Goutham M K – RVCE24MSE001
Nityasree G – RVCE24MIT004

Sirinaadu

Royal Heritage of Wodeyar Dynasty Using RAG AI

Objectives

The main goals of the **Sirinaadu** project are:

- **Making Wodeyar's History Accessible**: Provide an AI-powered retrieval system for Wodeyar's historical data.
- **Supporting Kannada Language Queries**: Enable users to interact with the system using Kannada and English.
- **Efficient Information Retrieval**: Use Pinecone-based vector search for retrieving historical facts.
- **Interactive AI Chatbot**: Implement a chatbot using Google Gemini 2.0 Flash [4] for dynamic content generation.
- **Multi-Modal Input Support**: Accept both speech and text inputs to enhance accessibility.
- **Language Translation**: Automatically translate responses between Kannada and English using Google Translate.
- **Context-Aware Responses [11]**: Retrieve contextually relevant content from the dataset to improve answer accuracy.
- **User-Friendly Interface**: Provide a ChatGPT-like UI for ease of use.
- **Real-Time Updates**: Ensure live updates for historical queries and responses.
- **Handling Speech Failures**: Implement retry mechanisms for speech input failure and fallback to text input if needed.
- **Data Optimization**: Optimize text Chunking [15] and Embeddings [3] for efficient storage and retrieval.

Technical Details

Technologies Used

- **AI & NLP Models**: Google Gemini 2.0 Flash [4], intfloat/multilingual-e5-large [5]
- **Vector Database [2]**: Pinecone
- **PDF Processing**: PyMuPDF (fitz) [6]
- **Speech Recognition**: Google Speech-to-Text [7]
- **Programming Languages used**: Python 3
- **Web Framework**: Flask [13], Bootstrap [12]
- **Python Libraries**: Flask [13]==3.1.0, google-generativeai==0.8.4, google-api-python-client==2.161.0, huggingface-hub==0.28.1, numpy==2.2.3, pinecone-client==5.0.1, protobuf==5.29.3, scikit-learn==1.6.1, scipy==1.15.1, sentence-transformers==3.4.1, SpeechRecognition==3.14.1, torch==2.6.0, transformers==4.48.3, tqdm==4.67.1
- **Frontend**: HTML, CSS, JavaScript
- **Cloud Compute Service**: AWS EC2 [14]

Workflow

1. **Data Extraction**: Extract historical content from PDFs using PyMuPDF.
2. **Text Processing**: Split extracted text into manageable chunks.
3. **Embedding Generation**: Convert text into Embeddings [3] using a multilingual transformer model.
4. **Indexing**: Store Embeddings [3] in Pinecone for efficient retrieval.
5. **Query Handling**:
    - Convert user input into an embedding.
    - Retrieve relevant historical chunks from Pinecone.
    - Pass retrieved data to Google Gemini 2.0 Flash [4] for generating responses.
6. **Translation**: Translate responses if needed.
7. **Speech Support**:
    - Recognize speech input and convert it to text.
    - Retry speech recognition up to three times before switching to text mode.

8. **User Interface**: Provide an intuitive and interactive experience.

