Ingénierie des connaissances Web sémantique

Projet Final
Rapport

Réalisé par :

Ali HAITAM

Anass ELAYACHI

1) Ontology :

The PICE Ontology is designed to provide a user-friendly platform where individuals can search for event organizers and event locations within a city, while also displaying weather information for that city. This makes the event planning process more integrated and convenient by combining essential elements like location, organization, and weather forecasts in one easy-to-use system.

a) Classes and Hierarchies:

The ontology's class structure is meticulously designed to reflect the diverse aspects of event planning. The core classes — ***dbo:Event, pice:Venue, pice:WeatherForecast, and dbo:ServiceProvider*** — serve as the foundation for modeling the event planning domain.

- ***dbo:Event:*** This class is essential as it encapsulates the concept of an event in its entirety, which is the focal point of the ontology. The flexibility to represent various event types, from formal conferences to personal celebrations like weddings, allows for wide applicability within the event planning domain.
- ***pice:Venue:*** By defining venues as spaces where events occur, the ontology acknowledges the critical role of location in event planning. Differentiating venues by categories such as ***hotels or restaurants*** provides a nuanced understanding of the setting, which is vital for event logistics and ambiance.
- ***pice:WeatherForecast***: Including weather forecasts addresses the often overlooked but crucial aspect of planning—environmental conditions. By foreseeing weather-related factors, event planners can mitigate risks and ensure attendee comfort.
- ***dbo:ServiceProvider:*** Recognizing service providers as a class highlights the collaborative nature of event management, where multiple vendors contribute services like catering or entertainment, essential for the event's success.
- ***foaf:Person*** and ***pice:Participant:*** The distinction between a general person and a participant underscores the ontology's attention to the individuals involved in an event, recognizing their varied roles and contributions.
- ***pice:Equipment:*** Acknowledging equipment as a separate class reflects the importance of logistical details in creating the desired event experience, from audio-visual setups to specialized decor.

b) Properties inside the schema:

The ontology's properties bridge the classes, creating a network of relationships that reflect the event planning process. The properties are not just connections but are imbued with semantic richness that gives meaning to the links they establish.

1. ***dbo:location:*** This property is pivotal as it connects an event to its venue, providing a

geographical anchor and context for the event.

2. ***pice:organizedBy and dbo:organize:*** These properties form the reciprocal relationship between events and service providers, reflecting the real-world interactions where service providers are engaged in organizing events and vice versa.
3. ***foaf:knows:*** Including social relationships within the ontology allows for modeling networks of participants, which can be crucial for events like conferences where networking is a goal.
4. ***pice:hasEquipment:*** This property ensures that the logistical aspect of event planning is not overlooked, tying events to the equipment used, which is essential for execution.
5. Date and Theme Properties (***dbo:startDate, dbo:endDate, dbo:theme***): These properties capture the temporal and thematic dimensions of events, which are critical for scheduling and branding.
6. Weather Forecast Properties (***pice:hasTemperature, pice:hasWeatherDescription***, etc.)***:*** These properties enrich the ontology with environmental data, allowing event planners to make informed decisions based on forecasted conditions.
7. Venue and Service Provider Properties (***pice:hasAddress, pice:hasCategory, pice:hasPrice***, etc.): These properties provide detailed descriptors for venues and service providers, facilitating a comprehensive view of the options available for event planning.

- **Benefits of the PICE ontology:**

- Enhanced Coordination: By integrating various components of event planning such as venues, service providers, and weather forecasts, the PICE Ontology ensures seamless coordination, leading to more efficient event management.
- Comprehensive Data Access: With its extensive properties and classes, the ontology allows for a detailed understanding and representation of all aspects of an event, from logistical details to participant interactions.
- Improved Decision Making: The ability to link weather forecasts with event dates and locations aids in making informed decisions, ensuring better preparedness for environmental factors.
- Service Provider Optimization: The detailed classification of organizations and their services allows for a tailored selection of service providers, ensuring that the best possible options are chosen for each event.

c) Utilization of SHACL:

SHACL (Shapes Constraint Language) is used in this ontology to enforce data integrity, guide data entry, and validate RDF data against a set of conditions. Below are detailed explanations of the SHACL choices made for this ontology:

- *Event Shape Constraints:*

- Start and End Dates: The use of ***sh:minCount and sh:maxCount for dbo:startDate and dbo:endDate*** ensures that each event has exactly one start date and one end date, preventing ambiguity around when the event is scheduled to occur.

