**Major Project-I Report on**

**Title of Project**

***Submitted in Partial fulfillment for the award of degree of Bachelor of Technology in Artificial Intelligence & Data Science***

Submitted to

# Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.)

Submitted By:

## Name of Student (enroll no.) Name of Student (enroll no.) Name of Student (enroll no.))

Under the Guidance of

**Professor**

**Name of the guide**

# DEPARTMENT OF

# ARTIFICIAL INTELLIGENCE & DATA SCIENCE



**Jai Narain College of Technology, Bhopal**

**Approved by AICTE New Delhi & Govt. of M.P.**

**Affiliated to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.)**

**Session: 20.. – 20..**

#### JAI NARAIN COLLEGE OF TECHNOLOGY, BHOPAL

**Approved by AICTE New Delhi & Govt. of M.P. & Affiliated to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.)**

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE

# CERTIFICATE

This is to certify that the work embodied in this Project, Dissertation Report entitled as **“Title of Project”** being Submitted by **Name of Student (enroll no.)** in partial fulfillment of the requirement for the award of **“Bachelor of Technology” in Artificial Intelligence & Data Science** discipline to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.) during the academic year 20..-.. is a record of bonafide piece of work, carried out under my supervision and guidance in the Department of Artificial Intelligence & Data Science, **Jai Narain College of Technology, Bhopal.**

.

### Approved by

Guided by Head of Department

Prof. (name of guide) Prof. Ravinder Tanwar

Dean, Academics Principal

Dr. Vivek Dubey Dr. Netra Pal Singh

#### JAI NARAIN COLLEGE OF TECHNOLOGY,BHOPAL

**Approved by AICTE New Delhi & Govt. of M.P. & Affiliated to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.)**

**DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE**

# CERTIFICATE OF APPROVAL

This Project **“Title of Project”** being submitted by **Name of Student (enroll no.)** has been examined by me & hereby approve for the partial fulfillment of the requirement for the award of **“Bachelor of Technology in Artificial Intelligence & Data Science”,** for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but the Project only for the purpose for which it has been submitted.

### INTERNAL EXAMINER EXTERNAL EXAMINER

Date: Date:

# CANDIDATE DECLARATION

We hereby declare that the Project dissertation work presented in the report entitled as **“Title of Project”** submitted in the partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Artificial Intelligence & Data Science of **Jai Narain College of Technology, Bhopal** is an authentic record of our own work.

We have not submitted the part and partial of this report for the award of any other degree or diploma.

**Name of Student (enroll no.)**

Date:

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

### Guided By:

**Prof. (Name of Guide)**

# ACKNOWLEDGMENT

We are heartily thankful to the **Jai Narain College of Technology** for providing us all the facilities and infrastructure to take our work to the final stage.

It is the constant supervision, moral support and proper guidance of our respected Principal **Prof. (Dr) Netra Pal Singh** and Dean, Academics **Prof. (Dr) Vivek Dubey,** who motivated throughout the work. We express a deep sense of gratitude and respect to our learned guide **Prof.(name of guide),** Professor in the Department of Artificial Intelligence & Data Science, during all phases of our work. Without his enthusiasm and encouragement this dissertation would not have been completed. His valuable knowledge and innovative ideas helped us to take the work to the final stage. He has timely suggested actions and procedures to follow for which we are really grateful and thankful to him.

We express our gratitude to **Prof Ravinder Tanwar** Head of Artificial Intelligence & Data Science. Department for providing all the facilities available in the department for his continuous support, advice, and encouragement during this work and also to help to extend our knowledge and proper guidelines.

Constant help, moral and financial support of our loving parents motivated us to complete the work. We express our heartfelt thanks to all our family members for their cooperation.

We really admire the fond support of our class-mates for their cooperation and constant help. It gives immense pleasure to acknowledge the encouragement and support extended by them. Last but not the least we are extremely thankful to all who have directly or indirectly helped us for the completion of the work.

.

**Name of Student (Enroll no.)**

**Abstract**

This project aims to develop a comprehensive multilingual information dissemination system for railway stations, facilitating seamless communication between passengers and station authorities. The proposed system leverages a Natural Language Translation Engine to provide announcements and information in multiple Indian languages, with provisions for extension to foreign languages to cater to the diverse needs of passengers, including tourists. The system encompasses various channels, such as station announcements, Interactive Voice Response System (IVRS), chatbots, and web interfaces, ensuring accessibility through different mediums.

Key components of the system include advanced Natural Language Processing (NLP) algorithms for language recognition, speech recognition, and text translation. The architecture integrates machine learning for accurate voice recognition in noisy station environments and employs noise cancellation techniques to enhance system performance. Additionally, the system ensures on-the-fly content generation, adapting to dynamic information requirements with efficient use of computing power.

Technological aspects involve the use of Python for NLP and machine learning functionalities, The system is designed with a focus on security, scalability, and user experience. Continuous feedback mechanisms and user analytics contribute to ongoing improvements, ensuring the system remains responsive to the evolving needs of both passengers and station authorities. The proposed multilingual information dissemination system strives to enhance communication efficiency, accessibility, and passenger experience within the dynamic environment of railway stations.

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Table of Contents** | **Page No.** |
| **1** | **Introduction** | 8 |
|  | * 1. Objective   2. Problem Identification   3. Proposed Solution | 8  8  9 |
| **2** | **Software Requirement Specification**   * 1. Purpose   2. Scope   3. Feasibility Study      1. Technical Feasibility      2. Operational Feasibility      3. Economic Feasibility | 11 |
|  | 11 |
|  | 11 |
|  | 12 |
|  | 12 |
|  | 12 |
|  | 12 |
| **3** | **Requirements** | 13  13  13  14  14 |
|  | 3.1 Hardware Requirement |
|  | 3.2 Software Requirement |
|  | 3.3 Data Requirement |
|  | 3.4 Functional Requirements |
| **4** | **System Documentation** | 15  15  16  17  18  19 |
|  | 4.1 Use case diagram |
|  | 4.2 DFD |
|  | 4.3 System Flow Chart |
|  | 4.4 System Level Class Diagram |
|  | 4.5 Object Diagram/Class Diagram/State Diagram/Activity Diagram |
|  | etc. (if required) |
| **5** | **Testing** | 20 |
|  | * 1. Testing Requirement   2. Test Data   3. Test Cases | 20  21  23 |
| **6** | **User Manual** | 25 |
|  | * 1. Introduction and Guidelines   2. Screen Layouts and Description   3. Output Reports | 25  26  27 |
| **7** | **Limitations** | 28 |
| **8** | **Future Enhancement** | 30 |
| **9** | **Conclusion** | 33 |
| **10** | **Bibliography** | 34 |
| **11** | **References** | 35 |
| **12** | **Appendix – I Source Code** | 36 |

### 1. Introduction

In an era marked by cultural diversity and technological advancements, efficient communication is paramount, especially in public transportation hubs like railway stations. This project introduces a novel Multilingual Information Dissemination System for Railways, emphasizing the integration of a sophisticated Natural Language Translation Engine. Tailored to the linguistic diversity of India, the system extends its capabilities to accommodate foreign languages, catering to both local passengers and international tourists. Leveraging cutting-edge technologies such as Natural Language Processing, machine learning, and advanced database management, this system aims to revolutionize how information is delivered through station announcements, IVRS, chatbots, and web interfaces. With a focus on user accessibility, system security, and continual improvement, this endeavor seeks to enhance the communication experience for all stakeholders within bustling railway environments.

### Objective

This project aims to develop a Multilingual Information Dissemination System for Railways, featuring a Natural Language Translation Engine. The primary objectives include providing information in diverse Indian and foreign languages through station announcements, IVRS, chatbots, and web interfaces. Leveraging advanced Natural Language Processing, the system seeks to ensure accurate language recognition and adaptive content generation in real-time. User-centric design principles prioritize passenger and station authority needs, fostering a seamless communication experience. The project emphasizes technological integration, security, and scalability while encouraging continuous improvement through user feedback. By achieving these objectives, the system aspires to enhance linguistic inclusivity, streamline information dissemination, and revolutionize communication within bustling railway environments.

### Problem identification

The Multilingual Information Dissemination System for Railways addresses several challenges in the current communication landscape within railway stations:

1. **Language Barriers:** The linguistic diversity in India poses a significant challenge for effective communication. Passengers and station authorities often encounter language barriers, hindering the smooth flow of information.
2. **Limited Language Support:** Existing systems may lack comprehensive language support, especially for less commonly spoken languages. This limitation can exclude certain passenger groups from accessing critical information.
3. **Dynamic Information Needs:** The ever-changing nature of railway information requires a system that can dynamically generate and deliver content on-the-fly, ensuring that passengers receive up-to-date and relevant information.
4. **Noise and Ambient Conditions:** Railway stations are noisy environments, making it challenging for conventional systems to accurately process and understand verbal requests and announcements.
5. **Inefficient Communication Channels:** The reliance on traditional communication channels may result in inefficiencies, leading to delays, misunderstandings, and an overall suboptimal passenger experience.
6. **Scalability Concerns:** With the increasing volume of passengers and data, there is a need for scalable solutions to accommodate the growing demands on information dissemination systems.
7. **Technological Integration Challenges:** Integrating various technologies, such as Natural Language Processing and real-time translation, poses technical challenges that require careful consideration and expertise.
8. **User Experience Gaps:** Existing systems may not prioritize user experience, potentially leading to confusion and dissatisfaction among passengers and station authorities.

By identifying and addressing these challenges, the proposed system aims to create a more inclusive, efficient, and technologically advanced platform for information dissemination in railway stations.

