



"الآنسة فصيحة"

نظرة عامة

الآنسة فصيحة هو نظام ترجمة ذكي يقدم ترجمة دقيقة وفصيحة من اللغة الإنجليزية إلى العربية، مدعوماً بشات بوت ذكي يعمل بتقنية ' Gemini AI ' للإجابة على استفسارات المستخدمين.

المميزات الرئيسية

- ترجمة نصوص مباشرة بجودة عالية
- ترجمة ملفات بصيغ متعددة (DOCX, TXT, EPUB)
- شات بوت ذكي مدعوم بـ Gemini AI
- واجهة جميلة بتصميم الساكورا الياباني
- تصميم متجاوب يعمل على جميع الأجهزة
- خصوصية تامة - لا يتم حفظ أي بيانات

التثبيت:

- تثبيت المكتبات # Backend

cd backend

pip install -r requirem.txt

- تثبيت المكتبات # Chatbot

cd ../chatbot

pip install -r requirements.txt

البدء السريع

مثال تشغيل نظام الترجمة:

bash

تشغيل # Backend (Flask)

cd backend

python app.py

الخادم سيعمل على: http://localhost:5000

مثال تشغيل الشات بوت

bash

إنشاء ملف وإضافة المفتاح # .env

echo "GEMINI_API_KEY=your_api_key_here" > .env

تشغيل # Chatbot (FastAPI)

cd chatbot

uvicorn app.chatapp:app --reload --port 8000

الخادم سيعمل على: http://localhost:8000

مثال استخدام API الترجمة:

python

import requests

ترجمة نص

```
response = requests.post('http://localhost:5000/api/translate-text',  
    json={  
        "text": "Machine learning is fascinating",  
        "use_two_stage": True  
    }  
)
```

```
print(response.json()['translation'])
```

#الناتج: "التعلم الآلي رائع"

مثال ترجمة ملف:

python

```
import requests
```

رفع وترجمة ملف

```
with open('document.pdf', 'rb') as file:
```

```
    response = requests.post('http://localhost:5000/api/translate-file',  
        files={'file': file},  
        data={'use_two_stage': 'true'})
```

حفظ الملف المترجم

```
with open('document_translated.pdf', 'wb') as output:
```

```
    output.write(response.content)
```

مثال استخدام الشات بوت:

python

```
import requests
```

إرسال رسالة للشات بوت

```
response = requests.post('http://localhost:8000/api/chat',
    json={
        "message": "كيف أترجم ملف كبير؟"
    }
)
```

```
print(response.json()['reply'])
```

مثال النصوص المتعددة

```
python
```

```
from translator import translate_texts
```

```
texts = [
    "Hello, how are you?",
    "Machine learning is the future.",
    "Python is a great programming language."
]
translations = translate_texts(texts, batch_size=5)
for original, translated in zip(texts, translations):
    print(f"{original} → {translated}")
```

للشآت بوت System Prompt مثال

ملاحظة: Gemini AI مدعوم في System Prompt

```
python
```

```
gemini_chat.py # في ملف
```

```
SYSTEM_INSTRUCTION = """
```

أنتِ "الآنسة فصيحة" — مساعدة افتراضية متخصصة في الترجمة والإجابة على الأسئلة.

"""

مرجع API :

Chat Completions (ترجمة نص)

POST http://localhost:5000/api/translate-text

الطلب:

```
json
{
  "text": "Your English text here",
  "use_two_stage": true
}
```

الاستجابة:

```
json
{
  "success": true,
  "translation": "النص العربي المترجم"
}
```

File Translation (ترجمة ملف)

POST http://localhost:5000/api/translate-file

: Form Data الطلب

- file: الملف المراد ترجمته
- use_two_stage: true/false

الاستجابة: ملف مترجم بنفس الصيغة

Chatbot (الشات بوت)

POST http://localhost:8000/api/chat

الطلب:

```
json
{
  "message": "رسالتك هنا"
}
```

الاستجابة:

```
Json {
  "source": "FAQ" أو "Gemini",
  "reply": "الرد هنا"
```

}

Health Check (فحص الحالة)

GET http://localhost:5000/api/health

GET http://localhost:8000/

التقنيات المستخدمة:

Backend

- Python 3.9+
- Flask 3.0.0 - نظام الترجمة
- FastAPI - الشات بوت
- Transformers 4.35.0 - نموذج الترجمة
- Helsinki-NLP/opus-mt-en-ar - نموذج الترجمة العصبي
- Google Gemini AI - الذكاء الاصطناعي

Frontend

- HTML5 / CSS3 / JavaScript (ES6+)
- docx.js 8.0.4 - توليد ملفات Word

المكتبات الرئيسية:

flask==3.0.0

transformers==4.35.0

torch==2.1.0

fastapi

google-generativeai

python-docx==1.1.0

ebooklib==0.18

هيكل المشروع :

```
miss-fasiha/
| — backend/          # نظام الترجمة
| | | — app.py
| | | — translator.py
| | | — requirem.txt
| | | — file_handlers/
| | |   | — txt_handler.py
| | |   | — docx_handler.py
| | |   | — epub_handler.py
| | — chatbot/        # الشات بوت
| |   | — chatapp.py
| |   | — requirements.txt
| |   | — app/
| |   | — gemini_chat.py
| |   | — faq.json
| — frontend/        # الواجهة
| | — index.html
| | — style.css
| | — script.js
| | — assets/
| |   | — logo.png
| |   | — sakura-tree-bg.png
| |   | — sakura-flower-petal.png
| — Start_all.bat     # مختصر لتشغيل السيرفرات
| — docs/ # التوثيق
| — تقرير_المشروع.pdf
```

طريقة الاستخدام

ترجمة نص

١. اكتب أو الصق النص الإنجليزي
٢. اضغط على "ترجمة الآن"
٣. ستظهر الترجمة العربية الفصيحة

ترجمة ملف

١. اضغط على أيقونة رفع ملف
٢. اختر الملف (DOCX, TXT, EPUB)
٣. اضغط "ترجمة الآن"
٤. حمل الملف المترجم بنفس الصيغة

استخدام الشات بوت

١. اضغط على أيقونة المساعدة
٢. اكتب سؤالك بالعربية (فصحى أو عامية)
٣. ستحصل على إجابة فورية

مقاييس الأداء

النتيجة	المقياس
97.7%	دقة الترجمة (EM Rate)
2.0 S	متوسط وقت الاستجابة
2.27%	معدل الهلوسة
95%+	نجاح العمليات

الخصوصية والأمان

- لا يتم حفظ الملفات أو النصوص
- التخزين المؤقت فقط (يُحذف بعد تحديث الصفحة)
- لا يتم جمع بيانات المستخدمين
- مفاتيح API محمية في ملف .env

- def cleanup_old_files():
- for folder in [UPLOAD_FOLDER, OUTPUT_FOLDER]:
- for filename in os.listdir(folder):
- if current_time - os.path.getmtime(filepath) > 3600:
- os.remove(filepath) # حذف بعد ساعة

حل المشاكل الشائعة

خطأ في الاتصال بالBackend

#تأكد من تشغيل الخوادم

```
python backend/app.py # المنفذ ٥٠٠٠
```

```
uvicorn chatbot.app.chatapp:app --reload --port 8000
```

الترجمة بطيئة

- الملفات الكبيرة (>5MB) قد تستغرق دقائق
- تأكد من اتصال الإنترنت لـ Gemini AI

خطأ في Gemini API

#تأكد من صحة المفتاح في .env

```
cat chatbot/.env
```

فريق العمل

- رحمه موسى الأسمرى
- وسن حسن حقي
- رغد حسن الزهراني
- فاطمة عبد الرحمن الغامدي

بإشراف / مزنة العقلا

التواصل

- البريد : missfasihah@gmail.com

الترخيص

هذا المشروع تعليمي تطبيقي من الكلية التقنية الرقمية للبنات بجدة
دبلوم تقنية برمجة وتطوير الويب - الفصل التدريبي الأول ١٤٤٧ هـ

تم التطوير بواسطة فريق الأنسة فصيحة

المراجع

Hugging Face - نموذج Helsinki-NLP

Google AI - Gemini API

Flask

FastAPI

الأنسة فصيحة - جميع الحقوق محفوظة ©2025