Bhagavad Geeta?

The screenshot shows the same web browser window with the KaushalAI: Bhagavad Gita Chatbot interface. The message from the bot reads: "Krishna: This path is for those with a devotional nature, who". Below this, another message from the bot reads: "Krishna: The Bhagavad Gita is a sacred text that contains the teachings of Lord Krishna to his devotee and friend, Arjuna. It is a dialogue between the two, taking place on the battlefield of Kurukshetra. The Gita is a part of the epic Mahabharata and is considered one of the most important and influential Hindu scriptures. In the Gita, Krishna explains to Arjuna the true nature of the self, the purpose of life, and the path to liberation. He teaches about the different types of yoga and the importance of acting in accordance with one's duty, without attachment to the results." The interface includes the same input fields for 'OpenAI API Key', 'Temperature' (set to 0.7), 'Max Tokens' (set to 3000), and 'Select Model' (set to GPT-4) on the left, and a message input field at the bottom containing the question "Give me a Summary of Geeta?".

OpenAI API Key:
.....

Temperature: 0.7

Max Tokens: 150

Select Model: GPT-4

KaushalAI
Bhagavad Gita Chatbot
"Chat with Krishna, Anytime."

You 🙏 : Who am I?

Krishna 🕉️ : According to the Bhagavad-gita, you are a part and parcel of the Supreme Lord and your true identity is not as the creator or enjoyer of the material world, but as a cooperative part of the whole. Your consciousness is "I am," but the Supreme Lord is both the creator and the enjoyer, while you are neither. Your true essence is eternal and spiritual, and you should not lament or identify with the temporary material body and its changes.

Type your message... Send

- **What is Yoga? And how many types of Yogas are there?**

OpenAI API Key:
.....

Temperature: 0.7

Max Tokens: 3000

Select Model: GPT-4

KaushalAI
Bhagavad Gita Chatbot
"Chat with Krishna, Anytime."

You 🙏 : What is Yoga? and How many types of yoga are there?

Krishna 🕉️ : Yoga is the process of connecting with the Supreme Soul, Krishna. It is the ladder that leads to spiritual realization and liberation. There are three main types of yoga: jñāna-yoga (the path of knowledge and understanding), dhyāna-yoga (the path of meditation and concentration), and bhakti-yoga (the path of devotion and surrender). Each path has its own practices and methods, but they all ultimately lead to the same goal of connecting with Krishna.

You 🙏 : Explain me all three yogas.

Type your message... Send

Appendix: File Structure

Bhagvad-Geeta-chatbot/

```
| — frontend/
|   | — src/
|   |   | — components/
|   |   |   | — Sidebar.jsx
|   |   |   | — ChatScreen.jsx
|   |   |   | — ChatMessage.jsx
|   |   |   | — ChatApp.jsx (Main Chat Application)
|   |   |   | — App.jsx
|   |   |   | — main.jsx
|   |   | — public/
|   |   | — package.json
|   |   | — tailwind.config.js
|
| — backend/
|   | — run.py (Main entry point for backend)
|   | — app/
|   |   | — __init__.py (Necessary initialization)
|   |   | — routes.py (API route definitions)
|   |   | — src/
|   |   |   | — helper.py (Utility functions for API operations)
|   |   |   | — prompt.py (Predefined prompts for chatbot)
|   |   | — data/ (PDF containing Bhagavad Gita verses)
|   |   | — research/ (Notebooks for primary research on the app's functionality)
|   |   | — requirements.txt
|
| — README.md
```