TECHNICAL REPORT FOR DEVELOPING A SOLUTION OF FILTERING AUTO ENTREPREUNERIA ACTIVITIY SUGGESTIONS

Problem

With the growing number of auto-entrepreneurs in Algeria, the National Agency for Auto-Entrepreneurs receives thousands of suggestions for new activities to be added to the official list of authorized businesses. However, the current manual validation process is time-consuming and inefficient, given that over 15,000 suggestions have been submitted. leading most of the activities to be either redundant activities that they already exist in the list of activity, or activities that are not legible for Auto Entrepreneuriat.

Objective

A key objective of our project is to develop an intelligent digital solution that will streamline the process of processing and filtering these activity suggestions. By leveraging Natural Language Processing (NLP), Clustering and Classification techniques, the system will ensure that only unique, relevant, and authorized activities are considered to be confirmed by the Agency.

Methodology

1. Approach used to solve the problem

a. Rejected Dataset:

We collect a dataset of rejected activities by performing web scraping to extract relevant data from various sources. The scraped data is then cleaned and structured before being stored in an Excel file. This file serves as an input for the model, allowing it to analyze patterns in rejected activities. By continuously updating the dataset, we improve the model's ability to detect and predict similar cases in the future.

b. Labeling Activities:

To organize extracted activities into meaningful groups, we applied clustering techniques. This step helps in categorizing activities under relevant domains, improving searchability and classification.

the labels (domains) are

- activities starting with 01 would be for domain: **Conseil, Expertise et formation**
- activities starting with 02 would be for domain: **Services numériques et des activités connexes**
- activities starting with 03 would be for domain: **Services à la personne**
- activities starting with 04 would be for domain: **Prestations à domicile**
- activities starting with 05 would be for domain: **Services de loisirs et de récréation**
- activities starting with 06 would be for domain: **Services aux entreprises**
- activities starting with 07 would be for domain: **Services culturels, de communication et d'audiovisuel.**

The clustering process involved:
Assigning labels to each cluster to define its domain.
Handling variations in wording within clusters to ensure consistency.

example:
In the data of activities, code activity from 072232 to 070101 the activities belongs all to the same domain Services culturels, de communication et d'audiovisuel and so on for others

c. Pre-processing

The preprocessing pipeline consists of language-specific techniques for Arabic and French text. The steps involved are:

1. General pre processing approach :
    - Dynamically identify text columns in the dataset.
    - Detect language using a character-based regex pattern.
    - Apply preprocessing based on the detected language.
    - Store processed text and language in new data frame columns.
2. Language Detection : To process text correctly, an initial language detection step is necessary. The approach relies on regular expressions
3. French text preprocessing : French text preprocessing follows standard text normalization practices:
    - Convert all characters to lowercase for uniformity.
    - Remove punctuation and digits to eliminate unnecessary symbols and numbers.
    - Tokenize text using `word_tokenize` from the NLTK library.
    - Remove stop words using the `NLTK` French stop word list.
    - Apply `SnowballStemmer` to reduce words to their base form.
4. Arabic text preprocessing : Arabic text presents unique challenges due to its script and morphological richness. The preprocessing steps include:
    - Remove diacritics using `strip_tashkeel` from `arabic_reshaper`.
    - Strip elongated characters with `strip_tatweel` for standardization.
    - Normalize characters by converting letters to a standard form (e.g., → أ ه → ة ,ا).
    - Remove non-Arabic characters to keep only Arabic script.
    - Tokenize text using whitespace-based splitting.

2. Explanation of Models/Algorithms Used

the algorithm is designed to match and validate new activities against redundant (existing) and declined activities that do not fit for auto entrepreneurs.

- it is designed to handle large datasets efficiently while ensuring high accuracy.

Field statistic calculation :

- `TF-IDF` vectorization : a TF-IDF vectorizer is trained on all existing activity names to capture term importance.
- BERT embeddings : pre computed embeddings for existing activities are generated using a multilingual BERT model.

for each field (domain), the following statistic are computed :

1. number of activities
2. average length of activity names
3. most common words
4. pre computed TF-IDF and BERT embeddings

purpose : used to validate new activities and compute similarity scores efficiently

Activity Validation

- length check
- word overlap check

purpose : ensures that new activities follow the general patterns observed in their respective field

Similarity Calculation

similarities are computed in batches to optimize performance.

Multi Metric Similarity : we use different similarity metrics to ensure best results, using :

1. TF-IDF similarity : using cosine similarity on TF-IDF vectors
2. Fuzzy Matching : Used for high-similarity matches to account for minor variations (ex: typos)
3. semantic similarity: using cosine similarity on BERT embeddings to capture contextual meaning.

combined score : the final score is weighted combination of the three metrics : (TF-IDF: 30%, Fuzzy: 30%, Semantic: 40%).

