

BÁO CÁO CHI TIẾT

Phần Cứng và Phần Mềm

Máy Chủ (Server) và Máy Khách (Client)

Dự án: CSV Translator Pro

Ứng dụng dịch thuật đa ngôn ngữ sử dụng AI

Mô hình: google/translategemma-27b-it

Kiến trúc: Client–Server (Web Application)

Backend: FastAPI + PyTorch + Hugging Face Transformers

Frontend: React 19 + Vite 7

Mô hình AI: TranslateGemma 27B-IT (27 tỷ tham số)

Ngôn ngữ hỗ trợ: 55 ngôn ngữ

Ngày lập: Ngày 6 tháng 2 năm 2026

Mục lục

1 Tổng quan dự án

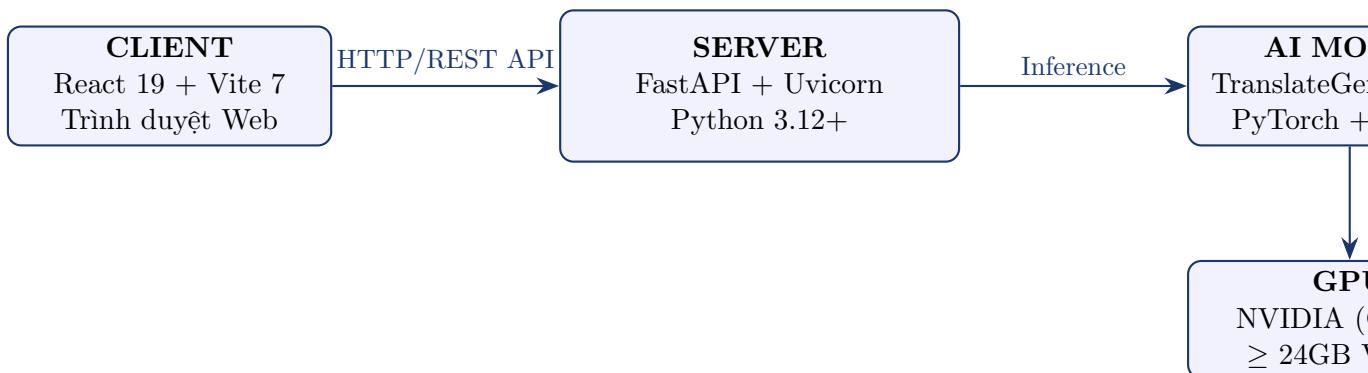
1.1 Giới thiệu

CSV Translator Pro là một ứng dụng web dịch thuật đa ngôn ngữ sử dụng trí tuệ nhân tạo, được thiết kế với kiến trúc Client–Server. Ứng dụng sử dụng mô hình ngôn ngữ lớn **TranslateGemma-27B-IT** của Google — một mô hình dịch thuật tiên tiến thuộc họ Gemma 3 với 27 tỷ tham số, hỗ trợ dịch thuật chính xác qua 55 ngôn ngữ.

1.2 Các chức năng chính

- Dịch file CSV hàng loạt:** Tải lên file CSV, dịch tự động cột “Text” và tải về kết quả.
- Dịch văn bản trực tiếp:** Nhập văn bản và nhận bản dịch theo thời gian thực.
- Dịch văn bản từ ảnh (OCR):** Trích xuất và dịch văn bản nhúng trong hình ảnh.
- Hỗ trợ triển khai offline:** Đóng gói toàn bộ mô hình và thư viện để chạy không cần Internet.

1.3 Kiến trúc hệ thống tổng quan



Hình 1: Sơ đồ kiến trúc tổng quan hệ thống CSV Translator Pro

2 Thông số mô hình TranslateGemma-27B-IT

2.1 Tổng quan mô hình

TranslateGemma-27B-IT là mô hình dịch thuật mã nguồn mở tiên tiến nhất trong họ TranslateGemma của Google, được xây dựng trên nền tảng kiến trúc Gemma 3. Đây là phiên bản lớn nhất, mang lại độ chính xác dịch thuật cao nhất, phù hợp cho môi trường production server.

Bảng 1: Thông số kỹ thuật mô hình TranslateGemma-27B-IT

Thông số	Giá trị
Tên mô hình	google/translategemma-27b-it
Số tham số (Parameters)	27 tỷ (27B)
Kiến trúc	Decoder-only Transformer (Gemma 3)
Số lớp (Layers)	46
Hidden Dimension	4096
Attention Heads	64
Key-Value Heads	16 (Grouped-Query Attention)
Position Embedding	RoPE (Rotary Position Embedding)
Normalization	RMS Normalization
Context Length (Đầu vào)	2,048 tokens
Hỗ trợ ảnh	Có (896×896 , mã hóa 256 tokens/ảnh)
Số ngôn ngữ hỗ trợ	55 ngôn ngữ
Định dạng trọng số	Safetensors
Giấy phép	Gemma License
Framework huấn luyện	JAX + ML Pathways
Phần cứng huấn luyện	Google TPUv4p, TPUv5p, TPUv5e
Dữ liệu SFT	4.3 tỷ tokens
Dữ liệu RLHF	10.2 triệu tokens

2.2 So sánh các phiên bản TranslateGemma

Bảng 2: So sánh benchmark giữa các phiên bản TranslateGemma

Benchmark	4B	12B	27B
WMT24++ MetricX ↓	5.32	3.60	3.09
WMT24++ COMET ↑	81.6	83.5	84.4
WMT25 MQM ↓	N/A	7.94	5.86
Vistra MetricX ↓	2.57	2.08	1.57