- Event Location Validation: By specifying ***sh:class pice:Venue*** for the ***dbo:location*** path, the shape restricts the location of an event to be an instance of ***pice:Venue***, ensuring that events are associated with proper venues within the ontology's scope.
- Temporal Logic: The ***sh:lessThan*** constraint applied to ***dbo:startDate*** relative to ***dbo:endDate*** enforces logical temporal sequencing where an event must start before it ends.

➢ *Venue Shape Constraints:*

- Address Requirement: The ***sh:minLength*** constraint applied to ***pice:hasAddress*** ensures that a venue's address is not left blank, which is crucial for locating the venue and planning logistics.

➢ *Organization Shape Constraints:*

- Service Provider Naming: The ***sh:minLength*** constraint for ***pice:hasName*** under ***dbo:ServiceProvider*** guarantees that service providers have meaningful names, which is essential for identification and communication.
- Service Price Validation: The ***sh:minInclusive*** constraint on ***pice:hasPrice*** ensures that the service price is a non-negative value, reflecting the real-world expectation that prices cannot be negative.

➢ *Weather Forecast Shape Constraints:*

- Temperature Constraints: The shape properties for ***pice:hasTemperature,*** ***pice:hasMinTemperature***, and ***pice:hasMaxTemperature*** with ***xsd:float*** datatype ensure that temperature values are appropriately formatted as numerical data.
- Consistency in Temperature Reporting: The ***sh:lessThanOrEquals*** constraint between ***pice:hasMinTemperature*** and ***pice:hasMaxTemperature*** prevents illogical data entry where the minimum temperature could erroneously be greater than the maximum temperature.

➢ *Venue Category Validation:*

- Predefined Categories: The ***sh:in*** constraint within the ***pice:VenueShape*** ensures that the venue categories assigned fall within a predefined list, thus maintaining consistency in categorization across the dataset.

d) Utilization of SKOS:

The Simple Knowledge Organization System (SKOS) is leveraged within the PICE ontology to provide a structured way of organizing knowledge related to event planning through concepts, semantic relationships, and collections. Here's how SKOS is utilized:

➢ *Concept Definitions and Hierarchy:*

- Base Concepts: ***pice:EventSpace*** and ***pice:City*** serve as foundational concepts that other concepts can link to, offering a context for where events occur and the urban settings they take place in.

- Event Types: ***pice:Party, pice:Conference, pice:Wedding, and pice:Birthday*** are defined as narrower concepts under ***pice:EventSpace***, signifying their specificity to the type of event space they represent. This hierarchical arrangement facilitates easy navigation through the ontology's taxonomy.
- Service Types: Similarly, ***pice:Catering, pice:Entertainment, and pice:AudioVisualEquipment*** are categorized under ***pice:EventSpace***, denoting their role in providing services during events.

➢ *Semantic Relationships:*

- Related Concepts: The ***skos:related*** property is used to associate cities like ***pice:Paris***, ***pice:London, and pice:Tokyo*** with various event activities and services, enriching the semantic network and indicating the kind of events and services commonly found in these cities.
- Collections: SKOS collections such as ***pice:EventTypes, pice:EventServices, and pice:Cities*** group relevant concepts, providing an organized view of the ontology that can be used for categorization and retrieval of information.

➢ *New Collection for Event Organization:*

- Organizing Events: A new SKOS collection has been introduced to encapsulate the elements usually involved in organizing an event. This includes ***pice:Venue*** for the location, ***pice:Attendees*** for the people present, and ***pice:ServiceProvider*** for the entity providing services. Grouping these concepts under a single collection offers a holistic view of the resources needed for event planning.

➢ *Meteorological Concepts:*

- Weather-Related Concepts: The ontology includes meteorological concepts like ***pice:Temperature, pice:Humidity, and pice:WeatherCondition*** under the broader concept ***pice:Meteorology***. This mirrors the real-world interplay between event planning and weather considerations.

➢ *Venue Categories:*

- Specific Venue Types: Concepts such as ***pice:BanquetRooms, pice:Cinema, and pice:Hotel*** are described as specific venue types that fall under the broader concept ***pice:Venue***, enabling users to classify venues based on their characteristics and intended use.
- SKOS Collection for Venue Categories: The ***pice:VenueCategories*** collection groups all venue types, making it easier to query and analyze the ontology based on venue classifications.

2) Data Lifting :

a) APIs:

In our report, the data lifting process is a critical component, enriched by the integration of twoAPIs: ***TomTom*** and ***OpenWeatherMap***, which significantly enhance the functionality of the PICE Ontology.