- Similarities are computed in batches of activities (depending on fields/domains) to optimize performance, especially for large datasets.
- BERT embeddings are computed in batches of 64 to balance memory usage and speed.

Activity matching

we have decided to add forbidden activities (i.e. activities that are not legible for auto entrepreneuriat) in order to make sure that we will look for similarities for each activity proposed, with different activities (either already existing (redundant) or not legible)

Activities with a combined similarity score less than `0.7` are flagged as potential activities that should be checked.
Semantic similarities above
`0.85` are considered strong matches. which mean that the proposed activity is redundant (matching with already existing activities) or declined (matching with activities that are not legible)

Handling of users with fields (domains) that do not match their activity

- Rejected activities could be due to fields not matching their activity and description.
    - for that, we use an nlp model to classify those activities to their real field (one of the 7s mentioned before)
    - after that, we run the program again on these activities for better and accurate filtering of the activities.

Workflow

1. Input :
    - **New Activities**: A DataFrame containing new activities to be validated.

- **Existing Activities**: A DataFrame containing existing activities for comparison.
- **Forbidden Activities**: A DataFrame containing activities that are not allowed.

2. Steps :

   a. **Preprocessing**:

      - Normalize all text inputs.
      - Train the TF-IDF vectorizer on existing activities.

   b. **Field Statistics Calculation**:

      - Compute statistics and embeddings for each field.

   c. **Activity Validation**:

      - Validate each new activity against its field's statistics.

   d. **Similarity Calculation**:

      - Compute similarities between new activities and existing activities in the same field.
      - Flag activities that exceed the similarity threshold as potential duplicates.

   e. **Forbidden Activity Matching**:

      - Compute similarities between new activities and forbidden activities.
      - Flag activities that exceed the similarity threshold as forbidden matches.

   f. **Output Results**:

      - Generate a DataFrame with the results, including:
        - Identifier for the new activity.
        - Activity details.
        - Status (`rejected` or `to_review`).
        - Reasons for rejection (if any).
        - Similarity matches and scores.

System Design for Activity Display Website

Interface Overview

This is a full-stack implementation of an interface for our activity filtering systems . The backend is built using Django, and the frontend is developed with React. The system is designed to filter activities that are entered by the user depending on redundancy, non legibility or to check out.

the interface allows the user to see activities filtered by the model, for him/her to either accept/reject one by one, or accept/reject all at once

- this was done because we wanted to combine AI with human interactions for efficient activity filtering.

Project Structure

```
hackathon/
├── backend/              # Django backend implementation
├── frontend/             # React frontend implementation
│   ├── public/           # Public assets
│   └── src/              # Source files
```

Installation and Setup

Prerequisites

- Python 3.x
- Django 4.x
- SQLite
- npm

Backend Setup

1. Navigate to the backend directory:

   `cd backend/hackathon`

2. Create a virtual environment:

```
python -m venv venv
source venv/bin/activate  # On Windows: venv\Scripts\activate
```

1. Install dependencies:

   `pip install -r requirements.txt`

   ADDED BY MAISSA

2. Run migrations:

   `python` `manage.py` `migrate`

3. Start the development server:

   `python` `manage.py` `runserver`

Frontend Setup

1. Navigate to the frontend directory:

   `cd frontend`

2. Install dependencies:

   `npm install`

3. Start the development server:

   `npm start`

4. Access the application at http://localhost:3000/.

Testing

Dataset Importing

import dataset (csv) that is stored with the zip to get data classified using the model.

Backend Testing

Run the test suite using:

python manage.py test

Frontend Testing

Run tests using:

`npm test`

Architecture Overview

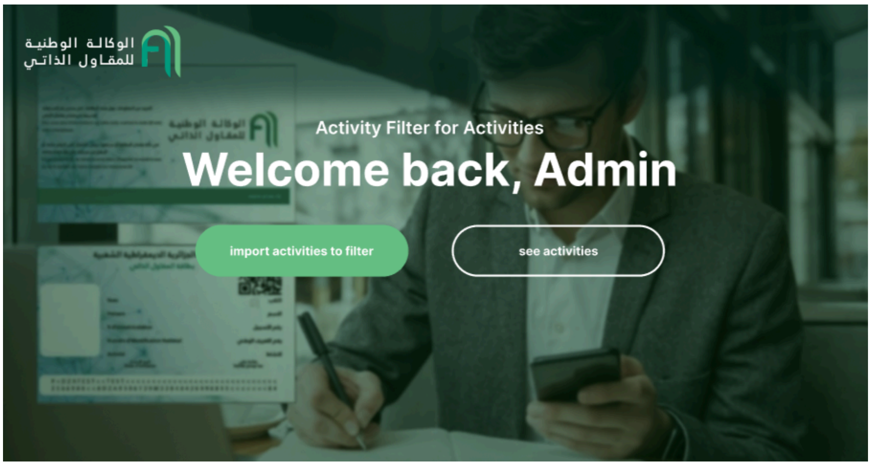
The system follows a **client-server architecture**, where:

- The **frontend** (React) provides an interactive UI for users.
- The **backend** (Django) handles API requests, authentication, and database operations.
- A **database** (SQLite) stores activity details, user interactions, and media.