Lưu ý về mô hình 27B so với 4B hiện tại

Dự án hiện đang sử dụng `google/translategemma-4b-it` (4 tỷ tham số). Việc nâng cấp lên `google/translategemma-27b-it` (27 tỷ tham số) sẽ mang lại:

- Chất lượng dịch thuật tốt hơn đáng kể (MetricX giảm từ 5.32 xuống 3.09)
- Yêu cầu phần cứng cao hơn nhiều (đặc biệt về VRAM GPU)
- Tốc độ suy luận (inference) chậm hơn do kích thước mô hình lớn hơn 6.75 lần

2.3 Yêu cầu VRAM theo phương pháp lượng tử hóa

Bảng 3: Yêu cầu VRAM GPU cho TranslateGemma-27B-IT

Phương pháp	VRAM (Model)	VRAM (Thực tế)	Ghi chú
FP16 / BF16	~54 GB	~58–62 GB	Chất lượng tốt nhất, cần GPU cao cấp (A100/H100)
INT8 (8-bit)	~27 GB	~30–34 GB	Cân bằng chất lượng/hiệu năng, cần GPU \geq 32 GB
INT4 / NF4 (4-bit)	~14.1 GB	~18–22 GB	Giảm chất lượng nhẹ, chạy được trên RTX 3090/4090
QAT INT4	~14.1 GB	~18–20 GB	Google QAT — giữ chất lượng tốt hơn PTQ

VRAM thực tế cao hơn trọng số mô hình

VRAM thực tế khi suy luận luôn cao hơn kích thước trọng số mô hình vì cần thêm bộ nhớ cho:

- KV-Cache (Key-Value Cache) cho attention mechanism
- Activation memory cho quá trình forward pass
- Buffer của CUDA runtime và framework
- Context window (cửa sổ ngữ cảnh) — lên đến 2K tokens

3 Phần cứng máy chủ (Server)

Máy chủ là thành phần cốt lõi, chịu trách nhiệm chạy mô hình AI TranslateGemma-27B-IT và xử lý tất cả yêu cầu dịch thuật. Yêu cầu phần cứng phụ thuộc chủ yếu vào phương pháp lượng tử hóa được chọn.

3.1 GPU — Card đồ họa (Thành phần quan trọng nhất)

Bảng 4: Các cấu hình GPU đề xuất cho TranslateGemma-27B-IT

GPU	VRAM	Quantization	Mức	Ghi chú
Cấu hình tối thiểu				
RTX 3090	24 GB	NF4 (4-bit)	Tối thiểu	VRAM vừa đủ, có thể gặp OOM với văn bản dài
RTX 4090	24 GB	NF4 (4-bit)	Tối thiểu	Nhanh hơn 3090, kiến trúc Ada Lovelace
Cấu hình khuyến nghị				
RTX A5000	24 GB	NF4 (4-bit)	Khuyến nghị	ECC memory, dòng workstation ổn định
RTX A6000	48 GB	INT8 (8-bit)	Khuyến nghị	Đủ VRAM cho 8-bit, chất lượng tốt hơn
RTX 6000 Ada	48 GB	INT8 / FP16	Khuyến nghị	Thế hệ mới, hiệu năng cao
Cấu hình tối ưu (Production)				
NVIDIA A100	80 GB	FP16 / BF16	Tối ưu	Tiêu chuẩn datacenter, Tensor Cores thế hệ 3
NVIDIA H100	80 GB	BF16	Tối ưu	Hiệu năng AI tốt nhất, Tensor Cores thế hệ 4
NVIDIA L40S	48 GB	INT8 / FP16	Tối ưu	Ada Lovelace datacenter, giá tốt hơn A100

Khuyến nghị GPU cho dự án này

- **Ngân sách hạn chế:** NVIDIA RTX 4090 (24 GB) + NF4 4-bit quantization
- **Cân bằng:** NVIDIA RTX A6000 (48 GB) + INT8 8-bit quantization
- **Production/Doanh nghiệp:** NVIDIA A100 80 GB hoặc H100 80 GB + FP16/BF16
- **Multi-GPU:** 2× RTX 4090 hoặc 2× A6000 với `device_map="auto"` cho

model sharding

3.2 CPU — Bộ xử lý trung tâm

Bảng 5: Yêu cầu CPU cho máy chủ

Thông số	Tối thiểu	Khuyến nghị	Tối ưu
Bộ xử lý	Intel Core i7-12700 hoặc AMD Ryzen 7 5800X	Intel Xeon W-2255 hoặc AMD EPYC 7313	Intel Xeon Gold 6338 hoặc AMD EPYC 7543
Số nhân/luồng	≥ 8 nhân / 16 luồng	≥ 16 nhân / 32 luồng	≥ 32 nhân / 64 luồng
Tần số	≥ 3.0 GHz	≥ 3.4 GHz	≥ 2.8 GHz (nhiều nhân)
Hỗ trợ PCIe	PCIe 4.0 x16	PCIe 4.0 x16	PCIe 5.0 x16

Vai trò CPU trong hệ thống

CPU chủ yếu đảm nhiệm:

- Chạy FastAPI web server (Uvicorn ASGI)
- Tiền xử lý dữ liệu (đọc CSV, xử lý ảnh với Pillow)
- Quản lý hàng đợi công việc (job queue) và background tasks
- Tokenization văn bản đầu vào
- Truyền dữ liệu giữa CPU \leftrightarrow GPU qua PCIe bus

Phần lớn tính toán nặng (inference) được GPU đảm nhiệm, nên CPU không cần quá mạnh.

