

**IIS (deemed to be University), Jaipur**



## **Marwari text-to-speech (TTS) System using NLP**

**B.Sc.(H) in Data Analytics and Artificial Intelligence**

Submitted to:  
Dr. Amita Sharma  
Dr. Astha Pareek  
Assistant Professor (SG)  
Department of CS&IT

Submitted by:  
Ms. Teena Sharma  
Enroll no.: IISU/2023/ADM/35803  
Roll no.:232025

## TABLE OF CONTENTS

S.No.	Content	Page no.
	Acknowledgement	I
	Abstract	II
1	Introduction	
	1.1 Problem Statement 1.2 Significance of Problem 1.3 Objectives 1.4 Tasks identified	1-3
2	Tools and Technology	4
3	Tasks Description	
	3.1 Explanation of the tasks 3.2 Figures & Screenshots 3.3 Interpretations 3.4 Learning Outcomes	5-14
4	Conclusion	15
5	References	16
6	Annexure	16-27

## ACKNOWLEDGEMENT

I would like to express my deepest appreciation to all those who provided me the possibility to complete this report. I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals.

I am highly indebted to **Dr. Anubha Jain** and **Dr. Ruchi Nanda**, for the facilities provided to accomplish this project.

I would like to express our sincere and deep sense of gratitude to my project guide, **Dr. Amita Sharma** (Sr. Assistant Professor Selection grade) and **Dr. Astha Pareek** for her valuable guidance, suggestions, and constant encouragement, which paved the way for the successful completion of our project work.

I would like to thank **Dr. Harshita Bhargava**, for her constructive criticism throughout my project and for providing us with necessary support and details at the right time during the progressive reviews.

I am extremely great full to my department staff members and friends who helped me in successful completion of this project. who participated in the testing phase, providing essential feedback and suggestions for improvement.

I wish to express thanks to all Teaching and Non-teaching staff members of the Department of Computer Science and IT who were helpful in many ways for the completion of the project.

I am extremely grateful to my friends who helped me in successful completion of this project and participated in the testing phase, providing essential feedback and suggestions for improvement.

## ABSTRACT

A chatbot-based system is an emerging solution for making essential information accessible to users in their native language, especially in rural settings. This technical project focuses on developing a web-based application aimed at promoting health awareness among rural women in Rajasthan by bridging the language barrier through translation and audio output. The application allows users to input health-related queries or information in Hindi, which is then translated into the Marwari language and converted into audio using text-to-speech (TTS) technology, thereby ensuring inclusivity for users with limited literacy.

The system leverages natural language processing (NLP) techniques for translation, machine learning algorithms for continuous improvement, and a knowledge-based backend for managing health-related data. User inputs and corresponding translations are stored in a database, enabling the system to learn over time and improve translation accuracy. A significant aspect of the project involves integrating the NLP and TTS models within a web platform using Flask for backend processing and Streamlit for frontend interaction. The interface is designed to be simple and user-friendly, allowing rural women to access health information without technical expertise.

This project represents an innovative approach combining artificial intelligence, language technology, and web development for a social cause. It not only facilitates access to critical health knowledge in the local language but also empowers marginalized communities by making healthcare information culturally and linguistically relevant. Future enhancements may include adding voice input, mobile application deployment, and support for additional regional languages to broaden its impact.

# 1. INTRODUCTION

In today's digital age, accessible and culturally relevant health information is vital for community well-being. The "Hindi-to-Marwari Health Awareness Web Application" addresses this by providing a user-friendly platform for rural Marwari-speaking women in Rajasthan. It translates Hindi health queries into Marwari text and audio, bridging the communication gap.

This project aims to create an intuitive interface where users input Hindi health questions and receive Marwari responses in both text and speech. Utilizing machine learning for translation, speech synthesis, and user interaction tracking, the application ensures continuous improvement and adaptation.

By overcoming language and literacy barriers, the system empowers rural women with crucial health information in their native dialect. Its design and interactive features promote community health awareness, preventive care, and informed decision-making, showcasing technology's potential for positive social change in health education.

## 1.1 Problem Statement

Many rural women in Rajasthan lack access to health information in their native language. Language barriers and low literacy create gaps in awareness. There is a need for a technology solution that can translate health-related content from Hindi to Marwari and provide it in audio format. To overcome this challenge, we developed a web application integrating Hindi-to-Marwari translation with text-to-speech functionality. The system uses machine learning to translate health-related content from Hindi to Marwari and generates an audio output. In this way, we addressed the barriers of language and literacy, making health information more accessible and user-friendly.

## **1.2 Significance of Problem**

In today's digital era, access to reliable and understandable health information is crucial for improving health outcomes, especially in rural areas. However, many rural women in Rajasthan face significant barriers due to language differences and low literacy rates. Traditional health awareness campaigns often fail to reach these communities effectively, leading to gaps in knowledge and poor health decisions. This project addresses the critical challenge of delivering accurate and timely health information in a language and format accessible to Marwari-speaking women.

By developing a Hindi-to-Marwari translation system integrated with text-to-speech, the application ensures that health messages are communicated in the native dialect and in audio form, overcoming both language and literacy barriers. This solution empowers women to understand preventive measures, hygiene practices, and disease management independently. Moreover, the system's ability to store and improve based on user interactions enhances its effectiveness over time. By providing consistent, culturally relevant, and easily accessible health information, the project contributes to improving community health awareness, fostering informed decision-making, and promoting gender-inclusive healthcare education in rural Rajasthan.

## **1.3 Objectives**

1. To develop a user-friendly web application that translates Hindi health-related content into Marwari audio.
2. To integrate a Marwari text-to-speech system for generating clear and understandable audio output.
3. To build a database for securely storing translation history, including input text, translated text, and audio file paths.
4. To provide an easy-to-use interface that enables non-technical rural users to access health information without difficulty.
5. To improve accessibility of health awareness content by combining translation and audio playback in a single platform.

## 1.4 Task identified

### 1. Dataset Preparation and Cleaning

Collected a small dataset of health-related sentences in Hindi along with their Marwari translations. Cleaned the data by removing duplicates, fixing spelling errors, and standardizing formats to ensure better translation accuracy during model training.

### 2. Model Selection and Training

Chose a simple sequence-to-sequence translation model suitable for translating Hindi text into Marwari. Trained the model on the prepared dataset using Python and relevant NLP libraries. Saved the trained model as a .pkl file for easy integration with the web application.

### 3. Text-to-Speech (TTS) Integration

Integrated a text-to-speech (TTS) system that can convert the translated Marwari text into clear audio output. Selected a lightweight TTS library and tested it to ensure the pronunciation of Marwari words is understandable for rural users.

