

Bài toán ứng dụng AI trong tạo banner ads

design.aiservice.vn là 1 công cụ thiết kế giúp người dùng tạo ra các thiết kế quảng cáo tự động dựa vào các thông tin và ảnh sản phẩm được nhập đồng thời đó là phát triển các banner tương đồng từ một banner đầu vào. Điều này có thể giúp người dùng tạo ra nhiều ý tưởng mới từ một banner gốc hoặc tạo lại một banner rõ nét hơn từ một banner bị mờ.

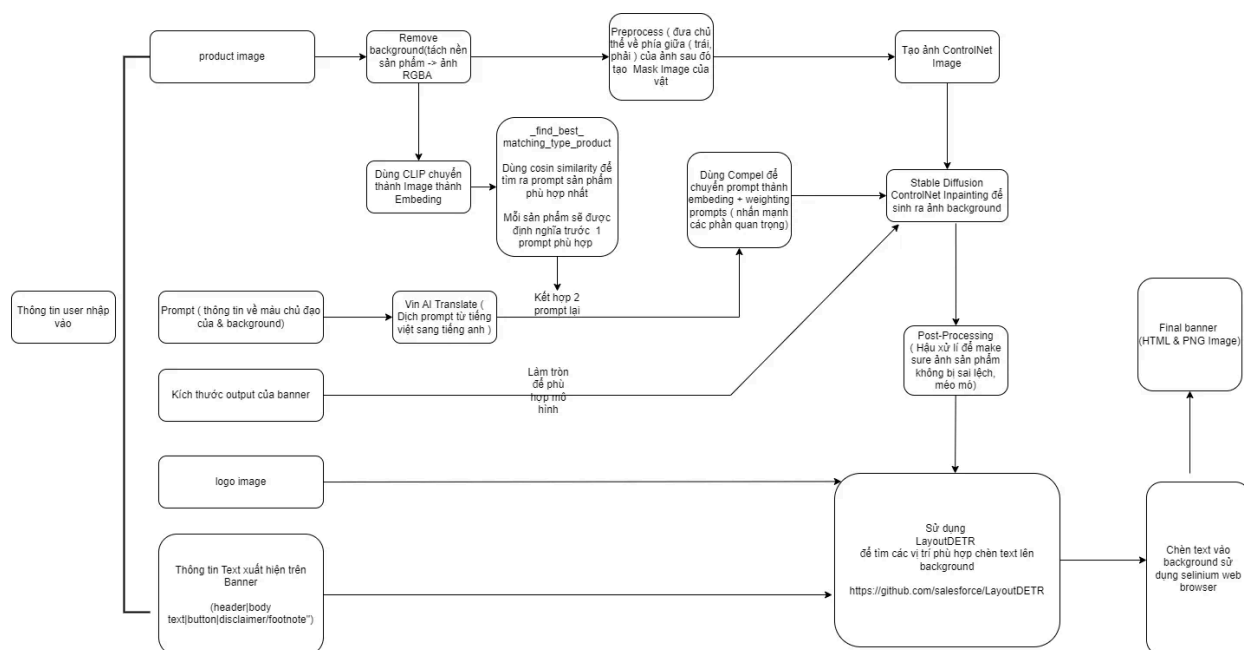
design.aiservice.vn có các chức năng như là tạo banner/quote mới từ banner/quote của user hoặc user có thể nhập prompt để sinh ra các banner/quote theo ý muốn của mình. Trang web cũng nhận figma(các node được convert cho phù hợp với fabric.js) làm ảnh đầu vào cộng với prompt của user để sinh ra ảnh theo mong muốn.

Cụ thể, trang web được xây dựng dựa trên 4 chức năng chính sau:

1. Gen banner ads bằng hệ thống AI đóng kín(local)
2. Gen banner ads bằng API AI
3. Gen similar banner image
4. Gen dynamic banner

Sau đây là chi tiết về 4 chức năng trên

1. Gen banner ads bằng hệ thống AI đóng kín(local)



API của chức năng : <http://172.18.5.44:8000/mlbigdata/aidn/banner/generate-banner-sd>

Tham số của api:

- **ori_img (UploadFile, required):** Hình ảnh của vật dưới dạng file, không phải là chuỗi base64.
- **prompt_detail_color (string, required):** Mô tả màu sắc chủ đạo của banner, ví dụ "Đỏ".
- **product_info (string, required):** Thông tin chi tiết về sản phẩm cần hiển thị trên banner.
- **Thông tin về sản phẩm theo định dạng:** header|body text|button|disclaimer/footnote. Phần nào không muốn xuất hiện thì để trống.
- **width (int, required):** Chiều rộng mong muốn của banner (tính bằng pixel).
- **height(int, required):** Chiều cao mong muốn của banner (tính bằng pixel).