3.3 RAM — Bộ nhớ hệ thống

Bảng 6: Yêu cầu RAM cho máy chủ

Thông số	Tối thiểu	Khuyến nghị	Tối ưu
Dung lượng	32 GB	64 GB	128 GB
Loại	DDR4-3200	DDR4-3600 / DDR5-4800	DDR5-5600 ECC
Kênh	Dual Channel	Dual Channel	Quad Channel

Lý do cần nhiều RAM:

- Mô hình 27B cần tải trọng số vào RAM trước khi chuyển sang GPU (\sim 27–54 GB tùy precision)
- Pandas DataFrame khi xử lý CSV lớn chiếm bộ nhớ
- Xử lý ảnh (Pillow) cần buffer trong RAM
- Hệ điều hành và các dịch vụ nền
- Overhead cho Python runtime và garbage collection

3.4 Ổ cứng — Lưu trữ

Bảng 7: Yêu cầu lưu trữ cho máy chủ

Thành phần	Mô tả	Dung lượng ước tính
Trọng số mô hình (FP16)	Cache Hugging Face: ~/.cache/huggingface/hub/	\sim 54 GB
Trọng số mô hình (4-bit)	Phiên bản đã lượng tử hóa	\sim 15 GB
Hệ điều hành	Windows Server / Ubuntu	\sim 30–50 GB
Python + Thư viện	Virtual environment, PyTorch, CUDA	\sim 15–20 GB
CUDA Toolkit	Driver + Runtime libraries	\sim 5–8 GB
Dữ liệu tạm (uploads/outputs)	File CSV tải lên và kết quả dịch	\sim 10–50 GB
Tổng cộng		\sim130–240 GB

Yêu cầu SSD bắt buộc

Bắt buộc sử dụng SSD NVMe cho ổ chứa mô hình:

- Tải mô hình 27B từ ổ cứng lần đầu mất 30–120 giây (SSD) vs 5–15 phút (HDD)
- Khuyến nghị: SSD NVMe Gen4 \geq 500 GB, tốc độ đọc \geq 5,000 MB/s
- Dung lượng tối thiểu: 256 GB (chỉ đủ cho mô hình 4-bit + hệ thống)

3.5 Nguồn điện (PSU)

Bảng 8: Yêu cầu nguồn điện theo cấu hình GPU

Cấu hình GPU	TDP GPU	TDP Hệ thống	PSU khuyến nghị
1x RTX 4090	450W	~650W	≥ 850W (80+ Gold)
1x RTX A6000	300W	~550W	≥ 750W (80+ Gold)
1x A100 PCIe	300W	~550W	≥ 750W (80+ Platinum)
2x RTX 4090	900W	~1200W	≥ 1600W (80+ Platinum)

3.6 Mạng (Network)

Bảng 9: Yêu cầu mạng cho máy chủ

Thông số	Yêu cầu
Tốc độ mạng LAN	≥ 1 Gbps (khuyến nghị 10 Gbps cho nhiều client đồng thời)
Internet	Chỉ cần khi tải mô hình lần đầu. Sau đó có thể chạy offline hoàn toàn
Cổng mạng	Port 8000 (FastAPI mặc định), có thể tùy chỉnh
Giao thức	HTTP/HTTPS (khuyến nghị HTTPS với reverse proxy)

3.7 Tổng hợp cấu hình phần cứng máy chủ

Bảng 10: Tổng hợp 3 mức cấu hình phần cứng máy chủ

Thành phần	Tối thiểu	Khuyến nghị	Tối ưu (Production)
GPU	RTX 4090 24GB	RTX A6000 48GB	A100 80GB / H100 80GB
Quantization	NF4 4-bit	INT8 8-bit	FP16 / BF16
CPU	i7-12700 (8C/16T)	Xeon W-2255 (16C/32T)	EPYC 7543 (32C/64T)
RAM	32 GB DDR4	64 GB DDR4/DDR5	128 GB DDR5 ECC
SSD	256 GB NVMe	500 GB NVMe Gen4	1 TB NVMe Gen4/5
PSU	850W 80+ Gold	750W 80+ Gold	750W+ 80+ Platinum
Mạng	1 Gbps	1 Gbps	10 Gbps
Chi phí ước tính	~\$3,000–4,000	~\$6,000–10,000	~\$15,000–30,000

4 Phần mềm máy chủ (Server)

4.1 Hệ điều hành

Bảng 11: Hệ điều hành được hỗ trợ

Hệ điều hành	Phiên bản	Ghi chú
Ubuntu Server	22.04 LTS / 24.04 LTS	Khuyến nghị – Hỗ trợ tốt nhất cho CUDA & bitsandbytes
Windows Server	2022	Hỗ trợ, nhưng bitsandbytes (quantization) hạn chế
Windows 10/11	Pro/Enterprise	Dùng cho phát triển/thử nghiệm
Rocky Linux	9.x	Thay thế CentOS cho môi trường enterprise