- TomTom API Integration:

We employ the TomTom API to gather comprehensive information about event-related venues in various cities. This API, with its extensive category range, allows us to focus specifically on locations pertinent to event planning. The integration of this data into the PICE Ontology means users have access to an updated and diverse array of venue options in their chosen city, tailored to the specific needs of their event. The robustness of this API is the variety of labels that it gives about the venue like its category, … and the number of filters before launching the call which increase the match between what desire the user and what he gets.

```
; Configuration du micro-service TomTom Category Search

custom_parameter[] = lat ; 43.7101728
custom_parameter[] = lon ; 7.261953200000001
custom_parameter[] = categorySet ; POI

; Chaîne de requête de l'API Web
; Personnalisez le paramètre key avec votre propre clé API obtenue de TomTom
api_query = "https://api.tomtom.com/search/2/categorySearch/.json?key=wIthmoUf6toN66cs7cEfSiuS52fAtAbI&lat={lat}&lon={lon}&limit=15&categorySet={

; Temps maximum (en secondes) pour mettre en cache les réponses de l'API Web. Par défaut : 2592000 = 30 jours. 604800 = 7 jours
cache_expires_after = 604800

; Ajouter des informations de provenance au graphe généré à chaque invocation. Par défaut : false
add_provenance = false
```

```
CONSTRUCT {
    ?Place
        a schema:Place;
        schema:identifier ?placeId;
        schema:name ?placeName;
        schema:category ?categories;
        schema:city ?municipality;
        schema:country ?country;
        schema:countryCode ?countryCode;
        schema:freeFormAddress ?freeformAddress;
}

WHERE {
    [] :results [
            :type ?placeType;
            :id ?placeId;
            :poi [
                :name ?placeName;
                :categorySet [
                    :id ?categoryId;
                ];
                :categories ?categories;
            ];
            :address [
                :streetName ?streetName;
                :municipality ?municipality;
                :postalCode ?postalCode;
                :country ?country;
                :countryCode ?countryCode;
                :freeformAddress ?freeformAddress;
            ]
```

☑	categorySet	7317
☑	lat	48.8588897
☑	limit	1000
☑	countrySet	FR
☑	radius	20000000
☑	language	FR, ANG
☑	lon	2.3200410217200766


```
"info": "search:ta:250007000406031-FR",
"poi": {
    "name": "Hôtel de Salm",
    "categorySet": [
        {
            "id": 7317
        }
    ],
    "categories": [
        "museum"
    ],
    "classifications": [⋯
    ]
},
"address": {
    "streetNumber": "64",
    "streetName": "Rue de Lille",
    "municipalitySubdivision": "7ème Arrondissement",
    "municipality": "Paris",
    "neighbourhood": "Les Invalides",
    "countrySecondarySubdivision": "Paris",
    "countrySubdivision": "Île-de-France",
```


- <u>OpenWeatherMap API Usage:</u>

For real-time weather data, we rely on the OpenWeatherMap API. This tool provides vital weather forecasts and current conditions for different cities over the next five coming days, a crucial aspect of event planning. Integrating this weather data into the ontology equips users with the necessary information to make informed decisions about event scheduling and preparation, adapting to weather conditions as needed.

```
prefix : <http://ns.inria.fr/sparql-micro-service/api#>
prefix xsd: <http://www.w3.org/2001/XMLSchema#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX schema: <http://schema.org/#>
prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>

CONSTRUCT {
    ?forecast
        a schema:forecast;
        schema:dt ?dt;
        schema:dt_txt ?dt_txt;
        schema:temp ?temp;
        schema:feels_like ?feels_like;
        schema:temp_min ?temp_min;
        schema:temp_max ?temp_max;
        schema:pressure ?pressure;
        schema:humidity ?humidity;
        schema:id ?id;
        schema:main ?main;
        schema:description ?description;
        schema:icon ?icon;
        schema:windSpeed ?speed;
        schema:cityName ?cityName;
        schema:cityCountry ?cityCountry;
}
```

