A PROJECT REPORT ON

AI BASED REAL TIME NEWS TRANSLATOR

Submitted in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

In

Computer Science and Engineering

By

CHIDARALA DURGAPRASAD (22A81A0575)

PINDI SASI DHARANI (22A81A05B0)

YALAMARTHI LAHARI (22A81A05D2)

SANABOINA VAMSI NAGA SAI (22A81A05B5)

NALLURI NUTANA VENKATA SAI (22A81A05A5)

Under the Esteemed Supervision of

Mr.N.V.Ratnakishor Gade ,MTech,(Ph.D)

Asst. Professor

Department of Computer Science and Engineering (Accredited by N.B.A.)

SRI VASAVI ENGINEERING COLLEGE

(Autonomous)

Pedatadepalli, Tadepalligudem-534101, A.P

2024-25

SRI VASAVI ENGINEERING COLLEGE (Autonomous)

Department Of Computer Science and Engineering

Pedatadepalli, Tadepalligudem

Certificate

This is to certify that the Mini Project Report entitled **"AI BASED REAL TIME NEWS TRANSLATOR"** submitted by **CHIDARALA DURGA PRASAD (22A81A0575) , PINDI SASI DHARANI (22A81A05B0) , YALAMARTHI LAHARI (22A81A05D2) , SANABOINA VAMSI NAGA SAI (22A81A05B5) , NALLURI NUTANA VENKATA SAI (22A81A05A5)** for the award of the degree of Bachelor of Technology in the Department of Computer Science and Engineering during the academic year 2023 - 2024.

Name of Project Guide	**Head of the Department**
Mr.N.V.Ratnakishor Gade, MTech., Ph.D	Dr.D Jaya Kumari MTech., Ph.D.
Assistant Professor	Professor & HOD.

External Examiner

DECLARATION

We hereby declare that the project report entitled **AI BASED REAL TIME NEWS TRANSLATOR** submitted by us to Sri Vasavi Engineering College (Autonomous), Tadepalligudem , affiliated to JNTUK Kakinada in partial fulfillment of the requirement for the award of the degree of B. Tech in Computer Science and Engineering is a record of Bonafide project work carried out by us under the guidance of , **Mr. N.V.Ratnakishore Gade** MTech., Ph.D, Asst. Professor. We further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree in this institute or any other institute or University.

Project Associates

Chidarala DurgaPrasad (22A81A0575)

Pindi Sasi Dharani (22A81A05B0)

Yalamarthi Lahari (22A81A05D2)

Sanaboina Vamsi Naga Sai (22A81A05B5)

Nalluri Nutana Venkata Sai(22A81A05A5)

ACKNOWLEDGEMENT

First and foremost, we sincerely salute to our esteemed institute **SRI VASAVI ENGINEERING COLLEGE**, for giving us this golden opportunity to fulfill our warm dream to become an engineer. Our sincere gratitude to our project guide **Mr. N.V.Ratnakishore Gade** MTech., Ph.D, Assistant Professor, Department of Computer Science and Engineering, for his timely cooperation and valuable suggestions while carrying out this project.

We express our sincere thanks and heartful gratitude to **Dr. D. Jaya Kumari**, Professor & Head of the Department of Computer Science and Engineering, for permitting us to do our project.

We express our sincere thanks and heartful gratitude to **Dr. G.V.N.S.R. Ratnakara Rao**, Principal, for providing a favorable environment and supporting us during the development of this project.

Our special thanks to the management and all the teaching and non-teaching staff members, the Department of Computer Science and Engineering, for their support and cooperation in various ways during our project work. It is our pleasure to acknowledge the help of all those respected individuals. We would like to express our gratitude to our parents, friends who helped to complete this project.

Project Associates

Chidarala DurgaPrasad (22A81A0575)

Pindi Sasi Dharani (22A81A05B0)

Yalamarthi Lahari (22A81A05D2)

Sanaboina Vamsi Naga Sai (22A81A05B5)

Nalluri Nutana Venkata Sai(22A81A05A5)

ABSTRACT

The main aim of this project is an AI-based News Translator is designed to enhance accessibility and understanding of global News. This system collects News articles from various News API and allows users to search for relevant content using keyword - based queries.

The system analyzes collected news to identify the main themes and content of headlines and articles. It then provides audio translations in the user's preferred language, improving accessibility for non-native speakers. After processing, the system generates audio translations in the user's preferred language, making news more accessible to non - native speakers. This project bridges the gap between language barriers and digital information, providing a streamlined, multilingual audio of News experience.

TABLE OF CONTENTS

CHAPTER – 1
INTRODUCTION

1.1. INTRODUCTION

In our connected world, the ability to overcome language barriers is more important than ever. AI-powered real-time news translation offers a groundbreaking way to make news accessible worldwide by instantly translating articles, articles description, URL, posts into multiple languages. This technology goes beyond simple translation, capturing context, tone, and subtle meanings so people everywhere can understand important information. As global audiences seek timely updates on events that matter to them, AI helps news organizations reach more people by offering content in their native languages. This supports a more informed public, encourages cultural exchange, and allows everyone to stay connected to global events. AI's ability to quickly and accurately translate also means that people can access news as it happens, making real-time information accessible to all.

1.2. MOTIVATION

In our fast-paced world, real-time news translation has become essential. Language barriers still limit access to vital information, especially during important events. As social media and digital platforms drive demand for instant updates, real-time translation systems provide quick, accurate news in multiple languages, helping people stay informed and engaged globally.

1.3. SCOPE

The scope of the AI News Translation System project is defined by specific boundaries that encompass the languages supported, the types of news content being translated, and the target audience it aims to serve. The system will prioritize real-time dissemination of news, addressing the increasing demand for immediate updates across diverse platforms.

1.4. PROJECT OUTLINE

This project aims to develop an AI-driven real-time news translation system that breaks down language barriers, making global news accessible to non-native speakers instantly. The project will involve building a translation engine trained on vast multilingual datasets, implementing context and sentiment analysis, and developing a user-friendly interface to deliver instant, culturally relevant translations.

CHAPTER – 2
LITERATURE SURVEY

LITERATURE SURVEY

- **Translating News in the Era of Globalization: Challenges and Strategies"(2018)**
  Author: Roberto A. Valdeón , *Journal*: Across Languages and Cultures, *Summary*: This article examines the impact of globalization on news translation, discussing strategies for handling cultural references, idiomatic expressions, and political nuances.

- **"Gatekeeping and Translation: A Study of News Translation Practices in International News Agencies"(2016)**
  Author: Piotr Blumczynski ,*Journal*: Translation Studies, *Summary*: Focuses on the role of translators as gatekeepers, exploring the decisions they make and their influence on how news is presented to different cultural audiences.

- **"Ideological Manipulation in News Translation"(2017)**
  Author: Jonathan Ross, *Journal*: Journal of Language and Politics, *Summary*: This article discusses how translation can subtly or overtly alter the ideological slant of news, analyzing cases where political bias is introduced or mitigated through translation.

- **"Translation and Power in International News Media"(2019)**
  Author: Mona Baker, *Journal*: Critical Discourse Studies, *Summary*: Examines how power dynamics influence translation choices in global media, particularly how certain narratives are promoted or suppressed.

- **"Handling Cultural Specificity in News Translation"(2015)**
  Author: Esperança Bielsa, *Journal*: Language and Intercultural Communication, *Summary*: Focuses on the challenges of translating culturally specific references and concepts, proposing strategies for balancing accuracy with accessibility.

- **"The Ethics of News Translation: A Comparative Study of English and Arabic News Outlets"(2020)**
  Author: Ahmed Al-Salmi, *Journal*: Translation and Interpreting Studies ,*Summary*: Discusses ethical considerations in news translation, especially regarding sensitive topics in English and Arabic media, with case studies from news events.

- **"Machine Translation in Newsrooms: Boon or Bane?"(2021)**
  Author: Mary C. Liu, *Journal*: Journalism Studies ,*Summary*: Investigates the growing use of machine translation in newsrooms, addressing its benefits and limitations in preserving the original meaning and tone of news stories.

- **"Transediting: Translating and Editing News for Multilingual Audiences"(2016)**
  Author: Federico Zanettin, *Journal*: Perspectives: Studies in Translation Theory and Practice,*Summary*: Introduces the concept of "transediting" – combining translation and editing – and its application in making news content suitable for various audiences.

CHAPTER – 3
SYSTEM STUDY AND ANALYSIS

3.1. Problem Statement

The main problem statements for an AI-based real-time news translator include the need for seamless translation of news articles across multiple languages without manual intervention, which is essential for users who consume news in diverse languages. Additionally, there is often a lack of sentiment analysis, which could help users quickly grasp the emotional tone or bias of the news. Another challenge is categorizing news based on topics in real time, making it difficult for users to find relevant information efficiently. Finally, the absence of text-to-speech integration limits accessibility for users who prefer or require audio summaries, especially on mobile devices or while multitasking.

3.2. Existing System

The existing system for an AI-based real-time news translator typically involves using online news platforms and translation services separately, where users manually search for news in different languages and then use tools like Google Translate for translation. This system is often inefficient, as it doesn't support automatic language detection, real-time sentiment analysis, or categorization. Users also lack the ability to listen to news summaries, as conventional systems lack integrated text-to-speech functionality. Consequently, the experience is fragmented, requiring multiple steps for translation, sentiment analysis, and playback, with limited automation or personalization.

3.3. Limitations of Existing System

The existing system for news translation is limited by its reliance on separate tools for language translation, sentiment analysis, and news categorization, making the process cumbersome and time-consuming. It often requires users to manually translate each article and lacks the capability for real-time, automatic language detection and translation. Additionally, there is no integrated sentiment analysis, which limits users' ability to quickly assess the tone or bias of articles. The absence of automatic categorization means users must sift through unorganized content, and without a text-to-speech feature, accessibility is hindered for users who prefer audio formats or need hands-free access. These limitations reduce efficiency and accessibility, particularly for multilingual users seeking a seamless, real-time news experience.

3.4. Proposed System

It aims to integrate real-time language detection, translation, sentiment analysis, topic categorization, and text-to-speech capabilities into a single platform. This system will automatically translate news articles from multiple languages, categorize them by sentiment and topic, and provide audio summaries, offering a streamlined experience for users. By combining these features, the proposed system will make it easier for users to access, understand, and engage with global news, regardless of language barriers or format preferences.