4.2 CUDA Toolkit và NVIDIA Driver

Bảng 12: Yêu cầu CUDA và Driver

Thành phần	Phiên bản yêu cầu
NVIDIA Driver	$\geq 550.x$ (hỗ trợ CUDA 12.4)
CUDA Toolkit	12.4 (tương thích với PyTorch 2.6.0+cu124)
cuDNN	≥ 8.9 (đi kèm với CUDA Toolkit)
NCCL	≥ 2.20 (cần cho multi-GPU, nếu sử dụng)

4.3 Python Runtime và môi trường ảo

Bảng 13: Môi trường Python

Thành phần	Chi tiết
Python	≥ 3.10 , khuyến nghị 3.12.x
Pip	≥ 23.0
Virtual Environment	<code>python -m venv .my-env</code> hoặc Conda

4.4 Thư viện Python — Backend Dependencies

Bảng 14: Danh sách thư viện Python chính

Thư viện	Phiên bản	Mục đích
Framework Web		
<code>fastapi</code>	latest	Framework web API hiệu năng cao (ASGI)
<code>uvicorn[standard]</code>	latest	ASGI server chạy FastAPI
<code>python-multipart</code>	latest	Xử lý upload file (multipart form data)
<code>aiofiles</code>	latest	Đọc/ghi file bất đồng bộ (async I/O)
Machine Learning / AI		
<code>torch</code>	$\geq 2.6.0+\text{cu124}$	PyTorch với hỗ trợ CUDA 12.4
<code>torchvision</code>	$\geq 0.21.0+\text{cu124}$	Xử lý ảnh cho PyTorch
<code>torchaudio</code>	$\geq 2.6.0+\text{cu124}$	Xử lý âm thanh (dependency)
<code>transformers</code>	latest	Hugging Face Transformers — tải và chạy mô hình
<code>accelerate</code>	latest	Tối ưu hóa tải mô hình và device mapping
<code>bitsandbytes</code>	latest	Lượng tử hóa 4-bit/8-bit (NF4, INT8)
<code>sentencepiece</code>	latest	Tokenizer cho mô hình Gemma
<code>huggingface_hub</code>	latest	Tải/quản lý mô hình từ Hugging Face Hub
<code>sacremoses</code>	latest	Tiền xử lý văn bản (tokenization, detokenization)
Xử lý dữ liệu		
<code>pandas</code>	latest	Đọc/ghi và xử lý file CSV
<code>Pillow</code>	latest	Xử lý hình ảnh (resize, convert, OCR input)
<code>requests</code>	latest	HTTP client (tải ảnh từ URL)
<code>packaging</code>	latest	Quản lý phiên bản thư viện

4.5 Cài đặt PyTorch với CUDA

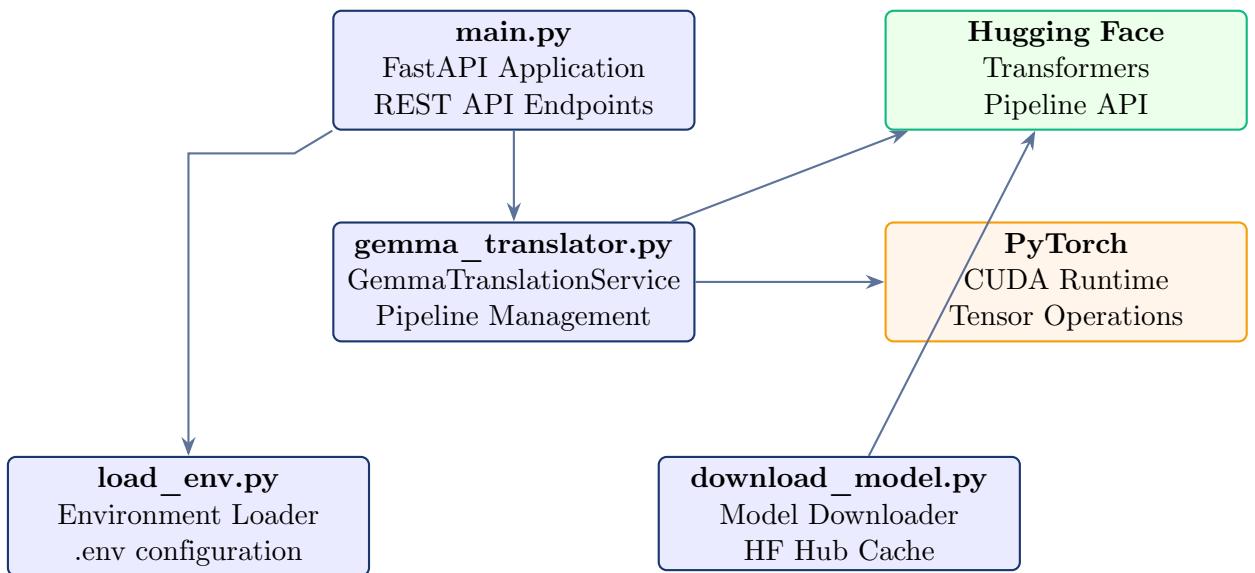
```

1 # CUDA 12.4 (khuyn ngh )
2 pip install torch==2.6.0+cu124 \
3     torchvision==0.21.0+cu124 \
4     torchaudio==2.6.0+cu124 \
5     --index-url https://download.pytorch.org/whl/cu124
6
7 # CPU only (ch dung khi kh ng c GPU - rt ch m)
8 pip install torch==2.6.0 torchvision==0.21.0 torchaudio==2.6.0