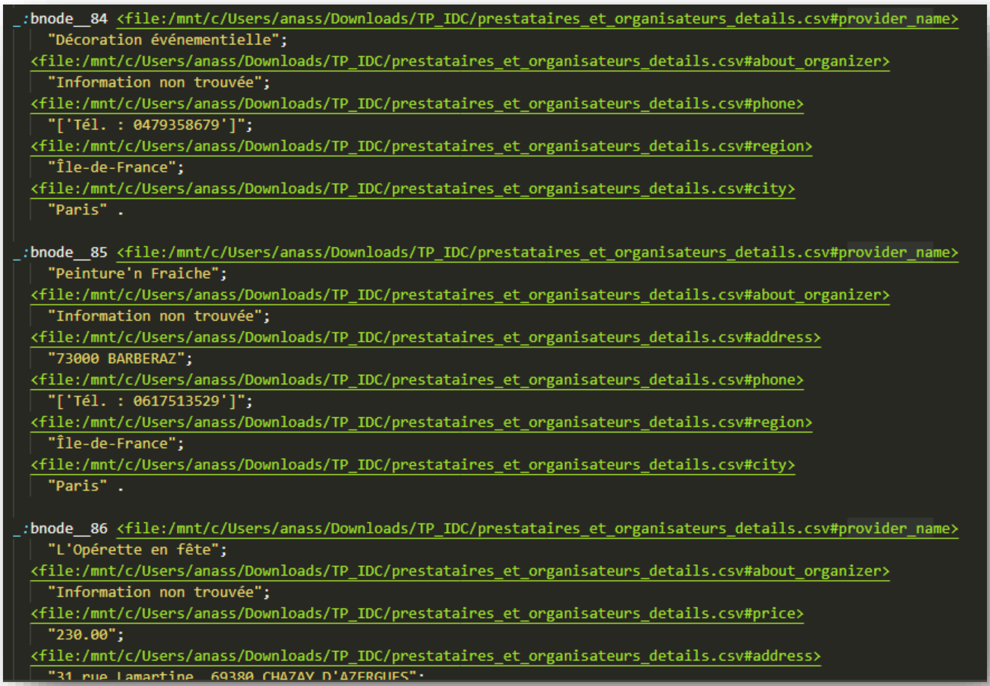

```
"list": [
    {
        "dt": 1706022000,
        "main": {
            "temp": 7.63,
            "feels_like": 7.63,
            "temp_min": 7.14,
            "temp_max": 7.63,
            "pressure": 1027,
            "sea_level": 1027,
            "grnd_level": 875,
            "humidity": 60,
            "temp_kf": 0.49
        },
        "weather": [
            {
                "id": 804,
                "main": "Clouds",
                "description": "overcast clouds",
                "icon": "04d"
            }
        ],
```


b) CSV:

For real-time weather data, we rely on the OpenWeatherMap API. This tool provides vital weather forecasts and current conditions for different cities, a crucial aspect of event planning. Integrating this weather data into the ontology equips users with the necessary information to make informed decisions about event scheduling and preparation, adapting to weather conditions as needed.

For the second aspect of our data lifting process, we utilized a CSV file generated from data scraping of the website 'https://www.lemagdelevenementiel.com/prestations-region-11.html'. This website is a comprehensive resource for event service providers across all regions of France

The website 'Le Mag de l'Événementiel' serves as a rich repository of event service providers in France, offering detailed listings across various categories. We employed web scraping techniques to extract relevant data from this website. This process involved programmatically navigating the site and collecting information on event service providers, such as names, services offered, and regional locations. The CSV file serves as a static dataset that complements the dynamic data obtained from the APIs. It provides a broad base of information about various event service providers in France, enriching the PICE Ontology with more localized and specific data.



```
_:bnode__84 <file:/mnt/c/Users/anass/Downloads/TP_IDC/prestataires_et_organisateurs_details.csv#provider_name>
  "Décoration événementielle";
 <file:/mnt/c/Users/anass/Downloads/TP_IDC/prestataires_et_organisateurs_details.csv#about_organizer>
  "Information non trouvée";
 <file:/mnt/c/Users/anass/Downloads/TP_IDC/prestataires_et_organisateurs_details.csv#phone>
  "['Tél. : 0479358679']";
 <file:/mnt/c/Users/anass/Downloads/TP_IDC/prestataires_et_organisateurs_details.csv#region>
  "Île-de-France";
 <file:/mnt/c/Users/anass/Downloads/TP_IDC/prestataires_et_organisateurs_details.csv#city>
  "Paris" .

_:bnode__85 <file:/mnt/c/Users/anass/Downloads/TP_IDC/prestataires_et_organisateurs_details.csv#provider_name>
  "Peinture'n Fraiche";
 <file:/mnt/c/Users/anass/Downloads/TP_IDC/prestataires_et_organisateurs_details.csv#about_organizer>
  "Information non trouvée";
 <file:/mnt/c/Users/anass/Downloads/TP_IDC/prestataires_et_organisateurs_details.csv#address>
  "73000 BARBERAZ";
 <file:/mnt/c/Users/anass/Downloads/TP_IDC/prestataires_et_organisateurs_details.csv#phone>
  "['Tél. : 0617513529']";
 <file:/mnt/c/Users/anass/Downloads/TP_IDC/prestataires_et_organisateurs_details.csv#region>
  "Île-de-France";
 <file:/mnt/c/Users/anass/Downloads/TP_IDC/prestataires_et_organisateurs_details.csv#city>
  "Paris" .

_:bnode__86 <file:/mnt/c/Users/anass/Downloads/TP_IDC/prestataires_et_organisateurs_details.csv#provider_name>
  "L'Opérette en fête";
 <file:/mnt/c/Users/anass/Downloads/TP_IDC/prestataires_et_organisateurs_details.csv#about_organizer>
  "Information non trouvée";
 <file:/mnt/c/Users/anass/Downloads/TP_IDC/prestataires_et_organisateurs_details.csv#price>
  "230.00";
 <file:/mnt/c/Users/anass/Downloads/TP_IDC/prestataires_et_organisateurs_details.csv#address>
  "31 rue Lamartine, 69380 CHAZAY D'AZERGUES";
```