3.5. Advantages of Proposed System

Seamless Language Translation: It automatically translates news articles across various languages, making global news more accessible to users without language constraints.

Real-Time Processing: Users can receive translations, sentiment analysis, and topic categorization instantly, which keeps them updated with news in real time.

Sentiment Analysis: By analyzing the sentiment of articles, the system allows users to quickly understand the emotional tone or bias, aiding in better decision-making and awareness of media perspectives.

Text-to-Speech Functionality: The integrated text-to-speech feature enhances accessibility by allowing users to listen to article summaries, making it convenient for those on the go or with visual impairments.

3.6. Functional Requirements

1. **News Fetching**: The application should fetch news articles from the News API based on user-defined keywords or trending topics.
2. **Text Translation**: Users should be able to translate article titles and descriptions into their preferred languages.
3. **Sentiment Analysis**: The application should analyse the sentiment of news articles and classify them as Positive, Neutral, or Negative.
4. **Search Functionality**: Users should be able to search for articles using keywords and receive results that match their query.
5. **Display News Articles**: The application should present fetched news articles in an organised format, including titles, descriptions, and links to full articles.
6. **Text-to-Speech**: The application should provide a "Speak" button that allows users to listen to the article content.

3.7. Non– Functional Requirements

Non-functional requirements specify the criteria that judge the operation of a system, rather than specific behaviours. These may include:

1. **Performance:** The application should be able to handle up to 100 concurrent users without significant performance degradation.
2. **Scalability:** The application should be scalable to accommodate a growing number of users and articles.
3. **Security:** User data should be securely stored, and communications with the server should be encrypted (e.g., HTTPS).
4. **Usability:** The user interface should be intuitive and easy to navigate, ensuring a good user experience.

5. **Accessibility:** The application should be accessible to users with disabilities, complying with WCAG (Web Content Accessibility Guidelines).
6. **Compatibility:** The application should work across multiple browsers and devices (desktop, tablets, smartphones).
7. **Maintainability:** The code should be well-documented and modular, allowing for easy updates and maintenance.

3.8. System Requirements

System requirements detail the necessary software and hardware for the application to run effectively.

3.8.1. Software Requirements

- Web Framework: Flask (for building the web application).
- Translation Library: Google Translate API (for translating text).
- Natural Language Processing: TextBlob (for sentiment analysis).
- HTTP Requests: Requests library (for making API calls to fetch news).
- Caching: Flask-Caching (for caching frequently accessed data).
- Database: (If applicable) SQLAlchemy or a NoSQL database for storing user data and application state.
- Development Tools: Python 3.x, a code editor (like Visual Studio Code), and version control (like Git).

3.8.2. Hardware Requirements

- Server Specifications:
  - CPU: Minimum of 2 cores.
  - RAM: Minimum of 4 GB (8 GB recommended for better performance).
  - Storage: At least 20 GB of free disk space for application files and data.
- Development Environment: Development machine with Python installed, as well as all necessary libraries and dependencies.
- Network: Stable internet connection for API access and user interactions.

CHAPTER–4
SYSTEM DESIGN

4.1. System Architecture

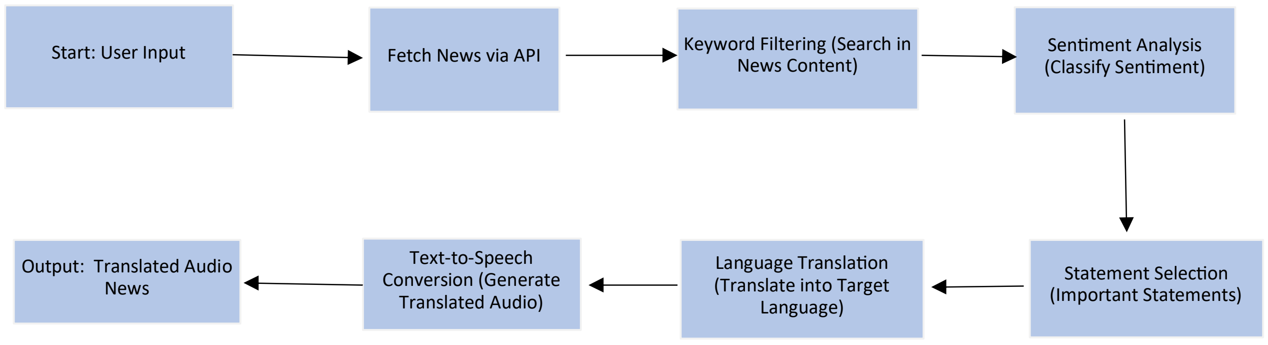

Start: User Input
Fetch News via API
Keyword Filtering (Search in News Content)
Sentiment Analysis (Classify Sentiment)
Statement Selection (Important Statements)
Language Translation (Translate into Target Language)
Text-to-Speech Conversion (Generate Translated Audio)
Output: Translated Audio News

Figure 4.1

The diagram illustrates a system architecture for processing news content. The system starts with user input, likely a search query or topic of interest. It then fetches relevant news articles using an API. Keyword filtering is applied to the retrieved news content to extract specific information or themes. Sentiment analysis is performed to classify the news content based on its emotional tone (e.g., positive, negative, neutral). The processed news content is then translated into the target language, and text-to-speech conversion generates translated audio output for the user.

4.2. System Flow Design

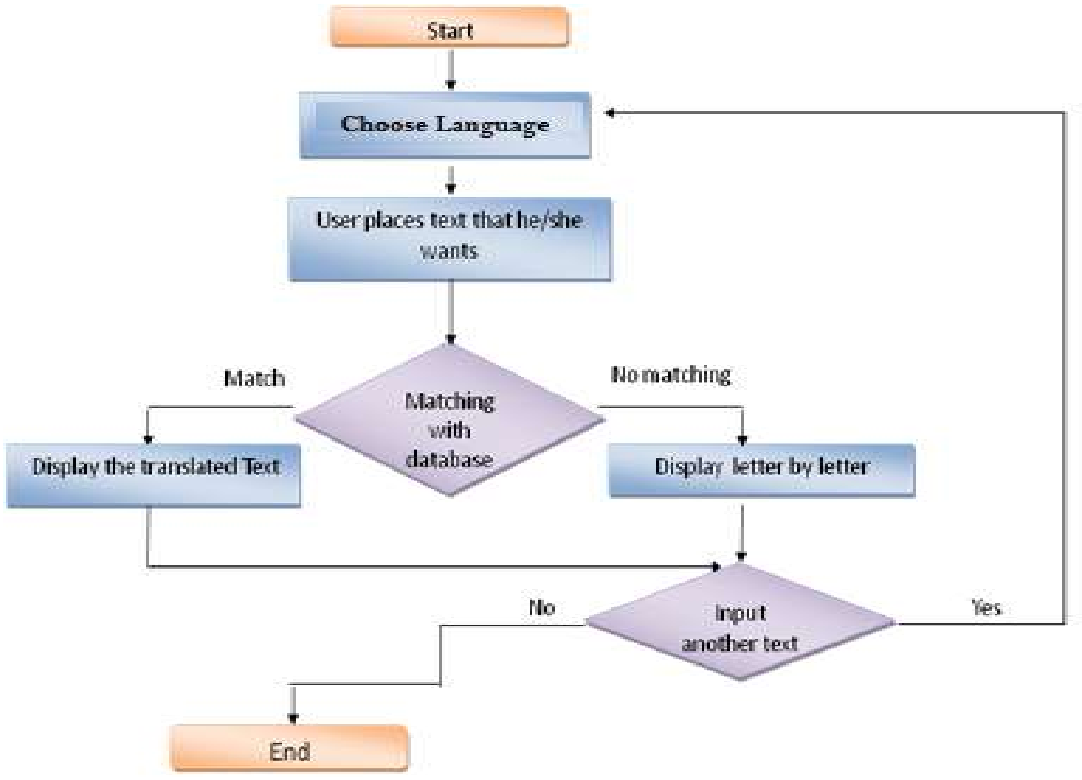


Figure 4.2. System Flow Design

The diagram illustrates a system flow design for a language translation application. The system starts by prompting the user to choose the desired language for translation. The user then inputs the text they want to translate. The system checks its database for an exact match of the input text. If a match is found, the translated text is immediately displayed. If no match is found, the system displays the input text letter by letter, possibly as a visual representation of the translation process. After each translation, the user is asked if they want to translate another text. The system continues this process until the user decides to stop, at which point the system ends.

4.3. UML Diagrams

A UML (Unified Modeling Language) diagram is a tool that provides a standardized, visual way to represent the structure and behavior of a system. These diagrams make it easier for developers, stakeholders, and business teams to understand and communicate the design, logic, and processes of complex systems. Here's how UML diagrams contribute to problem-solving, productivity, and understanding in the context of software development and business processes.

Use Case Diagram

UML use case diagram, illustrating the interactions between a user and various APIs. The user can search for news articles, view trending news, translate articles, analyze sentiment, and listen to audio translations. These actions are facilitated by APIs for news retrieval, translation, sentiment analysis, and text-to-speech. This diagram provides a visual representation of how a user can leverage these APIs to access and understand information from diverse sources.