Methodology

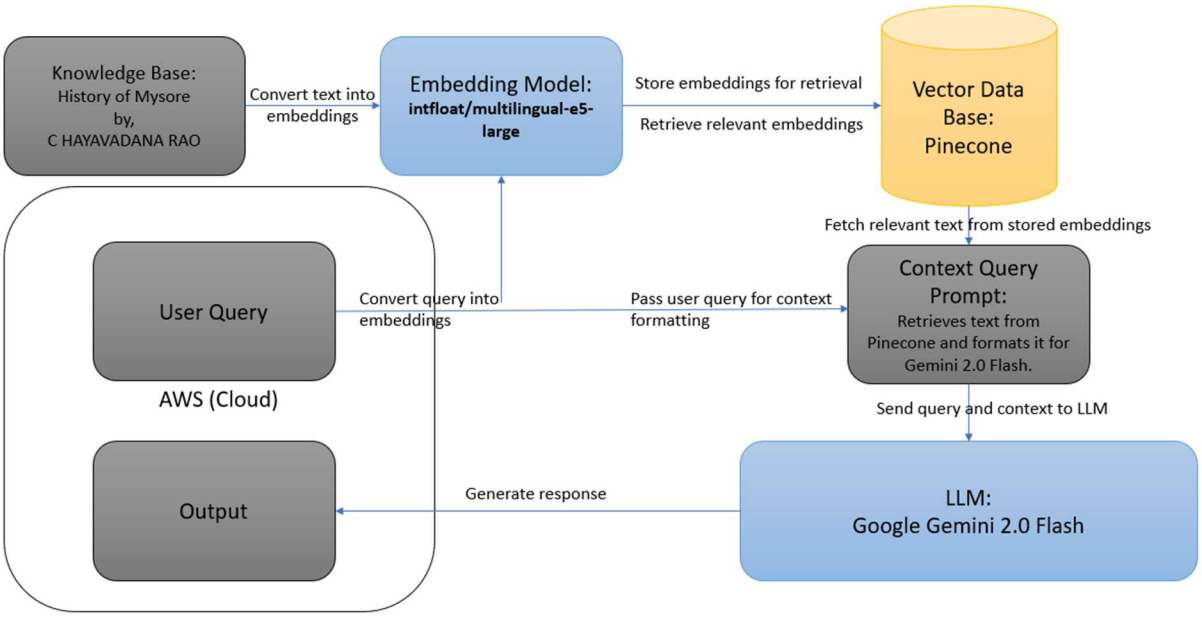

Fig. 1. Architecture Diagram

1. Data Collection & Preprocessing

- Gather **historical PDFs** related to Karnataka's history.
- Generate **vector Embeddings [3]** using **intfloat/multilingual-e5-large [5]**.
- Store processed Embeddings [3] in **Pinecone** for fast retrieval.

2. Backend Development (Flask [13] API)

- Develop API endpoints using **Flask [13]** for:
  - Querying **Pinecone** for relevant historical data.
  - Sending queries to **Google Gemini 2.0 Flash [4]** for dynamic responses.
  - Handling **multilingual translations** and voice input.
- Integrate **Google Speech-to-Text [7]** for converting spoken queries into text.

3. Frontend Development (UI/UX)

- Build an **UI** using **JavaScript, HTML, CSS**.
- Implement **speech recognition UI** to support voice queries.

4. Hosting & Deployment

- Deploy the Flask [13] backend on **AWS EC2 [14]**.
- Use **Nginx [8]** as a Reverse Proxy [16].
- Configure **Gunicorn [9]** for production.

5. System Functionalities

- **Retrieval-Augmented Generation [1] (RAG)** → Combines **historical data retrieval** (Pinecone) with **dynamic AI responses** (Gemini).
- **Multilingual Support** → Converts AI-generated responses into the selected language.

Prototype Analysis

Initial Prototype:

- The initial prototype encompassed content for the entire state of Karnataka, based on a comprehensive document provided by the Government of Karnataka. While the software functioned as expected, the primary flaw was the lack of assurance or proof regarding the accuracy of the information provided.
- Additionally, the image carousel on the left side displayed the name of each image only when hovered over by the cursor, which resulted in poor user experience due to its limited ease of use.

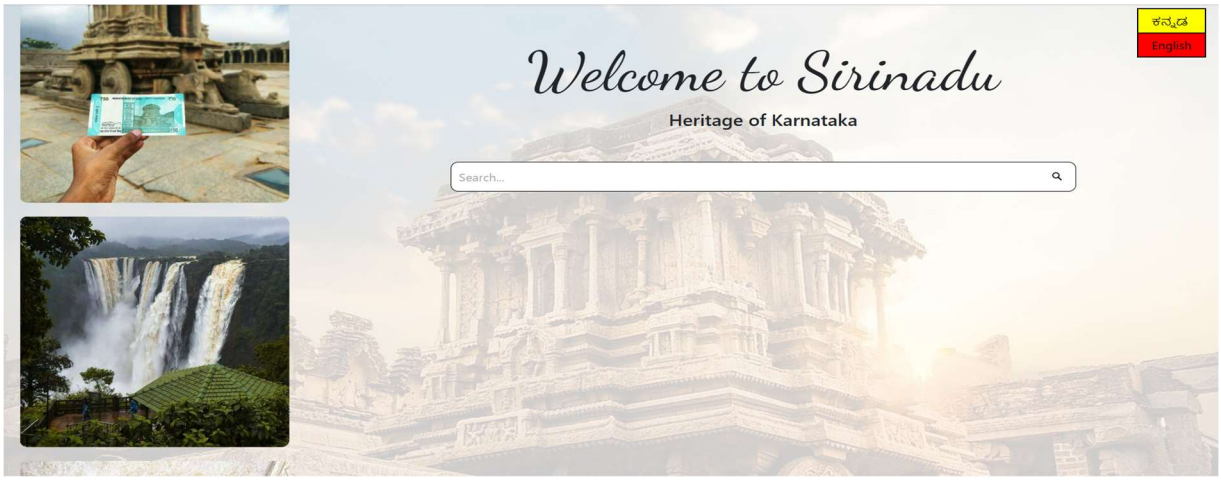

Fig. 2. Initial Prototype

Final Prototype:

- To address these issues, the scope was narrowed to focus solely on the history of Mysore during the Wodeyar dynasty, prioritizing accuracy.
- Verifiable data from the historian C. Hayavadana Rao [19]'s book[2] covering the period from 1399 to 1799 A.D. was utilized.
- The Carousel Functionality [18] was improved by making image information constantly visible, significantly enhancing user comfort.
- Furthermore, an additional webpage was introduced to cite sources for both data and images, reinforcing the credibility of the content.

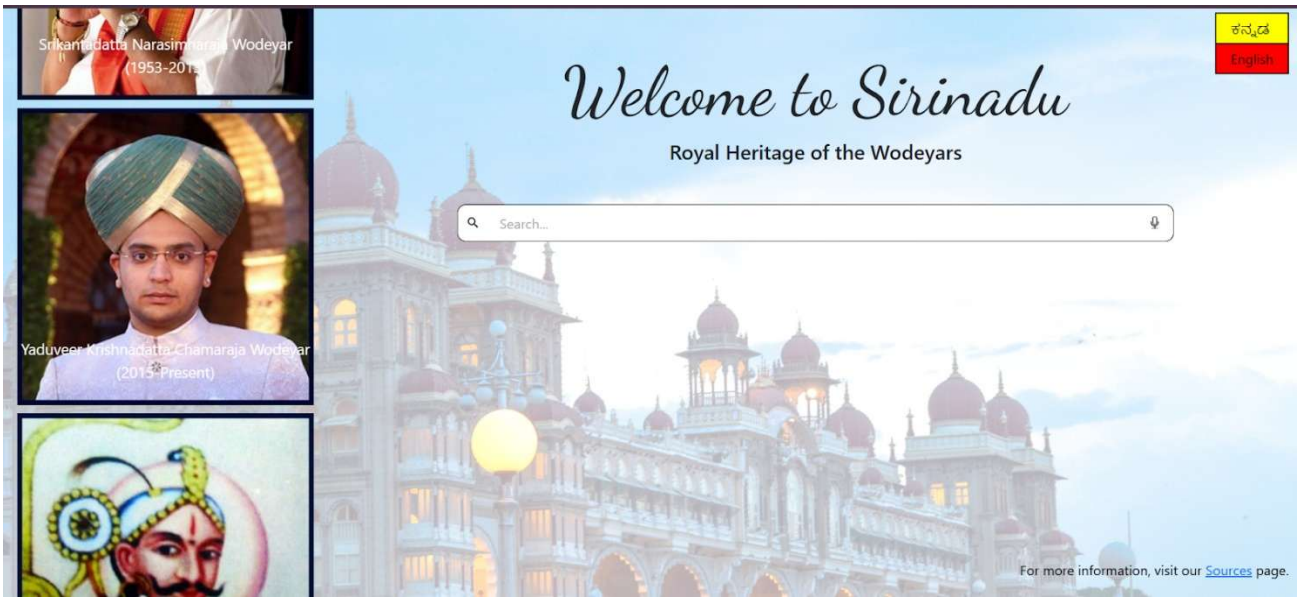

Fig. 3. Final Prototype

Comparison of Initial and Final Prototype through Design Thinking Principles

Design thinking is a user-centered approach that focuses on understanding the needs, challenges, and pain points of the end users. It consists of five key stages: *empathize*, *define*, *ideate*, *prototype*, and *test*. Here's how these steps apply to the project:

1. **Empathize**: In the initial stage, the team tried to address the needs of users by offering a broad range of information about Karnataka's history. However, users were primarily concerned with the accuracy and usability of the content. This feedback helped in understanding their desire for credible, easy-to-navigate historical information.
2. **Define**: After gathering user insights, the problem was redefined—users needed a more focused, reliable source of information rather than broad coverage with potential inaccuracies. The team identified the key issues as the need for verifiable data and a more user-friendly interface, especially regarding the image carousel.
3. **Ideate**: Once the problem was clearly defined, the ideation phase involved brainstorming ways to improve accuracy and usability. The decision to narrow the scope to the Wodeyar dynasty, which allowed the use of credible sources like C. Hayavadana Rao's book, emerged from this stage. Additionally, improvements to the image carousel and source citation were proposed to enhance user trust and experience.
4. **Prototype**: The revised version of the prototype incorporated these ideas, focusing solely on the Wodeyar dynasty, ensuring a more focused and accurate representation of history. The carousel was also updated to display image information continuously, making it more intuitive for users.
5. **Test**: The final prototype was tested, and user feedback showed that narrowing the scope and improving both accuracy and interface usability significantly enhanced user experience. The addition of a dedicated citation page reinforced trust in the content, validating the effectiveness of these changes.