c) CSV:

- *Information extraction: Challenges and strategic shift:*

In the information extraction phase of our project, our initial approach involved processing natural language texts to extract details about event service providers and their related information. This task was undertaken with the intention of leveraging both the REBEL and SpaCy Matcher tools. However, the outcomes of this approach did not meet our expectations due to several challenges.

- *Challenges Encountered:*

- Accuracy Issues: We observed that the precision of information extraction from natural language texts using REBEL and SpaCy Matcher was not sufficiently high. This led to a significant amount of erroneous or incomplete data, impacting the overall quality of our dataset.

Given these challenges, we made a strategic decision to pivot our approach. Instead of relying on text extraction from natural language sources, we chose to utilize data directly from our integrated APIs and a specially curated CSV file.

3) Federrated Query :

This federated SPARQL query performs a multi-dataset search to gather comprehensive information about venues in Paris, including their names, addresses, and categories. It integrates data from a local dataset, which lists venues and their attributes, with live weather data from the OpenWeatherMap service for Paris. The query filters for venues where the provider city matches "paris" (case-insensitive) and the temperature is above freezing, showcasing the flexibility of SPARQL in combining local static data with dynamic external data sources. The use of `LIMIT 10` ensures that only the first ten results are returned, and the `ORDER BY` clause sorts these results by date and time, providing a temporal context to the query's results. This exemplifies the rich construction of SPARQL queries in accessing and synthesizing information from distributed sources to aid in decision-making processes.

```
WHERE {
    ?venue a schema:Place ;
      schema:name ?venueName ;
      schema:category ?venueCategory ;
      schema:freeFormAddress ?venueAddress ;
      schema:city ?venueCity .

    ?organizer org:provider_name ?providerName ;
           org:city ?providerCity .

  SERVICE <http://localhost/service/openweathermap/forecast?city=paris> {
    ?forecast schema:temp ?temp ;
      schema:dt_txt ?dt_txt ;
      schema:description ?description ;
      schema:cityName ?cityName ;
      schema:dt ?dt .
  }
  FILTER(LCASE(STR(?providerCity)) = "paris")
  FILTER(?temp > 0)
}
ORDER BY (?dt)
LIMIT 10
```

Graph | XML/RDF | Table | Validate

num	?venueName	?venueAddress	?venueCategory
1	L'Indigo	Avenue Princesse Grace 10, Monaco, 98000	banquet rooms
2	L'Indigo	Avenue Princesse Grace 10, Monaco, 98000	restaurant
3	Le Mas des Bartavelles	Avenue de l'Aumarinier, 83600 FrA©jus	banquet rooms
4	Le Mas des Bartavelles	Avenue de l'Aumarinier, 83600 FrA©jus	restaurant

4) Application :

a) Two possible scenarios:

➢ *Scenario 1: Comprehensive Event Planning*

- User Journey:

- The user enters the name of a city where they want to organize an event.
- The application retrieves and displays the weather forecast for the selected city to help plan the event date.
- The user selects an event category (e.g., "banquet Rooms") and views a list of available venues.
- They can then view a list of service providers in that city, filtered by the selected category, complete with pricing and contact details.
- The user finalizes the venue and service providers and proceeds to book them through the platform.

- Interface Features:

- Weather forecast display with temperature, humidity, and conditions.
- Dynamic lists of venues and service providers with details and images.
- Booking and inquiry forms integrated with each service provider and venue listing.
- A calendar feature to select and view available dates based on the weather forecast and venue availability.