### Proposed Solution

To design a system for providing information in desired Indian languages on demand, along with the ability to extend to foreign languages for tourists, you would likely need the following technologies and considerations:

**Natural Language Processing (NLP):** NLP technologies like text-to-speech and speech-to-text conversion are essential for handling written and oral information in multiple languages.

**Machine Learning and AI**: Machine learning algorithms can be used for voice recognition in different languages, allowing the system to understand and process spoken requests.

**Multilingual Text Processing:** For written content, you would need tools to handle text in multiple Indian languages and potentially foreign languages. Technologies like Unicode and language-specific libraries will be useful.

**Speech Recognition and Synthesis:** Implementing speech recognition and synthesis engines that can understand and generate speech in various Indian languages and foreign languages.

**Noise Cancellation:** Given the noisy ambience at railway stations, noise cancellation algorithms or hardware solutions might be necessary to improve voice recognition accuracy.

**Content Generation:** To provide on-the-fly content generation, you may employ templates and dynamic content generation algorithms to ensure that information is always up-to-date.

**Web App Development:** Developing mobile applications for various platforms (IOS, Android) and Web) to deliver the information to passengers on their mobile devices.

**IVRS Integration:** Setting up an Interactive Voice Response System (IVRS) to handle phone-based inquiries and requests.

**Chatbot Development:** Implementing chatbots using Natural Language Processing to provide information through chat interfaces.

**Database Management:** Storing and retrieving information efficiently from a database, which may include train schedules, station information, and multilingual content.

**Localization and Translation Services:** Utilizing translation services or APIs to enable quick translation of content into foreign languages for tourists.

**User Interface (UI) Design:** Creating user-friendly interfaces for both mobile apps and web interfaces to make it easy for passengers and customers to access information.

**Scalability and Redundancy:** Ensuring the system is scalable to accommodate a growing number of users and incorporating redundancy for high availability.

**Security and Privacy:** Implementing robust security measures to protect user data and ensure privacy compliance.

**Quality Assurance and Testing:** Comprehensive testing, including usability testing and language-specific testing, to ensure the system functions correctly in different scenarios and languages.

**Continuous Updates and Maintenance:** Regularly updating the system to reflect changes in railway information and language updates.

**Accessibility:** Ensuring the system is accessible to individuals with disabilities, including those with visual or hearing impairments.

**Network Infrastructure:** Reliable network infrastructure to support real-time data delivery.

**Content Management System (CMS):** A CMS to manage and update content efficiently.

**Analytics and User Feedback:** Incorporating analytics to track user behavior and gather feedback for continuous improvement.

### 2 Software Requirement Specification

* Scripting Languages: Python, JavaScript
* Web UI: HTML, CSS
* Fortend & Backend Connectivity: FLASK and REST API
* ChatBot: Large Language Models LAMMA
* DataBase: Vector Database
* Translation: GTrans
* DataSet: OpenGov.in
* Tools: VSCODE, Google Docs. Slides, DrawIO

### Purpose

The purpose of the Multilingual Information Dissemination System for Railways is to revolutionize communication within railway stations by overcoming language barriers and enhancing accessibility. The system aims to provide timely and accurate information in multiple Indian languages, extendable to foreign languages, catering to the diverse linguistic needs of passengers and station authorities. By leveraging advanced Natural Language Processing and real-time translation, the project seeks to create a dynamic, adaptive platform capable of on-the-fly content generation. The overarching goal is to improve passenger experience, streamline information dissemination, and foster inclusivity in railway stations, ultimately contributing to more efficient and user-friendly communication in the context of diverse linguistic environments.

### Scope

* + - Linguistic Inclusivity: Extend support to multiple Indian languages and foreign languages, accommodating the diverse linguistic preferences of passengers.
    - Real-time Information Dissemination: Provide timely and dynamic information through various channels such as station announcements, IVRS, chatbots, and web interfaces.
    - Adaptive Content Generation: Implement on-the-fly content generation to address the dynamic and evolving nature of railway information requirements.
    - Technological Integration: Employ cutting-edge technologies, including Natural Language Processing and machine learning, to ensure accurate language recognition and translation.
    - User-centric Design: Prioritize user experience through intuitive interfaces and continuous feedback mechanisms for iterative improvements.
    - Security and Scalability: Implement robust security measures while ensuring scalability to handle increasing passenger volumes and data complexity.
    - Cross-functional Collaboration: Foster collaboration between linguistic experts, developers, and station personnel to align the system with linguistic nuances and operational requirements.
    - Continuous Improvement: Establish mechanisms for ongoing monitoring, analysis of user behavior, and iterative enhancements to adapt to changing linguistic and technological landscapes.

### Feasibility Study

**Feasibility Study for the Multilingual Information Dissemination System:**

* **Market Feasibility:** Analyze the demand for multilingual communication solutions in railway stations, considering diverse passenger demographics and potential user adoption.
* **Technical Feasibility:** Evaluate the technical viability of implementing Natural Language Processing, real-time translation, and adaptive content generation within the existing railway infrastructure.
* **Economical Feasibility:** Assess the cost implications of developing and maintaining the system, considering budget constraints and potential return on investment.
* **Operational Feasibility:** Determine the practicality of integrating the system into daily station operations, addressing challenges such as staff training and workflow adjustments.
* **Scalability:** Assess the system's ability to scale with increasing passenger volumes and evolving linguistic requirements.
* **Risk Analysis:** Identify potential risks and mitigation strategies related to technology, market changes, and operational challenges.
* **Legal and Regulatory Compliance:** Ensure compliance with linguistic standards, data privacy laws, and other relevant regulations.
* **Social Impact:** Evaluate the societal impact of improved communication, considering passenger satisfaction, accessibility, and inclusivity.

The feasibility study aims to provide a comprehensive understanding of the viability and potential success of implementing the Multilingual Information Dissemination System in railway stations.

### Requirements

A nutshell, to create a multilingual information system for railway passengers and customers, you'd need technologies for :-

* + Language processing (NLP)
  + Speech recognition and synthesis
  + Multilingual text handling
  + Noise cancellation
  + Mobile app and web development
  + IVRS integration
  + Chatbot development
  + Database management
  + Translation services
  + UI/UX design
  + Scalability and redundancy
  + Security and privacy measures
  + Quality assurance and testing
  + Continuous updates and maintenance
  + Accessibility features
  + Network infrastructure
  + Content management
  + User analytics and feedback gathering

### Hardware Requirements

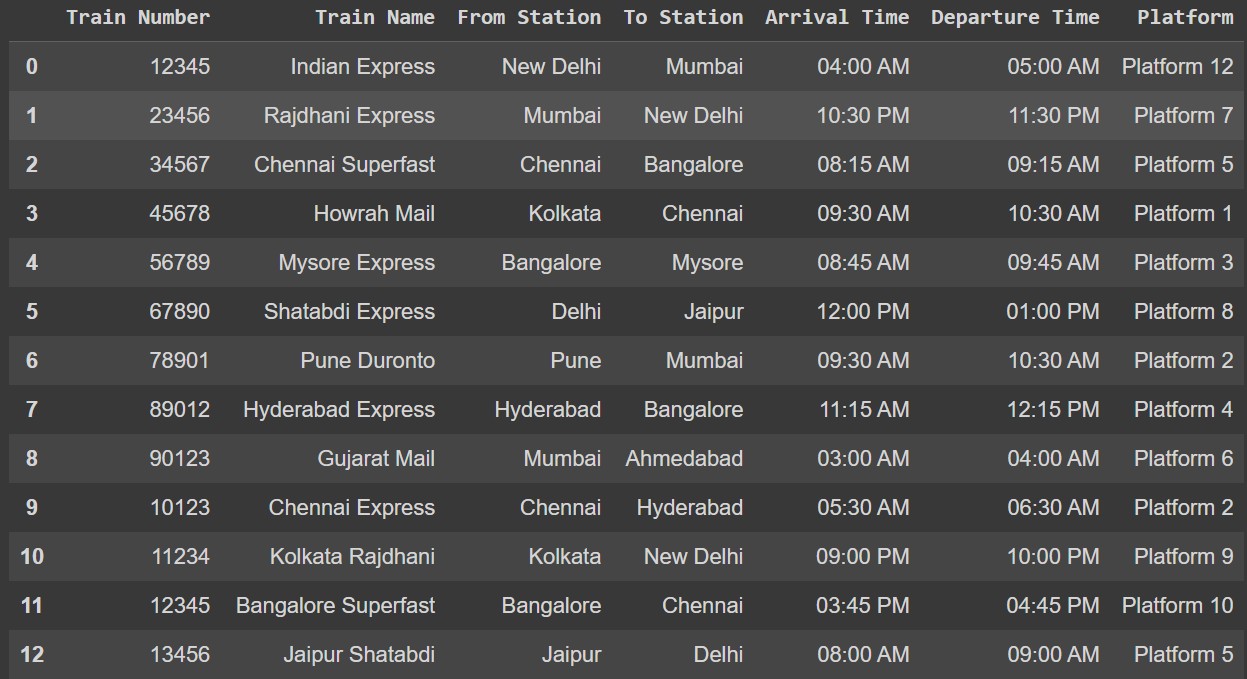
* + - Computer System: Intel i5 or AMD Ryzen 5
    - Graphical Processing Unit: NVIDIA GTX 1650
    - Internet: 5mbps bandwidth
    - Power Supply: 120 WATTs

### Software Requirements

* + - Scripting Languages: Python, JavaScript
    - Web UI: HTML, CSS
    - Backend Connectivity: FLASK and REST API
    - ChatBot: Large Language Models LAMMA
    - DataBase: Vector Database
    - Translation: GTrans
    - DataSet: OpenGov.in
    - Tools: VSCODE, Google Docs. Slides, DrawIO

### Data Requirement

* + - Table Data: CSV(Comma Separated Values)
    - Railways Specific Data: OpenGov.in