Components and Interactions

Frontend (User Interface)

- **Homepage:** Displays upcoming and past activities.

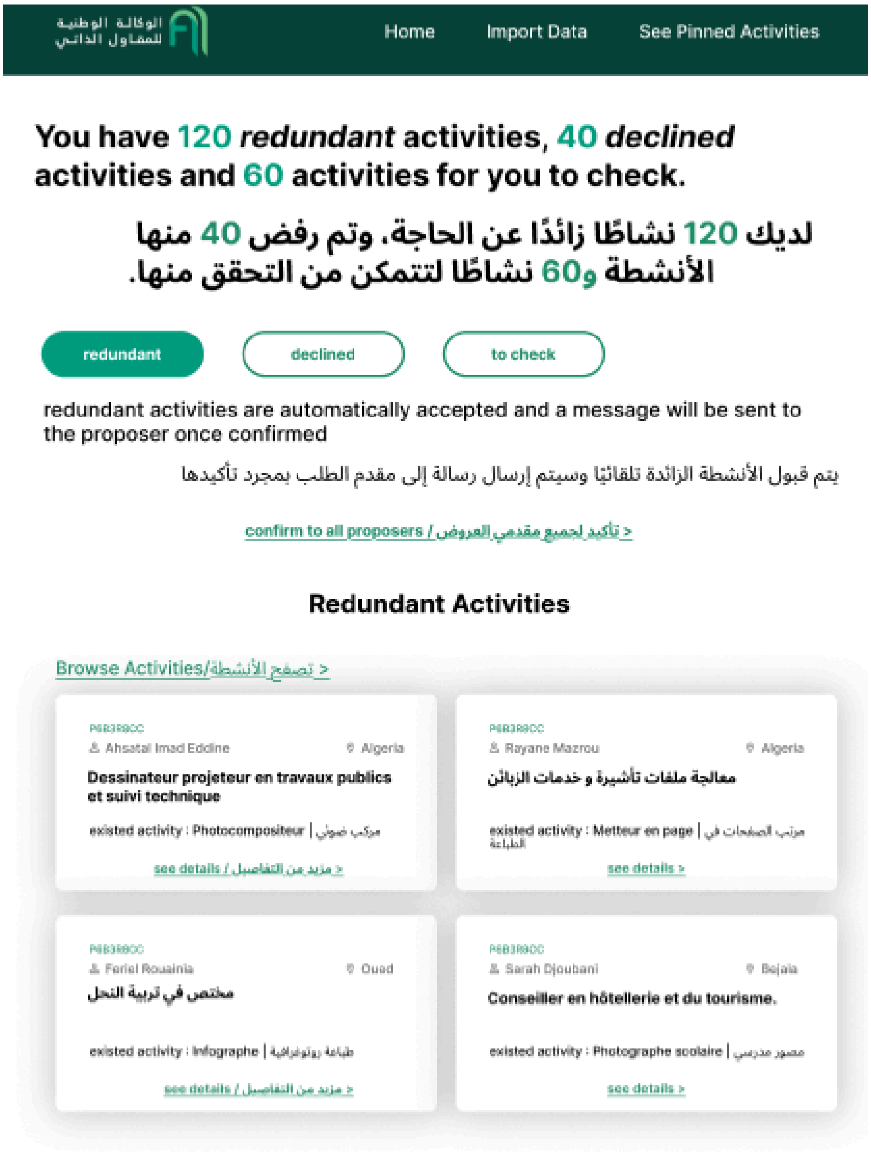


Activity Filter for Activities

Welcome back, Admin

- **activities Page:** A page that displays activities proposed by users, being filtered and classified to redundant, declined and the ones to be checked.

> ⚠️ the proposer’s personal information are random for better understanding of the interface, since the given dataset do not have the personal information for the proposer



You have 120 *redundant* activities, 40 *declined* activities and 60 activities for you to check.