➢ *Scenario 2: Venue Discovery and Comparison*

- User Journey:

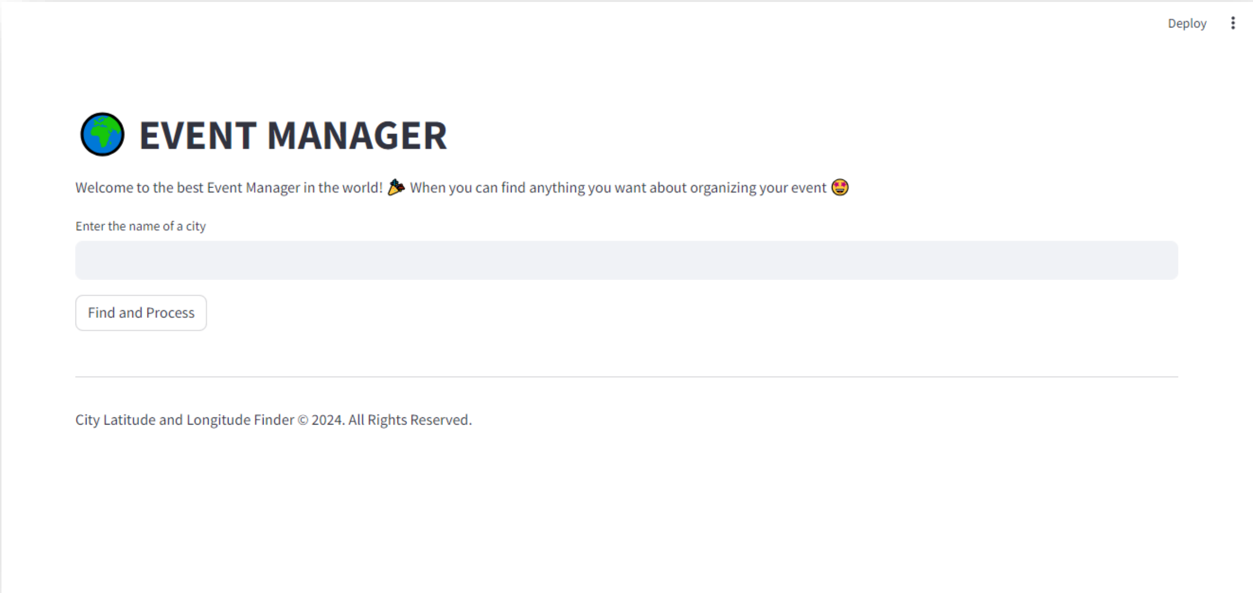
- The user is exploring options for a venue in a specific city.
- They can compare different venues based on user reviews, ratings, and weather conditions on preferred dates.
- The interface provides a comparison tool to overlay information about various venues, including capacity, facilities, and proximity to city landmarks.
- After selecting a venue, the user can send a request for a proposal directly to the venue managers.

- Interface Features:

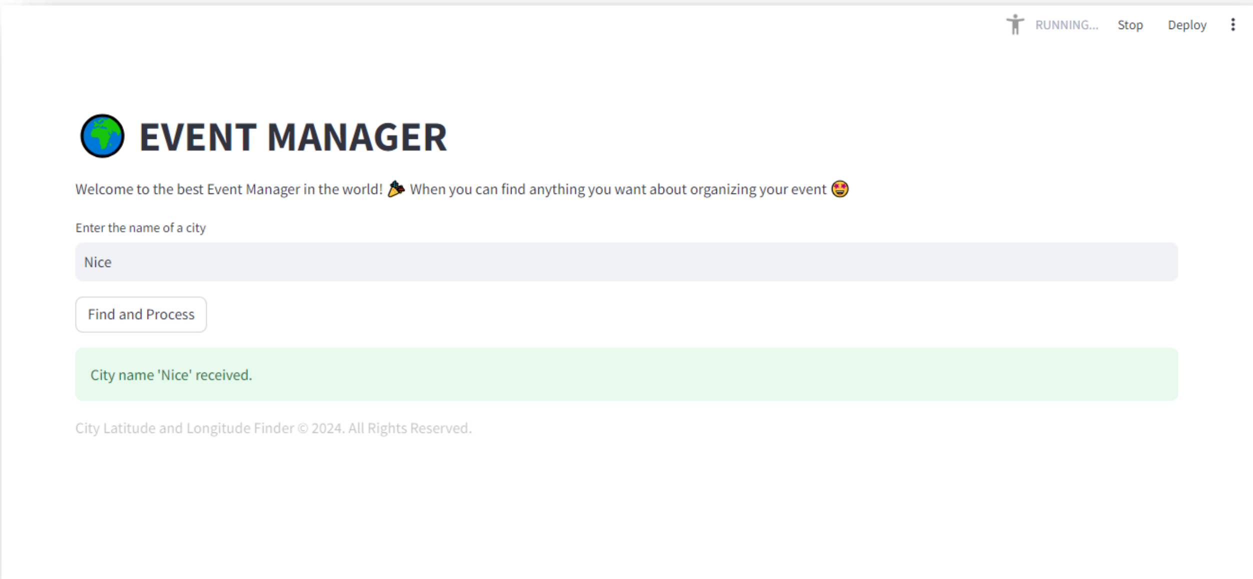
- Venue comparison charts and tables.
- Interactive map showing venue locations and weather conditions.
- User review and rating system for each venue.
- Direct messaging system to contact venue managers.

