```
# 1. Kill the old server to fix "Port 5000 is in use"
import os
print("Stopping old server...")
!fuser -k 5000/tcp

# 2. Install libraries (if needed)
!pip install flask deep-translator gTTS pyngrok

# 3. Setup Tunnel
!npm install -g localtunnel

# 4. Import libraries
from flask import Flask, render_template_string, request, jsonify
from deep_translator import GoogleTranslator
from gtts import gTTS
import io
import base64
import threading
import requests

app = Flask(__name__)

# --- GET THE PASSWORD (IP ADDRESS) ---
# We print this clearly so you can copy it
try:
    public_ip = requests.get('https://api.ipify.org').text
    print("\n" + "="*40)
    print(f"🔐 TUNNEL PASSWORD:  {public_ip}")
    print("="*40 + "\n")
except:
    print("Could not get IP automatically. Run '!curl ipv4.icanhazip.com' if needed.")

# --- HTML Frontend ---
HTML_TEMPLATE = """
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Voice Translator India</title>
    <style>
        body { font-family: 'Segoe UI', sans-serif; background: #eef2f3; text-align: center; padding: 20px; }
        .container { background: white; padding: 30px; border-radius: 20px; max-width: 600px; margin: auto; box-shadow: 0 10
        h1 { color: #ff9933; margin: 0; }
        h3 { color: #138808; margin-top: 5px; margin-bottom: 25px; }

        .input-group { position: relative; margin-bottom: 20px; }
        textarea { width: 100%; height: 100px; padding: 15px; border-radius: 10px; border: 1px solid #ddd; resize: none; fon

        /* Microphone Button */
        .mic-btn {
            position: absolute; bottom: 10px; right: 10px;
            background: #dc3545; color: white; border: none;
            width: 40px; height: 40px; border-radius: 50%;
            cursor: pointer; font-size: 20px; box-shadow: 0 4px 6px rgba(0,0,0,0.2);
            transition: transform 0.2s, background 0.2s;
        }
        .mic-btn:hover { transform: scale(1.1); background: #c82333; }
        .mic-btn.listening { animation: pulse 1.5s infinite; background: #28a745; }

        @keyframes pulse {
            0% { box-shadow: 0 0 0 0 rgba(40, 167, 69, 0.7); }
            70% { box-shadow: 0 0 0 10px rgba(40, 167, 69, 0); }
            100% { box-shadow: 0 0 0 0 rgba(40, 167, 69, 0); }
        }

        .controls { display: flex; gap: 10px; margin-bottom: 20px; }
        select { flex: 1; padding: 12px; border-radius: 8px; border: 1px solid #ddd; font-size: 16px; }
        button.main-btn { flex: 0.5; background: #007bff; color: white; border: none; border-radius: 8px; font-weight: bold;
        button.main-btn:hover { background: #0056b3; }

        .output-area { background: #f8f9fa; border-radius: 10px; padding: 15px; text-align: left; min-height: 100px; positio
        .speak-btn {
            position: absolute; bottom: 10px; right: 10px;
            background: #28a745; color: white; border: none;
            padding: 8px 15px; border-radius: 20px; font-size: 14px; cursor: pointer;
        }
    </style>
</head>
<body>
    <div class="container">
```

```
<div class="container">
    <h1>IN Voice Translator</h1>
    <h3>Speak & Listen</h3>

    <div class="input-group">
        <textarea id="inputText" placeholder="Type or click the Mic to speak..."></textarea>
        <button class="mic-btn" id="micBtn" onclick="toggleMic()" title="Speak Now">🎤</button>
    </div>

    <div class="controls">
        <select id="langSelect">
            <option value="hi">Hindi (हिंदी)</option>
            <option value="bn">Bengali (বাংলা)</option>
            <option value="te">Telugu (తెలుగు)</option>
            <option value="mr">Marathi (मराठी)</option>
            <option value="ta">Tamil (தமிழ்)</option>
            <option value="gu">Gujarati (ગુજરાતી)</option>
            <option value="kn">Kannada (ಕನ್ನಡ)</option>
            <option value="ml">Malayalam (മലയാളം)</option>
            <option value="pa">Punjabi (ਪੰਜਾਬੀ)</option>
        </select>
        <button class="main-btn" onclick="translateText()">Translate</button>