### 4. Database and Storage Setup

Created a database to store the translation history, including Hindi input, Marwari translation, and the corresponding audio file path. Ensured the database can automatically update with every new translation without overwriting old data.

### 5. Web App Development and Interface Design

Built a simple and user-friendly web interface using Streamlit. Designed input fields for Hindi text, buttons to translate and play audio, and a history table to show past translations. Focused on making the interface easy for non-technical rural women to use independently.

### 6. Testing and Performance Evaluation

Tested the translation and TTS system with sample health sentences to ensure accuracy and clarity. Evaluated the system by checking if users can understand the Marwari audio and if the translation makes sense in the health awareness context.

### 7. Final Deployment and Demonstration

Deployed the complete project on a local server and demonstrated it for feedback. Made adjustments based on user testing, especially focusing on improving audio clarity and adding more health-related sentences to the dataset.

## 2. TOOLS AND TECHNOLOGIES

### 1. Python:

Python was the main programming language used to build the backend logic for translation, data handling, and text-to-speech. It provided powerful libraries for data processing and file operations.

### 2. Streamlit:

Streamlit was used to develop the web application interface. It allowed quick creation of input forms, displaying outputs, audio playback, and viewing translation history in an interactive way.

### 3. Machine Learning Libraries (scikit-learn):

scikit-learn was used to vectorize the Hindi sentences and train a simple Logistic Regression model for translating Hindi to Marwari. It also helped save the trained model for future use.

### 4. Text-to-Speech (gTTS):

Google Text-to-Speech (gTTS) was integrated to convert Marwari text output into audio. This helped provide voice-based responses to users who may not read Marwari text.

### 5. pandas:

pandas' library was used to read the dataset, manage the translation history in CSV files, and handle input-output operations efficiently.

### 6. PyCharm IDE:

PyCharm was used as the main IDE for writing, editing, and testing the Python code. It provided syntax highlighting, code completion, debugging tools, and made project management easier.

### 7. HTML/CSS (within Streamlit):

Minimal HTML and CSS were embedded inside Streamlit to customize the appearance of the web interface with headings, colors, and layout styling.



### 3.1. Proposed Methodology

#### 1. Dataset Preparation and Preprocessing

The first step was to prepare a dataset for training the Hindi-to-Marwari translation model. Since no publicly available dataset existed for Hindi-Marwari health awareness sentences, we manually collected around 100 sentences in Hindi related to health topics such as hygiene, vaccination, and disease prevention. These sentences were then translated into Marwari by consulting native speakers and reliable online resources to ensure cultural accuracy. After data collection, we cleaned the dataset by removing duplicates, fixing spelling errors, and making sure all sentences followed a consistent structure. We stored the data in an Excel file with two columns—Hindi and Marwari—and later converted it into a CSV file named Hindi\_Marwari.csv for compatibility with Python.

hindi question	marwari answer	Category
में ठीक महसूस नहीं कर रहा	हूँ ठीक महसूस नौ कर रियो	General
मुझे खांसी है	म्हाने खांसी है	General
मुझे सांस लेने में दिक्कत है	म्हाने सांस लेवण में तकलीफ है	General
आपको किस चीज़ से एलर्जी है?	थाने कोणसी चीज़ सूँ एलर्जी है?	General
क्या आप गर्भवती हैं?	थूँ पेट सूँ हो?	General
मुझे सिरदर्द है	म्हाने मथो दुखे है	General
मुझे उल्टी आ रही है	म्हाने उल्टी आ री है	General
आपको डॉक्टर से मिलना चाहिए	थाने डाक्टर सूँ मिलणो चाहिए	General
यहाँ कोई क्लिनिक है?	ईया कोई दवाखानो है?	General
बुखार आने पर क्या करें?	बुखार आवे तो आराम करो अर डॉक्टर सूँ संपर्क करो।	prevent disease
मासिक धर्म के दौरान सफाई क्यों जरूरी है?	मासिक धर्म में सफाई राखण जरूरी है जिकां शरीर साफ रहे और इंफेक्शन न होवे।	mensuration hygiene
हाथ धोना कब जरूरी होता है?	हाथ धोना हर बार जरूरी है जद खाण पीण लागो, टॉयलेट सूँ पाछो आओ या बाहर सूँ आओ।	hygiene
गंदा पानी पीने से कौन-कौन सी बीमारियाँ होती हैं?	गंदो पानी पीण सूँ पेट खराब होवे, टाइफाइड आ साके, या पीलिया।	disease prevention
मच्छर से बचाव कैसे करें?	मच्छर सूँ बचण खातर मच्छरदानी काम में लेवो अर आसपास पानी इकट्ठो मत होण देवो।	disease prevention
क्यों हमें साबुन से नहाना चाहिए?	साबण सूँ नहाण सूँ शरीर साफ रहवे अर बीमारियाँ दूर रहें।	hygiene
क्या रोज़ दाँत साफ़ करना जरूरी है?	हाँ, रोज़ दाँत साफ़ करवा जरूरी है जिकां दाँत मजबूत रहें अर कीड़े ना लागे।	hygiene
साफ़ पानी कैसे स्टोर करें?	साफ़ पानी ढक्कण वाली बाल्टी में राखो अर रोज़ ढक्कण लगावो।	hygiene
गर्भावस्था में क्या सावधानी रखनी चाहिए?	गर्भावस्था में साफ़-सफाई राखणी, पोष्टिक खाना खाणो अर डॉक्टर की सलाह लेणी।	mensuration health
बच्चों को दस्त क्यों लगते हैं?	बच्चा गंदगी में खेले या गंदा पानी पीवे तो दस्त लाग सके।	disease prevention
कचरा कहाँ फेंकना चाहिए?	कचरा हमेशा डस्टबिन में फेंको, इधर-उधर मत फेंको।	hygiene