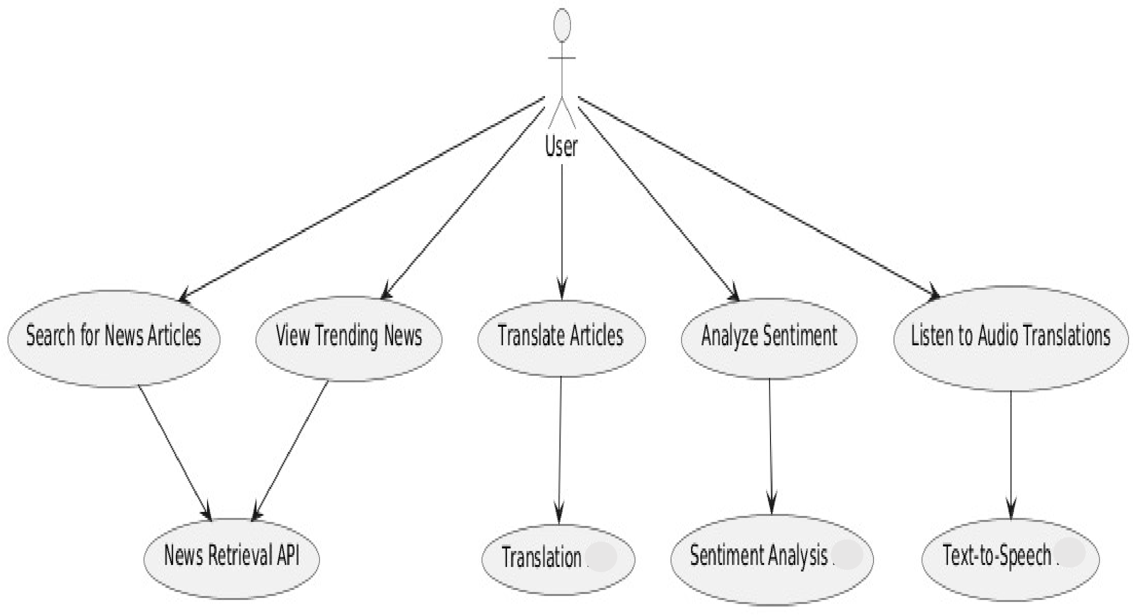


Figure 4.3. Use Case Diagram

Activity Diagram

This activity diagram illustrates the workflow of an AI-powered news translation system. It starts with the user requesting articles, providing either a specific keyword or language preference. Based on the user's input, the system either fetches articles from a news API using the keyword or retrieves trending articles if no keyword is specified. The fetched articles are then filtered to refine the selection. Next, a sentiment analysis is performed on the articles to determine their overall tone. Simultaneously, the articles are translated into the user's preferred language. If the user requests audio output, the summaries of the articles are converted into audio format. Finally, the system displays the translated articles along with their sentiment analysis, and if audio conversion was requested, it plays the audio summaries.

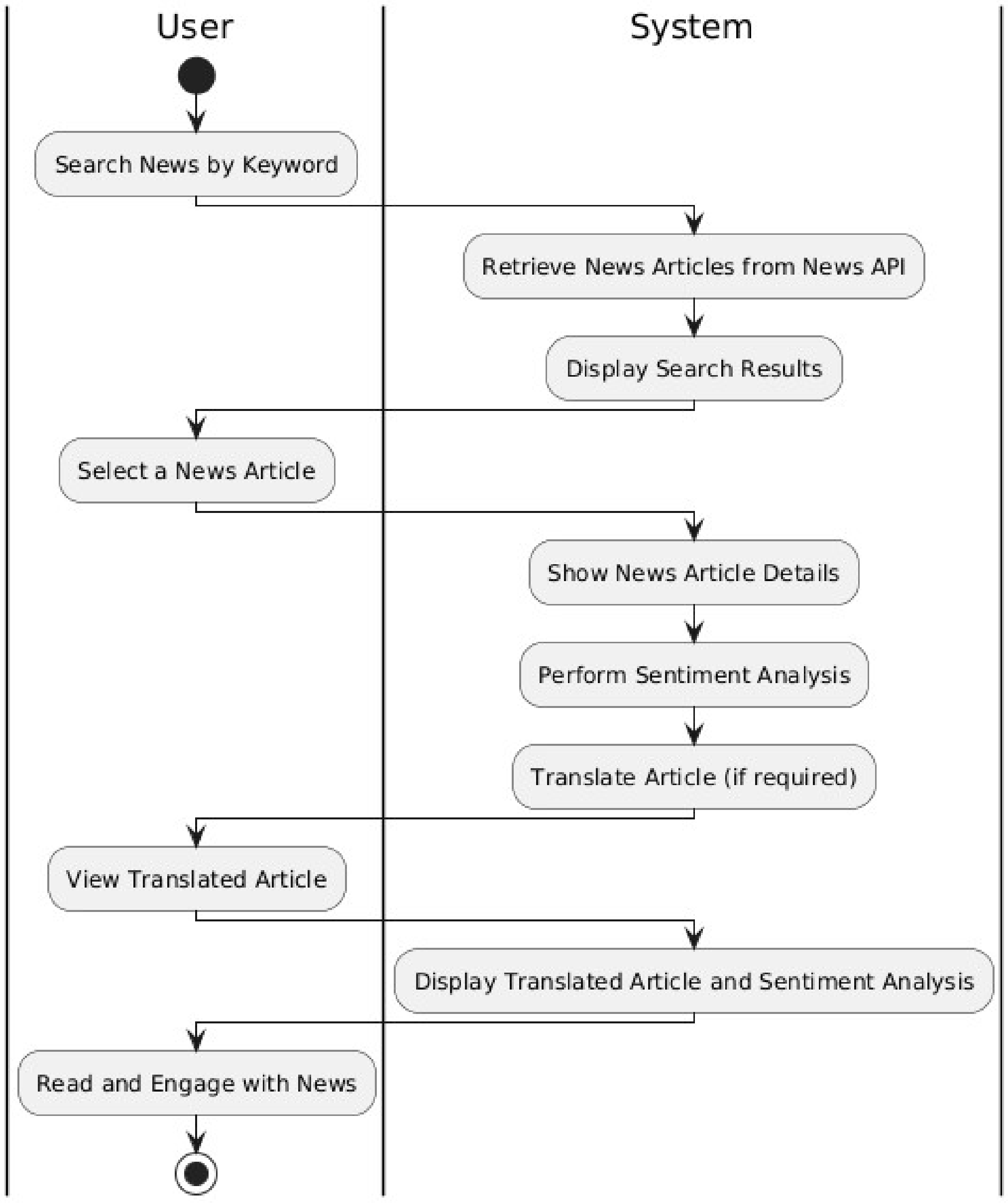

User
System
Search News by Keyword
Retrieve News Articles from News API
Display Search Results
Select a News Article
Show News Article Details
Perform Sentiment Analysis
Translate Article (if required)
View Translated Article
Display Translated Article and Sentiment Analysis
Read and Engage with News

Figure 4.3. Activity Diagram

Class Diagram

The class diagram outlines the components of a news application. It shows how a User can search for news, which is fetched by the News Service. The fetched articles are then analyzed by the Sentiment Analysis Service to determine their sentiment. If needed, the Translation Service can translate the articles into the User's preferred language. The Sentiment object represents the result of the sentiment analysis, indicating the type and intensity of the sentiment.

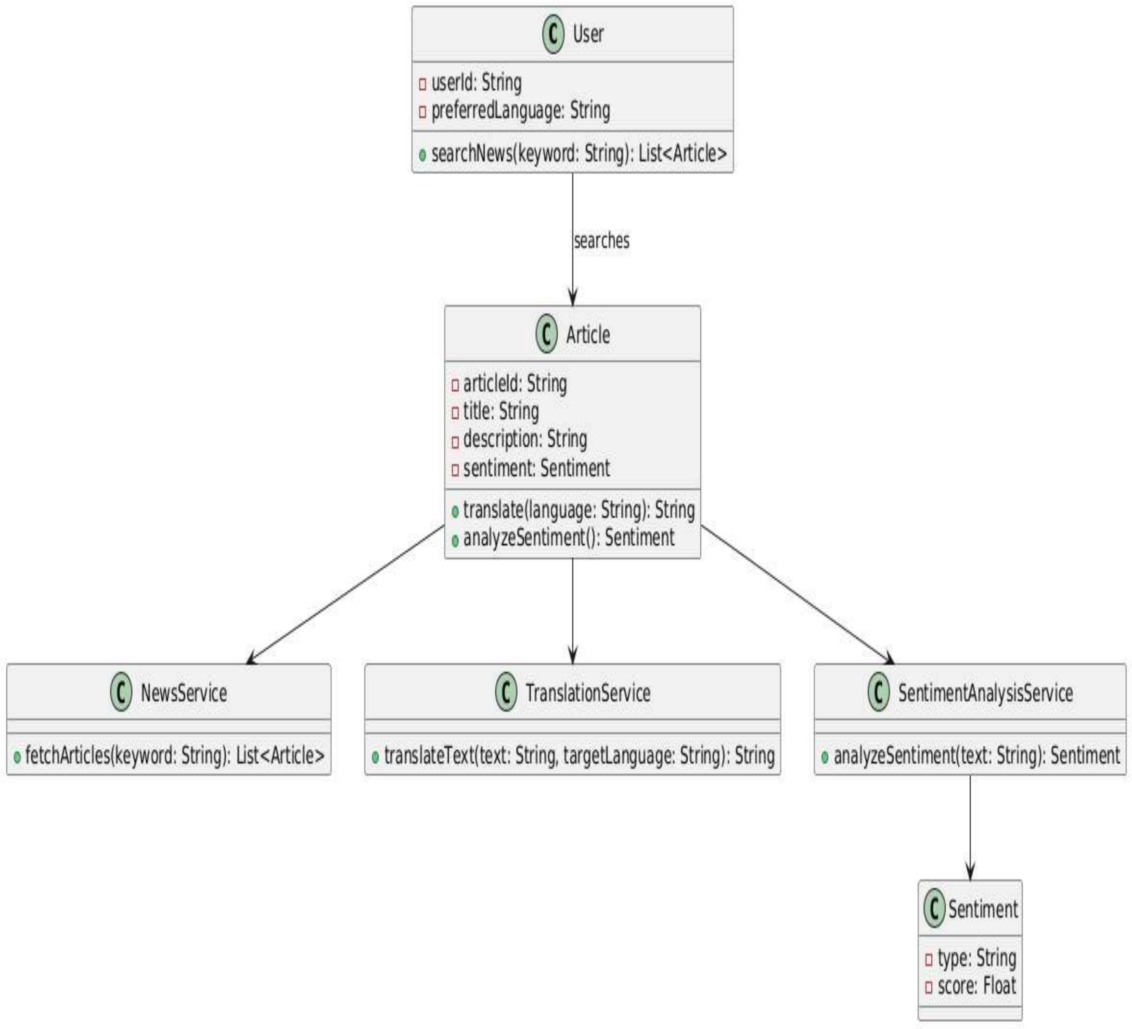

User
userId: String
preferredLanguage: String
searchNews(keyword: String): List<Article>
searches
Article
articleId: String
title: String
description: String
sentiment: Sentiment
translate(language: String): String
analyzeSentiment(): Sentiment
NewsService
fetchArticles(keyword: String): List<Article>
TranslationService
translateText(text: String, targetLanguage: String): String
SentimentAnalysisService
analyzeSentiment(text: String): Sentiment
Sentiment
type: String
score: Float

Figure 4.3. Class Diagram

Sequence Diagram

The sequence diagram illustrates how a user's news search is processed. The user starts by searching for news on the website. The website then sends this request to its controller. The controller fetches relevant articles from the News Service and analyzes their sentiment using the Sentiment Service. If necessary, the articles are translated by the Translation Service. Finally, the analyzed and potentially translated articles are displayed to the user on the website.