    </div>

    <div class="output-area">
        <p id="outputText" style="margin: 0; color: #333;">Translation will appear here...</p>
        <button class="speak-btn" onclick="speakText()">🔊 Listen</button>
    </div>

    <audio id="audioPlayer"></audio>
</div>

<script>
    // --- 1. Speech to Text (Microphone) ---
    function toggleMic() {
        const micBtn = document.getElementById("micBtn");
        const inputBox = document.getElementById("inputText");

        // Check browser support
        if (!('webkitSpeechRecognition' in window)) {
            alert("Microphone not supported in this browser. Please use Chrome.");
            return;
        }

        const recognition = new webkitSpeechRecognition();
        recognition.lang = "en-US"; // Input language (English)
        recognition.interimResults = false;
        recognition.maxAlternatives = 1;

        recognition.start();
        micBtn.classList.add("listening");

        recognition.onresult = function(event) {
            const speechResult = event.results[0][0].transcript;
            inputBox.value = speechResult;
            micBtn.classList.remove("listening");
            setTimeout(translateText, 500);
        };

        recognition.onspeechend = function() {
            recognition.stop();
            micBtn.classList.remove("listening");
        };

        recognition.onerror = function(event) {
            console.error(event.error);
            micBtn.classList.remove("listening");
        };
    }

    // --- 2. Translation Logic ---
    async function translateText() {
        const text = document.getElementById("inputText").value;
        const target = document.getElementById("langSelect").value;
        const outText = document.getElementById("outputText");

        if(!text) return;
        outText.innerText = "Translating...";

        const res = await fetch('/translate', {
            method: 'POST',
            headers: {'Content-Type': 'application/json'},
            body: JSON.stringify({text, target})
        });
```

```
            });
            const data = await res.json();
            outText.innerText = data.translated_text;
        }

        // --- 3. Text to Speech (Audio Output) ---
        async function speakText() {
            const text = document.getElementById("outputText").innerText;
            const lang = document.getElementById("langSelect").value;

            if(!text || text === "Translating...") return;

            const res = await fetch('/speak', {
                method: 'POST',
                headers: {'Content-Type': 'application/json'},
                body: JSON.stringify({text, lang})
            });
            const data = await res.json();
            if(data.audio) {
                const audio = document.getElementById("audioPlayer");
                audio.src = "data:audio/mp3;base64," + data.audio;
                audio.play();
            }
        }
    </script>
</body>
</html>
"""

# --- Backend API ---
@app.route('/')
def home(): return render_template_string(HTML_TEMPLATE)

@app.route('/translate', methods=['POST'])
def translate():
    data = request.json
    try:
        translated = GoogleTranslator(source='auto', target=data.get('target', 'hi')).translate(data.get('text', ''))
        return jsonify({'translated_text': translated})
    except:
        return jsonify({'translated_text': "Error during translation"})

@app.route('/speak', methods=['POST'])
def speak():
    try:
        fp = io.BytesIO()
        gTTS(text=request.json.get('text'), lang=request.json.get('lang')).write_to_fp(fp)
        fp.seek(0)
        return jsonify({'audio': base64.b64encode(fp.read()).decode('utf-8')})
    except:
        return jsonify({'error': 'Audio error'})

# --- Run Server ---
if __name__ == '__main__':
    threading.Thread(target=app.run, kwargs={'port': 5000}).start()
    print("Starting... Click the link below!")
    !lt --port 5000
```

```
Stopping old server...