b) Web Interface:

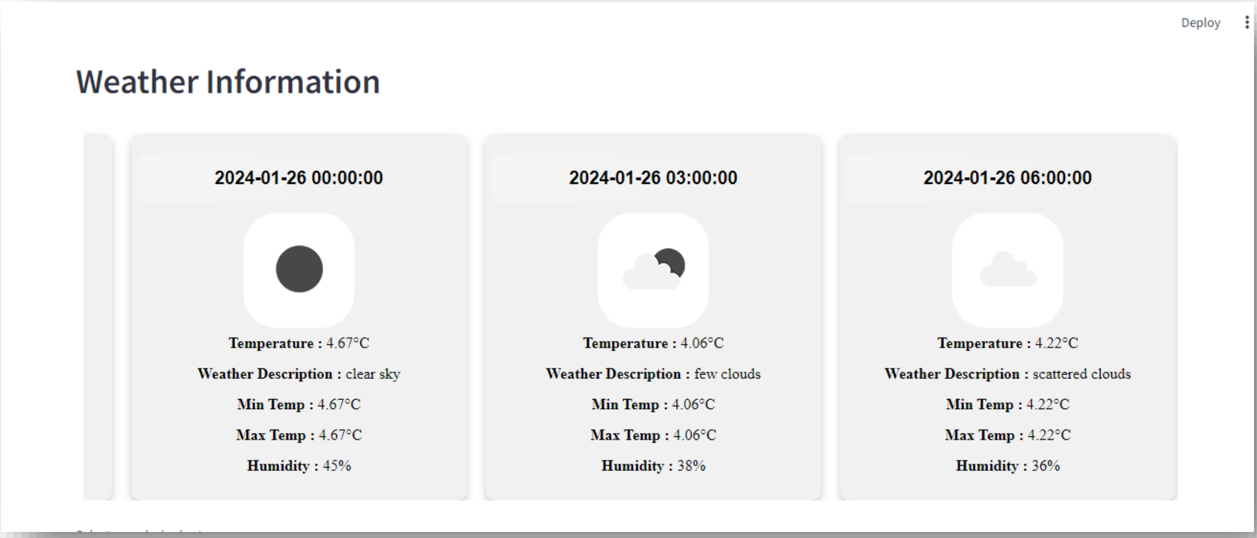
1) The Welcome Menu Face



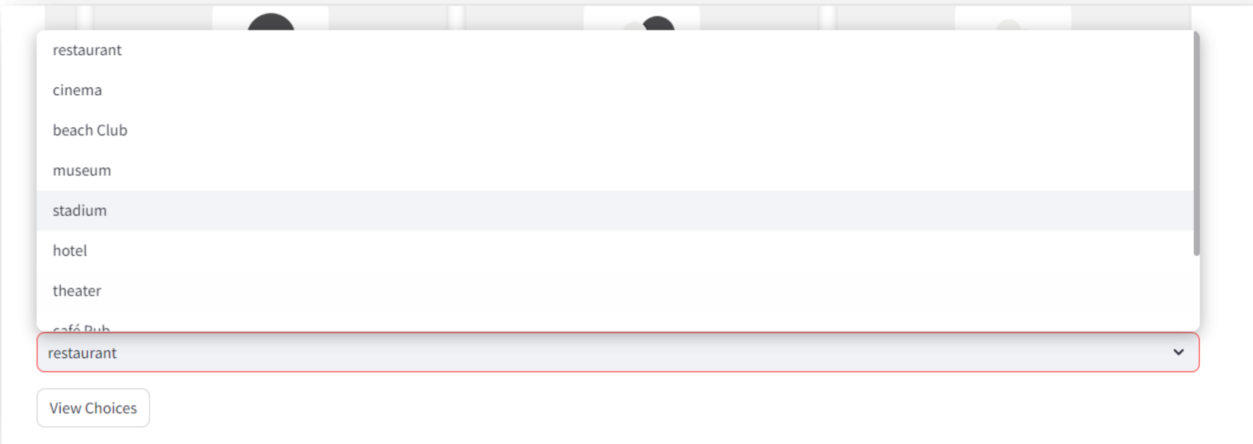
2) The User selects a specific city



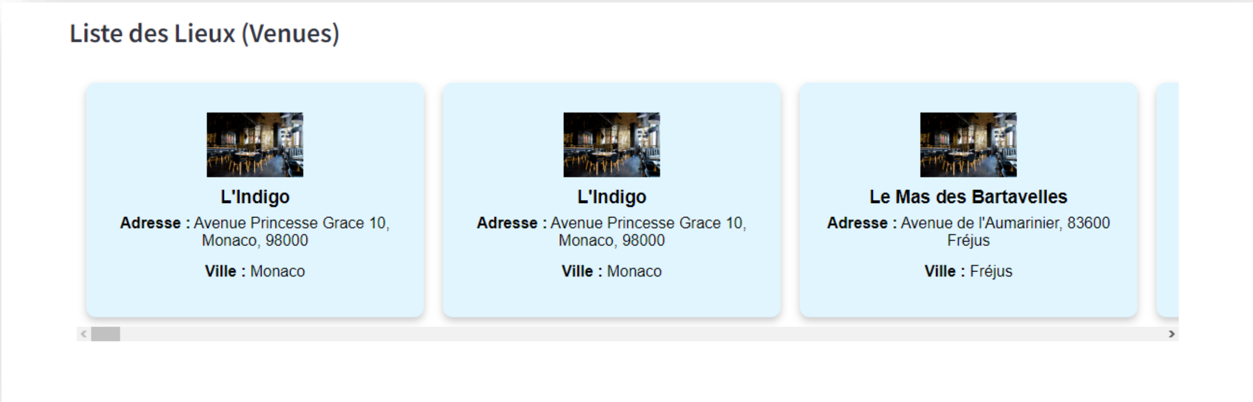
3) The user sees a detailed weather forecast for the chosen city