```

Listing 1: Lệnh cài đặt PyTorch cho CUDA 12.4

4.6 Kiến trúc phần mềm Backend



Hình 2: Sơ đồ module phần mềm Backend

4.7 API Endpoints

Bảng 15: Danh sách API Endpoints

Method	Endpoint	Mô tả
GET	/api/languages	Lấy danh sách ngôn ngữ được hỗ trợ
POST	/api/translate-text	Dịch một đoạn văn bản
POST	/api/translate-image	Trích xuất và dịch văn bản từ ảnh
POST	/api/upload	Tải lên file CSV và bắt đầu dịch hàng loạt
GET	/api/status/{job_id}	Kiểm tra tiến trình dịch CSV
GET	/api/download/{job_id}	Tải file CSV đã dịch
GET	/api/health	Kiểm tra trạng thái hoạt động server

4.8 Cấu hình lượng tử hóa cho mô hình 27B

Khi nâng cấp từ mô hình 4B lên 27B, cần điều chỉnh logic lượng tử hóa trong `gemma_translator.py`:

```

1 MODEL_NAME = "google/translategemma-27b-it"
2
3 # Quantization logic for 27B model
4 if vram_gb < 24:
5     # Không đủ VRAM cho 27B, cần sử dụng 4B thay thế
6     print("[Gemma] VRAM < 24GB -> Cannot run 27B model")
7     raise RuntimeError("Insufficient VRAM for 27B model")
8 elif vram_gb < 48:
9     # 24-47 GB: Bắt buộc dùng 4-bit quantization
  
```

```

10     print(f"[Gemma] VRAM < 48GB -> Using 4-bit NF4")
11     self._use_quantization = "4bit"
12 elif vram_gb < 64:
13     # 48-63 GB: Co the dung 8-bit
14     print(f"[Gemma] VRAM < 64GB -> Using 8-bit")
15     self._use_quantization = "8bit"
16 else:
17     # >= 64 GB: Chay FP16/BF16 day du
18     print(f"[Gemma] VRAM >= 64GB -> BFfloat16")
19     self._use_quantization = None

```

Listing 2: Logic lượng tử hóa điều chỉnh cho translategemma-27b-it

4.9 Cấu hình Offline Mode

```

1 # Offline mode - khong can Internet
2 HF_HUB_OFFLINE=true
3 TRANSFORMERS_OFFLINE=true
4
5 # Token Hugging Face (can khi tai model lan dau)
6 HF_TOKEN=hf_xxxxxxxxxxxxxxxxxxxxxx
7
8 # CUDA configuration
9 CUDA_VISIBLE_DEVICES=0

```

Listing 3: NỘI DUNG FILE .ENV CHO CHẾ ĐỘ OFFLINE

5 Phần cứng máy khách (Client)

Máy khách (Client) chỉ cần chạy trình duyệt web để truy cập giao diện ứng dụng. Toàn bộ tính toán AI được thực hiện trên máy chủ, nên yêu cầu phần cứng rất nhẹ.

5.1 Yêu cầu phần cứng Client

Bảng 16: Yêu cầu phần cứng máy khách

Thành phần	Tối thiểu	Khuyến nghị
CPU	Intel Core i3 / AMD Ryzen 3 hoặc tương đương	Intel Core i5 / AMD Ryzen 5 trở lên
RAM	4 GB	8 GB trở lên
Ổ cứng	1 GB trống (cho cache trình duyệt)	SSD bất kỳ
GPU	Không yêu cầu (integrated graphics đủ dùng)	Không yêu cầu
Màn hình	Dộ phân giải $\geq 1280 \times 720$	$\geq 1920 \times 1080$ (Full HD)
Mạng	Kết nối đến máy chủ (LAN/WiFi)	LAN Gigabit hoặc WiFi 5/6

Client rất nhẹ

Vì toàn bộ xử lý AI diễn ra trên Server, Client chỉ cần:

- Hiển thị giao diện web (HTML/CSS/JavaScript)
- Gửi file CSV / văn bản / ảnh đến Server qua HTTP
- Nhận và hiển thị kết quả dịch
- Polling tiến trình (mỗi 1 giây cho dịch CSV)

Bất kỳ máy tính, laptop, hoặc tablet nào có trình duyệt web hiện đại đều có thể làm Client.

5.2 Thiết bị Client được hỗ trợ

Bảng 17: Các loại thiết bị Client được hỗ trợ

Loại thiết bị	Mô tả	Hỗ trợ
PC Desktop	Windows / macOS / Linux	Đầy đủ
Laptop	Bất kỳ với trình duyệt web hiện đại	Đầy đủ
Tablet	iPad, Android tablet	Đầy đủ
Điện thoại	iPhone, Android	Cơ bản (responsive)
Thin Client	Chrome OS, Raspberry Pi	Đầy đủ

6 Phần mềm máy khách (Client)

6.1 Trình duyệt web yêu cầu

Bảng 18: Trình duyệt web được hỗ trợ

Trình duyệt	Phiên bản tối thiểu	Ghi chú
Google Chrome	≥ 90	Khuyến nghị — Hiệu năng tốt nhất
Mozilla Firefox	≥ 90	Hỗ trợ đầy đủ
Microsoft Edge	≥ 90	Chromium-based, tương tự Chrome
Safari	≥ 15	macOS / iOS
Opera	≥ 80	Chromium-based

Yêu cầu trình duyệt

Trình duyệt cần hỗ trợ:

- **ES2020+** JavaScript (async/await, optional chaining)
- **Fetch API** cho HTTP requests
- **File API** cho upload file CSV và ảnh
- **FileReader API** cho đọc và chuyển đổi base64 ảnh
- **Drag and Drop API** cho kéo thả file CSV
- **CSS Grid/Flexbox** cho responsive layout

6.2 Kiến trúc Frontend

Bảng 19: Công nghệ Frontend

Công nghệ	Phiên bản	Mục đích
Runtime Dependencies		
React	19.2.0	Thư viện UI, component-based architecture
React DOM	19.2.0	Render React components vào DOM
Development Dependencies		
Vite	7.2.4	Build tool và dev server (HMR)
@vitejs/plugin-react	5.1.1	Plugin React cho Vite (JSX transform)
ESLint	9.39.1	Linting và code quality
eslint-plugin-react-hooks	7.0.1	Kiểm tra React Hooks rules
eslint-plugin-react-refresh	0.4.24	Hỗ trợ React Fast Refresh

6.3 Cấu trúc thư mục Frontend

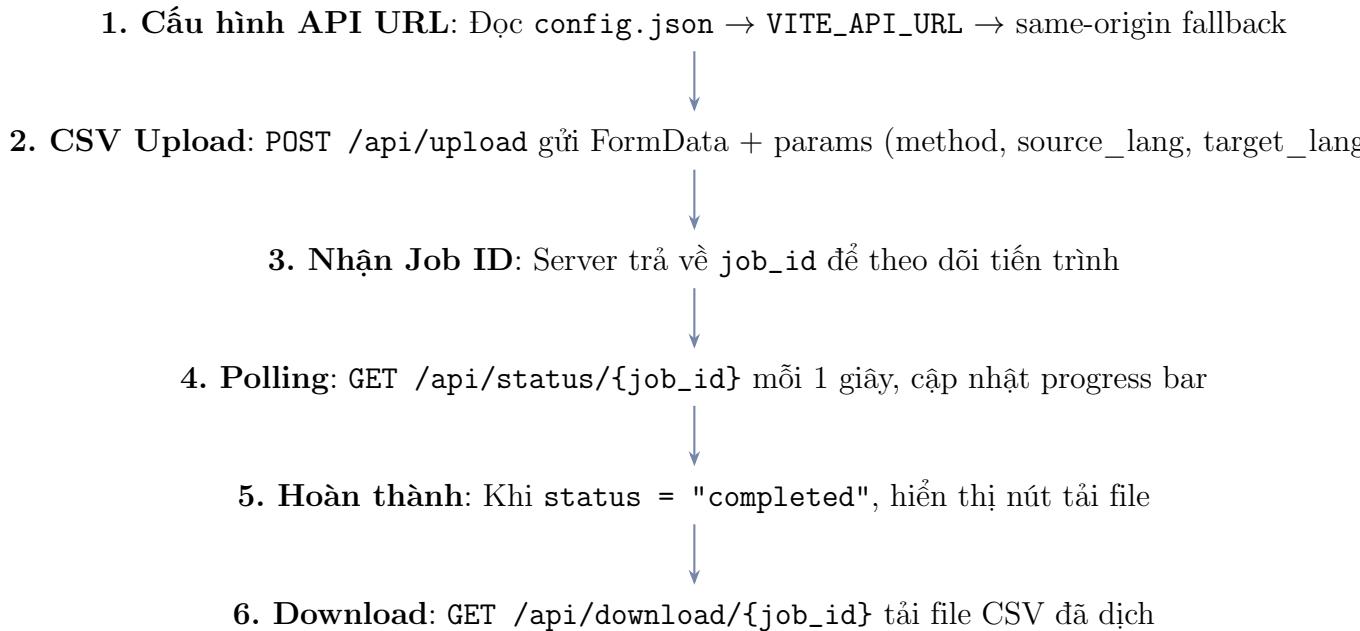
```

1 frontend/
2     public/
3         config.json          # Runtime API URL config
4         vite.svg
5     src/
6         App.jsx              # Component chính (3 tab)
7     )
8         App.css              # Styles
9         main.jsx              # Entry point
10        index.css             # Global styles
11        assets/
12            react.svg
13        index.html             # HTML template
14        package.json           # Dependencies
15        vite.config.js        # Vite configuration
16        eslint.config.js      # ESLint configuration

```

Listing 4: Cấu trúc thư mục Frontend

6.4 Cơ chế giao tiếp Client–Server



Hình 3: Luồng giao tiếp Client–Server khi dịch CSV

6.5 Cấu hình Runtime (config.json)

```

1 {
2   "apiUrl": "http://192.168.1.100:8000/api"
3 }
  
```

Listing 5: File config.json cho cấu hình API URL runtime

Thứ tự ưu tiên phân giải API URL:

- Runtime config:** Đọc từ ./config.json (có thể sửa sau khi build)
- Build-time env:** Biến môi trường VITE_API_URL lúc build
- Same-host fallback:** {window.location.origin}/api

7 Triển khai và vận hành

7.1 Quy trình triển khai Server

Bước 1: Cài đặt NVIDIA Driver + CUDA Toolkit 12.4

```

1 # Ubuntu
2 sudo apt update
3 sudo apt install nvidia-driver-550
4 # Reboot, then install CUDA 12.4

```

Bước 2: Cài đặt Python 3.12 và tạo virtual environment

```

1 python -m venv .my-env
2 source .my-env/bin/activate # Linux
3 .my-env\Scripts\activate # Windows