Mục tiêu: Sinh ra banner (ảnh nền) bằng AI dựa trên ảnh sản phẩm và thông tin người dùng.

Luồng hoạt động:

1. **Xử lý ảnh sản phẩm:**
 - Remove background → ảnh RGBA.
 - Preprocess (đưa đối tượng về giữa ảnh) → tạo mask image.
 - Tạo ControlNet Image từ mask.
2. **Sinh background:**
 - Dùng ControlNet (Stable Diffusion) để inpainting background theo mask.
 - Post-processing để tránh méo mó ảnh sản phẩm.
3. **Xử lý Prompt:**
 - Prompt được nhập từ user → dịch sang tiếng Anh bằng Vin AI.
 - Embed bằng CLIP → tìm prompt phù hợp từ tập mẫu (_find_best_matching_type_product).
 - Combine prompt + weighting bằng Compel → sinh prompt tối ưu.
4. **Sinh layout text:**
 - Dùng LayoutDETR để xác định vị trí phù hợp để chèn text lên ảnh background.
5. **Chèn text và xuất banner:**
 - Chèn text (HTML, PNG) bằng Selenium web browser.
 - Kết quả là Final banner.

Input:

Người dùng sẽ nhập các thông tin liên quan như sau:

- **Kích thước Banner:** Kích thước mong muốn của Banner đầu ra.

- **Company name:** Tên của công ty hoặc thương hiệu.
- **Product name:** Tên của sản phẩm hoặc thương hiệu.
- **More Info:** Các thông tin thêm giới thiệu về sản phẩm.
- **Website:** Địa chỉ website của sản phẩm đang được quảng cáo.
- **Phone:** Số điện thoại.
- **Call to action:** Một câu ngắn nhấn mạnh người xem mua hàng (Mua ngay, Số lượng có hạn...).
- **Discount:** Thông tin về giảm giá của sản phẩm (Sale 50%...).
- **Logo:** Ảnh logo của sản phẩm (Đang cập nhật).
- **Ảnh sản phẩm:** Upload 1 ảnh sản phẩm rõ ràng để tool có thể nhận diện và tự động tách nền của sản phẩm.
- **Prompt:** Thông tin về background, sản phẩm...

Output:

Người dùng sẽ được trả về 2 kết quả:

1. Ảnh của banner (định dạng PNG).
2. File HTML của Banner (Hiện tại đang cập nhật).

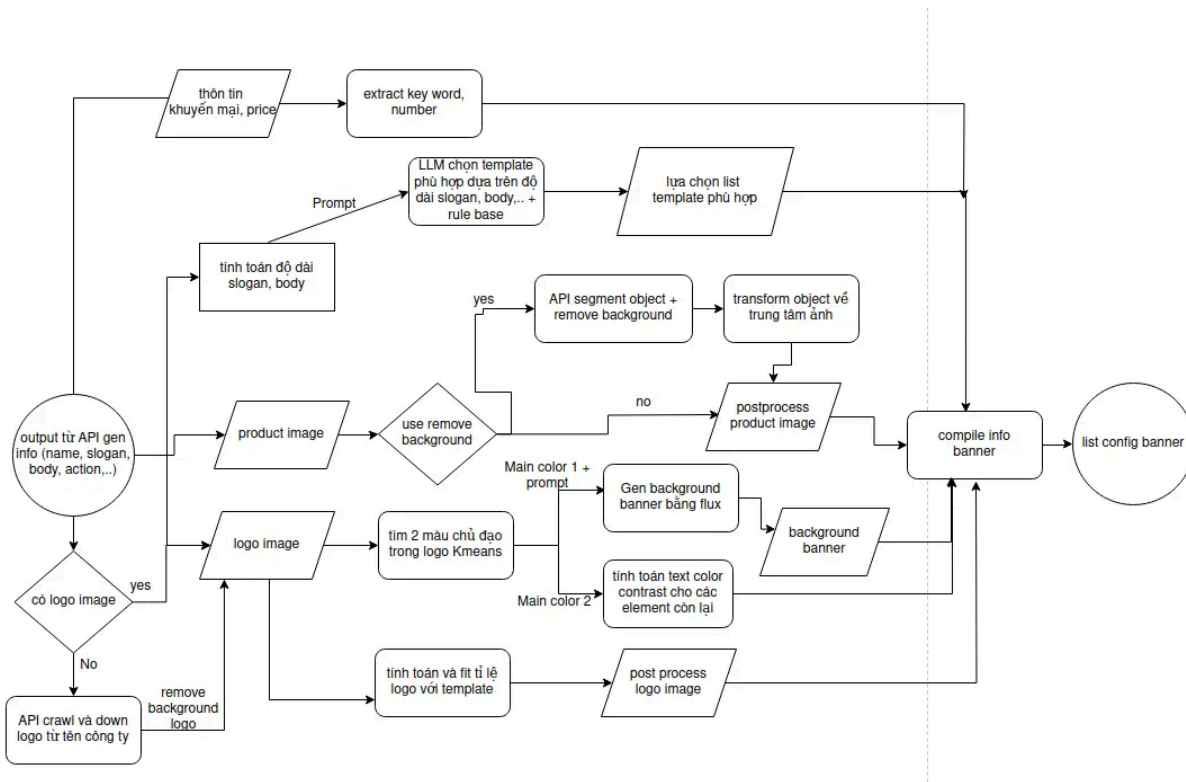
Yêu cầu của thiết kế:

- Kích thước:
 - (300, 600)
 - (512, 512)
 - (300, 250)
 - (180, 600)

Ưu điểm & nhược điểm:

- **Ưu điểm:**
 - Có khả năng sáng tạo mạnh.
 - Có thể điều chỉnh màu background theo ý muốn.
- **Nhược điểm:**
 - Tốn tài nguyên (GPU).
 - Cần hiểu biết về prompt (output đầu ra phụ thuộc vào prompt khá nhiều).
 - Cần thử nhiều lần để đạt kết quả tốt.
 - Khó kiểm soát được output.

2. Gen banner ads bằng API AI



API gen banner info:

http://172.18.5.44:8000/mlbigdata/vision/quote-gen/v1/api/gen_banner_info

API gen banner:

http://172.18.5.44:8000/mlbigdata/vision/quote-gen/v1/api/gen_banner

Mục tiêu: Tạo banner theo template cố định, không dùng AI sinh background.

Luồng hoạt động:

1. **Input từ API:**
 - o Bao gồm: name, slogan, logo,....
2. **Xử lý text:**

- Tách key info (slogan, price, etc).
Tính độ dài slogan/body.
- Chọn **template phù hợp** bằng rule hoặc LLM.
- 3. **Xử lý ảnh sản phẩm:**
 - Nếu cần: Segment + Remove background.
 - Đưa sản phẩm về giữa ảnh.
 - Post-process ảnh sản phẩm.
- 4. **Xử lý logo:**
 - Crawl nếu chưa có.
 - Remove background.
 - Tìm màu chủ đạo bằng K-means → chọn 2 màu chính.
- 5. **Tạo background:**
 - Dùng màu chính + prompt để **generate background** (bằng flux).
 - Tính contrast màu chữ (text color).
- 6. **Ghép lại thành banner:**
 - Compile info.
 - Xuất ra danh sách cấu hình banner.

Input: Các thông tin output từ API gen info + ảnh sản phẩm, logo

Output: List config banner (các config này được compile và hiển thị sang phía user và người dùng có thể sử dụng thêm các công cụ trên tool để edit và export sản phẩm thiết kế)