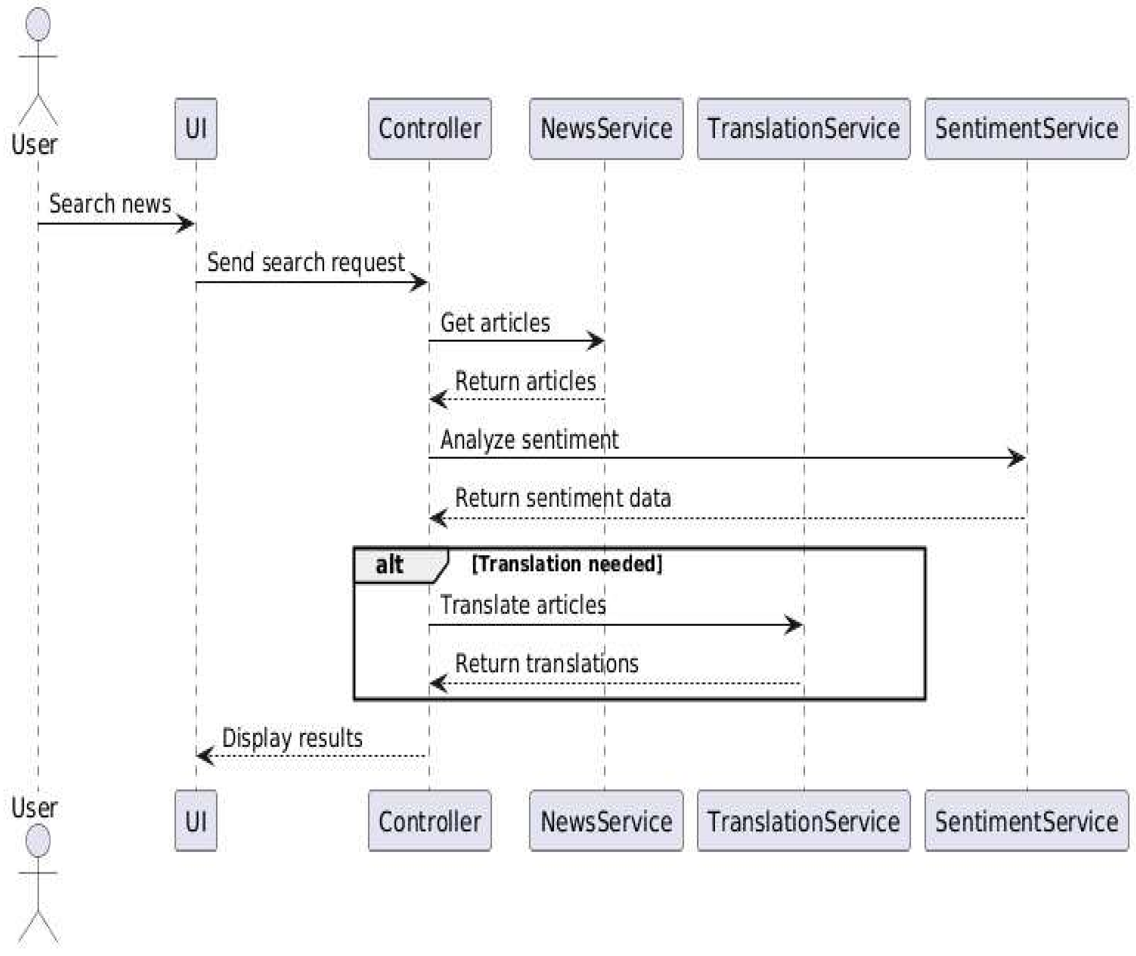


Figure 4.3. Sequence Diagram

CHAPTER–5
LANGUAGE & TOOLS

5.1. HYPER TEXT MARKUP LANGUAGE (HTML)

About HTML: HTML was invented by Tim Berners-Lee, a physicist at CERN, in 1991 as part of a project to facilitate information sharing among researchers. The initial version of HTML was quite basic. Hyper Text Markup Language, is the foundational markup language used to create and structure content on the World Wide Web. Its history is closely tied to the development of the web itself.

Advantages:

● Simplicity and Ease of Use.

● Compatibility Across Browsers

Applications:

● GUIbaseddesktop applications (Games, Scientific Applications)

● Web frame works and applications

● Enterprise and Business applications

● Operating Systems

● Language Development

5.2. CASCADING STYL SHEETS(CSS)

About CSS: CSS stands for Cascading Style Sheets. It is a stylesheet language that describes how HTML elements should be displayed on screen, paper, or in other media. CSS allows developers to apply styles such as colors, fonts, spacing, and positioning to HTML documents, enabling a separation of content from design.

Advantages:

● Better User Experience

● Platform Independence

Applications:

● Responsive Web Design 15

● Styling User Interfaces for Applications

5.3. JavaScript for Frontend

1. JavaScript:

- JavaScript, specifically the Web Speech API, enables text-to-speech (TTS) functionality on the "Speak" button.

- JavaScript is also used to handle click events on the "Speak" button, trigger TTS, and control the flow of text reading.

2. Java Script Web Speech API

a. SpeechSynthesisUtterance:

- This is part of the Web Speech API, used to create spoken utterances in the browser.
- The SpeechSynthesisUtterance object enables text-to-speech for the article title and description in the selected language, enhancing accessibility.

b. speechSynthesis.speak:

- **T**his method is invoked to read out the text in the browser.
- The method is triggered by JavaScript and reads the text set in the SpeechSynthesisUtterance object, making the application more interactive and providing an auditory experience.

5.4. Required Python Libraries

Technologies:

1. Python Libraries and Flask Framework

a. Flask:

- Flask is a lightweight web framework in Python used here to build the server-side of the application.
- It provides functionalities for routing (e.g., /index, /search) and rendering templates (like start.html and results.html) to handle HTTP requests and responses.

b. Flask-Caching:

- Used to store frequently requested data (like trending news) temporarily to reduce load times and avoid redundant API requests.
- Configured with a simple cache, which stores data in memory, helping improve application performance.

c. requests:

- This library is used to send HTTP requests to the NewsAPI to fetch articles for news translation. It handles API calls, retrieves data, and handles errors, including retries in case of failures.

d. google translate:

- The googletrans library is used to translate text content into different languages via Google Translate.
- It provides an interface to the Google Translate API, enabling language translations for article titles and descriptions.

e. TextBlob:

- TextBlob is used to perform sentiment analysis on news articles. It helps classify text as positive, neutral, or negative based on polarity, which aids in categorising news by sentiment.

5.5. News API

- The application integrates with NewsAPI to retrieve current news articles based on search queries or trending topics.
- It's used to get up-to-date content that the app then translates and categorizes based on sentiment. The NewsAPI endpoint provides article data that includes titles, descriptions, URLs, and image URLs**.**

Technology	**Purpose**
Flask	Web framework for handling routing, HTTP requests, and rendering HTML templates.
Flask-Caching	Caches frequently requested data to improve load times.
requests	Sends HTTP requests to NewsAPI to fetch articles.
google translate	Translates news articles into the selected language.
TextBlob	Performs sentiment analysis on news text to classify as positive, neutral, or negative.
JavaScript	Handles UI interactions, including triggering text-to-speech via the Speak button.
Web Speech API	Provides text-to-speech functionality for reading articles aloud.
NewsAPI	Provides real-time news content for search and trending news features.
HTML & CSS	Structures and styles the interface, providing a user-friendly layout and design.

CHAPTER–6
IMPLEMENTATION

6.1. Implementation Phases

Coding Phase

- **Environment Setup**:
    - Install Python and create a virtual environment to isolate your project.
    - Use pip to install necessary libraries: Flask for the web framework, requests for API calls, google translate for translation, textblob for sentiment analysis, and Flask-Caching for caching responses.

- **Application Structure**:
    - Create the main Flask application with defined routes:
        - **Home Page (/)**: Displays a welcome page.
        - **Index Page (/index)**: Shows trending news articles based on user-selected language.
        - **Search Endpoint (/search)**: Accepts user queries, retrieves articles, analyses their sentiment, and displays the results.

- **Functionality Development**:
    - Integrate the **News API** to fetch news articles based on keywords and trending topics.
    - Implement **sentiment analysis** using **TextBlob** to classify articles as positive, neutral, or negative.
    - Use **Google Translate** to convert articles into different languages as specified by the user.

- **Error Handling and Caching**:
    - Add error handling to manage API request failures, including retry logic for robustness.
    - Configure caching to store results and minimise redundant API calls, enhancing performance.