Through the design thinking process, the project evolved by continually refining the solution based on user needs and feedback, leading to a product that was both reliable and user-friendly.

Incorporation of Agile Methodology

Agile methodology was utilized throughout the project to ensure continuous progress and flexibility in responding to user feedback. The project was divided into iterative sprints, with each sprint focusing on specific goals and tasks. Initially, sprints focused on gathering and analyzing user feedback, which was then used to refine the product backlog. As the project evolved, features such as narrowing the scope to the Wodeyar dynasty and improving the image carousel were developed incrementally. Regular sprint reviews and user testing ensured that each iteration met user needs and expectations. This Agile approach allowed the team to adapt quickly, prioritize features based on user feedback, and make timely adjustments to improve the product's accuracy and usability, ultimately delivering a more user-centered and functional solution.

Results and Observation

- **Interactive User Engagement**: The platform facilitates dynamic exploration of the Wodeyar dynasty through both text and voice inputs, offering a seamless and user-friendly experience.
- **Precision in Information Retrieval:** The RAG model ensures the delivery of accurate, context-driven responses sourced directly from the historical text, guaranteeing trustworthy information.
- **Enhanced Visual Learning:** The vertical image scroller gallery provides a smooth, visually rich method for users to navigate and learn about the kings of Mysore.
- **Efficient RAG Model Execution:** The model operates effectively, retrieving and generating contextually relevant responses that enhance user interaction.
- **Increased Interest in Heritage:** The platform encourages deeper interest and interaction with Karnataka's rich historical heritage, especially regarding the Wodeyar dynasty.

