Ajay Kumar Singh

ajaysingh002@gmail.com

Capstone Project PGP ILT\_B2

Developed a web application using Streamlit that enables users to translate text into various languages, convert the translated text into speech, and download the resulting audio file.

Generative AI & ML : Capstone project

22-Oct-2025

Contents

[Project Overview 2](#_Toc212044610)

[Features 2](#_Toc212044611)

[Core Functionality 2](#_Toc212044612)

[Supported File Formats 2](#_Toc212044613)

[Installation & Setup 3](#_Toc212044614)

[Prerequisites 3](#_Toc212044615)

[Step-by-Step Installation 3](#_Toc212044616)

[Usage Guide 4](#_Toc212044617)

[Basic Workflow 4](#_Toc212044618)

[Example Use Cases 4](#_Toc212044619)

[Technical Architecture 5](#_Toc212044620)

[Components 5](#_Toc212044621)

[Code Structure 5](#_Toc212044622)

[API Integration Details 6](#_Toc212044623)

[Gemini API 6](#_Toc212044624)

[gTTS API 6](#_Toc212044625)

[Limitations & Considerations 6](#_Toc212044626)

[Challenges & Solutions 7](#_Toc212044627)

[Error Handling 8](#_Toc212044628)

[Troubleshooting 8](#_Toc212044629)

[Security Considerations 9](#_Toc212044630)

[Technologies Used 9](#_Toc212044631)

[Screenshots 10](#_Toc212044632)

Multi-Language Text Translator & Speech Synthesizer

# Project Overview

A web-based application built with Streamlit that translates text into multiple languages using Google's Gemini API and converts the translated text into speech using Google Text-to-Speech (gTTS). The application supports both direct text input and file uploads in various formats. Project Code is available at <https://github.com/ajaysingh0021/translation_tts_app.git>

# Features

## Core Functionality

* **Multi-language Translation**: Supports 20+ languages including Spanish, French, German, Japanese, Chinese, Arabic, Hindi, and more
* **Text-to-Speech Conversion**: Generates natural-sounding audio from translated text
* **Multiple Input Methods**:
  + Direct text input via text area
  + File upload support (TXT, PDF, CSV, Excel)
* **Audio Download**: Download generated speech as MP3 files
* **User-friendly Interface**: Clean, intuitive design with clear instructions

## Supported File Formats

* **TXT**: Plain text files
* **PDF**: Portable Document Format files
* **CSV**: Comma-separated values
* **Excel**: .xlsx and .xls formats

# Installation & Setup

## Prerequisites

* Python 3.8 or higher
* Google Gemini API key (free tier available)
* Internet connection for API calls

## Step-by-Step Installation

1. **Clone or Download the Project**

mkdir translation\_tts\_app

cd translation\_tts\_app

1. **Install Required Dependencies**

pip install -r requirements.txt

1. **Obtain Gemini API Key**
   * Visit [Google AI Studio](https://aistudio.google.com/app/api-keys)
   * Sign in with your Google account
   * Click "Create API Key"
   * Copy the generated API key
2. **Run the Application**

streamlit run translation\_tts\_app.py

1. **Access the Application**
   * The app will automatically open in your browser
   * Default URL: <http://localhost:8501>
   * Enter your API key in the sidebar

# Usage Guide

## Basic Workflow

1. **Enter API Key**
   * Paste your Gemini API key in the sidebar
   * Wait for confirmation message
2. **Choose Input Method**
   * **Option A**: Enter text directly in the text area
   * **Option B**: Upload a file (TXT, PDF, CSV, or Excel)
3. **Select Target Language**
   * Choose from the dropdown menu (20+ languages available)
4. **Translate**
   * Click the "Translate" button
   * View original and translated text side by side
5. **Generate & Download Audio**
   * Audio is automatically generated after translation
   * Play audio directly in the browser
   * Download as MP3 file

## Example Use Cases

**Use Case 1: Document Translation**

1. Upload a PDF document
2. Select "Spanish" as target language
3. Click translate
4. Download audio for pronunciation practice

**Use Case 2: Quick Translation**

1. Type or paste text directly
2. Select "Japanese"
3. Translate and listen to pronunciation

**Use Case 3: Data Translation**

1. Upload CSV/Excel file with text data
2. Select target language
3. Translate structured data
4. Generate audio for each entry

# Technical Architecture

## Components

1. **Frontend (Streamlit)**

* User interface and interaction handling
* File upload processing
* Audio playback

1. **Translation Engine (Gemini API)**

* Natural language processing
* Context-aware translation
* Multi-language support