app.py

```
from flask import Flask, render_template, request, jsonify, redirect, url_for
import requests
import time
from googletrans import Translator
from textblob import TextBlob

app = Flask(_name_)

# Replace with your actual NewsAPI key
api_key = '24be048e36114998a46000c06c02d10a'
news_api_url = 'https://newsapi.org/v2/everything'
trending_news_url = 'https://newsapi.org/v2/top-headlines'

# Initialize the translator
translator = Translator()

def analyze_sentiment(text):
    """Analyze sentiment using TextBlob on English text."""
    analysis = TextBlob(text)
    if analysis.sentiment.polarity > 0.1:
        return 'Positive'
    elif analysis.sentiment.polarity < -0.1:
        return 'Negative'
    else:
        return 'Neutral'

def translate_text(text, dest_language):
    """Translate text to the desired language."""
    if not text:
        return "No content available"
    try:
        translated = translator.translate(text, dest=dest_language)
        return translated.text
    except Exception as e:
        print(f"Translation error: {e}")
        return text  # Fallback to original if translation fails

def fetch_with_retries(url, params, max_retries=3, delay=5):
    """Fetch data with retries to handle API rate limits."""
    for attempt in range(max_retries):
        try:
            response = requests.get(url, params=params)
            response.raise_for_status()
            articles = response.json().get('articles', [])
            # Filter out unwanted articles
            available_articles = [
            article for article in articles
            if article.get('title') and article.get('description') and article.get('url') and
            article['title'] != '[Removed]' and article['description'] != '[Removed]'
        ]
        return available_articles
    except requests.exceptions.RequestException as e:
        print(f"Request error on attempt {attempt + 1}: {e}")
        if attempt < max_retries - 1:
            print(f"Retrying in {delay} seconds...")
            time.sleep(delay)
    return []

@app.route('/')
def home():
    return render_template('start.html')

@app.route('/index')
def index():
    """Render the home page with trending articles in the default language."""
    language = request.args.get('language', 'en')
    trending_articles = get_translated_trending_articles(language)

    # Debugging: Print articles being passed to template
    print("Trending Articles for index route:", trending_articles)

    return render_template('index.html', trending_articles=trending_articles,
selected_language=language)

@app.route('/search', methods=['POST'])
def search():
    """Search for articles based on keyword, then translate and classify by sentiment."""
    user_language = request.form.get('language', 'en')
    keyword = request.form.get('keyword', '')
    page_size = int(request.form.get('pageSize', 15))
    target_per_sentiment = max(1, page_size // 3)

    params = {
        'apiKey': api_key,
        'q': keyword,
        'pageSize': page_size * 2,
        'language': 'en'
    }

    articles = fetch_with_retries(news_api_url, params)

    classified_articles = {'Positive': [], 'Neutral': [], 'Negative': []}

    for article in articles:
        if all(len(classified_articles[sentiment]) >= target_per_sentiment for sentiment in
classified_articles):
```

```
            break

        title = article['title']
        description = article['description']
        url = article['url']
        image_url = article.get('urlToImage', '')

        full_text = f"{title} {description}"
        sentiment = analyze_sentiment(full_text)

        translated_title = translate_text(title, user_language) if user_language != 'en' else title
        translated_description = translate_text(description, user_language) if user_language !=
'en' else description

        if len(classified_articles[sentiment]) < target_per_sentiment:
            classified_articles[sentiment].append({
                'title': translated_title,
                'description': translated_description,
                'url': url,
                'imageUrl': image_url
            })

    return render_template('results.html', articles=classified_articles,
language=user_language)

def get_translated_trending_articles(language):
    """Fetch trending articles and translate if necessary."""
    params = {
        'apiKey': api_key,
        'country': 'us',
        'pageSize': 10
    }
    trending_articles = fetch_with_retries(trending_news_url, params)

    # Debugging: Print fetched articles
    print("Trending articles fetched:", trending_articles)

    if language != 'en':
        for article in trending_articles:
            article['title'] = translate_text(article['title'], language)
            article['description'] = translate_text(article['description'], language)

    # Debugging: Print translated articles
    print("Trending articles after translation (if applied):", trending_articles)

    return trending_articles



@app.route('/trending')
```

```
def trending():
    """Fetch and return trending news articles in the selected language."""
    language = request.args.get('language', 'en')
    trending_articles = get_translated_trending_articles(language)
    return jsonify(trending_articles)

if _name_ == '_main_':
    app.run(debug=True)
```

start.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Welcome to News Translator</title>
    <style>
      * {
        margin: 0;
        padding: 0;
        box-sizing: border-box;
      }

      body {
        font-family: "Arial", sans-serif;
        background-color: #121212;
        display: flex;
        justify-content: center;
        align-items: center;
        min-height: 100vh;
        color: #ffffff;
      }

      .container {
        text-align: center;
        background-color: #1e1e1e;
        padding: 40px;
        border-radius: 15px;
        box-shadow: 0 0 30px rgba(0, 0, 0, 0.5);
        max-width: 800px;
        width: 90%;
      }

      h1 {
        font-size: 3em;
        margin-bottom: 20px;
  background: linear-gradient(45deg, #00ffff, #ff00ff);
  -webkit-background-clip: text;
  background-clip: text;
  color: transparent;
  text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.3);
}

.subtitle {
  color: #a0a0a0;
  font-size: 1.2em;
  margin-bottom: 40px;
}

.start-button {
  background: linear-gradient(45deg, #00ffff, #ff00ff);
  color: #ffffff;
  padding: 15px 40px;
  text-decoration: none;
  border-radius: 30px;
  font-size: 1.1em;
  font-weight: bold;
  transition: all 0.3s ease;
  display: inline-block;
  text-transform: uppercase;
  letter-spacing: 1px;
}

.start-button:hover {
  transform: translateY(-3px);
  box-shadow: 0 0 20px rgba(0, 255, 255, 0.5),
    0 0 20px rgba(255, 0, 255, 0.5);
}

.features {
  margin-top: 40px;
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
  gap: 20px;
  padding: 20px;
}

.feature-card {
  background-color: #2d2d2d;
  padding: 20px;
  border-radius: 10px;
  transition: transform 0.3s ease;
}

.feature-card:hover {
  transform: translateY(-5px);
```

```
    }

    .feature-card h3 {
      color: #00ffff;
      margin-bottom: 10px;
    }

    .feature-card p {
      color: #a0a0a0;
      font-size: 0.9em;
    }

    @media (max-width: 768px) {
      h1 {
        font-size: 2em;
      }

      .container {
        padding: 20px;
      }

      .features {
        grid-template-columns: 1fr;
      }
    }
  </style>
</head>
<body>
  <div class="container">
    <h1>Welcome to News Translator</h1>
    <p class="subtitle">Breaking Language Barriers in Global News</p>
    <a href="/index" class="start-button">Get Started</a>

    <div class="features">
      <div class="feature-card">
        <h3>Real-time Translation</h3>
        <p>Instant translation of news articles in multiple languages</p>
      </div>
      <div class="feature-card">
        <h3>Global Coverage</h3>
        <p>Access news from around the world in your preferred language</p>
      </div>
      <div class="feature-card">
        <h3>Easy to Use</h3>
        <p>Simple and intuitive interface for seamless translation</p>
      </div>
    </div>
  </div>
</body>
</html>
```

index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>News Translator</title>
    <link

href="https://fonts.googleapis.com/css2?family=Roboto:wght@300;400;700&display=swa
p"
      rel="stylesheet"
    />
    <style>
      * {
        margin: 0;
        padding: 0;
        box-sizing: border-box;
      }

      body {
        font-family: "Roboto", sans-serif;
        background-color: #121212;
        color: #ffffff;
        display: flex;
        flex-direction: column;
        align-items: center;
        min-height: 100vh;
        padding: 50px 20px;
      }

      .container {
        width: 100%;
        max-width: 800px;
      }

      .heading {
        font-size: 3em;
        margin-bottom: 20px;
        background: linear-gradient(45deg, #00ffff, #ff00ff);
        -webkit-background-clip: text;
        background-clip: text;
        color: transparent;
        text-align: center;
      }

      .search-bar-container {
        background-color: #1e1e1e;
  padding: 20px;
  border-radius: 15px;
  margin-top: 20px;
  display: flex;
  flex-wrap: wrap;
  gap: 10px;
}

.search-bar-container select,
.search-bar-container input,
.search-bar-container button {
  padding: 12px;
  border: none;
  border-radius: 8px;
  font-size: 16px;
  background-color: #2d2d2d;
  color: #ffffff;
  outline: none;
}

.search-bar-container input:focus,
.search-bar-container select:focus {
  background-color: #363636;
  box-shadow: 0 0 0 2px rgba(0, 255, 255, 0.3);
}

.search-bar-container select,
.search-bar-container input[type="number"] {
  flex: 1;
}

.search-bar-container input[type="text"] {
  flex: 3;
}

.search-bar-container button {
  background: linear-gradient(45deg, #00ffff, #ff00ff);
  color: #ffffff;
  cursor: pointer;
  transition: all 0.3s ease;
}

.search-bar-container button:hover {
  transform: translateY(-3px);
  box-shadow: 0 0 20px rgba(0, 255, 255, 0.5),
    0 0 20px rgba(255, 0, 255, 0.5);
}

.trending-section {
  margin-top: 30px;
  background-color: #1e1e1e;
  padding: 20px;
  border-radius: 15px;
}

.trending-section h2 {
  color: #00ffff;
  margin-bottom: 15px;
  text-align: center;
}

.trending-articles {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
  gap: 20px;
}

.trending-article {
  background-color: #2d2d2d;
  border-radius: 10px;
  padding: 15px;
  display: flex;
  flex-direction: column;
  transition: transform 0.3s ease, background-color 0.3s ease;
}

.trending-article:hover {
  transform: translateY(-5px);
  background-color: #363636;
}

.trending-article img {
  width: 100%;
  height: 150px;
  object-fit: cover;
  border-radius: 8px;
  margin-bottom: 10px;
}

.trending-article h3 {
  font-size: 18px;
  margin-bottom: 10px;
}

.trending-article a {
  color: #00ffff;
  text-decoration: none;
}

.trending-article a:hover {
  text-decoration: underline;
}

.speak-button {
  margin-top: auto;
  padding: 10px;
  background: linear-gradient(45deg, #00ffff, #ff00ff);
  color: #ffffff;
  border: none;
  border-radius: 5px;
  cursor: pointer;
  transition: all 0.3s ease;
}

.speak-button:hover {
  transform: translateY(-2px);
  box-shadow: 0 0 10px rgba(0, 255, 255, 0.5),
    0 0 10px rgba(255, 0, 255, 0.5);
}

.back-button-container {
  display: flex;
  justify-content: center;
  width: 100%;
  margin-top: auto;
  padding-top: 20px;
}

.back-button {
  background: linear-gradient(45deg, #00ffff, #ff00ff);
  color: #ffffff;
  padding: 10px 20px;
  border: none;
  border-radius: 8px;
  cursor: pointer;
  font-size: 16px;
  transition: all 0.3s ease;
}

.back-button:hover {
  transform: translateY(-2px);
  box-shadow: 0 0 10px rgba(0, 255, 255, 0.5), 0 0 10px rgba(255, 0, 255, 0.5);
}

@media (max-width: 600px) {
  .heading {
    font-size: 2em;
  }

  .search-bar-container {
        flex-direction: column;
      }

      .search-bar-container select,
      .search-bar-container input,
      .search-bar-container button {
        width: 100%;
      }
    }
  </style>
</head>
<body>
  <div class="container">
    <h1 class="heading">News Translator</h1>

    <form action="/search" method="post" class="search-bar-container">
      <select name="language" id="language" onchange="changeLanguage()">
        <option value="en">English</option>
        <option value="es">Spanish</option>
        <option value="fr">French</option>
        <option value="de">German</option>
        <option value="it">Italian</option>
        <option value="te">Telugu</option>
        <option value="hi">Hindi</option>
        <option value="kn">Kannada</option>
        <option value="ta">Tamil</option>
      </select>
      <input
        type="text"
        name="keyword"
        id="keyword"
        placeholder="Search news..."
        required
      />
      <input
        type="number"
        name="pageSize"
        id="pageSize"
        placeholder="Articles"
        min="1"
        value="5"
      />
      <button type="submit">Search</button>
    </form>

    <div class="trending-section">
      <h2>Trending News</h2>
      <div class="trending-articles" id="trendingArticles">
        <!-- Trending articles will be dynamically inserted here -->
      </div>
```

```
</div>

<script>
  async function loadTrendingArticles(language) {
    const trendingContainer = document.getElementById("trendingArticles");
    trendingContainer.innerHTML = ""; // Clear previous content

    try {
      const response = await fetch(/trending?language=${language});
      const articles = await response.json();

      articles.slice(0, 3).forEach((article) => {
        const articleElement = document.createElement("div");
        articleElement.className = "trending-article";
        articleElement.innerHTML = `
          <img src="${article.urlToImage}" alt="${article.title}">
          <h3>
            <a href="${article.url}" target="_blank">${article.title}</a>
          </h3>
          <button class="speak-button"
                  data-title="${article.title}"
                  data-language="${language}"
                  onclick="speakArticle(this)">Speak</button>
        `;
        trendingContainer.appendChild(articleElement);
      });
    } catch (error) {
      console.error("Error loading trending articles:", error);
    }
  }

  function speakArticle(button) {
    const title = button.getAttribute('data-title');
    const language = button.getAttribute('data-language');
    const speech = new SpeechSynthesisUtterance();
    speech.text = title;
    speech.lang = language === 'en' ? 'en-US' : language;
    window.speechSynthesis.speak(speech);
  }

  function changeLanguage() {
    const language = document.getElementById("language").value;
    loadTrendingArticles(language);
  }

  document.addEventListener("DOMContentLoaded", () => {
    const defaultLanguage = document.getElementById("language").value;
    loadTrendingArticles(defaultLanguage);
  });
</script>
```

```
    <div class="back-button-container">
      <button onclick="window.location.href='/'" class="back-button">Back to Homepage</button>
    </div>
  </div>
 </body>
</html>
```