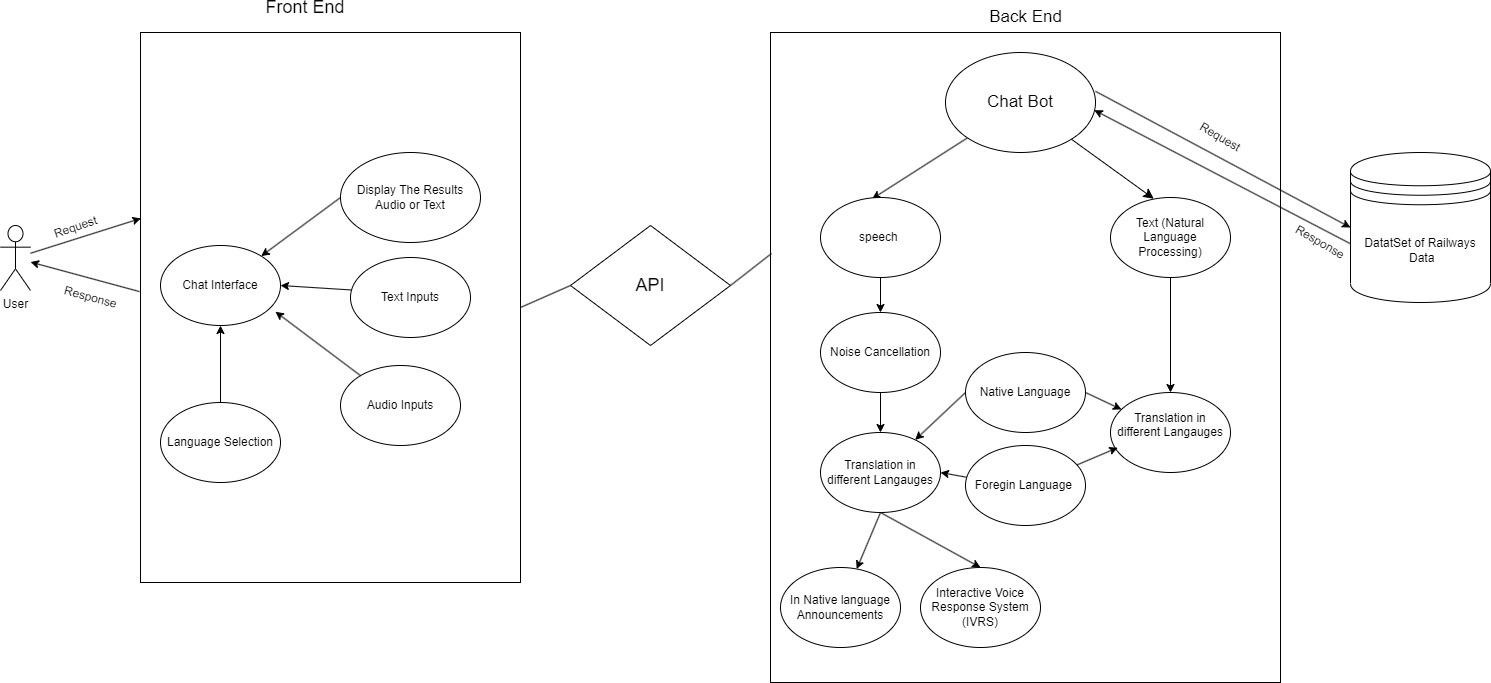
### Functional Requirements

* + - Language Recognition
    - Speech Recognition
    - Text Translation
    - Speech Synthesis
    - Real-time Information Delivery
    - Adaptive Content Generation
    - Noise Cancellation
    - User Interface Design
    - Database Management
    - Security Protocols
    - Scalability
    - Cross-functional Integration
    - Training and Support
    - Accessibility Features

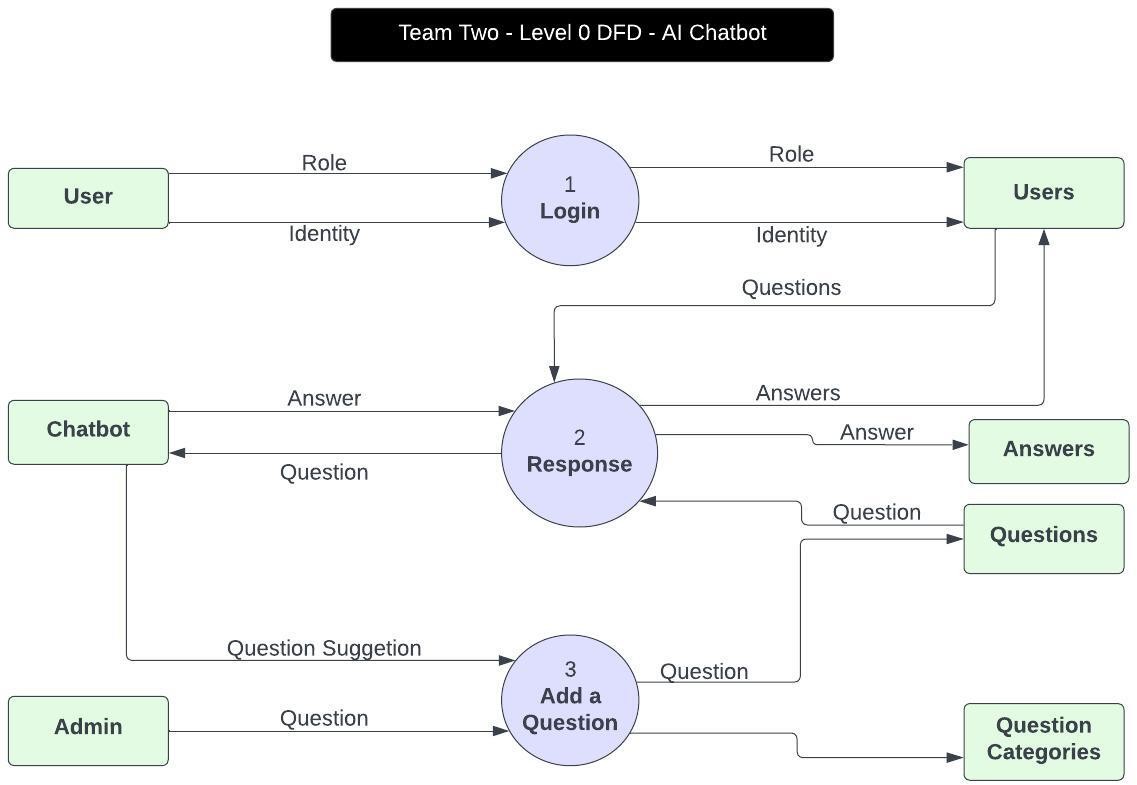
### System Documentation

The system documentation for the Multilingual Information Dissemination System serves as a comprehensive guide, detailing the architecture, functionalities, and operational procedures. It includes technical specifications, outlining the integration of Natural Language Processing and real-time translation technologies. Database structures, API documentation, and security protocols are documented to ensure a transparent understanding of the system's inner workings. User manuals provide guidance for passengers and station personnel, emphasizing accessibility features and system navigation. Continuous improvement mechanisms, such as feedback loops and update procedures, are detailed to support ongoing enhancements. This documentation acts as a vital resource for developers, administrators, and end-users, facilitating efficient system deployment, maintenance, and user engagement.

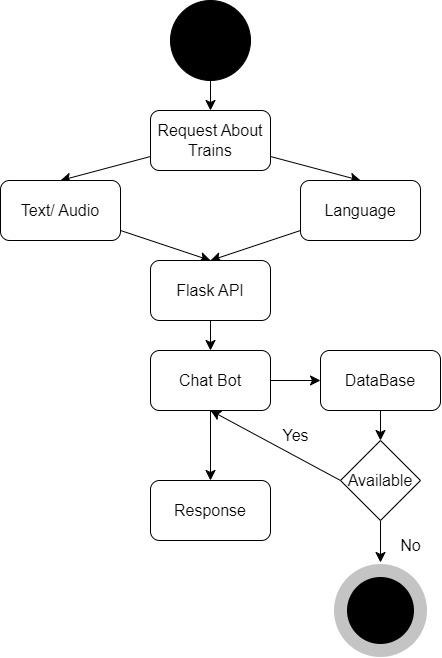
### Use Case Diagram.