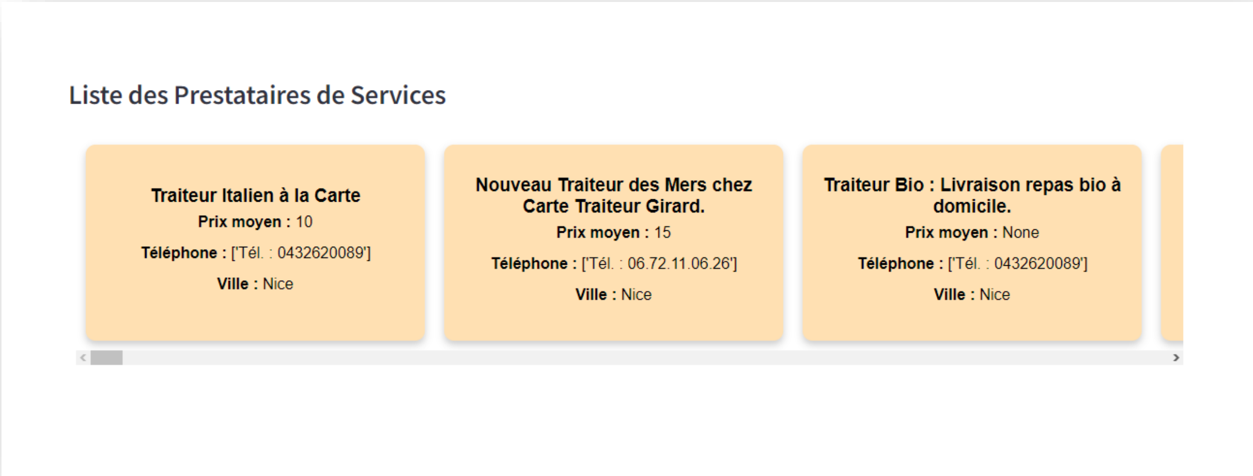


4) The user chooses a category of venues



5) The user can sees all the available venues with a variety of details and the service providers that he contact for organizing the event





- GITHUB:

  https://github.com/alihaitam/projet-final-IDC-WS