results.html

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="UTF-8" />
  <title>News Results</title>
  <style>
   body {
    font-family: Arial, sans-serif;
    color: white;
    background-color: #1a1a1a;
    padding: 20px;
    display: flex;
    flex-direction: column;
    align-items: center;
    min-height: 100vh;
   }

   h1 {
    font-size: 36px;
    text-align: center;
    color: #ffffff;
    background: linear-gradient(45deg, #00ffff, #ff00ff);
    -webkit-background-clip: text;
    -webkit-text-fill-color: transparent;
    margin-bottom: 30px;
   }

   #results {
    display: flex;
    justify-content: center;
    align-items: flex-start;
    width: 100%;
    max-width: 1200px;
    padding: 20px 10px;
   }
```

```
.sentiment-column {
  background-color: rgba(255, 255, 255, 0.1);
  padding: 20px;
  border-radius: 10px;
  width: 30%;
  max-height: 600px;
  overflow-y: auto;
  margin: 0 10px;
}

.sentiment-column::-webkit-scrollbar {
  width: 10px;
}

.sentiment-column::-webkit-scrollbar-thumb {
  background: linear-gradient(45deg, #00ffff, #ff00ff);
  border-radius: 10px;
  box-shadow: inset 0 0 6px rgba(0, 0, 0, 0.3);
}

.sentiment-column::-webkit-scrollbar-track {
  background: rgba(255, 255, 255, 0.1);
  border-radius: 10px;
}

.sentiment-column h2 {
  text-align: center;
  color: #ffffff;
  padding: 10px;
  border-radius: 8px;
  font-size: 24px;
  margin-bottom: 15px;
  background: linear-gradient(45deg, #00ffff, #ff00ff);
  text-shadow: none; /* No shadow by default */
  transition: text-shadow 0.3s ease, transform 0.3s ease;
}

.sentiment-column h2:hover {
  text-shadow: 0 0 10px rgba(0, 255, 255, 0.5),
    0 0 10px rgba(255, 0, 255, 0.5); /* Add shadow on hover */
  transform: translateY(-2px);
}

.news-article {
  border: 1px solid #333;
  padding: 15px;
  border-radius: 10px;
  margin-bottom: 15px;
  background-color: #2e2e2e;
  color: #ffffff;
  transition: transform 0.3s ease, box-shadow 0.3s ease;
}

.news-article:hover {
  transform: translateY(-2px);
  box-shadow: 0 0 10px rgba(0, 255, 255, 0.5),
    0 0 10px rgba(255, 0, 255, 0.5);
}

.news-link a {
  color: #ffcc00;
  font-weight: bold;
  text-decoration: none;
}

.news-link a:hover {
  text-decoration: underline;
}

.news-image {
  width: 100%;
  height: auto;
  border-radius: 10px;
  margin-bottom: 10px;
}

.news-article p {
  font-size: 16px;
  line-height: 1.5;
  color: #cccccc;
}

/* Speak button styling */
.speak-button {
  background: linear-gradient(45deg, #ff00ff, #00ffff);
  color: #ffffff;
  padding: 8px 16px;
  border: none;
  border-radius: 8px;
  cursor: pointer;
  font-size: 14px;
  transition: all 0.3s ease;
  display: block;
  margin: 10px auto 0;
}

.speak-button:hover {
  transform: translateY(-2px);
  box-shadow: 0 0 15px rgba(255, 0, 255, 0.6),
    0 0 15px rgba(0, 255, 255, 0.6);
    }

    .back-button-container {
      display: flex;
      justify-content: center;
      margin-top: auto;
      padding-top: 20px;
    }

    .back-button {
      background: linear-gradient(45deg, #00ffff, #ff00ff);
      color: #ffffff;
      padding: 10px 20px;
      border: none;
      border-radius: 8px;
      cursor: pointer;
      font-size: 16px;
      transition: all 0.3s ease;
    }

    .back-button:hover {
      transform: translateY(-2px);
      box-shadow: 0 0 10px rgba(0, 255, 255, 0.5),
        0 0 10px rgba(255, 0, 255, 0.5);
    }
  </style>
  <script>
    // Function to play text-to-speech using Web Speech API
    function playTextToSpeech(text, language) {
      const utterance = new SpeechSynthesisUtterance(text);
      utterance.lang = language; // Set language for speech synthesis
      speechSynthesis.speak(utterance); // Speak the text
    }

    document.addEventListener("DOMContentLoaded", function () {
      const speakButtons = document.querySelectorAll(".speak-button");
      speakButtons.forEach((button) => {
        button.addEventListener("click", function () {
          const articleTitle = button.dataset.title;
          const articleDescription = button.dataset.description;
          const lang = button.dataset.language;

          playTextToSpeech(articleTitle, lang);
          setTimeout(() => playTextToSpeech(articleDescription, lang), 3000);
        });
      });
    });
  </script>
</head>
<body>
```

```
    <h1>News Results</h1>
    <div id="results">
      {% for sentiment, article_list in articles.items() %}
      <div class="sentiment-column {{ sentiment|lower }}">
        <h2>{{ sentiment }} News</h2>
        {% for article in article_list %}
        <div class="news-article">
          <img
            src="{{ article.imageUrl }}"
            alt="News Image"
            class="news-image"
          />
          <h3 class="news-link">
            <a
              href="https://translate.google.com/translate?hl={{    language    }}&sl=auto&tl={{
language }}&u={{ article.url }}"
              target="_blank"
              >{{ article.title }}</a
            >
          </h3>
          <p>{{ article.description }}</p>
          <button
            class="speak-button"
            data-title="{{ article.title }}"
            data-description="{{ article.description }}"
            data-language="{{ language }}"
          >
            Speak
          </button>
        </div>
        {% endfor %}
      </div>
      {% endfor %}
    </div>