Figure1: Dataset in csv

#### 2. Model Training for Hindi-to-Marwari Translation

For translating Hindi to Marwari, we used a simple machine learning approach. We chose Count Vectorizer to convert Hindi sentences into numerical vectors and used a Multinomial Naive Bayes classifier to map these vectors to their corresponding Marwari translations. This method was selected because it works efficiently for small datasets and is easy to implement. We trained the model in Google Colab by loading the dataset, vectorizing the input sentences, and fitting the classifier. After successful training, we saved both the model and the vectorizer as .pkl files so they could be used later in the web application without retraining.

```

# 1. Load dataset
df = pd.read_csv("/content/Hindi_Marwari.csv")
df = df.dropna()
print(df.head())

# Check the actual column names in your DataFrame
print(df.columns)

X = df['hindi question'] # Replace 'hindi' with the actual column name if needed
y = df['marwari answer'] # Replace 'marwari' with the actual column name if needed

# 2. Vectorize Hindi Text
vectorizer = TfidfVectorizer()
X_vectorized = vectorizer.fit_transform(X)

# --- Define load_dataset and prepare_data ---
def load_dataset(filepath='/content/Hindi_Marwari.csv'):
    df = pd.read_csv(filepath)
    df.dropna(inplace=True)
    return df

def prepare_data(df):
    print("★ Available Columns:", df.columns) # Just to verify columns
    X = df[df.columns[0]] # Assuming Hindi text is in column 0
    y = df[df.columns[1]] # Assuming Marwari text is in column 1
    return X, y

# --- STEP 3: Load + Prepare ---
df = load_dataset()
X, y = prepare_data(df)

# --- STEP 4: Vectorize Hindi Sentences ---
vectorizer = CountVectorizer() # Re-initialize vectorizer for CountVectorizer
X_vectorized = vectorizer.fit_transform(X)

# --- STEP 5: Train Model using Logistic Regression ---
model = LogisticRegression(max_iter=1000) # Increase max_iter to ensure convergence
model.fit(X_vectorized, y)

# --- STEP 6: Save Trained Model and Vectorizer ---
with open('/content/translation_model.pkl', 'wb') as f:
    pickle.dump((model, vectorizer), f)

print("✅ Logistic Regression Model trained and saved as 'translation_model.pkl'!")

```

Figure 2: showing dataset loading, vectorization, and model training using Logistic Regression.”

### w3. Text-to-Speech Integration

To make the translated Marwari sentences accessible to users who may not be able to read Marwari text, we integrated a text-to-speech (TTS) system. Unfortunately, no dedicated Marwari TTS system was publicly available, so we used Google Text-to-Speech (gTTS) with the Hindi language code (lang='hi') since the script and pronunciation are very similar between Hindi and Marwari.

We created a Python function using gTTS that accepted a translated Marwari sentence and generated an audio file in MP3 format. For example, if the translation output was “हाथ धोना जरूरी है”, the function would save an audio file named something like audio\_1234567890.mp3 inside an Audio/ folder. Every audio file was saved with a unique filename using a timestamp to avoid overwriting earlier files. This audio output allowed users to listen to the translation even if they were not literate, making the application more inclusive and user-friendly.



Name	#	Title	Contributing artis
marwari_audio			
ई_दवई_कनन_वर_लन_ह			
गद_पन_पण_स_पट_खरब_...			
थ_कस_ह			
थर_उमर_कनन_ह_20250...			
थर_उमर_कनन_ह_20250...			
थर_नव_कइ_ह_20250505...			
धनयवद_202505051525...			
बखर_आव_त_आरम_कर_...			

Figure 3: Marwari Audios

### 4. Translation History Storage

To keep track of every translation performed by the application, we created a simple database using a CSV file named translation\_history.csv. Instead of using a complex relational database system, we decided a CSV was sufficient for the scope of this project.

This file stored three pieces of information for each translation:

- The original Hindi input sentence
- The translated Marwari output
- The file path to the generated audio file and Timestamp

For example, if a user entered “हाथ धोना ज़रूरी है”, the system saved that Hindi input, its Marwari translation “हाथ धोना जरूरी है”, and the path like Audio/audio\_1234567890.mp3 into a new row in the CSV. Every new translation appended a new row automatically, ensuring no previous data was lost.

This translation history was also displayed inside the web app as a table for users to view their past translations.

Timestamp	Hindi Input	Marwari Output	Audio Path
05-05-2025 11:45	हाथ धोना जरूरी है	हाथ धोना जरूरी है	C:\hindi_to_marwari_tts\Audio\हॉथ धोना जरूरी है.mp3
05-05-2025 11:46	हाथ धोना जरूरी है	हाथ धोना जरूरी है	C:\hindi_to_marwari_tts\Audio\हॉथ धोना जरूरी है.mp3
05-05-2025 11:47	हाथ धोना जरूरी है	हाथ धोना जरूरी है	C:\hindi_to_marwari_tts\Audio\हॉथ धोना जरूरी है.mp3
05-05-2025 11:47	हाथ धोना जरूरी है	हाथ धोना जरूरी है	C:\hindi_to_marwari_tts\Audio\हॉथ धोना जरूरी है.mp3
05-05-2025 11:48	हाथ धोना जरूरी है	हाथ धोना जरूरी है	C:\hindi_to_marwari_tts\Audio\हॉथ धोना जरूरी है.mp3
05-05-2025 11:48	हाथ धोना जरूरी है	हाथ धोना जरूरी है	C:\hindi_to_marwari_tts\Audio\हॉथ धोना जरूरी है.mp3
05-05-2025 11:50	हाथ धोना जरूरी है	हाथ धोना जरूरी है	C:\hindi_to_marwari_tts\Audio\हॉथ धोना जरूरी है.mp3
05-05-2025 11:59	हाथ धोना जरूरी है	हाथ धोना जरूरी है	C:\hindi_to_marwari_tts\Audio\हॉथ धोना जरूरी है.mp3
05-05-2025 14:44	हाथ धोना जरूरी है	हाथ धोना जरूरी है	C:\hindi_to_marwari_tts\Audio\हॉथ धोना जरूरी है.mp3
05-05-2025 14:46	हाथ धोना जरूरी है	हाथ धोना जरूरी है	C:\hindi_to_marwari_tts\Audio\हॉथ धोना जरूरी है.mp3
05-05-2025 14:57	हाथ धोना जरूरी है	हाथ धोना जरूरी है	C:\hindi_to_marwari_tts\Audio\हॉथ धोना जरूरी है.mp3
05-05-2025 15:25	हाथ धोना जरूरी है	हाथ धोना जरूरी है	C:\hindi_to_marwari_tts\Audio\हॉथ धोना जरूरी है.mp3
05-05-2025 15:25	हाथ धोना जरूरी है	हाथ धोना जरूरी है	C:\hindi_to_marwari_tts\Audio\हॉथ धोना जरूरी है.mp3
05-05-2025 19:42	हाथ धोना जरूरी है	हाथ धोना जरूरी है	C:\hindi_to_marwari_tts\dataset\Audio\हॉथ धोना जरूरी है.mp3

Figure 4: Translation history

## 5. Web Application Development

To make the application user-friendly and accessible, we built a web-based interface using Streamlit. Streamlit allowed us to create a fully functional web app with very little front-end coding. The web interface included an input box for Hindi sentences, a “Translate” button, an output area for the Marwari translation, and an audio playback control to listen to the translation.

For example, if the user typed “बच्चों को टीका लगावाएं” in the input box and clicked “Translate,” the app displayed the Marwari translation “बच्चा ने टीका लगाओ” along with a button to play the generated audio. Below this, the app also displayed a table showing all previous translations performed by the user.