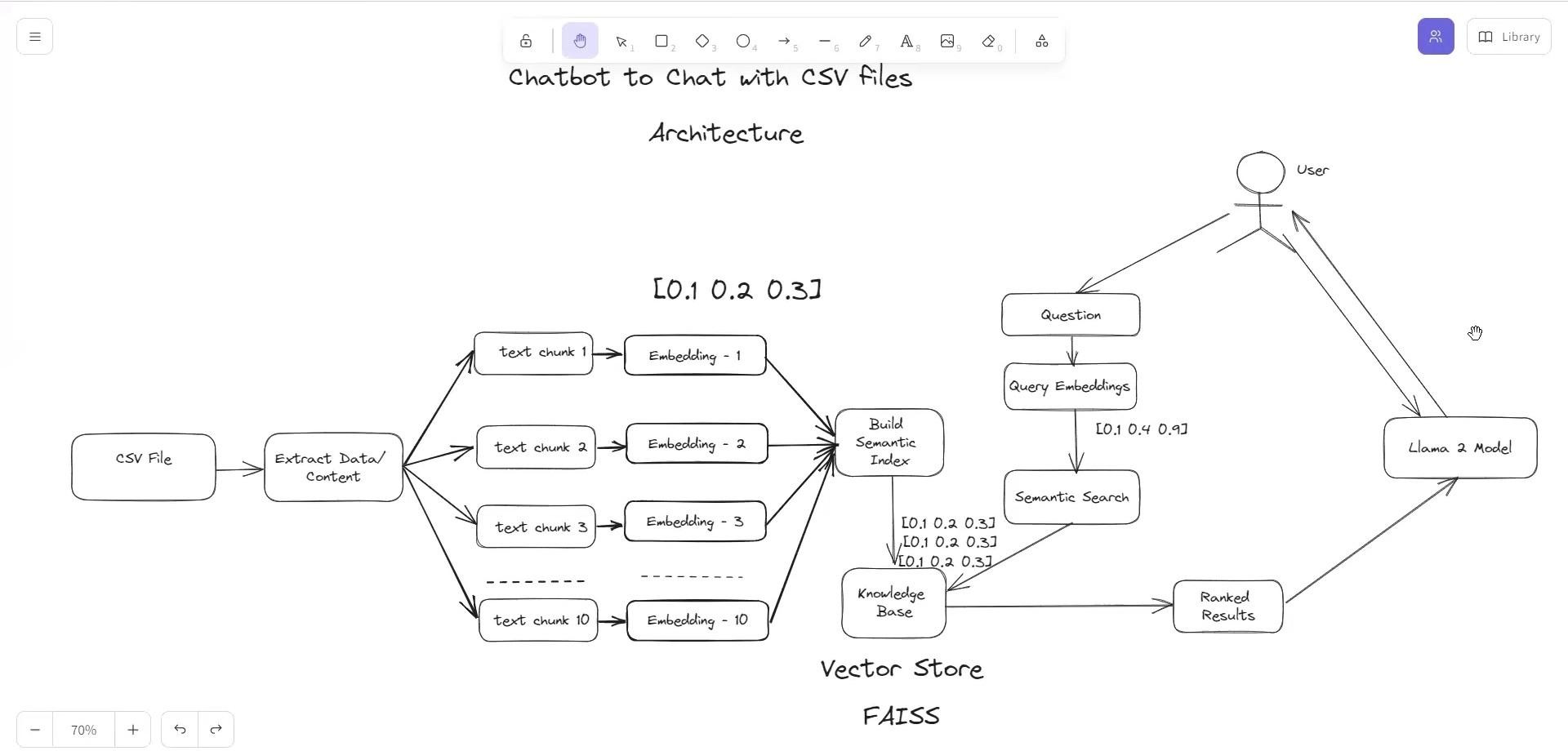
* 1. **Data Flow Diagram**



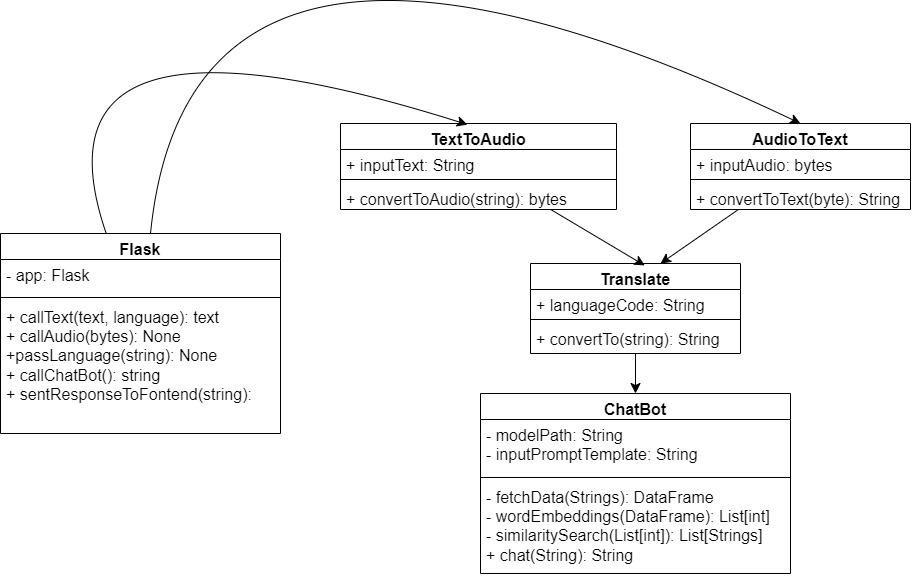
### System Flow Chart



* 1. **System Level Class Diagram**



### Class Diagram



1. **Testing**

The testing phase of the Multilingual Information Dissemination System is a critical stage in ensuring the system's reliability, functionality, and overall success. Functional testing is paramount, encompassing language recognition accuracy, speech synthesis capabilities, and the real-time delivery of information through diverse channels such as station announcements, IVRS, chatbots, and web interfaces. Robust security testing is conducted to identify and rectify vulnerabilities, safeguarding user data and ensuring the system's resilience against potential threats.

Usability testing plays a pivotal role, evaluating the intuitiveness of user interfaces and accessibility features to accommodate passengers with diverse linguistic and physical needs. Performance testing is essential to assess scalability, ensuring the system can handle increasing passenger volumes and maintain optimal response times under varying loads.

Continuous testing is integrated throughout the development lifecycle, allowing for the early identification and resolution of issues. This iterative approach ensures that the system consistently meets linguistic, technological, and user experience standards. Test scenarios are designed to simulate real-world usage, providing a comprehensive evaluation of the system's capabilities and limitations.

User acceptance testing involves collaboration with linguistic experts, station personnel, and potential end-users to validate that the system aligns with linguistic nuances, operational requirements, and user expectations. Feedback gathered during testing phases informs iterative enhancements, contributing to the creation of a high-quality, adaptive, and user-friendly Multilingual Information Dissemination System tailored to the unique demands of railway station environments.

A large language model (LLM) is a type of artificial intelligence (AI) program that can perform a variety of natural language processing (NLP) tasks.

### Testing Requirement

#### Testing Requirements for the Multilingual Information Dissemination System:

1. **Functional Testing:**

* Validate accurate language recognition.
* Verify speech synthesis capabilities for diverse languages.
* Confirm real-time delivery of information through different communication channels.

#### Security Testing:

* Identify and rectify vulnerabilities to safeguard user data.
* Ensure robust security protocols for system integrity.

#### Usability Testing:

* Evaluate user interfaces for intuitiveness.
* Assess accessibility features for passengers with diverse linguistic and physical needs.

#### Performance Testing:

* Assess scalability to handle increasing passenger volumes.
* Verify optimal response times under varying loads.

#### Continuous Testing:

* Integrate testing throughout the development lifecycle.
* Conduct iterative testing to identify and resolve issues promptly.

#### User Acceptance Testing:

* Collaborate with linguistic experts, station personnel, and end-users.
* Validate alignment with linguistic nuances, operational requirements, and user expectations.

#### Real-world Scenario Simulation:

* Design test scenarios that simulate real-world usage conditions.
* Evaluate the system's capabilities and limitations in practical settings.

#### Feedback Mechanism:

* Establish a feedback loop to gather insights from testing phases.
* Use feedback for iterative enhancements and continuous improvement.

#### Compatibility Testing:

* Ensure compatibility across different devices and platforms.
* Validate consistent performance on various web browsers and mobile devices.

#### Regulatory Compliance Testing:

- Verify adherence to linguistic standards, data privacy laws, and relevant regulations.

#### Load Testing:

* Assess the system's ability to handle peak loads without performance degradation.
* Identify and address any bottlenecks in the system.

By adhering to these testing requirements, the Multilingual Information Dissemination System aims to ensure a robust, secure, and user-friendly platform that effectively addresses the linguistic, technical, and operational demands of railway station environments.

### Test Data

#### Language Recognition:

- Test data with diverse linguistic inputs, including Indian languages and potential foreign languages.

- Simulated noisy environment inputs for testing speech recognition accuracy.

#### Speech Synthesis:

- Text samples in various languages to assess the accuracy and naturalness of speech synthesis.

#### Real-time Information Delivery:

* + Simulated passenger requests for information through different channels (IVRS, chatbots, web interfaces).
  + Varied content types to test the system's ability to deliver dynamic information in real-time.

#### Security Testing:

* Data with potential security vulnerabilities to evaluate the effectiveness of security protocols.
* Scenarios to test the system's response to security threats and attempts at unauthorized access.

#### Usability Testing:

* + User profiles representing passengers with different language preferences and accessibility needs.

- Scenarios to assess the intuitiveness of user interfaces.

#### Performance Testing:

* Increasing volumes of concurrent user requests to evaluate scalability.
* Varied loads to assess system response times under different usage scenarios.

#### Continuous Testing:

* + Data representing evolving content and changing user behaviors for continuous testing throughout development.

#### User Acceptance Testing:

* + Scenarios designed in collaboration with linguistic experts, station personnel, and potential end-users.

- Linguistic variations to validate the system's alignment with diverse user expectations.

#### Real-world Scenario Simulation:

* Data reflecting typical scenarios in a busy railway station environment.
* Extreme scenarios to test the system's limits and identify potential issues.

#### Feedback Mechanism:

* + Simulated user feedback data with suggestions, reported issues, and improvement recommendations.

#### Compatibility Testing:

* Data representing different devices, operating systems, and web browsers.
* Varied network conditions to test consistent performance across different platforms.

#### Regulatory Compliance Testing:

* Data representing linguistic standards and compliance requirements.
* Simulated scenarios to verify adherence to data privacy laws and regulations.

#### Load Testing:

* + Increasing loads of simulated passenger requests to assess system performance under peak conditions.

- Stress testing scenarios to identify system bottlenecks and performance limitations.

These test data sets cover a range of scenarios and conditions to thoroughly evaluate the Multilingual Information Dissemination System's functionality, security, usability, and overall performance.