    <div class="back-button-container">
      <button onclick="window.location.href='/'" class="back-button">
        Back to Homepage
      </button>
    </div>
  </body>
</html>
```

6.2. Testing Phase

- **Unit Testing**:
  - Develop tests for individual functions, such as sentiment analysis and translation, to ensure they perform correctly under various scenarios.
- **Integration Testing**:
  - Verify that all components of the application (routing, API fetching, sentiment analysis, and translation) work seamlessly together.
- **Debugging**:
  - Identify and fix any issues found during testing to ensure the application runs smoothly without errors.

CHAPTER–7
SCREENSHOTS

7.1. Screenshots

Home Page: Welcome to News Translator, your gateway to breaking news in multiple languages. With real-time translation, global coverage, and an easy-to-use interface, stay informed about what's happening around the world in your preferred language.

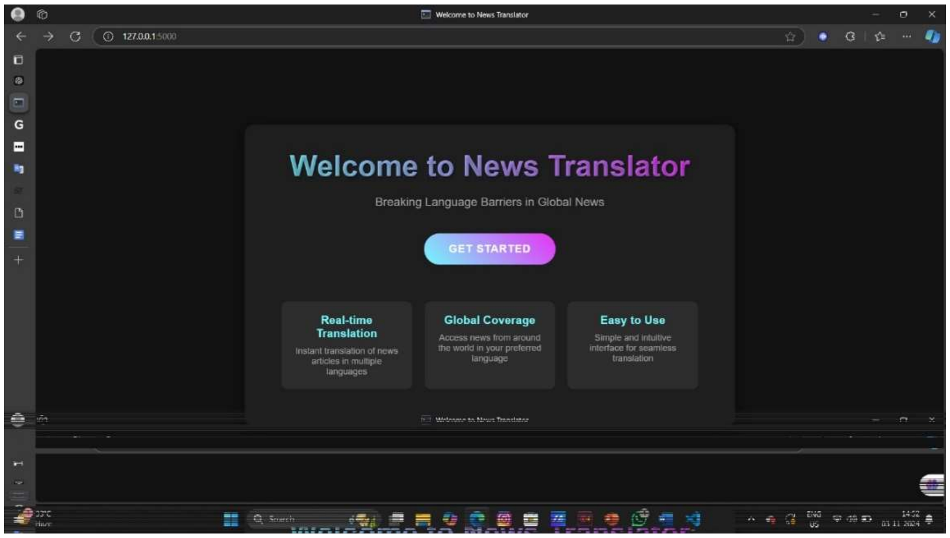

Welcome to News Translator
127.0.0.1:5000
Welcome to News Translator
Breaking Language Barriers in Global News
GET STARTED
Real-time Translation
Instant translation of news articles in multiple languages
Global Coverage
Access news from around the world in your preferred language
Easy to Use
Simple and intuitive interface for seamless translation

Figure 7.1.Home Page

Search Page: The Search Page in News Translator lets you explore trending news in your chosen language. Currently, top stories include the Gaza polio vaccination drive, rocket launches from Lebanon, and a controversial foul call in a recent basketball game.

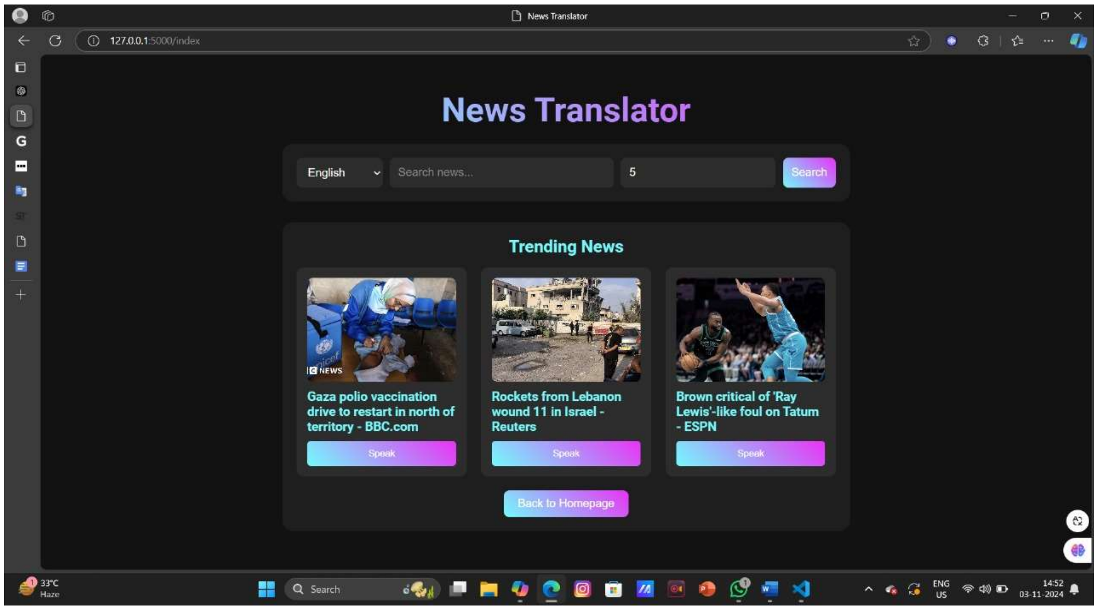
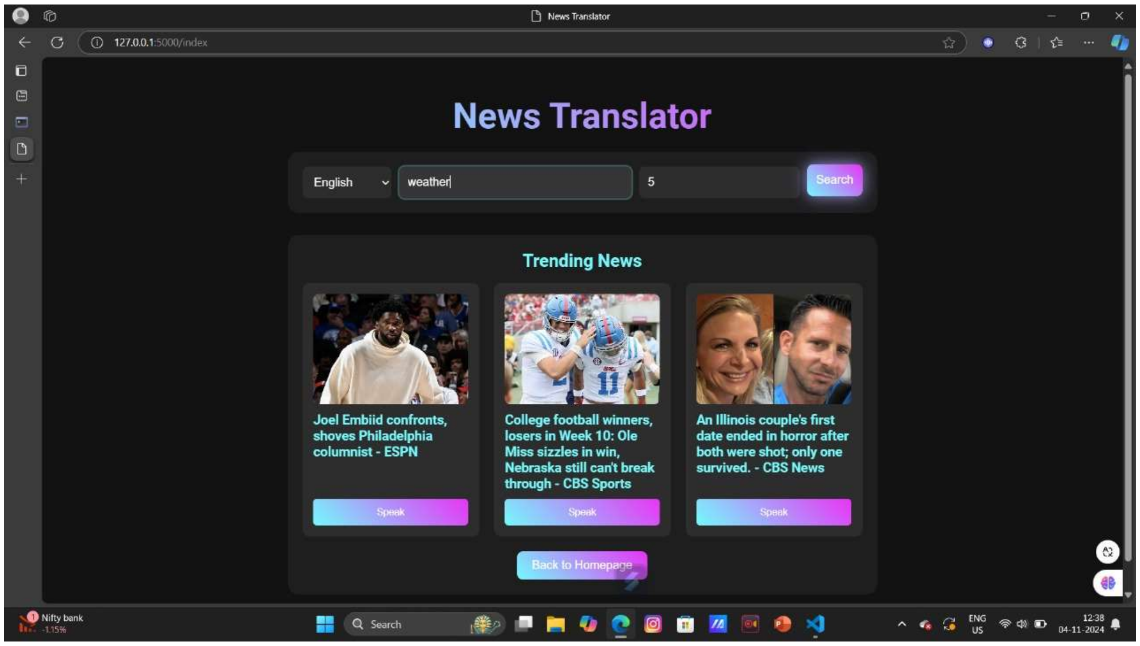

News Translator
127.0.0.1:5000/index
News Translator
English
Search news...
5
Search
Trending News
Gaza polio vaccination drive to restart in north of territory - BBC.com
Speak
Rockets from Lebanon wound 11 in Israel - Reuters
Speak
Brown critical of 'Ray Lewis'-like foul on Tatum - ESPN
Speak
Back to Homepage

Figure 7.1.Search Page

Example Page: For example, I want to search for weather in the search bar in English language, then the output will be given below



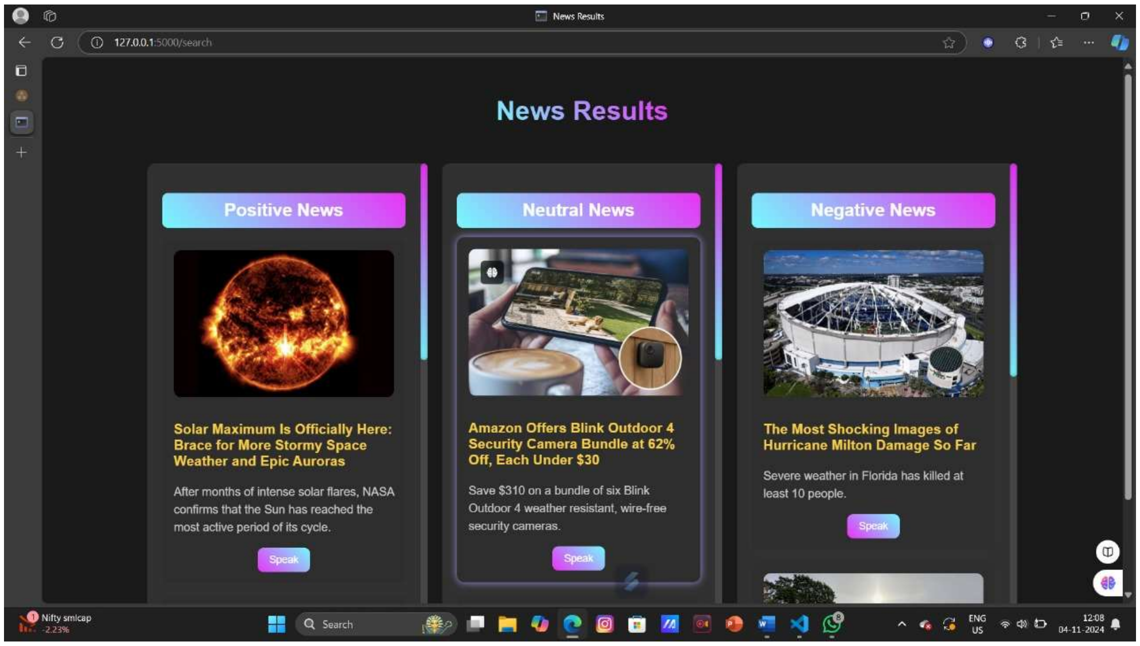
Joel Embiid confronts, shoves Philadelphia columnist - ESPN

College football winners, losers in Week 10: Ole Miss sizzles in win, Nebraska still can't break through - CBS Sports

An Illinois couple's first date ended in horror after both were shot; only one survived. - CBS News

Figure 7.1.Example Page

Results Page: The Result Page generates the results on the basis of the user as per their choose and it shows articles based on search criteria with sentiment analysis labels.



Solar Maximum Is Officially Here: Brace for More Stormy Space Weather and Epic Auroras

After months of intense solar flares, NASA confirms that the Sun has reached the most active period of its cycle.

Amazon Offers Blink Outdoor 4 Security Camera Bundle at 62% Off, Each Under $30

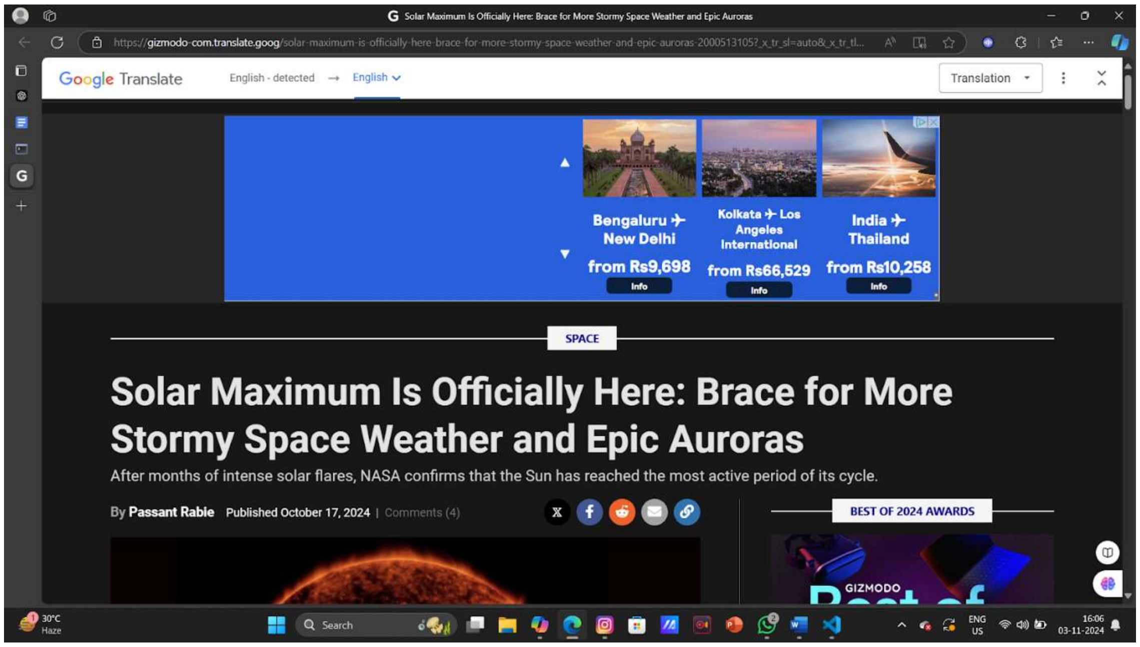
Save $310 on a bundle of six Blink Outdoor 4 weather resistant, wire-free security cameras.

The Most Shocking Images of Hurricane Milton Damage So Far

Severe weather in Florida has killed at least 10 people.

Figure 7.1.Result Page

After clicking any Article: It translates the NEWS into user preferred languages,generates the voice promt of the NEWS.



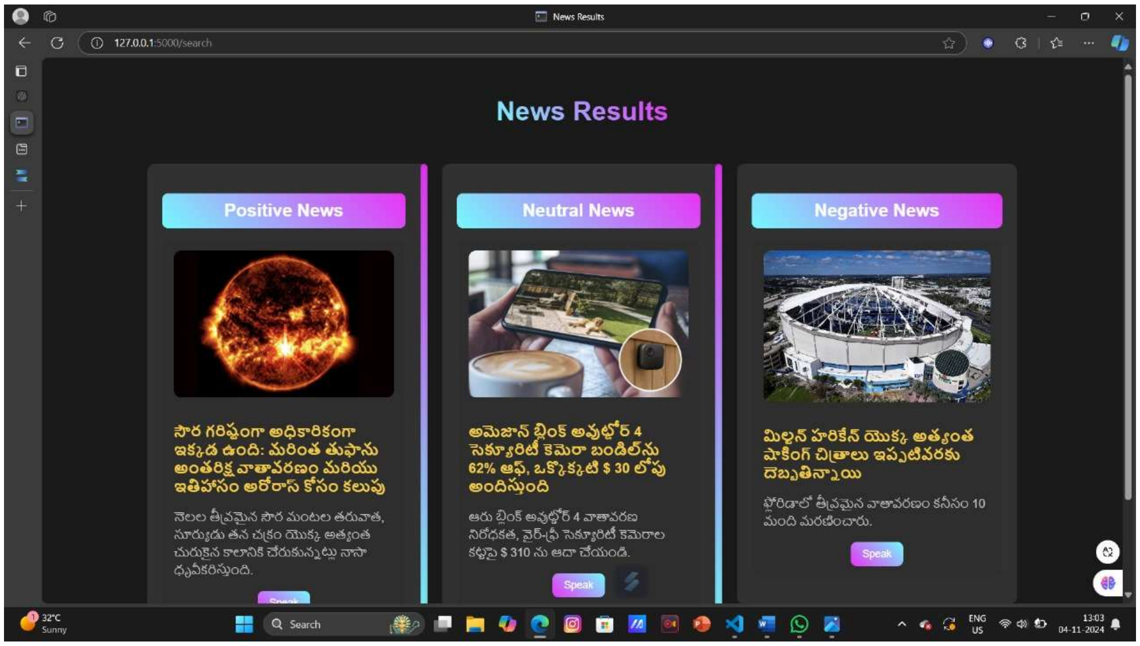
Solar Maximum Is Officially Here: Brace for More Stormy Space Weather and Epic Auroras

After months of intense solar flares, NASA confirms that the Sun has reached the most active period of its cycle.

By Passant Rabie Published October 17, 2024 | Comments (4)

Figure 7.1. Article Page

Translated L,.anguage: It translates the NEWS into user preferred languages.For example select Telugu language in search option.User can also change the language as per their wish.



సౌర గరిష్ఠంగా అధికారికంగా ఇక్కడ ఉంది: మరింత తుఫాను అంతరిక్ష వాతావరణం మరియు ఇతిహాసం అరోరాస్ కోసం కలుపు

నెలల తీవ్రమైన సౌర మంటల తరువాత, సూర్యుడు తన చక్రం యొక్క అత్యంత చురుకైన కాలానికి చేరుకున్నట్లు నాసా ధృవీకరిస్తుంది.

అమెజాన్ బ్లింక్ అవుట్డోర్ 4 సెక్యూరిటీ కెమెరా బండిల్‌ను 62% ఆఫ్, ఒక్కొక్కటి $ 30 లోపు అందిస్తుంది

ఆరు బ్లింక్ అవుట్డోర్ 4 వాతావరణ నిరోధకత, వైర్-ఫ్రీ సెక్యూరిటీ కెమెరాల కట్టపై $ 310 ను ఆదా చేయండి.

మిల్టన్ హరికేన్ యొక్క అత్యంత షాకింగ్ చిత్రాలు ఇప్పటివరకు దెబ్బతిన్నాయి

ఫ్లోరిడాలో తీవ్రమైన వాతావరణం కనీసం 10 మంది మరణించారు.

Figure 7.1.Translated Language

CHAPTER–8
CONCLUSION

8.1. Conclusion

The integration of these libraries and frameworks establishes a robust foundation for the AI News Translation System, empowering it to provide accurate and real-time translations seamlessly. Each component plays a crucial role in optimizing the system's performance and usability: from handling language processing tasks and performing sentiment analysis to converting text into natural-sounding audio. Together, they enhance the system's functionality, creating an efficient and user-friendly experience that effectively bridges language gaps in news delivery. This combination makes the platform a comprehensive solution for global news accessibility and understanding.

Purpose: The application allows users to search for news articles based on keywords, analyse their sentiment (positive, neutral, negative), and translate them into a desired language.

Key Features:

- **News Retrieval**: It fetches articles from the News API based on user-provided keywords and supports pagination.
