QUY TẮC CUỘC THI VISUALIZATION

1. MỤC TIÊU CHÍNH

Mục tiêu của cuộc thi là thử thách các đội tham gia xây dựng một chương trình có khả năng tự động hóa quá trình sinh giao diện người dùng (UI). Chương trình này phải có khả năng phân tích thông tin từ các "task problem" (bài toán cụ thể) do Ban Tổ Chức (BTC) cung cấp để tạo ra một giao diện phù hợp và có tính tương tác cao. Giao diện phải cho phép người dùng tương tác với các mô hình (models) được chỉ định trong "task problem" (ví dụ: tải lên dữ liệu, nhận và hiển thị kết quả từ model). Bên cạnh đó, giao diện cũng cần hiển thị kết quả đầu ra của mô hình đối với các dữ liệu test được cung cấp sẵn, đóng vai trò như các ví dụ minh họa (example cases). Quá trình này bao gồm việc phân tích yêu cầu bài toán từ file mô tả, phân tích cách sử dụng mô hình, xử lý dữ liệu đầu vào/đầu ra của mô hình, và quan trọng nhất là **một cách tự động** có được giao diện có khả năng tương tác và hiển thị dữ liệu.

2. DỮ LIỆU ĐẦU VÀO (INPUT) CUNG CẤP CHO MỖI ĐỘI

Với mỗi "task problem" (ví dụ: image classification,v.v), mỗi đội thi sẽ nhận được một thư mục chứa:

File `task.yaml`: Đây là file mô tả chính cho mỗi bài toán, bao gồm ba phần chính:

- **Mô tả bài toán (Task Description):**
  - `type`: Loại bài toán (ví dụ: `image_classification`, `image_to_text`, v.v).
  - `description`: Mô tả cơ bản về bài toán hiện tại, mục đích của nó là gì.
  - `input`: Mô tả định dạng dữ liệu đầu vào cho bài toán tổng thể mà người dùng sẽ tương tác qua giao diện.
  - `output:`: Mô tả định dạng kết quả đầu ra mong muốn hiển thị trên giao diện.
  - `visualize:` Mô tả và yêu cầu về giao diện
    - `visualize:` Mô tả tổng quan giao diện
    - `features:` Mô tả tính năng giao diện nên có
      - `list_display:` Mô tả chức năng hiển thị giao diện
      - `input_function:` Mô tả chức năng tương tác giao diện
- **Thông tin mô hình (Model Information):**
  - `api_url`: Địa chỉ API của mô hình.
  - `name`: Tên của mô hình được sử dụng.
  - `input_format`: Đặc tả định dạng dữ liệu đầu vào của mô hình.
  - `output_format`: Đặc tả định dạng dữ liệu đầu ra từ mô hình.
  - Một số trường khác (có thể có):

        - `sample_code`: Ví dụ cách gọi và sử dụng mô hình. Hoặc 1 số lưu ý khác về output của mô hình
- **Thông tin bộ dữ liệu (Dataset Information):**
    - `data_path`: Đường dẫn đến dữ liệu mẫu (thường là `./data` trong thư mục của task problem).
    - Mô tả các dữ liệu phụ trợ khác (nếu có).

Thư mục `data`: Chứa dữ liệu mẫu hoặc các tài nguyên cần thiết khác cho "task problem".

Một số file bổ sung khác (đối với một số problem đặc biệt).

Các đội có thể tham khảo ví dụ tại: LINK

3. KẾT QUẢ ĐẦU RA (OUTPUT) MONG MUỐN TỪ MỖI ĐỘI

Sản phẩm cuối cùng mà mỗi đội cần nộp là một chương trình hoàn chỉnh có khả năng:

- **Tự động phân tích `task.yaml`**: Đọc và hiểu các yêu cầu về bài toán, định dạng và thông tin mô hình từ file `task.yaml` của một "task problem" bất kỳ.
- **Tự động sinh giao diện (UI Generation)**: Dựa trên thông tin phân tích được, chương trình tự động tạo ra một giao diện người dùng phù hợp với `task_type` và các yêu cầu tương tác.
    - **Khả năng tương tác**: Giao diện phải cho phép người dùng thực hiện các hành động cần thiết (ví dụ: tải lên hình ảnh cho image classification, nhập văn bản, v.v.).
    - **Tích hợp mô hình**: Giao diện phải có khả năng gửi dữ liệu đầu vào đến mô hình (thông qua API được cung cấp hoặc sử dụng model trực tiếp nếu được chỉ định) và nhận kết quả trả về.
    - **Hiển thị kết quả**: Hiển thị kết quả từ mô hình một cách rõ ràng và thân thiện với người dùng trên giao diện (ví dụ: hiển thị nhãn và ảnh cho image classification, hiển thị văn bản được tạo ra cho image to text).
    - **Hiển thị ví dụ minh hoạ**: Hiển thị kết quả đầu ra của mô hình đối với các dữ liệu test được cung cấp sẵn, đóng vai trò như các ví dụ minh họa (example cases).
- **Xử lý dữ liệu**: Thực hiện các bước tiền xử lý dữ liệu đầu vào từ người dùng (nếu cần) để phù hợp với `input_format` của mô hình và hậu xử lý kết quả từ mô hình (nếu cần) để hiển thị theo `output` phù hợp trên giao diện.
- **Chi phí**: Chương trình cần được tối ưu để hoạt động trong giới hạn chi phí cho phép (tương tự như quy định về COST trong cuộc thi Coding).