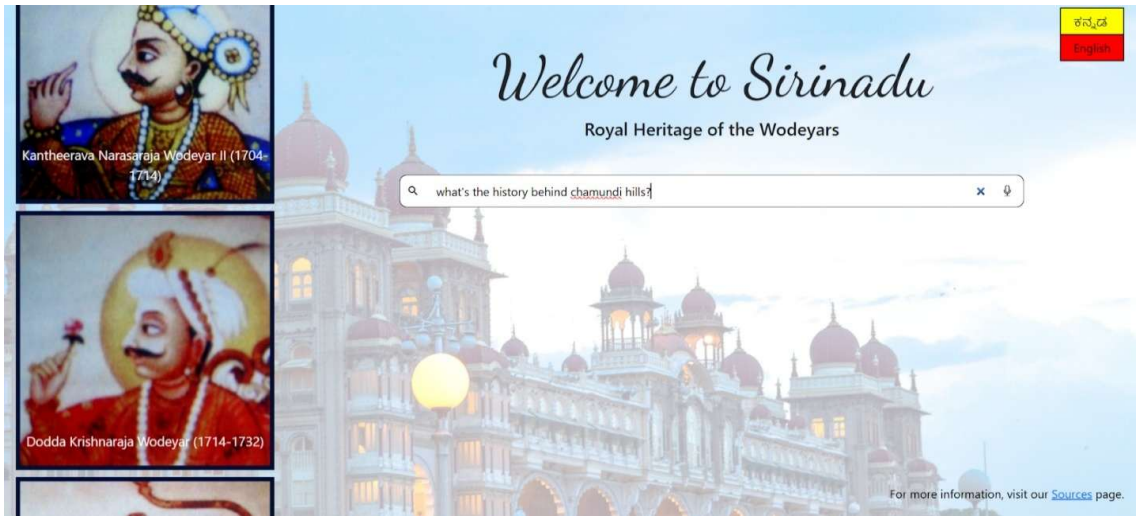

Fig. 4. Sirinaadu's Search Function

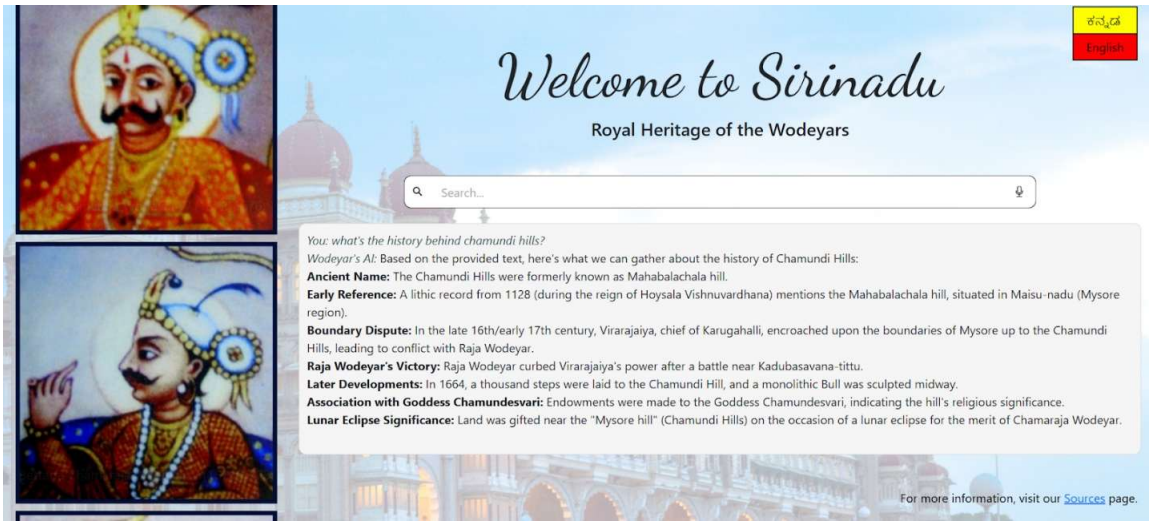

Fig. 5. Sirinaadu's English Responses

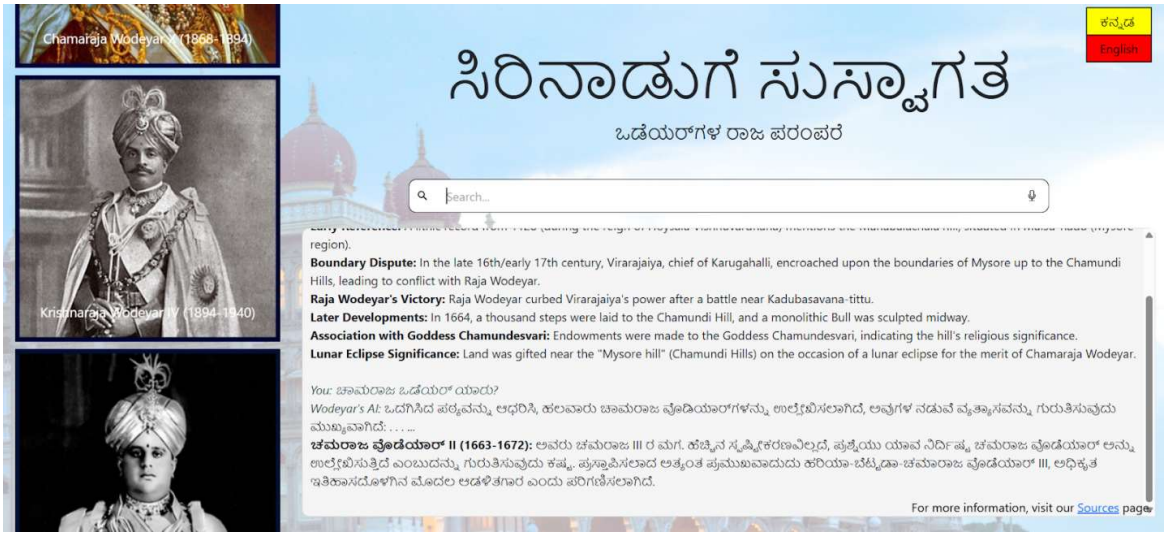

Fig. 6. Sirinaadu in Kannada

Inference

- The *Sirinaadu* platform demonstrates the potential of combining interactive technology (RAG model, voice input, cloud hosting and dynamic visuals) to **enhance the learning experience** for the user.
- By offering accurate, context-driven responses and a visually engaging interface, the project successfully **encourages user exploration** of the Wodeyar dynasty, fostering deeper engagement with Karnataka's heritage.