The app design was kept clean and simple, considering that it was targeted toward rural users who might not be familiar with complex interfaces. Streamlit was chosen because it allowed rapid development without requiring separate HTML, CSS, or JavaScript files.

# SwasthyaBot: A Rural Women's Health Chatbot

Empowering women through AI-driven healthcare conversations in Marwari.

## मुख्य विशेषताएँ








-  हिंदी से मारवाड़ी आवाज अनुवाद  
यह सुविधा हिंदी पाठ को मारवाड़ी बोली में बदलती है।
-  महिलाओं के स्वास्थ्य के लिए एआई-संचालित चैटबॉट  
कृत्रिम बुद्धिमत्ता से लैस चैटबॉट जो महिलाओं के स्वास्थ्य संबंधी जानकारी और सहायता प्रदान करता है।
-  फ़ोन पर आसान पहुँच  
इस चैटबॉट को फ़ोन पर आसानी से इस्तेमाल किया जा सकता है।
-  पहुँच के लिए ऑडियो प्रतिक्रियाएँ  
यह चैटबॉट ऑडियो के माध्यम से प्रतिक्रियाएँ देता है, जिससे पहुँच और आसान हो जाती है।

Figure: 5 UI Interface

## स्वस्थबॉट की विशेषताएँ

- ग्रामीण भारतीय महिलाओं के लिए सांस्कृतिक रूप से प्रासंगिक 
- आवाज समर्थन के साथ उपयोग में आसान 
- स्वास्थ्य सुझाव प्रदान करता है, सवालों के जवाब देता है और मार्गदर्शन देता है 

## कैसे इस्तेमाल करें ?





1. अपना स्वास्थ्य प्रश्न हिंदी में लिखिए 
2. बॉट इसे मारवाड़ी में अनुवाद करता है 
3. एआई संसाधित करता है और स्वास्थ्य सलाह के साथ उत्तर देता है 
4. आप ऑडियो प्रारूप में प्रतिक्रिया सुनते हैं 

Figure: 6 WEB Interface

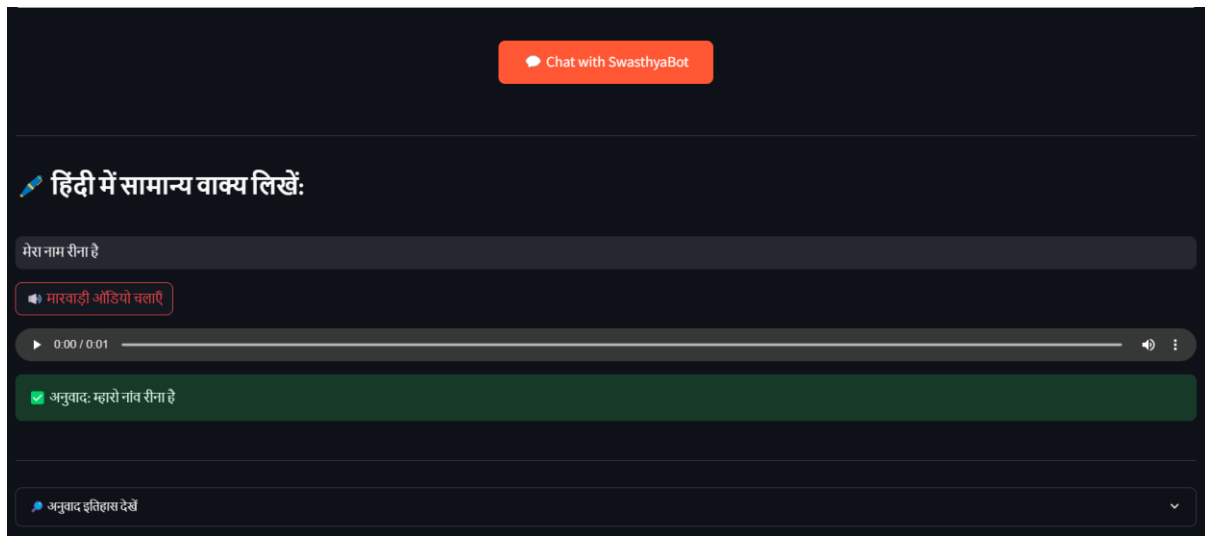


Figure: 7 Web page

## ☀ मुख्य विशेषताएँ

- 1. सामान्य और विशिष्ट स्वास्थ्य प्रश्नों का मारवाड़ी में ऑडियो उत्तर**  
 आपका चैटबॉट उपयोगकर्ताओं द्वारा पूछे गए सामान्य स्वास्थ्य संबंधी प्रश्नों और मासिक धर्म स्वच्छता व बीमारियों की रोकथाम जैसे विशिष्ट क्षेत्रों से संबंधित हिंदी पाठ को मारवाड़ी ऑडियो में उत्तर देगा। इससे उपयोगकर्ताओं को अपनी भाषा में जानकारी आसानी से समझने में मदद मिलेगी।
- 2. सुलभ और श्रव्य स्वास्थ्य मार्गदर्शन**  
 यह चैटबॉट स्वास्थ्य संबंधी जानकारी को सुनने योग्य प्रारूप में उपलब्ध कराएगा, जिससे उन लोगों के लिए भी यह सुलभ होगा जो पढ़ना पसंद नहीं करते या जिन्हें पढ़ने में कठिनाई होती है। मारवाड़ी ऑडियो प्रारूप जानकारी को अधिक व्यक्तिगत और समझने में आसान बना सकता है।
- 3. विशिष्ट स्वास्थ्य विषयों पर स्थानीय भाषा में जागरूकता**  
 आपका मॉडल मासिक धर्म स्वच्छता और बीमारियों की रोकथाम जैसे महत्वपूर्ण स्वास्थ्य विषयों पर मारवाड़ी भाषा में जानकारी प्रदान करके स्थानीय समुदाय में जागरूकता बढ़ाने में महत्वपूर्ण भूमिका निभाएगा। यह उन लोगों तक पहुंचने में मदद करेगा जो हिंदी में सहज नहीं हैं, जिससे स्वास्थ्य संबंधी महत्वपूर्ण संदेश प्रभावी ढंग से संप्रेषित किए जा सकेंगे।

Figure: 8 Web page



**Figure: 9 Dial Box**

## 6. Web Application Deployment

Finally, we deployed the web application locally by running the command `streamlit run app.py` inside the project folder using the command prompt. Once this command was executed, Streamlit launched a local server at `http://localhost:8501`, and the app became accessible in a web browser.

Users could interact with the application directly from the browser interface by typing Hindi sentences, viewing translations, listening to audio, and checking translation history, all without needing to use the command line themselves.