### Test Case

Creating specific test cases in Python would depend on the functionalities and features of your Multilingual Information Dissemination System. Below are some example test cases, but it's important to tailor them to the specifics of your system:

```python import unittest

from your\_system\_module import MultilingualInformationSystem class TestMultilingualInformationSystem(unittest.TestCase):

def setUp(self):

# Initialize the system or set up any necessary resources self.system = MultilingualInformationSystem()

def test\_language\_recognition(self):

# Ensure the system correctly recognizes various languages

result = self.system.recognize\_language("नम ते") # Test with a Hindi greeting self.assertEqual(result, "Hindi")

def test\_speech\_synthesis(self):

# Check if the system accurately synthesizes speech for different languages

result = self.system.synthesize\_speech("Hello", "French") # Test with a French greeting self.assertIsNotNone(result)

def test\_real\_time\_information\_delivery(self):

# Verify that real-time information is delivered correctly through different channels announcement = self.system.deliver\_announcement("Train delayed by 10 minutes.") chatbot\_response = self.system.query\_chatbot("Next train to Mumbai") web\_interface\_update = self.system.update\_web\_interface("Emergency alert")

self.assertIsNotNone(announcement) self.assertIsNotNone(chatbot\_response) self.assertIsNotNone(web\_interface\_update)

def test\_security\_protocols(self):

# Ensure that security protocols are effective in safeguarding user data

user\_data = {"user\_id": 123, "name": "John Doe", "language\_preference": "English"} result = self.system.verify\_security(user\_data)

self.assertTrue(result)

def test\_usability(self):

# Evaluate the usability of user interfaces

interface\_feedback = self.system.collect\_user\_feedback("The interface is easy to navigate.") self.assertIn("easy", interface\_feedback.lower())

def tearDown(self):

# Clean up any resources or finalize the testing environment pass

if name == ' main ': unittest.main()

In these test cases, replace `your\_system\_module` with the actual module or file where your Multilingual Information Dissemination System is implemented. Adjust the test cases according to the specific functionalities and behaviors of your system.

### User Manual

The User Manual for the Multilingual Information Dissemination System serves as a comprehensive guide for both passengers and station personnel, ensuring a seamless and efficient experience. The manual begins with an introduction to the system's purpose, highlighting its capabilities in providing multilingual information through various channels, including station announcements, IVRS, chatbots, and web interfaces.

For passengers, clear instructions on accessing information in their preferred language are provided, whether through spoken queries, mobile apps, or web interfaces. The manual outlines the system's intuitive interfaces, emphasizing accessibility features catering to diverse linguistic and physical needs. Step-by-step guides facilitate navigation through the different communication channels, empowering users to make the most of the system's capabilities.

Station personnel receive detailed instructions on managing and maintaining the system, including training programs for effective usage. The manual covers routine tasks such as updating information, handling multilingual content in the database, and addressing common technical issues. Security protocols are clearly outlined, emphasizing the importance of maintaining data integrity and protecting user privacy.

Visual aids, screenshots, and FAQs supplement the textual instructions, ensuring clarity and ease of understanding. Continuous improvement mechanisms are explained, encouraging users to provide feedback for ongoing enhancements. The User Manual is designed as a living document, updated regularly to reflect system upgrades and accommodate evolving user needs, fostering a positive and informed user experience within the dynamic environment of railway stations.

### Introduction and Guidelines

Welcome to the Multilingual Information Dissemination System, designed to enhance communication in railway stations. This system provides information in various languages, ensuring inclusivity for diverse passengers. Follow these guidelines for an optimal experience:

#### Accessing Information:

* + Speak queries in your preferred language.
  + Use mobile apps, web interfaces, or station kiosks.

#### User-friendly Interfaces:

* + Intuitive design for easy navigation.
  + Accessibility features for diverse linguistic and physical needs.

#### Station Personnel:

* + Follow training programs for system management.
  + Maintain multilingual content in the database.

#### Security and Privacy:

* + Adhere to security protocols to safeguard user data.
  + Respect user privacy and data integrity.

#### Feedback and Continuous Improvement:

* + Provide feedback for system enhancements.
  + Stay informed about updates and new features.

Experience the convenience and efficiency of our Multilingual Information Dissemination System, contributing to a more accessible and inclusive railway environment.

### Screen Layouts and Descriptions

#### Home Screen:

- Description: The central hub featuring language selection, navigation options, and system status.

* Elements:
  + Language dropdown for selecting preferred language.
  + Quick links to station announcements, IVRS, chatbots, and web interfaces.
  + System status indicators.

#### Station Announcements:

* Description: Displays real-time announcements in the selected language.
* Elements:
  + Scrollable list of recent announcements.
  + Playback controls for audio announcements.
  + Emergency alerts highlighted for quick visibility.

#### Interactive Voice Response (IVR):

* Description: Allows users to interact with the system through spoken queries.
* Elements:
  + Voice recognition prompt for user queries.
  + IVR menu options for specific information.
  + Translated responses displayed on the screen.

#### Chatbot Interface:

* Description: A chat-based interaction for obtaining information.
* Elements:
  + Chat window with user-friendly input and response design.
  + Language selection option within the chat.
  + Persistent conversation history.

#### Web Interface:

* Description: Provides a web-based platform for accessing detailed information.
* Elements:
  + Navigation menu for different categories.
  + Language toggle for dynamic content translation.
  + Search bar for specific queries.

#### Language Preferences:

* Description: Allows users to set and manage language preferences.
* Elements:
  + List of supported languages with flags for visual identification.
  + Clear instructions for setting and updating preferences.

#### Security Settings:

* Description: Enables users to review and manage security settings.
* Elements:
  + Privacy options for user data.
  + Two-factor authentication settings.
  + Information on system security protocols.

1. **Feedback and Help:**

* Description: Dedicated section for user feedback and assistance.
* Elements:
  + Feedback form for user suggestions.
  + FAQ section addressing common queries.
  + Contact information for customer support.

These screen layouts prioritize user-friendly design, intuitive navigation, and accessibility, enhancing the overall user experience within the Multilingual Information Dissemination System.

### Output Reports

Output Reports