4. PUBLIC TEST CASES (BỘ DỮ LIỆU THỬ NGHIỆM CÔNG KHAI)

- BTC sẽ cung cấp một số “task problem” mẫu dưới dạng các thư mục hoàn chỉnh (bao gồm `task.yaml` và thư mục `data`) làm public test.
- Mỗi public test sẽ là một ví dụ cụ thể về “task problem”.
- BTC cũng sẽ cung cấp mô tả về các chức năng và tính năng giao diện kỳ vọng cho mỗi public test này.
- **Mục đích**: Giúp các đội hiểu rõ hơn về yêu cầu của bài toán, cấu trúc `task.yaml`, và kiểm tra sơ bộ khả năng sinh giao diện tự động cũng như tính tương tác của chương trình.
- **Lưu ý**: Kết quả trên public tests không được sử dụng để tính điểm xếp hạng cuối cùng. Đây chỉ là công cụ hỗ trợ cho quá trình phát triển.

5. CÁCH THỨC ĐÁNH GIÁ VÀ XẾP HẠNG

Kết quả của các đội sẽ được đánh giá dựa trên hai tiêu chí chính: COST (Chi phí sử dụng tài nguyên) và RANK (Thứ hạng dựa trên chất lượng của chương trình sinh giao diện và giao diện được tạo ra).

5.1. COST (CHI PHÍ SỬ DỤNG TÀI NGUYÊN) (Áp dụng nếu có sử dụng API tốn phí)

- **Tài Nguyên Cung Cấp**: Mỗi đội có thể được cấp một ngân sách giới hạn (ví dụ: 20 USD) để sử dụng cho các API của mô hình (nếu các mô hình được cung cấp là dịch vụ có tính phí).
- **Tối Ưu Chi Phí**: Khuyến khích các đội phát triển giải pháp không chỉ mạnh mẽ mà còn hiệu quả về mặt chi phí. (Nếu tất cả models là local/miễn phí, mục này có thể được bỏ qua hoặc thay bằng tiêu chí hiệu suất tài nguyên của chương trình sinh UI).

5.2. RANK (THỨ HẠNG DỰA TRÊN CHẤT LƯỢNG)

- **Benchmark Task Problems (Các Bài Toán Đánh Giá Chuẩn)**:
    - BTC sẽ sử dụng các “task problem” riêng biệt, bí mật (private tests) để đánh giá chương trình của các đội. Các “task problem” này sẽ không được công bố trước.
    - Các benchmark này sẽ bao gồm nhiều loại bài toán khác nhau (như image classification, video classification, image to text, v.v.) để đánh giá khả năng thích ứng của chương trình sinh giao diện.
- **Tiêu Chí Đánh Giá cho mỗi Private Test**: Với mỗi “task problem” trong bộ private test, Ban Giám Khảo (BGK) sẽ đánh giá chương trình và giao diện được sinh ra dựa trên các tiêu chí sau:
    - **Tính Tự Động và Chính Xác của Chương Trình Sinh Giao Diện**:
        - Khả năng phân tích chính xác `task.yaml`.
        - Khả năng sinh ra giao diện phù hợp với `task_type` và các yêu cầu được mô tả.
    - **Chức Năng và Tính Tương Tác của Giao Diện Được Sinh Ra**:
        - Giao diện có đầy đủ các thành phần để người dùng nhập liệu theo `input` không?
        - Giao diện có gọi đúng tới mô hình và xử lý `input_format`, `output_format` không?

        - Kết quả có được hiển thị trực quan, dễ hiểu theo `output_format` không?
        - Các yếu tố tương tác (nút bấm, thanh trượt, vùng tải file, v.v.) có hoạt động đúng và mượt mà không?
    - **Trải Nghiệm Người Dùng (UX) và Tính Thẩm Mỹ của Giao Diện (Không bắt buộc - Điểm cộng)**:
        - Giao diện có dễ sử dụng, bố cục hợp lý không?
        - Tính thẩm mỹ chung của giao diện.
    - **Khả Năng Thích Ứng của Chương Trình**: Mức độ chương trình có thể sinh ra các giao diện tốt và khác biệt cho các loại `task_type` đa dạng. Có thể giải quyết được bao nhiêu loại "task_problem"
- **Tính Toán Rank**:
    - Chương trình của mỗi đội sẽ được chạy với từng private test. BGK sẽ cho điểm dựa trên các tiêu chí trên.
    - Với mỗi private test, các đội sẽ được xếp hạng dựa trên tổng điểm đánh giá. Đội có điểm số cao nhất sẽ nhận Rank 1, đội cao thứ hai nhận Rank 2, và cứ thế tiếp tục.
    - Trường hợp các đội có cùng điểm số trên một private test, có thể xét thêm những tiêu chí khác để đánh giá, hoặc không họ sẽ nhận cùng một hạng và hạng tiếp theo sẽ được bỏ qua (ví dụ: nếu 2 đội cùng Rank 1, đội tiếp theo sẽ là Rank 3).
- **Rank Cuối Cùng**:
    - Rank Cuối Cùng là trung bình cộng của tất cả các thứ hạng mà đội đó đạt được trên toàn bộ các private test.
- **Đội Chiến Thắng**: Đội có Rank Cuối Cùng thấp nhất sẽ là đội chiến thắng chung cuộc của cuộc thi.

6. QUY ĐỊNH CHUNG

- **Thời gian thi đấu & Hình thức nộp bài:** Các đội trình bày vào buổi chiều ngày **21/06/2025**
- **Quyết định của Ban Tổ Chức là quyết định cuối cùng** trong mọi trường hợp tranh chấp hoặc vấn đề phát sinh.