For example, during testing, we entered “साफ़ पानी पिएं” (Drink clean water) and confirmed that the Marwari translation and audio were generated successfully.

```

C:\Windows\system32\cmd.exe - streamlit run app.py
Microsoft Windows [Version 10.0.22631.5262]
(c) Microsoft Corporation. All rights reserved.

C:\Users\teena>cd C:\hindi_marwari_chatbot
C:\hindi_marwari_chatbot>streamlit run app.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.0.195:8501

2025-05-08 07:49:38.709 `label` got an empty value. This is discouraged for accessibility reasons and may be disallowed
in the future by raising an exception. Please provide a non-empty label and hide it with label_visibility if needed.
2025-05-08 07:54:42.408 `label` got an empty value. This is discouraged for accessibility reasons and may be disallowed
in the future by raising an exception. Please provide a non-empty label and hide it with label_visibility if needed.
2025-05-08 07:54:42.645 `label` got an empty value. This is discouraged for accessibility reasons and may be disallowed
in the future by raising an exception. Please provide a non-empty label and hide it with label_visibility if needed.

```

**Figure: 10 (Stremlit code)**



## 3.2 Figures & Screenshots

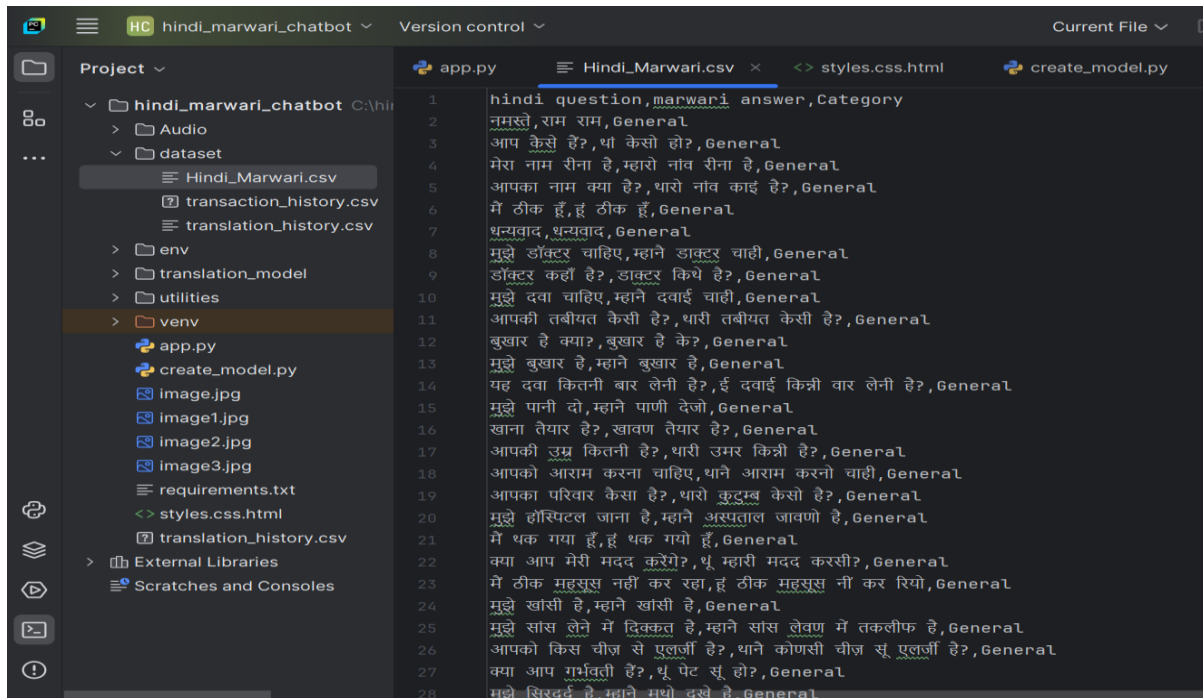


Figure 11: Dataset in PyCharm

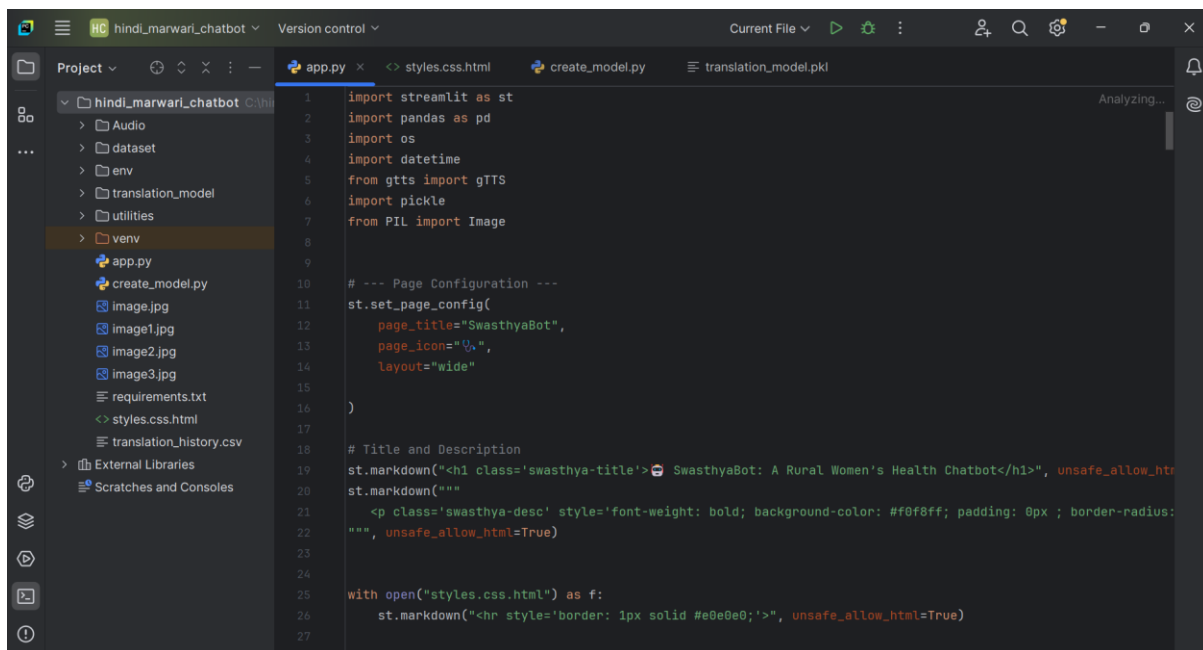


Figure 12: IDE(PyCharm) app.py code



### 3.3 Interpretations

This Python file is the main code that connects the translation model, text-to-speech system, and user interaction in a single web application. It loads the trained Hindi-to-Marwari translation model (which was saved earlier as a .pkl file) and the prepared dataset. When a user enters a Hindi sentence, the app checks this sentence in the dataset dictionary and fetches the corresponding Marwari translation if it exists. If the translation is found, it passes the Marwari text to the Google Text-to-Speech (gTTS) library, which generates an audio file. This audio file is then played back directly inside the web app so that users can listen to the translation instead of just reading it.