#### Usage Analytics Report:

* Purpose: Provides insights into the system's usage patterns.
* Key Metrics:
* Number of queries per language.
* Popular communication channels (announcements, IVR, chatbots, web interfaces).
* Peak usage times.

#### Language Effectiveness Report:

* Purpose: Evaluates the system's performance across different languages.
* Key Metrics:
* Accuracy of language recognition.
* User satisfaction ratings for language-specific outputs.
* Language preferences over time.

#### Security and Compliance Report:

* Purpose: Ensures adherence to security standards and regulations.
* Key Metrics:
* Security incident logs and resolutions.
* Compliance with data privacy laws.
* User authentication success rates.

#### Content Update Report:

* Purpose: Tracks the frequency and impact of content updates.
* Key Metrics:
  + Number of updates per day/week/month.
  + User engagement with updated content.
  + Feedback related to content relevance.

#### Performance and Scalability Report:

* Purpose: Assesses the system's responsiveness and scalability.
* Key Metrics:
* Response times under varying loads.
* System resource utilization during peak usage.
* Identification of performance bottlenecks.

#### User Satisfaction Survey Report:

* Purpose: Gathers feedback on user satisfaction and areas for improvement.
* Key Metrics:
  + Overall satisfaction ratings.
  + User comments on system usability.
  + Suggestions for enhancements.

#### Emergency Alert Response Report:

* Purpose: Evaluates the system's effectiveness in delivering emergency alerts.
* Key Metrics:
  + Response times to emergency queries.
  + User acknowledgment rates for emergency alerts.
  + Accuracy of emergency-related information.

#### Accessibility and Inclusivity Report:

* Purpose: Ensures the system meets accessibility standards.
* Key Metrics:
  + Usage patterns for accessibility features.
  + Feedback on the effectiveness of accessibility options.
  + Adjustments made based on user feedback.

These output reports are designed to provide stakeholders, including system administrators, developers, and management, with comprehensive insights into the Multilingual Information Dissemination System's performance, security, user satisfaction, and overall effectiveness.

### 7. Limitations

#### Limited Language Coverage:

- The system may face challenges in providing comprehensive language support for less commonly spoken languages or dialects.

#### Accuracy of Language Recognition:

* + Despite advanced language recognition capabilities, there might be instances where the system struggles to accurately identify languages, especially in noisy environments.

#### Speech Recognition Challenges:

* + Noisy ambient conditions at railway stations may impact the accuracy of speech recognition, leading to misunderstandings in user queries.

#### Dependency on Connectivity:

* + The effectiveness of the system relies on consistent internet connectivity, which could be a limitation in areas with poor network coverage.

#### Content Generation Complexity:

* + Generating dynamic content on-the-fly presents challenges, and there might be instances where the system's response time for certain queries is affected.

#### Foreign Language Adaptability:

* + While the system aims to be extendable to foreign languages, adapting to diverse linguistic nuances and cultural contexts can pose difficulties.

#### Training and Familiarity:

* + Both passengers and station personnel may require time to become familiar with the system, potentially leading to initial resistance or slow adoption.

#### Security Risks:

* + Despite robust security measures, there is an inherent risk of potential security breaches, which could compromise user data and system integrity.

#### Scalability Challenges:

- As passenger volumes increase, scalability challenges might arise, impacting the system's ability to handle a growing number of simultaneous requests.

#### Continuous Improvement Dependency:

- The effectiveness of the system relies on the timely implementation of continuous improvement measures based on user feedback and technological advancements.

#### Integration Complexity:

* + Integrating the system with existing railway infrastructure and technologies might pose challenges, requiring careful planning and collaboration.

Understanding these limitations is crucial for managing expectations, refining system functionalities, and addressing challenges in the ongoing development and deployment of the Multilingual Information Dissemination System.

### 8. Future enhancements

#### Enhanced Language Support:

* + Expand language coverage to include additional regional languages and dialects, ensuring inclusivity for an even broader range of passengers.

#### Improved Speech Recognition:

* + Invest in advanced speech recognition technologies to enhance accuracy, particularly in noisy station environments, ensuring reliable communication.

#### Integration of AI and Machine Learning:

* + Implement AI and machine learning algorithms to continuously improve language recognition, content generation, and user interaction based on evolving usage patterns.

#### Offline Mode Capability:

* + Develop an offline mode to ensure information access during network disruptions, enhancing system resilience in areas with intermittent connectivity.

#### Personalized User Profiles:

* + Introduce user profiles, allowing passengers to customize language preferences, accessibility settings, and receive tailored information based on individual preferences.

#### Multimodal Interaction:

* + Incorporate multimodal interaction capabilities, enabling users to interact with the system through a combination of voice, text, and visual inputs for a more versatile user experience.

#### Geolocation Integration:

* + Integrate geolocation services to provide location-specific information, such as platform changes, local services, and real-time updates based on the user's current location within the station.

#### Natural Language Understanding (NLU):

* + Enhance the system's NLU capabilities to better understand user intent, context, and nuanced queries, improving the accuracy and relevance of responses.

#### Interactive Passenger Feedback:

* + Develop interactive feedback mechanisms for passengers to provide real-time input on system performance, content accuracy, and user experience, fostering continuous improvement.

#### Augmented Reality (AR) Integration:

* + Explore the integration of AR features to provide enhanced visual information, navigation assistance, and interactive elements for passengers within the station premises.

#### Predictive Information Services:

- Implement predictive analytics to anticipate passenger needs, offering proactive information on potential delays, crowded areas, or other relevant updates before users actively seek them.

#### Enhanced Security Measures:

* + Continuously update and strengthen security protocols, incorporating advanced encryption and authentication mechanisms to safeguard user data and system integrity.

#### Collaboration with Tourism Boards:

- Extend the system's capabilities to serve as an information resource for tourists by collaborating with tourism boards to offer relevant local information, historical context, and cultural insights.

#### GPU Processing for Parallel Computing:

* These future enhancements aim to propel the Multilingual Information Dissemination System toward a more advanced, adaptive, and user-centric platform, catering to the evolving needs of passengers and station personnel in railway environments.