Future Enhancements

- **Expanding Dataset**: Add more historical documents and sources.
- **Improved Query Understanding**: Enhance AI comprehension using advanced NLP techniques.
- **User Personalization**: Provide tailored responses based on user history.
- **Integration with Educational Platforms**: Partner with institutions for academic applications.
- **Mobile App Development**: Extend functionality to mobile platforms.

Conclusion

The **Sirinaadu** project successfully demonstrates the potential of advanced AI technologies, such as **Retrieval-Augmented Generation [1] (RAG)** and **Multimodal Input [10]s**, to revolutionize how we access and interact with Karnataka's historical heritage, specifically the **Wodeyar dynasty**. Through an innovative combination of Vector Database [2] (Pinecone), dynamic AI response generation (Google Gemini 2.0 Flash [4]), and seamless language translation between **Kannada** and **English**, the platform provides users with a unique, interactive, and accessible means to explore history.

The project efficiently handles both **text** and **voice inputs**, making it accessible to a broader audience while ensuring real-time, contextually relevant responses. The integration of a **visually enriched interface**, including a **vertical image scroller gallery**, further enhances the user experience, promoting deeper engagement with the historical content. By blending cutting-edge AI technology with a user-friendly design, **Sirinaadu** offers an engaging, dynamic, and informative exploration of the Wodeyar dynasty's rich history.

Significance of the Work

The **Sirinaadu** platform is significant for several reasons:

1. **Cultural Preservation**: By providing easy access to historical data about the Wodeyar dynasty, the project helps preserve and promote Karnataka's rich cultural heritage.
2. **Accessibility**: The use of **multilingual support** and **voice input capabilities** makes the platform accessible to a diverse range of users, including native Kannada speakers and those with varying levels of digital literacy.
3. **Educational Impact**: The platform has great potential as an educational tool, enabling students, researchers, and history enthusiasts to explore and learn in an interactive manner.
4. **Technological Innovation**: By utilizing the latest advancements in **AI, natural language processing (NLP), and vector search**, Sirinaadu sets a new standard for historical data retrieval and AI-assisted learning platforms.
5. **User Engagement**: The project successfully fosters a deepened interest in heritage through its engaging design and accurate, context-driven responses, providing an excellent foundation for future historical explorations.

Overall, **Sirinaadu** exemplifies how AI can transform the way we engage with and learn from historical data, making it more accessible, dynamic, and personalized for users.

References:

1. **Retrieval-Augmented Generation (RAG)** – A technique that combines information retrieval from a database with AI-generated responses to provide accurate and context-aware answers.
2. **Vector Database** – A specialized database that stores high-dimensional vector Embeddings to enable efficient similarity searches, such as Pinecone.
3. **Embeddings** – Numerical representations of words, phrases, or documents in a multi-dimensional space, allowing AI models to understand their relationships.
4. **Google Gemini 2.0 Flash** – A large language model from Google that generates dynamic responses based on input queries.
5. **intfloat/multilingual-e5-large** – A multilingual embedding model used to convert text into vector representations for AI processing.
6. **PyMuPDF (fitz)** – A Python library used for extracting text, images, and other elements from PDF documents.
7. **Google Speech-to-Text** – A cloud-based API that converts spoken words into text using machine learning models.
8. **Nginx** – A web server and Reverse Proxy [16] used to handle incoming traffic and improve application performance.
9. **Gunicorn** – A Python WSGI HTTP server used for deploying Flask [13] applications in a production environment.
10. **Multimodal Input** – A system that supports multiple types of user inputs, such as text and speech, enhancing accessibility.
11. **Context-Aware Responses** – AI-generated answers that consider the surrounding context to improve accuracy and relevance.
12. **Bootstrap** – A front-end framework used to create responsive and mobile-friendly web designs.
13. **Flask** – A lightweight Python web framework used to develop backend APIs and web applications.
14. **AWS EC2** – Amazon Web Services Elastic Compute Cloud, a cloud-based service for deploying and running applications.

15. **Chunking** – The process of splitting large text documents into smaller, manageable pieces for efficient storage and retrieval.
16. **Reverse Proxy** – A server that sits between client devices and backend servers to manage traffic and enhance security.
17. **Query Embeddings** – The process of converting user queries into numerical representations for efficient search and retrieval.
18. **Carousel Functionality** – A UI feature that displays images or content in a sliding format for better user engagement.
19. **C. Hayavadana Rao** – A historian known for documenting the history of Mysore, whose works were used for validating data in the project.
20. **Semantic Search** – A search technique that understands the meaning of words rather than just matching keywords, improving the relevance of retrieved results.