In addition to translation and audio generation, this code also saves a history of every interaction. Each time the user translates a sentence, the Hindi input, Marwari output, timestamp, and the path to the audio file are saved into a CSV file. This helps keep a record of all translations made by the app, which can later be viewed in a table inside the app itself. Overall, this Python code combines machine learning (translation model), text-to-speech, and simple data storage to create a complete and easy-to-use web-based translator for rural women to access health information in their native Marwari language.

### 3.4 Learning Outcomes

#### 1. Python Programming:

- Proficiency in Python was significantly improved, especially in writing clean, efficient, and maintainable code.
- Developed a deeper understanding of Python's standard libraries and third-party packages.

#### 2. Flask Framework:

- Gained expertise in using Flask to create web applications and APIs.
- Learned to set up routes, handle requests, and manage sessions in Flask.

#### 3. Natural Language Processing (NLP):

- Enhanced knowledge of NLP concepts and techniques such as tokenization, stemming, lemmatization, and entity recognition.
- Hands-on experience with NLP libraries like NLTK and spaCy for processing and analyzing text data.

#### 4. Machine Learning:

- Acquired skills in applying machine learning algorithms to classify and interpret user intents.
- Familiarity with TensorFlow for implementing and training machine learning models.

#### 5. Web Development:

- Improved skills in HTML, CSS, and JavaScript for designing and developing a responsive and user-friendly chatbot interface.
- Experience with AJAX for enabling asynchronous communication between the client-side interface and the server.

#### 6. Integrated Development Environment (IDE) - PyCharm:

- Mastery of PyCharm features such as code navigation, debugging, refactoring, and virtual environment management.
- Utilized PyCharm's tools to enhance productivity and maintain high code quality.

## **5. CONCLUSION**

The development of the College Enquiry Chatbot represents a significant step towards automating the admission and registration process at IIS (Deemed to be University). By providing instant, accurate responses to user inquiries, the chatbot reduces the workload on university staff and enhances the overall experience for prospective and current students. The use of advanced technologies and a robust system architecture ensures that the chatbot is scalable, efficient, and secure. This project serves as a model for leveraging technology to improve administrative processes in educational institutions.

## REFERENCES

1. Gawade, H., Patil, V., Vishe, P., & Kolpe, S. (2020). College enquiry chat-bot system. *International Journal of Engineering Research & Technology (IJERT)*, 9(09), 741.
2. Adamopoulou, E., & Moussiades, L. (2020). An overview of chatbot technology. In *IFIP international conference on artificial intelligence applications and innovations* (pp. 373-383). Springer, Cham.
3. Tarun Lalwani, Shashank Bhalotia, Ashish Pal, et al. "Implementation of a Chatbot System using AI and NLP". In: International Journal of Innovative Research in Computer Science & Technology (IJIRCST) Volume- 6, Issue-3 (2018)
4. <https://www.geeksforgeeks.org/top-10-projects-for-beginners-to-practice-html-and-css-skills/>
5. <https://www.ibm.com/topics/natural-language-processing>

## ANNEXURE

### app.py

```
import streamlit as st
import pandas as pd
import os
import datetime
from gtts import gTTS
import pickle
from PIL import Image

# --- Page Configuration ---
st.set_page_config(
    page_title="SwasthyaBot",
    page_icon="🏠",
    layout="wide"
)

# Title and Description
st.markdown("<h1 class='swasthya-title'>🏠 SwasthyaBot: A Rural Women's Health
```

```
Chatbot</h1>", unsafe_allow_html=True)
st.markdown("""
    <p class='swasthya-desc' style='font-weight: bold; background-color: #f0f8ff; padding: 0px ;
    border-radius: 0px; color: black; text-align: center;'>Empowering women through AI-driven
    healthcare conversations in Marwari.</p>
    """, unsafe_allow_html=True)
```

```
with open("styles.css.html") as f:
    st.markdown("<hr style='border: 1px solid #e0e0e0;'>", unsafe_allow_html=True)
```

## # Features

```
st.markdown("<div class='features'>", unsafe_allow_html=True)
st.markdown("<h2> ✨ मुख्य विशेषताएँ </h2>", unsafe_allow_html=True)
st.markdown("""
<ul>
    <li> 🗣️ हिंदी से मारवाड़ी आवाज अनुवाद
        <dl>यह सुविधा हिंदी पाठ को मारवाड़ी बोली में बदलती है। </dl>
    </li>
    <li> 👩 महिलाओं के स्वास्थ्य के लिए एआई-संचालित चैटबॉट
        <dl>कृत्रिम बुद्धिमत्ता से लैस चैटबॉट जो महिलाओं के स्वास्थ्य संबंधी जानकारी और सहायता प्रदान
        करता है।</dl>
    </li>
    <li> 📞 फ़ोन पर आसान पहुँच
        <dl>इस चैटबॉट को फ़ोन पर आसानी से इस्तेमाल किया जा सकता है।</dl>
    </li>
    <li> 🎧 पहुँच के लिए ऑडियो प्रतिक्रियाएँ
        <dl>यह चैटबॉट ऑडियो के माध्यम से प्रतिक्रियाएँ देता है, जिससे पहुँच और आसान हो जाती
        है।</dl>
    </li>
</ul>
    """, unsafe_allow_html=True)
st.markdown("</div>", unsafe_allow_html=True)
```

## # Characteristics

```
st.markdown("<h3> 🤖 📦 स्वस्थबॉट की विशेषताएँ </h3>", unsafe_allow_html=True)
```

```

st.markdown("""
- ग्रामीण भारतीय महिलाओं के लिए सांस्कृतिक रूप से प्रासंगिक □
- आवाज समर्थन के साथ उपयोग में आसान 🗣️
- स्वास्थ्य सुझाव प्रदान करता है, सवालों के जवाब देता है और मार्गदर्शन देता है 🏠
""", unsafe_allow_html=True)

# Steps
st.markdown("<h3>🌀 कैसे इस्तेमाल करें ? </h3>", unsafe_allow_html=True)
st.markdown("""
1. अपना स्वास्थ्य प्रश्न हिंदी में लिखिए □
2. बॉट इसे मारवाड़ी में अनुवाद करता है 🗣️
3. एआई संसाधित करता है और स्वास्थ्य सलाह के साथ उत्तर देता है 🧠□
4. आप ऑडियो प्रारूप में प्रतिक्रिया सुनते हैं 🗣️
""", unsafe_allow_html=True)