```

Bước 3: Cài đặt PyTorch với CUDA

```

1 pip install torch==2.6.0+cu124 \
2     torchvision==0.21.0+cu124 \
3     torchaudio==2.6.0+cu124 \
4     --index-url https://download.pytorch.org/whl/
      cu124

```

Bước 4: Cài đặt dependencies

```

1 pip install -r requirements.txt
2 pip install bitsandbytes # Linux only, for
  quantization

```

Bước 5: Tải mô hình TranslateGemma-27B-IT

```

1 # Set Hugging Face token (required for gated model)
2 export HF_TOKEN=hfxxxxxxxxxxxxxx
3 python download_model.py

```

Bước 6: Khởi chạy Server

```

1 cd backend
2 python main.py
3 # Server starts at http://0.0.0.0:8000

```

7.2 Quy trình triển khai Client

Bước 1: Build Frontend (trên máy phát triển)

```

1 cd frontend
2 npm install # hoc : bun install
3 npm run build # To th m c dist/

```

Bước 2: Cấu hình API URL

Sửa file `dist/config.json` trỏ đến địa chỉ Server:

```
1 { "apiUrl": "http://<server-ip>:8000/api" }
```

Bước 3: Phân phối cho Client

Chỉ cần gửi đường dẫn URL hoặc deploy `dist/` lên web server (nginx, Apache). Client mở trình duyệt và truy cập.

7.3 Triển khai Offline

Dự án hỗ trợ đóng gói toàn bộ để triển khai trên máy không có Internet:

Bảng 20: Thành phần trong gói offline

Thư mục	Nội dung	Kích thước
<code>backend/</code>	Mã nguồn Python	~50 KB
<code>frontend_dist/</code>	Frontend đã build (HTML/CSS/JS)	~500 KB
<code>packages/</code>	Python wheels (.whl)	~2–3 GB
<code>model_cache/</code>	Trọng số mô hình TranslateGemma-27B	~54 GB
<code>setup.bat/.sh</code>	Script cài đặt tự động	~5 KB
Tổng cộng		~56–58 GB

Gói offline cho mô hình 27B rất lớn

Với mô hình 27B (FP16), trọng số nặng khoảng 54 GB. Khi đóng gói offline:

- Tổng dung lượng gói: ~56–58 GB
- Cần USB/ổ cứng di động ≥ 64 GB hoặc truyền qua mạng LAN
- Thời gian cài đặt offline: 15–30 phút (phụ thuộc tốc độ ổ cứng)
- So sánh: Gói offline mô hình 4B hiện tại chỉ ~10 GB

8 Hiệu năng và tối ưu hóa

8.1 Ước tính tốc độ suy luận (Inference Speed)

Bảng 21: Ước tính tốc độ suy luận TranslateGemma-27B-IT

GPU	Quantization	Tokens/giây	Ghi chú
RTX 4090 (24GB)	NF4 4-bit	~20–35	Tốc độ chấp nhận được
RTX A6000 (48GB)	INT8 8-bit	~25–40	Chất lượng tốt hơn NF4
A100 (80GB)	BF16	~40–70	Hiệu năng production
H100 (80GB)	BF16	~60–100	Hiệu năng tối ưu nhất

8.2 Ước tính thời gian dịch CSV

Giả sử trung bình mỗi ô văn bản dài ~50 tokens đầu ra:

Bảng 22: Thời gian ước tính dịch CSV (TranslateGemma-27B)

Số dòng CSV	RTX 4090 (4-bit)	A6000 (8-bit)	A100 (BF16)	H100 (BF16)
100 dòng	~4–6 phút	~3–5 phút	~2–3 phút	~1–2 phút
500 dòng	~18–25 phút	~15–20 phút	~8–12 phút	~5–8 phút
1,000 dòng	~35–50 phút	~25–35 phút	~15–22 phút	~10–15 phút
5,000 dòng	~3–4 giờ	~2–3 giờ	~1–2 giờ	~45–75 phút

8.3 Các phương pháp tối ưu hiệu năng

- Quantization-Aware Training (QAT):** Sử dụng phiên bản QAT INT4 do Google cung cấp, giữ chất lượng tốt hơn Post-Training Quantization (PTQ).
- FlashAttention-2:** Tích hợp trong PyTorch ≥ 2.6 để giảm bộ nhớ attention và tăng tốc.
- torch.compile():** Biên dịch mô hình với TorchInductor backend để tối ưu hóa graph.
- Batching thông minh:** Gom nhiều câu có độ dài tương tự để dịch cùng lúc.
- Model Sharding:** Chia mô hình qua nhiều GPU với `device_map="auto"` khi dùng Accelerate.
- Caching KV:** Tái sử dụng Key-Value cache giữa các request liên tiếp cùng context.

9 Bảo mật hệ thống

9.1 Bảo mật Server

- **CORS Policy:** Hiện đang cho phép tất cả origins (`allow_origins=["*"]`). Trong production, nên giới hạn chỉ các domain cụ thể.
- **HF_TOKEN:** Token Hugging Face cần được bảo mật trong file `.env`, không commit vào version control.
- **File Upload:** Kiểm tra phần mở rộng file (`.csv`), giới hạn kích thước upload.
- **HTTPS:** Khuyến nghị sử dụng reverse proxy (`nginx`) với SSL/TLS certificate.
- **Firewall:** Chỉ mở port 8000 (hoặc port tùy chỉnh) cho mạng nội bộ.
- **Rate Limiting:** Nên thêm rate limiter để tránh quá tải server.