1. **Speech Synthesis (gTTS)**

* Text-to-speech conversion
* MP3 audio generation
* Multiple language support

## Code Structure

translation\_tts\_app.py

├── Configuration & Setup

├── Helper Functions

│ ├── extract\_text\_from\_pdf()

│ ├── extract\_text\_from\_excel()

│ ├── extract\_text\_from\_csv()

│ ├── translate\_text()

│ └── text\_to\_speech()

├── UI Components

│ ├── Sidebar (API key, settings)

│ ├── Input section

│ ├── Translation display

│ └── Audio player & download

└── Error Handling

# API Integration Details

## Gemini API

* **Model Used**: gemini-flash-latest
* **Purpose**: Text translation with context awareness
* **Error Handling**: Connection errors, invalid keys, rate limiting

## gTTS API

* **Service**: Google Text-to-Speech
* **Output Format**: MP3
* **Languages**: 50+ languages supported
* **Limitations**: Requires internet connection

# Limitations & Considerations

1. **Text Length**

* Maximum 10,000 characters per translation
* Long documents may need to be split

1. **File Size**

* PDF: Recommended max 10 MB
* Excel/CSV: Recommended max 5 MB

1. **API Dependencies**

* Requires active internet connection
* Subject to API rate limits
* API key required for each session

1. **Translation Quality**

* Context may be lost in very long texts
* Technical jargon might not translate perfectly
* Idiomatic expressions may be literal

1. **Audio Quality**

* gTTS uses automated voice (not human)
* Pronunciation may vary by language
* No control over voice speed or pitch in current version

# Challenges & Solutions

**Challenge 1: File Format Handling**

**Problem**: Different file formats require different extraction methods

**Solution**: Implemented separate handlers for each format (PyPDF2 for PDF, pandas for CSV/Excel)

**Challenge 2: API Rate Limiting**

**Problem**: Gemini API has rate limits that could affect user experience

**Solution**:

* Added error handling with user-friendly messages
* Implemented text length validation
* Added loading spinners for better UX

**Challenge 3: Audio Generation Delays**

**Problem**: gTTS can be slow for long texts

**Solution**:

* Added progress indicators
* Implemented efficient buffering
* Provide immediate feedback to users

**Challenge 4: API Key Security**

**Problem**: Protecting user API keys

**Solution**:

* Password-type input field
* Session-based storage (not persistent)
* Clear warnings about key security

**Challenge 5: Large PDF Processing**

**Problem**: Some PDFs contain images or complex formatting

**Solution**:

* Implemented try-catch blocks
* Added text preview functionality
* Clear error messages for unsupported content

# Error Handling

**Implemented Error Checks**

1. **Input Validation**

* Empty text detection
* Minimum character requirement
* Maximum length enforcement

1. **File Upload Errors**

* Unsupported file format detection
* Corrupted file handling
* File size warnings

1. **API Errors**

* Invalid API key detection
* Network connectivity issues
* Rate limit exceeded handling

1. **Translation Errors**

* Language not supported
* Translation failure recovery
* Timeout handling

# Troubleshooting

**Common Issues**

**Issue 1: "Invalid API Key" Error**

* Solution: Verify API key is correct, check for extra spaces

**Issue 2: Translation Not Working**

* Check internet connection
* Verify API key has not exceeded quota
* Try with shorter text

**Issue 3: Audio Not Playing**

* Check browser audio settings
* Try different browser.
* Ensure speakers/headphones are connected

**Issue 4: File Upload Failing**

* Verify file format is supported
* Check file is not corrupted
* Try smaller file size

# Security Considerations

1. **API Key Protection**
   * Never share your API key
   * Use password-type input
   * Key is not stored permanently
2. **Data Privacy**
   * Text is sent to Google APIs
   * No data is stored on server
   * Session-based processing only
3. **File Upload Security**
   * Files are processed in memory
   * No permanent storage
   * Automatic cleanup after session

# Technologies Used

* **Framework**: Streamlit
* **Translation**: Google Gemini API
* **Text-to-Speech**: Google Text-to-Speech (gTTS)
* **PDF Processing**: PyPDF2
* **Data Processing**: Pandas

# Screenshots

**Default layout with API Key entered**

A screenshot of a computer

AI-generated content may be incorrect.

**Upload file layout**

A screenshot of a computer

AI-generated content may be incorrect.

**Text-file upload feature**

A screenshot of a computer

AI-generated content may be incorrect.

**Translation result**

A screenshot of a computer

AI-generated content may be incorrect.