- **Sentiment Analysis**: Each article is analysed for sentiment using TextBlob, categorising them as positive, neutral, or negative.
- **Translation**: Articles can be translated into the user's preferred language using the Google Translate API.
- **Trending News**: The application also retrieves trending news articles and provides them in the selected language.

Caching: To improve performance and reduce the load on the external API, the application uses Flask-Caching, storing results for a limited time.

User Interaction:

- Users can enter keywords and select a language for their searches.
- The results are displayed in a structured format, allowing for easy navigation and interaction with the articles

8.2. Future Work

- Language Preference Saving: Allow users to save their preferred language, so they don't need to select it every time they open the app.
- Dark Mode Toggle: Add a button to switch between dark and light mode, making the app easier to read in different lighting conditions.
- Additional Sorting Options: Enable sorting news articles by date, relevance, or popularity, giving users more control over how they view their feed.
- Share News Articles: Add a "Share" button to each article, so users can easily share translated articles with friends via social media or messaging apps.

- Bookmark Articles: Let users bookmark articles to save for later reading, giving them a simple way to keep track of interesting news.
- Basic User Feedback Form: Implement a simple feedback form for users to report translation errors or suggest improvements, making it easy to gather user insights.
- Multiple Keyword Search: Enable searching with multiple keywords or phrases, helping users find articles on specific topics more accurately.
- Language Learning Feature: Add a “Word of the Day” from the news content to help users learn a new word in their selected language each day.
- Article Refresh Button: Include a refresh button to quickly load the latest articles without reloading the whole page, improving the user experience.

CHAPTER–9
REFERENCES

1. Translating News in the Era of Globalization: Challenges and Strategies"(2018)
    - *Author*: Roberto A. Valdeón
    - *Journal*: Across Languages and Cultures
    - *Summary*: This article examines the impact of globalization on news translation, discussing strategies for handling cultural references, idiomatic expressions, and political nuances.
2. "Gatekeeping and Translation: A Study of News Translation Practices in International News Agencies"(2016)
    - *Author*: Piotr Blumczynski
    - *Journal*: Translation Studies
    - *Summary*: Focuses on the role of translators as gatekeepers, exploring the decisions they make and their influence on how news is presented to different cultural audiences.
3. "Ideological Manipulation in News Translation"(2017)
    - *Author*: Jonathan Ross
    - *Journal*: Journal of Language and Politics
    - *Summary*: This article discusses how translation can subtly or overtly alter the ideological slant of news, analyzing cases where political bias is introduced or mitigated through translation.
4. "Translation and Power in International News Media"(2019)
    - *Author*: Mona Baker
    - *Journal*: Critical Discourse Studies
    - *Summary*: Examines how power dynamics influence translation choices in global media, particularly how certain narratives are promoted or suppressed.
5. "Handling Cultural Specificity in News Translation"(2015)
    - *Author*: Esperança Bielsa
    - *Journal*: Language and Intercultural Communication
    - *Summary*: Focuses on the challenges of translating culturally specific references and concepts, proposing strategies for balancing accuracy with accessibility.
6. "The Ethics of News Translation: A Comparative Study of English and Arabic News Outlets"(2020)
    - *Author*: Ahmed Al-Salmi
    - *Journal*: Translation and Interpreting Studies

- *Summary*: Discusses ethical considerations in news translation, especially regarding sensitive topics in English and Arabic media, with case studies from news events.

7. "Machine Translation in Newsrooms: Boon or Bane?"(2021)
    - *Author*: Mary C. Liu
    - *Journal*: Journalism Studies
    - *Summary*: Investigates the growing use of machine translation in newsrooms, addressing its benefits and limitations in preserving the original meaning and tone of news stories.
8. "Transediting: Translating and Editing News for Multilingual Audiences"(2016)
    - *Author*: Federico Zanettin
    - *Journal*: Perspectives: Studies in Translation Theory and Practice
    - *Summary*: Introduces the concept of "transediting" – combining translation and editing – and its application in making news content suitable for various audiences.
9. "Cultural Filtering in News Translation"(2019)
    - *Author*: Dionysios Kapsaskis
    - *Journal*: Translation Spaces
    - *Summary*: Analyzes the process of "cultural filtering," where translators adapt content to be culturally acceptable and comprehensible to the target audience.
10. "Linguistic and Cultural Challenges in Translating Breaking News"(2018)
    - *Author*: Sara Olmos
    - *Journal*: International Journal of Translation and Interpreting
    - *Summary*: This article looks at the unique pressures involved in translating breaking news, focusing on the quick decision-making required to ensure both accuracy and speed.