9.2 Bảo mật Client

- Frontend là ứng dụng static (HTML/CSS/JS), không chứa logic nhạy cảm.
- Dữ liệu dịch được gửi qua HTTP request — cần HTTPS trong môi trường production.
- `config.json` chứa API URL, có thể bị sửa đổi — cần validate ở server-side.

10 Kết luận

10.1 Tổng kết

Dự án CSV Translator Pro sử dụng mô hình **google/translategemma-27b-it** là một hệ thống dịch thuật AI mạnh mẽ với kiến trúc Client–Server rõ ràng:

- **Server:** Đòi hỏi phần cứng chuyên dụng, đặc biệt GPU với VRAM lớn (≥ 24 GB). Phần mềm bao gồm FastAPI, PyTorch, Hugging Face Transformers, và hệ sinh thái CUDA. Mô hình 27B cho chất lượng dịch thuật vượt trội so với phiên bản 4B, nhưng đánh đổi bằng yêu cầu tài nguyên cao hơn đáng kể.
- **Client:** Nhẹ nhàng, chỉ cần trình duyệt web hiện đại. Frontend được xây dựng với React 19 và Vite 7, giao tiếp với server qua REST API. Hỗ trợ 3 chế độ dịch: CSV hàng loạt, văn bản trực tiếp, và OCR từ ảnh.

10.2 So sánh khi nâng cấp từ 4B lên 27B

Bảng 23: So sánh tổng quan 4B vs 27B

Tiêu chí	TranslateGemma-4B	TranslateGemma-27B
Chất lượng (MetricX)	5.32	3.09 (tốt hơn 42%)
VRAM tối thiểu	~ 3 GB (4-bit)	~ 18 GB (4-bit)
VRAM khuyến nghị	8–16 GB	24–48 GB
GPU tối thiểu	RTX 3060 Ti (8GB)	RTX 3090/4090 (24GB)
Tốc độ	Nhanh	Chậm hơn $\sim 3\text{--}5\times$
Gói offline	~ 10 GB	$\sim 56\text{--}58$ GB
Chi phí Server	\$1,500–3,000	\$3,000–30,000
Client thay đổi	Không thay đổi — chỉ cần trình duyệt web	

10.3 Khuyến nghị

1. **Nếu ưu tiên chất lượng dịch:** Nâng cấp lên 27B với GPU A6000 (48 GB) + INT8 quantization.
2. **Nếu ưu tiên tốc độ + chi phí:** Giữ nguyên 4B hoặc xem xét 12B (cân bằng).
3. **Production deployment:** A100/H100 80 GB với BF16 để có cả chất lượng và tốc độ.
4. **Client:** Không cần thay đổi gì — giao diện web hoạt động giống hệt cho mọi phiên bản mô hình.

A Phụ lục A: Danh sách 55 ngôn ngữ hỗ trợ

TranslateGemma-27B-IT hỗ trợ dịch thuật qua 55 ngôn ngữ. Trong cấu hình hiện tại của dự án, các ngôn ngữ sau đã được kích hoạt:

Bảng 24: Ngôn ngữ được cấu hình trong hệ thống

Mã	Ngôn ngữ	Mã	Ngôn ngữ
ar	Tiếng Ả Rập	ko	Tiếng Hàn
vi	Tiếng Việt	ru	Tiếng Nga
en	Tiếng Anh	pt	Tiếng Bồ Đào Nha
de-DE	Tiếng Đức	it	Tiếng Ý
cs	Tiếng Séc	nl	Tiếng Hà Lan
fr	Tiếng Pháp	pl	Tiếng Ba Lan
es	Tiếng Tây Ban Nha	tr	Tiếng Thổ Nhĩ Kỳ
zh	Tiếng Trung	th	Tiếng Thái
ja	Tiếng Nhật		

B Phụ lục B: Lệnh kiểm tra phần cứng

```

1 import torch
2 print(f"PyTorch version: {torch.__version__}")
3 print(f"CUDA available: {torch.cuda.is_available()}")
4 if torch.cuda.is_available():
5     print(f"CUDA version: {torch.version.cuda}")
6     print(f"GPU: {torch.cuda.get_device_name(0)}")
7     vram = torch.cuda.get_device_properties(0).total_memory
8     print(f"VRAM: {vram / 1024**3:.1f} GB")
9     print(f"GPU count: {torch.cuda.device_count()}")
```

Listing 6: Script kiểm tra phần cứng GPU

C Phụ lục C: Tài liệu tham khảo

1. Google Translate Research Team, “TranslateGemma Technical Report”, arXiv:2601.09012, 2026.
2. Google DeepMind, “Gemma 3 Technical Report”, arXiv:2503.19786, 2025.
3. Hugging Face Model Card: <https://huggingface.co/google/translategemma-27b-it>
4. FastAPI Documentation: <https://fastapi.tiangolo.com/>
5. PyTorch Documentation: <https://pytorch.org/docs/stable/>
6. React 19 Documentation: <https://react.dev/>

7. Vite Documentation: <https://vitejs.dev/>
8. BitsAndBytes Quantization: <https://github.com/TimDettmers/bitsandbytes>
9. NVIDIA CUDA Toolkit: <https://developer.nvidia.com/cuda-toolkit>