```

```

# --- Custom CSS ---
st.markdown("""
<style>
    .main-title {
        display: flex;
        align-items: center;
        justify-content: center;
        gap: 15px;
        font-family: 'Trebuchet MS', sans-serif;
        color: #FF5733;
    }
    .description {
        text-align: center;
        font-family: 'Calibri', sans-serif;
        font-size: 16px;
        margin-bottom: 20px;
    }
    .features-heading {
        text-align: center;
        font-family: 'Trebuchet MS', sans-serif;
        font-size: 26px;
        color: #2E8B57;
        margin-top: 20px;
    }
    .feature-item {

```

```

    text-align: center;
    font-family: 'Calibri', sans-serif;
    font-size: 16px;
    margin-bottom: 10px;
}
.chatbot-button {
    display: block;
    margin: 30px auto;
    padding: 12px 30px;
    background-color: #FF5733;
    color: white;
    border: none;
    border-radius: 8px;
    font-size: 18px;
    cursor: pointer;
    transition: background-color 0.3s ease;
    text-align: center; /* Center the button text */
}
.dialog-box {
    border: 2px solid #FF5733;
    border-radius: 12px;
    padding: 20px;
    background-color: #fff0e6;
    text-align: center;
    font-family: 'Calibri', sans-serif;
}
.image-container {
    display: flex;
    flex-direction: column;
    align-items: center;
}
.image-caption-box {
    background-color: #f0f8ff; /* Light Alice Blue - You can change this */
    border: 1px solid #e0e0e0;
    padding: 10px;
    margin-top: 5px;
    border-radius: 5px;
    text-align: center;
    font-size: 14px;
    color: #333; /* Darker text for readability */
}
.detailed-features-container {

```

```

background-color: #E0FFFF
padding: 20px;
margin-top: 30px; /* Space from the above section */
border-radius: 5px;
color: #333; /* Dark text color */
}
.detailed-feature-heading {
text-align: center;
font-size: 24px;
font-weight: bold;
color: #2E8B57; /* Green heading color */
margin-bottom: 15px;
}
.detailed-feature-item-container {
background-color: #ffe4e1; /* Light Pink background for each point */
padding: 15px;
margin-bottom: 10px;
border-radius: 5px;
border: 1px solid #e0e0e0; /* Light grey border for each point */
color: #333; /* Dark text color */
}

```

```

.point-heading {
text-align: center;
font-weight: bold;
display: block;
}

```

</style>

""", unsafe\_allow\_html=True)

# --- Main Title ---

```

st.markdown(f"""
<div class="main-title">
  <h1>❀ SwasthyaBot ❀ </h1>
</div>

```



```
""", unsafe_allow_html=True)
```

```
# --- Description ---
```

```
st.markdown("""
```

```
<div class="description">
```

```
<p>हमारा SwasthyaBot एक AI-पावर्ड हेल्थ ट्रांसलेटर चैटबॉट है, जो ग्रामीण महिलाओं के लिए  
स्वास्थ्य शिक्षा
```

```
को सरल और प्रभावी तरीके से प्रस्तुत करता है। यहाँ आप हिंदी में पूछकर मारवाड़ी में स्वास्थ्य संबंधी  
सलाह और जानकारी प्राप्त कर सकते हैं।</p>
```

```
</div>
```

```
""", unsafe_allow_html=True)
```

```
# --- Representative Images ---
```

```
image_info = [
```

```
{ "url": "C:/hindi_marwari_chatbot/image1.jpg", "caption": "📱 मोबाइल से आसानी से उपयोग"},
```

```
{ "url": "C:/hindi_marwari_chatbot/image2.jpg", "caption": "🤖 AI आधारित स्मार्ट सहायक"},
```

```
{ "url": "C:/hindi_marwari_chatbot/image3.jpg", "caption": "👩🏻💡 महिलाओं के लिए विशेष रूप  
से"}]
```

```
]
```

```
cols = st.columns(3)
```

```
for idx, col in enumerate(cols):
```

```
    with col:
```

```
        st.markdown("<div class='image-container'>", unsafe_allow_html=True)
```

```
        st.image(image_info[idx]["url"], use_container_width=True) # Changed parameter here
```

```
        st.markdown(f"<div class='image-caption-box'>{image_info[idx]['caption']}</div>",
```

```
unsafe_allow_html=True)
```

```
        st.markdown("</div>", unsafe_allow_html=True)
```

```
# --- विस्तृत विशेषताएँ ---
```

```
# Features
```

```
st.markdown("<div class='features'>", unsafe_allow_html=True)
```

```
st.markdown("<h2><b>❀ मुख्य विशेषताएँ </b></h2>", unsafe_allow_html=True)
```

```
st.markdown("""
```

```
<div class="detailed-features-container">
```

```
<div class="detailed-feature-item-container">
```

```

<b>1. सामान्य और विशिष्ट स्वास्थ्य प्रश्नों का मारवाड़ी में ऑडियो उत्तर</b><br>
आपका चैटबॉट उपयोगकर्ताओं द्वारा पूछे गए सामान्य स्वास्थ्य संबंधी प्रश्नों और मासिक धर्म
स्वच्छता व बीमारियों की रोकथाम जैसे विशिष्ट क्षेत्रों से संबंधित हिंदी पाठ को मारवाड़ी ऑडियो में उत्तर
देगा। इससे उपयोगकर्ताओं को अपनी भाषा में जानकारी आसानी से समझने में मदद मिलेगी।
</div>
<div class="detailed-feature-item-container">
<b>2. सुलभ और श्रव्य स्वास्थ्य मार्गदर्शन</b><br>
यह चैटबॉट स्वास्थ्य संबंधी जानकारी को सुनने योग्य प्रारूप में उपलब्ध कराएगा, जिससे उन लोगों
के लिए भी यह सुलभ होगा जो पढ़ना पसंद नहीं करते या जिन्हें पढ़ने में कठिनाई होती है। मारवाड़ी
ऑडियो प्रारूप जानकारी को अधिक व्यक्तिगत और समझने में आसान बना सकता है।
</div>
<div class="detailed-feature-item-container">
<b>3. विशिष्ट स्वास्थ्य विषयों पर स्थानीय भाषा में जागरूकता</b><br>
आपका मॉडल मासिक धर्म स्वच्छता और बीमारियों की रोकथाम जैसे महत्वपूर्ण स्वास्थ्य विषयों पर
मारवाड़ी भाषा में जानकारी प्रदान करके स्थानीय समुदाय में जागरूकता बढ़ाने में महत्वपूर्ण भूमिका
निभाएगा। यह उन लोगों तक पहुंचने में मदद करेगा जो हिंदी में सहज नहीं हैं, जिससे स्वास्थ्य संबंधी
महत्वपूर्ण संदेश प्रभावी ढंग से संप्रेषित किए जा सकेंगे।
</div>
</div>
""", unsafe_allow_html=True)

# --- Chatbot Button ---
st.markdown("""
<div style="text-align: center;">
<button class="chatbot-button">🗣️ Chat with SwasthyaBot</button>
</div>
""", unsafe_allow_html=True)

# --- Paths ---
BASE_DIR = os.path.dirname(os.path.abspath(__file__))
DATASET_PATH = os.path.join(BASE_DIR, "dataset", "Hindi_Marwari.csv")
MODEL_PATH = os.path.join(BASE_DIR, "translation_model", "translation_model.pkl")
HISTORY_PATH = os.path.join(BASE_DIR, "translation_history.csv")
AUDIO_FOLDER = os.path.join(BASE_DIR, "Audio")

# --- Load Model ---
if not os.path.exists(MODEL_PATH):