API gen info sẽ extract thông tin banner từ nội dung người dùng nhập. Nhiệm vụ của api này là extract ra từ nội dung mô tả thành các trường thông tin tương ứng, bao gồm:

```
{
  "banner_info": {
    "product_name": "", // tên sản phẩm đầy đủ
    "product": "", // loại hình sản phẩm
    "company_name": "", // tên brand
    "distributor": "", // nhà phân phối
    "title": "", // title
    "slogan": "", // slogan
    "body": "", // thông tin mô tả chi tiết
    "action": "", // call to action
  }
}
```

```

"sales": "", // thông tin khuyến mại

"oldprice": "", // giá

"newprice": "",

"contact": "", // thông tin liên lạc

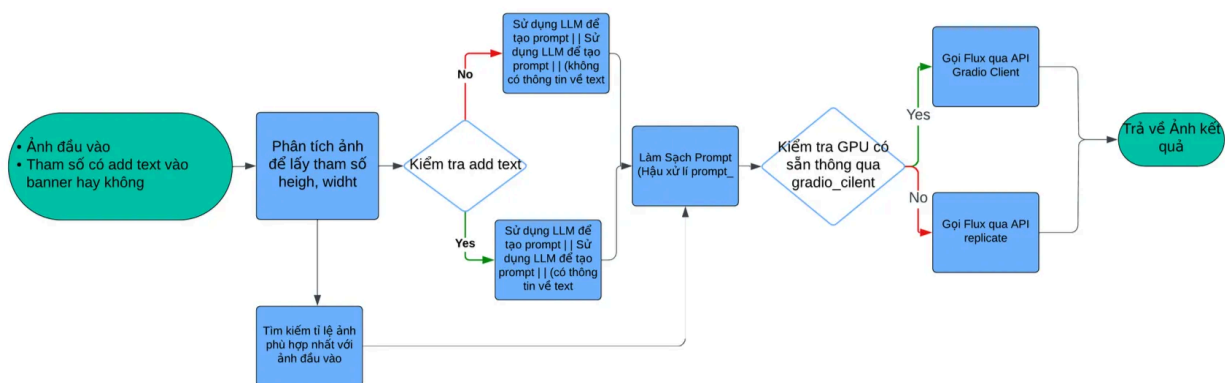
"size": "" // size banner

}

}

```

3. Gen similar banner image



API của chức năng: http://172.18.5.44:8000/mlbigdata/generative_ai/similar_banner/

Input:

- `image_base64` (bắt buộc): Chuỗi base64 của ảnh đầu vào.
- `include_text` (bắt buộc): Một giá trị boolean xác định liệu thông tin văn bản trong ảnh có được sử dụng hay không.
 - `True`: Sử dụng thông tin văn bản nếu có (chỉ áp dụng cho văn bản tiếng Anh).
 - `False`: Bỏ qua thông tin văn bản.

Output: Luồng dữ liệu (StreamingResponse) chứa ảnh banner mới được tạo dưới dạng base64

Mục tiêu: Quyết định nhanh có cần thêm text không và chọn cách gọi AI model phù hợp.

Luồng hoạt động:

1. **Phân tích ảnh đầu vào:**
 - Lấy thông tin về height, width.

2. Kiểm tra có add text không:

- Nếu **không có text** → prompt đơn giản.
- Nếu **có text** → prompt phức tạp hơn (LLM sinh prompt theo text).

3. Xử lý prompt:

- Làm sạch prompt sau khi LLM sinh ra.

4. Gọi model sinh background (Flux):

- Nếu có **GPU local** → dùng Gradio client.
- Nếu không có GPU → dùng dịch vụ replicate.

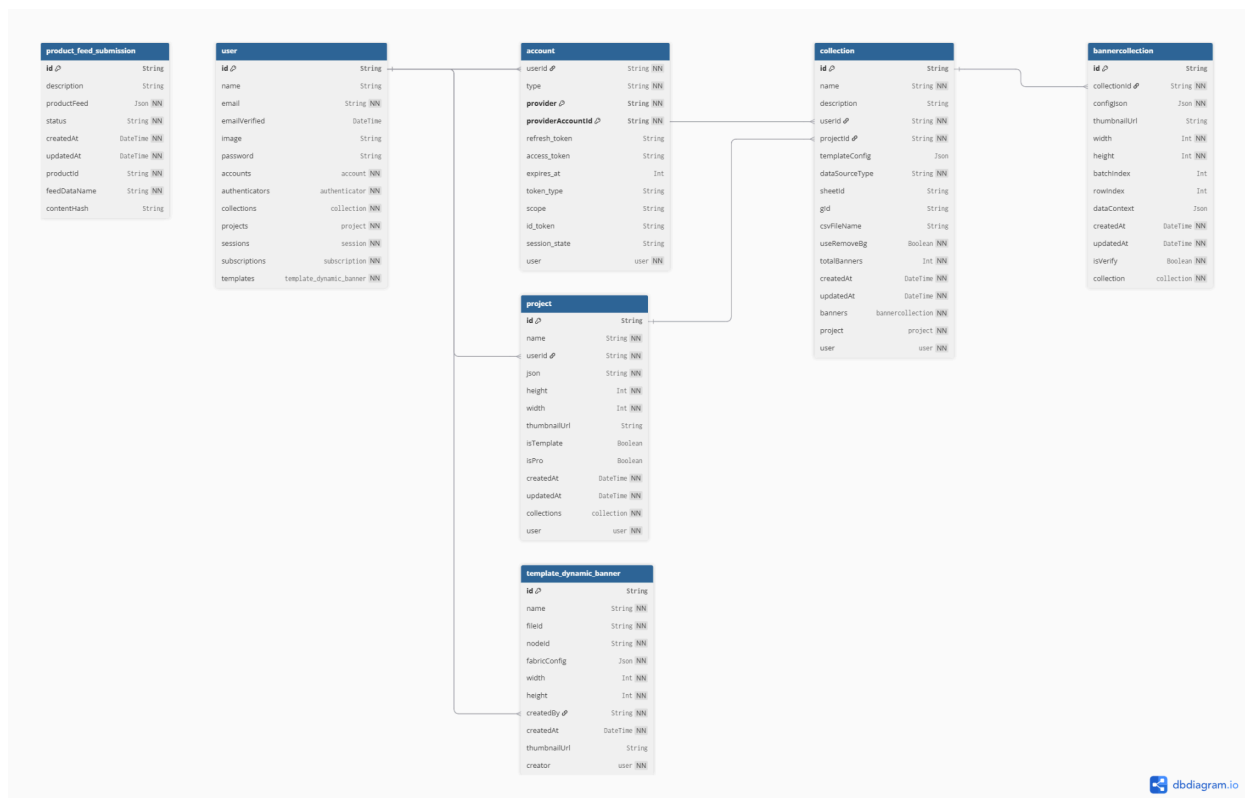
Flow và Kiến trúc:

- Quá trình bắt đầu từ việc nhận diện và trích xuất prompt từ banner đầu vào.
- Prompt sau đó được tiền xử lý để chuẩn bị cho quá trình tạo ảnh.
- Mô hình FLUX sẽ tạo ra banner đầu ra từ prompt đã xử lý.

Mô hình:

- Sử dụng LLM (OPENAI API) để trích xuất và xử lý prompt.
- Sử dụng mô hình FLUX để tạo ra các banner đầu ra tương ứng.

4. Gen dynamic banner



Input:

API yêu cầu tạo dynamic banner:

http://172.18.5.44:8000/mlbigdata/vision/quote-gen/v1/api/submit_product_feed

- Đầu vào: gửi yêu cầu tạo dynamic banner trên tool của adopt

```
{
  "feed_data_name": "
  "description":
  "products": [
    {
      "Id":
      "title":
      "description":
      "link":
      "price":
      "sale_price":
      "availability":
      "mobile_link":
      "image_link":
    }
  ]
}
```

- Đầu ra: API chạy xong sẽ trả về thống kê số lượng sản phẩm vừa được thêm vào feed_data.

```
{
  "status": ,
  "feed_data_name":
  "total_products":
  "statistics": {
    "new_products":
    "updated_products":
    "unchanged_products":
  },
  "details": {
    "new_product_ids": [],
    "updated_product_ids": [],
    "unchanged_product_ids": [
    ],
    "unchanged_with_status": [
      {
        "product_id":
```



```

        "status":
    },
    {
        "product_id":
        "status":
    }
]
},
"message":
"created_at":
}

```

Output:

API lấy dynamic banner:

http://172.18.5.44:8000/mlbigdata/vision/quote-gen/v1/api/gen_integrated_banner

Đầu vào: collection_name là data_feed_name + 'collection'

Đầu ra:

```

{
    "status":
    "collection_id":
    "collection_name":
    "banners": [
        {
            "banner_id":
            "thumbnail_url":
            "is_verify":
            "status":
        }
    ],
    "total": 2
}

```

Flow updated dữ liệu hiện tại:

Main process flow:

- Tạo 1 process trong process_updated_banners.py định kỳ 30 giây xử lý 1 lần với nhiệm vụ kiểm tra dữ liệu có status trong bảng product_feed_sub là "updated" sẽ được group by feed_data_name, max 20 product/batch để xử lý.

AI Batch processing:

- Tạo message tele noti thông báo cần update dữ liệu cho collection ..., từ feed_data_name lấy được collection_name, xong lấy được fabricConfig theo template_id. Dữ liệu update sẽ được gen lại với base config vừa lấy được.
- Hàm render_config_to_image thực hiện gọi tới server deploy tool_editor để thực hiện việc chuyển đổi config fabric ra image url.

Cập nhật kết quả:

- Url kết quả trong bảng bannercollection được update lại, status trong bảng product_feed_sub cũng được update lại từ “updated” thành “done”