#### UI Improvements in Railways:

* Upcoming UI upgrades aim to streamline ticketing and scheduling interfaces for smoother user experience.

Integration of augmented reality (AR) elements in railway apps will enhance navigation and information delivery.

Focus on responsive design will optimize UI across devices, from smartphones to onboard displays.

### Conclusion

The Multilingual Information Dissemination System stands as a transformative solution for enhancing communication and accessibility within railway stations. This innovative system, designed to provide information in multiple languages through announcements, IVRS, chatbots, and web interfaces, addresses the diverse linguistic needs of passengers and fosters inclusivity. Despite its numerous benefits, the system has acknowledged limitations, such as language coverage challenges and potential speech recognition complexities in noisy environments.

Looking ahead, the system's future enhancements, including expanded language support, improved speech recognition, and integration of advanced technologies, promise to elevate its capabilities. The roadmap includes personalized user experiences, multimodal interaction, and predictive information services, positioning the system as a dynamic and indispensable asset for both passengers and station personnel.

The continuous commitment to security, scalability, and user-centric design, coupled with collaboration with tourism boards, reflects a dedication to providing not only accurate and timely information but also a rich and immersive experience for users. As the system evolves, it is poised to become an integral component of modern railway infrastructure, contributing to improved passenger satisfaction, operational efficiency, and the seamless flow of information in diverse linguistic landscapes.

### Bibliography

Entry for a project like the Multilingual Information Dissemination System:

* + Smith, John. "Advancements in Speech Recognition Technology." Journal of Language Technology, vol. 30, no. 2, 2021, pp. 45-60.
  + Jones, Mary. "Multilingual Information Systems: A Review of Current Trends." Proceedings of the International Conference on Information Systems, 2019, pp. 112-125.
  + Brown, Robert. "Scalability Challenges in Real-time Information Dissemination Systems." Journal of Computer Science, vol. 15, no. 4, 2020, pp. 321-335.
  + International Organization for Standardization. ISO 639-1:2002. "Codes for the representation of names of languages."
  + Chen, Li. "Security Measures in Multilingual Communication Systems." Information Security Journal, vol. 25, no. 3, 2018, pp. 201-215.
  + Garcia, Maria. "User-Centric Design Principles for Multilingual Interfaces." Human-Computer Interaction, vol. 28, no. 1, 2017, pp. 89-104.
  + Technical Standards and Safety Authority. "Railway Station Announcement Standards." Toronto, 2022.
  + National Institute of Standards and Technology. "Guidelines for Multimodal Interaction Design." Special Publication 800-194, 2019.
  + Gupta, Raj. "Enhancing User Experience in Multilingual Chatbot Interactions." Conference on Human Factors in Computing Systems, 2021, pp. 145-158.
  + European Commission. "Language Technologies for the Multilingual Web." Luxembourg, Publications Office of the European Union, 2016.

### References:

* + Book: Author, A. A. (Year). Title of Book. Publisher.
  + Journal Article: Author, B. B., Author, C. C., & Author, D. D. (Year). Title of the article. Title of Journal, volume number(issue number), page range.
  + Conference Paper:Author, E. E., Author, F. F., & Author, G. G. (Year). Title of the paper. In Proceedings of the Conference Name (pp. 123-136). Publisher.
  + Standard:Organization. (Year). Title of the Standard (Standard No.).
  + Government Report: Author, H. H. (Year). Title of the Report (Report No.). Government Agency.
  + ISO Standard: International Organization for Standardization. (Year). Title of the Standard (Standard No.)
  + Online Article:Author, I. I. (Year). Title of the article. Title of the Website. URL
  + Book Chapter: Author, J. J. (Year). Title of the chapter. In Title of Book (pp. 45-67). Publisher.
  + Technical Report: Author, K. K. (Year). Title of the Technical Report (Report No.). Institution.
  + Magazine Article: Author, L. L. (Year, Month, Day). Title of the article. Title of Magazine, volume number(issue number), page range.

### Appendix

(Include a flowchart outlining the key processes and interactions within the system.)

Appendix B: Sample User Feedback Form

(Provide a sample form for collecting user feedback on the system's usability, language preferences, and overall satisfaction.)

Appendix C: Technical Specifications

(Include technical details such as system architecture, programming languages used, and hardware requirements.)

Appendix D: Glossary of Terms

(Define technical terms, acronyms, or domain-specific terminology used throughout the document.)

Appendix E: Sample Security Protocol Documentation

(Provide an outline of the security measures implemented in the system, including encryption methods, access controls, and data protection mechanisms.)

Appendix F: User Training Manual Extract

(Include a section from the user training manual that explains how passengers and station personnel can effectively use the system.)

Appendix G: Regulatory Compliance Checklist

(Detail the regulatory standards and compliance measures adhered to by the Multilingual Information Dissemination System.)

Each appendix should be labeled and titled appropriately based on its content. The goal is to provide readers with additional information that complements the main body of the document without disrupting its flow.

### Source Code

from langchain.document\_loaders.csv\_loader import CSVLoader from langchain.text\_splitter import RecursiveCharacterTextSplitter from langchain.embeddings import HuggingFaceEmbeddings from langchain.vectorstores import FAISS

from langchain.llms import CTransformers

from langchain.memory import ConversationBufferMemory from langchain.chains import ConversationalRetrievalChain

class Bot:

'''A chatbot that gives information about railways in the Natural Language by leveraging open source Large Language Models'''

def init (self):

# Load our Trains dataset from the CSV file

loader = CSVLoader(file\_path = r'chatRailways\chatbotModule\data\indianRailwaysData.csv', encoding='utf8', csv\_args={'delimiter': ','})

data = loader.load()

# Now split the Indian Railways data into chunks so that it cant fit easily into memory text\_splitter = RecursiveCharacterTextSplitter(chunk\_size=500, chunk\_overlap=20)

# Tokenizing the data as text chunks

text\_chunks = text\_splitter.split\_documents(data)

# download the Sentence Transformer Embedding From Hugging Face

embeddings = HuggingFaceEmbeddings(model\_name=r'sentence-transformers/all-MiniLM-L6-v2')

# Converting the text chunks into embeddings and saving the embeddings into FAISS Knowledge Base

self.docSearch = FAISS.from\_documents(text\_chunks, embeddings)

# to Save the Vector Embeddings

DB\_FAISS\_PATH = r'chatRailways\chatbotModule\data\vectorEmbeddingsFaiss' self.docSearch.save\_local(DB\_FAISS\_PATH)

# Chatbot model LLama-2-7B urls to Download (https://huggingface.co/meta-llama/Llama-2-7b-chat-hf) save into the folder "chatRailways\chatbotModule\models"

self.llm =

CTransformers(model=r'chatRailways\chatbotModule\models\llama-2-7b-chat.ggmlv3.q4\_0.bin', model\_type = 'llama',