```

```

st.error(f"📁 Model file not found at: {MODEL_PATH}")
st.stop()
try:
    with open(MODEL_PATH, 'rb') as f:
        translation_model = pickle.load(f)
except Exception as e:
    st.error(f"📁 Model load error: {e}")
    st.stop()

# --- Load Dataset ---
if not os.path.exists(DATASET_PATH):
    st.error(f"📁 Dataset file not found at: {DATASET_PATH}")
    st.stop()
try:
    df = pd.read_csv(DATASET_PATH)
    if 'hindi question' not in df.columns or 'marwari answer' not in df.columns:
        st.error(f"📁 Dataset में 'hindi question' और 'marwari answer' columns होने चाहिए।")
        st.stop()
    hindi_marwari_dict = pd.Series(df['marwari answer'].values, index=df['hindi
question']).to_dict()
except Exception as e:
    st.error(f"📁 Dataset load error: {e}")
    st.stop()

# --- Input ---
st.markdown("<hr>", unsafe_allow_html=True)
st.subheader("✍ हिंदी में सामान्य वाक्य लिखें:")
input_text = st.text_input("")

# --- Translate + Audio ---
if st.button("🔊 मारवाड़ी ऑडियो चलाएँ"):
    if input_text.strip() == "":
        st.warning("⚠ कृपया वाक्य दर्ज करें।")
    else:
        translated = hindi_marwari_dict.get(input_text.strip())
        if translated:
            try:
                tts = gTTS(translated, lang='hi')
                os.makedirs(AUDIO_FOLDER, exist_ok=True)

```

```

timestamp = datetime.datetime.now().strftime("%Y%m%d%H%M%S")
filename = "".join(e for e in translated if e.isalnum() or e.isspace()).replace(" ", "_")
audio_file = os.path.join(AUDIO_FOLDER, f'{filename}_{timestamp}.mp3')
tts.save(audio_file)
with open(audio_file, 'rb') as f:
    st.audio(f.read(), format='audio/mp3')
st.success(f"✅ अनुवाद: {translated}")
history_row = pd.DataFrame([[datetime.datetime.now(), input_text.strip(), translated,
audio_file]],
                           columns=["Timestamp", "Hindi Input", "Marwari Output", "Audio
Path"])
if os.path.exists(HISTORY_PATH):
    old_history = pd.read_csv(HISTORY_PATH)
    pd.concat([old_history, history_row], ignore_index=True).to_csv(HISTORY_PATH,
index=False)
else:
    history_row.to_csv(HISTORY_PATH, index=False)
except Exception as e:
    st.error(f"🚫 Audio error: {e}")
else:
    st.warning("⚠️ अनुवाद नहीं मिला।")

# --- Show History ---
st.markdown("<hr>", unsafe_allow_html=True)
with st.expander("🔍 अनुवाद इतिहास देखें"):
    if os.path.exists(HISTORY_PATH):
        try:
            history_df = pd.read_csv(HISTORY_PATH)
            st.dataframe(history_df)
        except Exception as e:
            st.error(f"🚫 History load error: {e}")
    else:
        st.info("❗ कोई अनुवाद इतिहास उपलब्ध नहीं है।")
        pass # Load CSV logic here

# --- Footer ---
st.markdown("""
<footer>
    📞 123-456-7890 | ✉ support@swasthyabot.in | 🌐 www.swasthyabot.in
</footer>
""", unsafe_allow_html=True)

```

## # Styles.css code

```
body {  
  background-color: #fef9f4;  
  font-family: 'Segoe UI', sans-serif;  
}
```

```
.swasthya-title {  
  color: #c0392b;  
  text-align: center;  
  font-size: 3rem;  
  margin-top: 20px;  
}
```

```
.swasthya-desc {  
  color: #555;  
  text-align: center;  
  font-size: 1.2rem;  
  margin-bottom: 30px;  
}
```

```
.features {  
  background-color: #fff5e6;  
  padding: 20px;  
  border-radius: 12px;  
  margin-bottom: 20px;  
}
```

## # create\_model.py

```
import pickle  
  
model = {  
  "नमस्ते": "राम राम",  
  "आप कैसे हैं?": "तू के करै सै?",  
  "धन्यवाद": "बधाई हो",  
  "बुखार आने पर क्या करें?": "बखर आन् पर कय करें?"  
}
```

```

with open("translation_model.pkl", "wb") as f:
    pickle.dump(model, f)

print("Model updated ✅")

# utils.py

import pandas as pd
from gtts import gTTS
import os

# Convert predicted text to audio
tts = gTTS(text=predicted_marwari, lang='hi') # For demo, using Hindi TTS voice
audio_path = "/content/marwari_audio.mp3"
tts.save(audio_path)

# Play audio (if running locally, this works)
from IPython.display import Audio
Audio(audio_path)

import pandas as pd
from gtts import gTTS
import os

def save_to_csv(hindi, marwari, audio_path):
    file = 'dataset/transaction_history.csv'
    df = pd.DataFrame([[hindi, marwari, audio_path]], columns=["Hindi", "Marwari", "Audio"])

    if os.path.exists(file):
        df.to_csv(file, mode='a', header=False, index=False)
    else:

```

```

df.to_csv(file, index=False)

print("✅ Data saved to CSV.")
def clean_text(text):
    return text.strip().lower()
def save_history(hindi_text, marwari_text, audio_path):
    record = {
        'Timestamp': datetime.now(),
        'Hindi Text': hindi_text,
        'Marwari Translation': marwari_text,
        'Audio File': audio_path
    }
    df = pd.DataFrame([record])
    df.to_csv('dataset/transaction_history.csv', mode='a', index=False, header=not
os.path.exists('dataset/transaction_history.csv'))

```