Requirement already satisfied: flask in /usr/local/lib/python3.12/dist-packages (3.1.2)
Requirement already satisfied: deep-translator in /usr/local/lib/python3.12/dist-packages (1.11.4)
Requirement already satisfied: gTTS in /usr/local/lib/python3.12/dist-packages (2.5.4)
Requirement already satisfied: pyngrok in /usr/local/lib/python3.12/dist-packages (7.5.0)
Requirement already satisfied: blinker>=1.9.0 in /usr/local/lib/python3.12/dist-packages (from flask) (1.9.0)
Requirement already satisfied: click>=8.1.3 in /usr/local/lib/python3.12/dist-packages (from flask) (8.1.8)
Requirement already satisfied: itsdangerous>=2.2.0 in /usr/local/lib/python3.12/dist-packages (from flask) (2.2.0)
Requirement already satisfied: jinja2>=3.1.2 in /usr/local/lib/python3.12/dist-packages (from flask) (3.1.6)
Requirement already satisfied: markupsafe>=2.1.1 in /usr/local/lib/python3.12/dist-packages (from flask) (3.0.3)
Requirement already satisfied: werkzeug>=3.1.0 in /usr/local/lib/python3.12/dist-packages (from flask) (3.1.5)
Requirement already satisfied: beautifulsoup4<5.0.0,>=4.9.1 in /usr/local/lib/python3.12/dist-packages (from deep-translat
Requirement already satisfied: requests<3.0.0,>=2.23.0 in /usr/local/lib/python3.12/dist-packages (from deep-translator) (
Requirement already satisfied: PyYAML>=5.1 in /usr/local/lib/python3.12/dist-packages (from pyngrok) (6.0.3)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.12/dist-packages (from beautifulsoup4<5.0.0,>=4.9.1
Requirement already satisfied: typing-extensions>=4.0.0 in /usr/local/lib/python3.12/dist-packages (from beautifulsoup4<5.
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.23.0->deep
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.23.0
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.23.0
⠙⠹⠸⠼⠴⠦⠧⠇⠏⠋⠙⠹⠸⠼⠴⠦⠧⠇⠏
changed 22 packages in 2s
⠏
⠏3 packages are looking for funding
⠏  run `npm fund` for details
⠏
========================================
🔐 TUNNEL PASSWORD:  35.233.198.137
========================================

Starting... Click the link below!
 * Serving Flask app '__main__'
 * Debug mode: off
INFO:werkzeug:WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI serve
 * Running on http://127.0.0.1:5000
INFO:werkzeug:Press CTRL+C to quit
your url is: https://strong-cats-flash.loca.lt
INFO:werkzeug:127.0.0.1 - - [16/Jan/2026 17:06:35] "GET /favicon.ico HTTP/1.1" 404 -
INFO:werkzeug:127.0.0.1 - - [16/Jan/2026 17:06:37] "GET / HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [16/Jan/2026 17:06:38] "GET /favicon.ico HTTP/1.1" 404 -
INFO:werkzeug:127.0.0.1 - - [16/Jan/2026 17:07:01] "POST /translate HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [16/Jan/2026 17:07:01] "POST /translate HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [16/Jan/2026 17:07:03] "POST /speak HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [16/Jan/2026 17:07:04] "GET /favicon.ico HTTP/1.1" 404 -
INFO:werkzeug:127.0.0.1 - - [16/Jan/2026 17:08:44] "POST /translate HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [16/Jan/2026 17:08:44] "POST /speak HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [16/Jan/2026 17:08:52] "POST /translate HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [16/Jan/2026 17:08:54] "POST /speak HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [16/Jan/2026 17:09:03] "POST /speak HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [16/Jan/2026 17:09:10] "POST /translate HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [16/Jan/2026 17:09:12] "POST /speak HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [16/Jan/2026 17:09:21] "POST /speak HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [16/Jan/2026 17:09:23] "POST /translate HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [16/Jan/2026 17:09:30] "POST /speak HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [16/Jan/2026 17:09:43] "POST /translate HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [16/Jan/2026 17:09:44] "POST /speak HTTP/1.1" 200 -
```