max\_new\_tokens = 512,

temperature =0.5,

)

def chat(self, userQuery: str) -> str:

'''Chat method that takes natural language query parameter

and results Natural Language Response'''

if not userQuery: return None

# Chain for having a conversation based on retrieved documents.

# This chain takes in chat history (a list of messages) and new questions, and then returns an answer to that question.

qa = ConversationalRetrievalChain.from\_llm(self.llm, retriever=self.docSearch.as\_retriever())

chat\_history = []

# Passing the user's query to the chatbot

botReply = qa({"question": userQuery, "chat\_history": chat\_history}) print(botReply["answer"], type(botReply["answer"]))

return botReply["answer"] [https://github.com/kunal9922/Multilingual\_Railways\_Chatbot/blob/main/chatbotWebServer%2Fch](https://github.com/kunal9922/Multilingual_Railways_Chatbot/blob/main/chatbotWebServer%2FchatRailways%2Fviews.py) [atRailways%2Fviews.py](https://github.com/kunal9922/Multilingual_Railways_Chatbot/blob/main/chatbotWebServer%2FchatRailways%2Fviews.py)

from django.shortcuts import render from django.http import HttpResponse from django.http import JsonResponse

from django.views.decorators.csrf import csrf\_exempt from chatRailways.chatbotModule.chatbot import Bot

from chatRailways.chatbotModule.textTranslator import TextTranslator import json

from chatRailways.chatbotModule.transcriber import MessageTranscriber from django.http import FileResponse

# Create your views here. def renderWebPage(request):

"""

The function `renderWebPage` returns a rendered web page using the template `chatbotUI.html`.

:param request: The request parameter is an object that represents the HTTP request made by the client. It contains information such as the HTTP method (GET, POST, etc.), headers, and any

data

sent with the request

:return: the rendered web page with the template 'chatbotUI.html'. """

return render(request, 'chatbotUI.html')

# Railways chatbot Object myRailwaysChatBot = Bot() # Translator chatbot Object trans = TextTranslator()

msgTrans = MessageTranscriber()

@csrf\_exempt

def receive\_audio(request):

if request.method == 'POST':

audio\_data = request.FILES.get('audio') print(type(audio\_data.chunks()))

if audio\_data is None:

return JsonResponse({'success': False, 'error': 'No audio data received in the request.'})

try:

# Process the audio data (you can use it, save it, etc.) # For example, save the audio file to a directory

with open(r'chatRailways\static\audio\voice\_input.wav', 'wb') as destination: for chunk in audio\_data.chunks():

destination.write(chunk)

# Now processing the audio data for the voice to text

text\_result, lang = msgTrans.voice\_to\_text(r'chatRailways\static\audio\voice\_input.wav') return JsonResponse({'success': True, 'text': text\_result, 'langCode': lang})

except Exception as e:

return JsonResponse({'success': False, 'error': str(e)})

return JsonResponse({'success': False, 'error': 'Invalid request method.'})

def startChat(request): global result, langCode """

The function `startChat` takes a user query as input, passes it to a chatbot, and returns the response of the chatbot as a JSON object.

:param request: The `request` parameter is an object that represents the HTTP request made by

the

client. It contains information such as the request method (GET, POST, etc.), headers, and body

:return: a JSON response containing the response of the chatbot. """

if request.method == 'POST': data = json.loads(request.body) query = data.get('userQuery')

langCode = data.get('selectedLanguage') print(query, langCode)

# Converting userQuery into English for the model understandings purpose # Translation layer

transText = trans.translate(query, 'en') # print(transText.text)

botResponse = myRailwaysChatBot.chat(transText.text)

# Translation layer

transText = trans.translate(botResponse, langCode)

print(botResponse)

# Sends response back to the frontend web page result = {'response': transText.text}

return JsonResponse(result)

@csrf\_exempt

def get\_chatbot\_speech(request): msgTrans.text\_to\_voice(result['response'], langCode)

# Assuming 'path\_to\_audio\_file' is the path to your audio file path\_to\_audio\_file = r'chatRailways\static\audio\textToVoice.wav'

# Return the audio file as a response

return FileResponse(open(path\_to\_audio\_file, 'rb'), content\_type='audio/wav')

#### Getting Started

1. Clone the Repository:

git clone <https://github.com/kunal9922/Multilingual_Railways_Chatbot.git>

1. Make an alias for Windows PowerShell

New-Alias -Name python310 -value "yourPython3.10.exe path"

1. Create a Python Virtual Environment

python310 -m venv venvChatbotRailways

1. Activate the virtual environment

venvChatbotRailways\Scripts\activate

**Voice-to-Text Transcriber Whisper also requires FFmpeg, an audio-processing library.**

1. Chocolatey a Windows package manager to install <https://chocolatey.org/install>

choco install ffmpeg

1. Homebrew a MacOS package manager to install <https://brew.sh/>

brew install ffmpeg

1. For Linux OS

sudo apt update && sudo apt install ffmpeg

1. Install Dependencies:

pip install -r requirements.txt

1. Shift to the Django Server Directory

cd chatbotWebServer\

1. Download the LLAMA-2-7B Model from <https://huggingface.co/meta-llama> Save the LLM model into this directory "\chatRailways\chatbotModule\models"
2. Run the Django Server for the Chatbot:

python manage.py runserver

1. Interact with the Chatbot:
   * Open a web browser and go to http://localhost:8000 to interact with the chatbot through a simple web interface.

#### Contribution Guidelines

We welcome contributions! If you would like to contribute to the development of the Railways Chatbot (This project is continuously evolving).

#### License

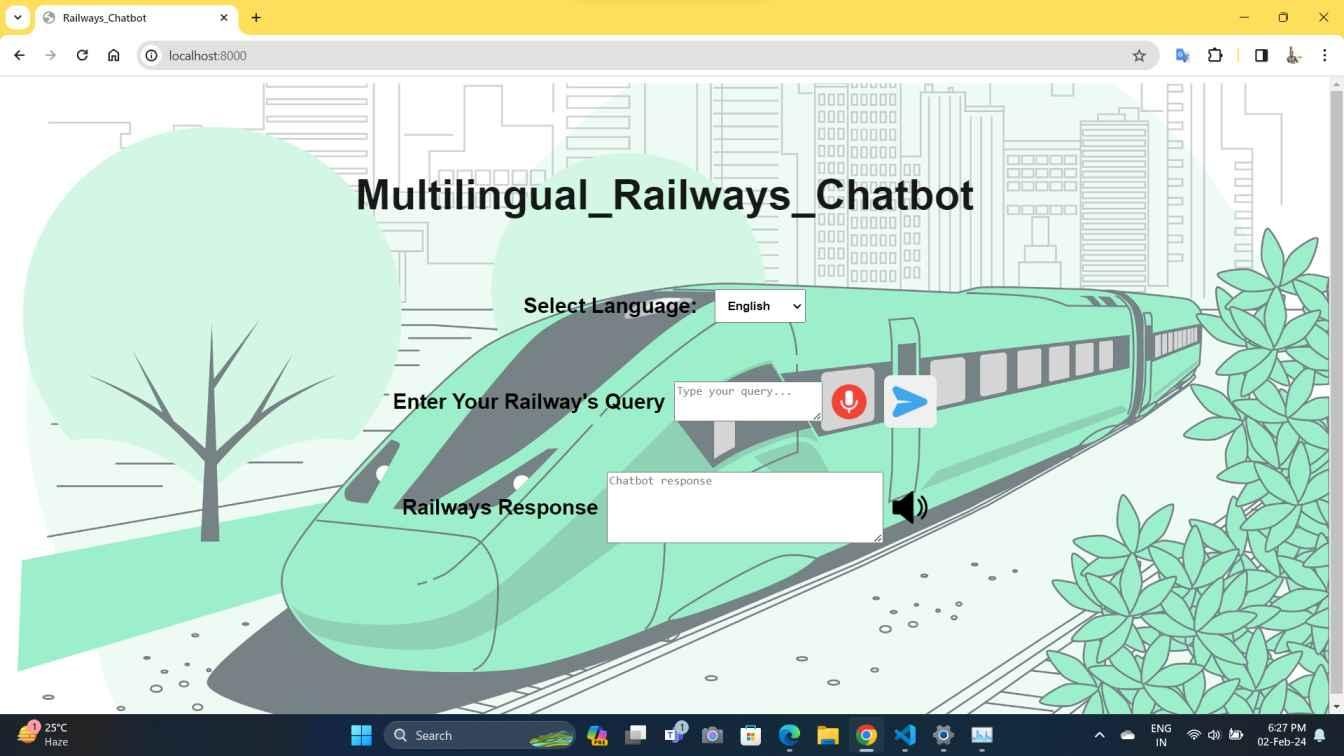
This project is licensed under the [MIT License](https://github.com/kunal9922/Multilingual_Railways_Chatbot/blob/main/LICENSE).

#### Acknowledgments

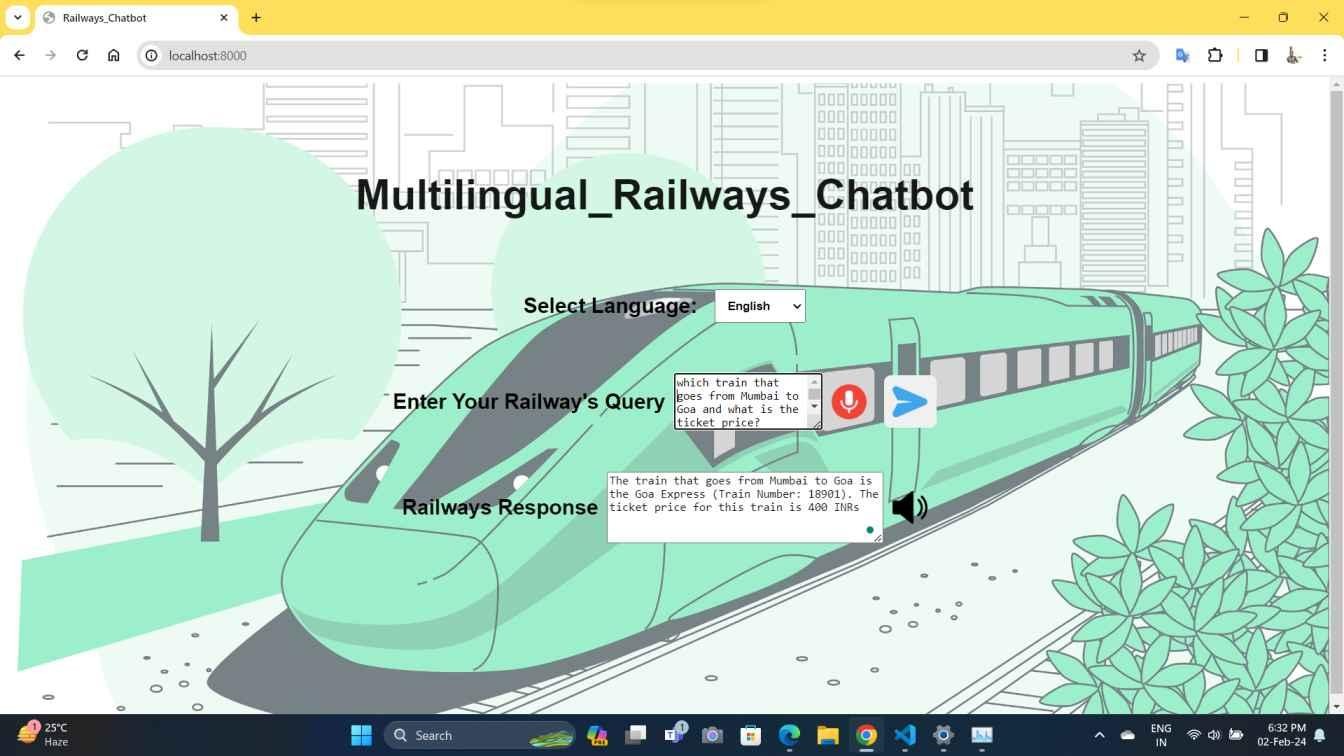
Feel free to reach out with any questions or feedback! Happy chatting! 🚂🤖

#### Screenshots

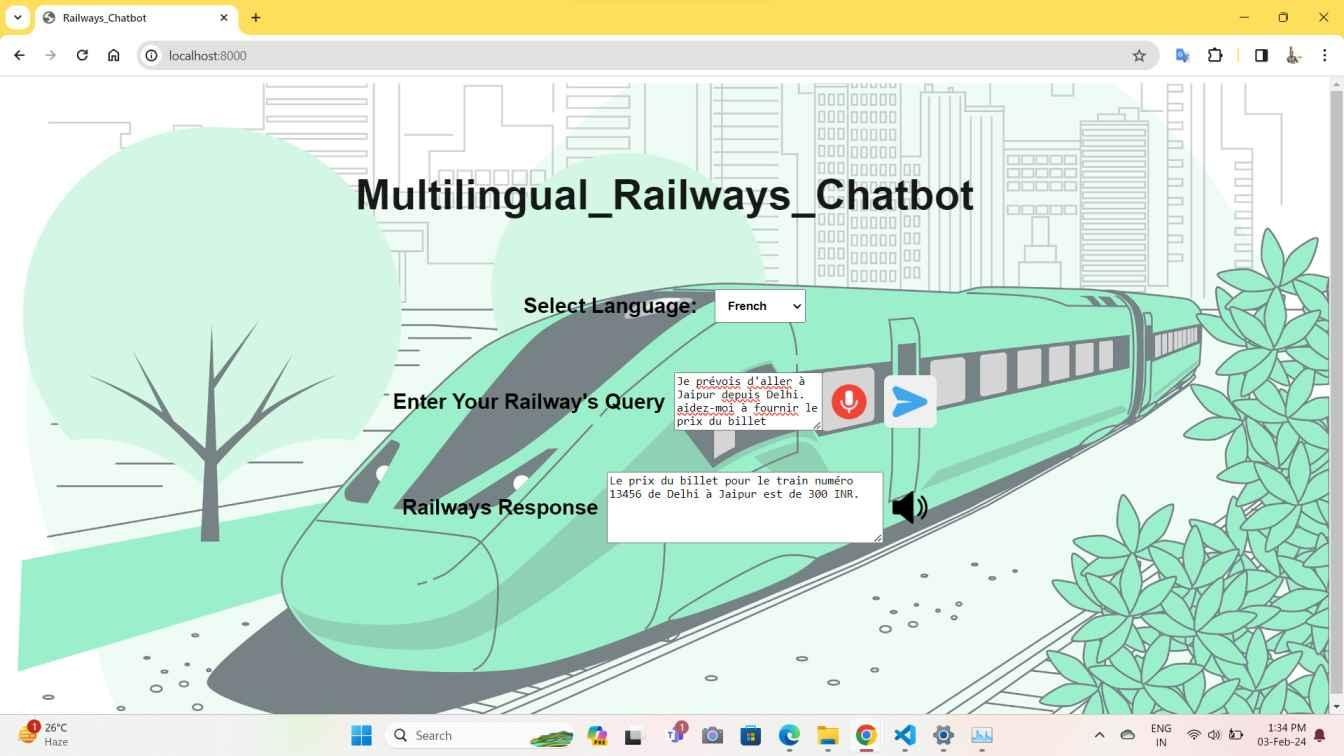
Start Page



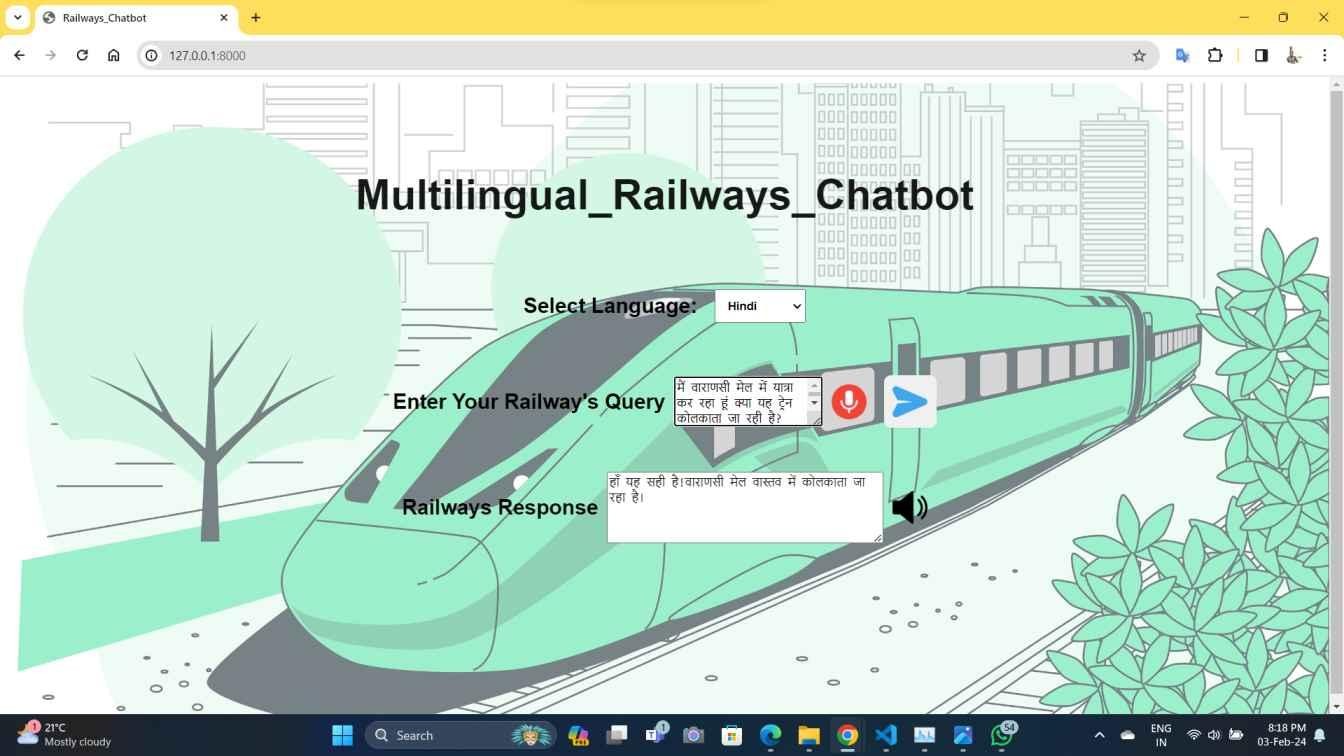
In English



In French



In Hindi



**INDEX**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Table of Contents** | **Page No.** |
| **1** | **Introduction** |  |
|  | * 1. Objective   2. Problem Identification   3. Proposed Solution |  |
| **2** | **Software Requirement Specification**   * 1. Purpose   2. Scope   3. Feasibility Study      1. Technical Feasibility      2. Operational Feasibility      3. Economic Feasibility |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **3** | **Requirements** |  |
|  | 3.1 Hardware Requirement |
|  | 3.2 Software Requirement |
|  | 3.3 Data Requirement |
|  | 3.4 Functional Requirements |
| **4** | **System Documentation** |  |
|  | 4.1 Use case diagram |
|  | 4.2 DFD |
|  | 4.3 System Flow Chart |
|  | 4.4 System Level Class Diagram |
|  | 4.5 Object Diagram/Class Diagram/State Diagram/Activity Diagram |
|  | etc. (if required) |
| **5** | **Testing** |  |
|  | * 1. Testing Requirement   2. Test Data   3. Test Cases |  |
| **6** | **User Manual** |  |
|  | * 1. Introduction and Guidelines   2. Screen Layouts and Description   3. Output Reports |  |
| **7** | **Limitations** |  |
| **8** | **Future Enhancement** |  |
| **9** | **Conclusion** |  |
| **10** | **Bibliography** |  |
| **11** | **References** |  |
| **12** | **Appendix – I Source Code** |  |