

AI Multilanguage Translator

Submitted by: Priyanshu
Project Report



Under the guidance of: Gaurav Singh

▼ Scroll to explore ▼

Objective

To build an AI-powered multilingual desktop translator that:

- Instantly translates across 12+ languages
- Preserves personal names using a custom name-masking algorithm
- Offers a modern, theme-aware graphical interface
- Saves user preferences (theme, last used languages, etc.) for a seamless experience



Technologies Used

Python 3.8+

Core programming language

Tkinter / TK

GUI framework from Python Standard Library

deep-translator

Google Translate API wrapper

JSON

Theme persistence and config management



Key Features

- Multilingual support (12+ languages)
- Auto-detection of source language
- Name-masking algorithm for privacy
- Light/Dark theme toggle (auto-saved)
- Swap button to flip source \Leftrightarrow target
- UI state saved via .theme config file



12+ Languages



Theme Support



Smart Algorithm

Name-Masking Algorithm





Steps:

1. Tokenize the input sentence
2. Mask capitalized words (likely names) with unique tokens
3. Translate the masked sentence using GoogleTranslator
4. Unmask the tokens back to original names

Time Complexity: $O(n)$ (linear, based on number of words)



UI Design Highlights

- **Font:** Segoe UI (clean & modern)
- **Window Size:** 900x600 px (minimum 850x500)
- **Themes:**
 -   Light Mode
 -   Dark Mode
- **Buttons:** Indigo #3151b5, bold with hover effect
- **Items:** Emoji-based controls (✓, 🟠) for intuitive UX

Translation Workflow

[UserInput]



[Name-Masking Algorithm]



[Auto Language Detection + Google Translate API]



[Unmask Names]



[Output Display]

How to Run

1. Ensure Python 3.8+ is installed

2. Install the required package:

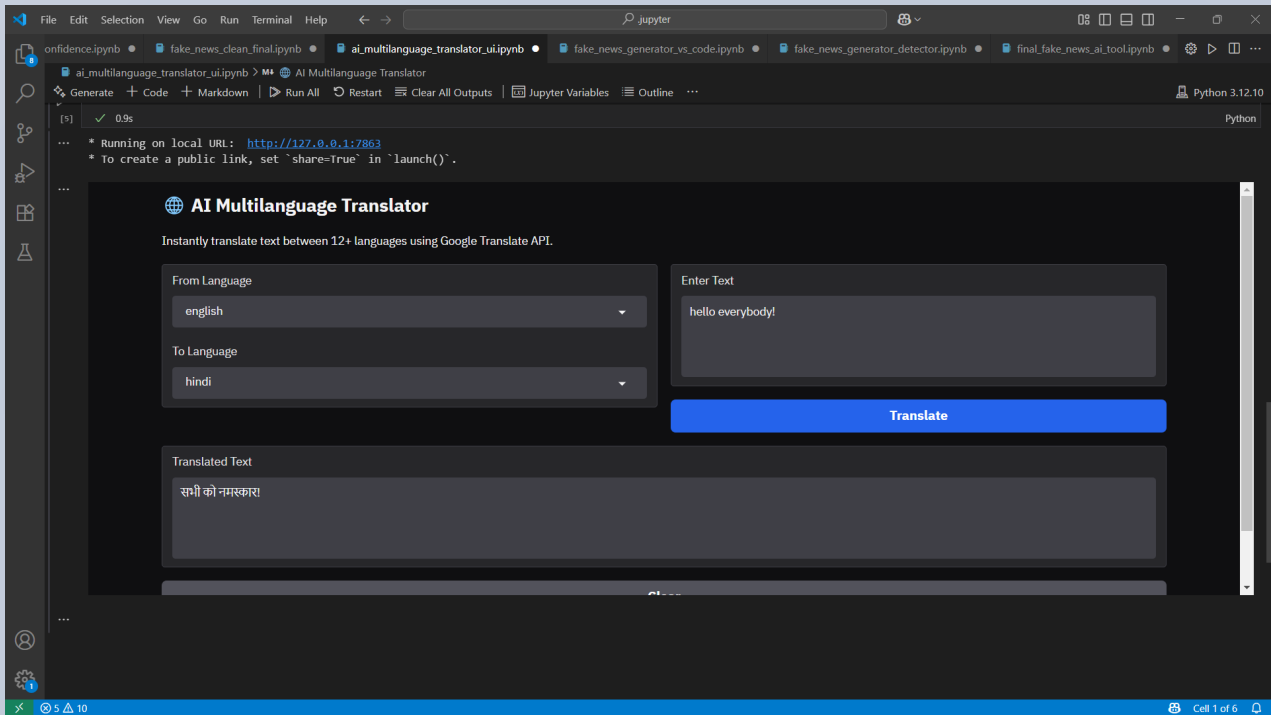
```
pip install deep-translator
```

3. Run the Translator GUI:

```
python translator_gui.py
```



Final Output



Future Enhancements

Copy-to-Clipboard Functionality

Add a dedicated button to copy translated text to clipboard for easy sharing and reuse.



Text-to-Speech Integration

Implement audio output capability to hear pronunciation of translated text.



OCR Translation

Add optical character recognition to translate text from images and scanned documents.



Standalone Executable

Package the application as a single executable file using PyInstaller for easier distribution.



10/10