لديك 120 نشاطًا زائدًا عن الحاجة، وتم رفض 40 منها
الأنشطة و60 نشاطًا لتتمكن من التحقق منها.

redundant activities are automatically accepted and a message will be sent to the proposer once confirmed

يتم قبول الأنشطة الزائدة تلقائيًا وسيتم إرسال رسالة إلى مقدم الطلب بمجرد تأكيدها

confirm to all proposers / تأكيد لجميع مقدمي العروض >

Redundant Activities

Browse Activities/تصفح الأنشطة >

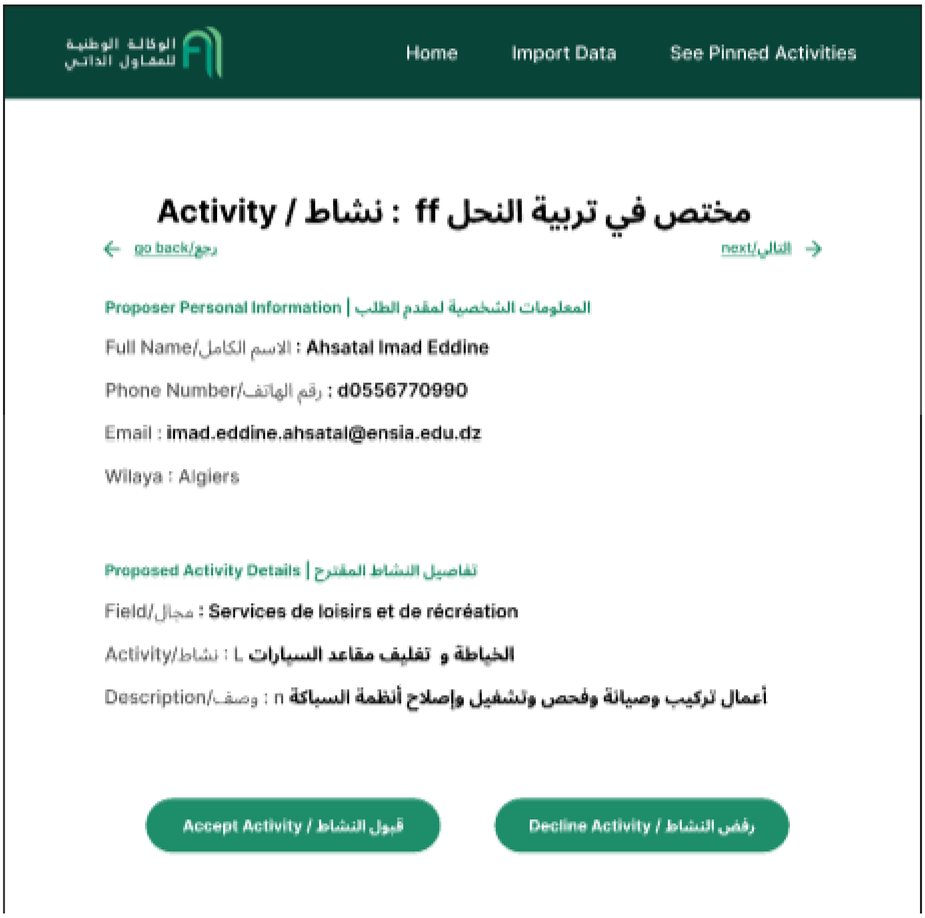
P6B3R9CC
Ahsatal Imad Eddine — Algeria
Dessinateur projeteur en travaux publics et suivi technique
existed activity : Photocompositeur | مركب ضوئي
see details / مزيد من التفاصيل >

P6B3R9CC
Rayane Mazrou — Algeria
معالجة ملفات تأشيرة و خدمات الزبائن
existed activity : Metteur en page | مرتب الصفحات في الطباعة
see details >

P6B3R9CC
Feriel Rouainia — Oued
مختص في تربية النحل
existed activity : Infographe | طباعة روتوغرافية
see details / مزيد من التفاصيل >

P6B3R9CC
Sarah Djoubani — Bejaia
Conseiller en hôtellerie et du tourisme.
existed activity : Photographe scolaire | مصور مدرسي
see details >

- **Activity Detail Page** : a page that displays details about a specific activity, displaying the proposer personal information, activity details including activity name, field and description and an option for similar activities depending if the activity is redundant or not.


الوكالة الوطنية للمقاول الذاتي
Home
Import Data
See Pinned Activities
go back/رجوع
next/التالي
Accept Activity / قبول النشاط
Decline Activity / رفض النشاط

Activity / نشاط : ff مختص في تربية النحل

Proposer Personal Information | المعلومات الشخصية لمقدم الطلب

Full Name/الاسم الكامل : Ahsatal Imad Eddine

Phone Number/رقم الهاتف : d0556770990

Email : imad.eddine.ahsatal@ensia.edu.dz

Wilaya : Algiers

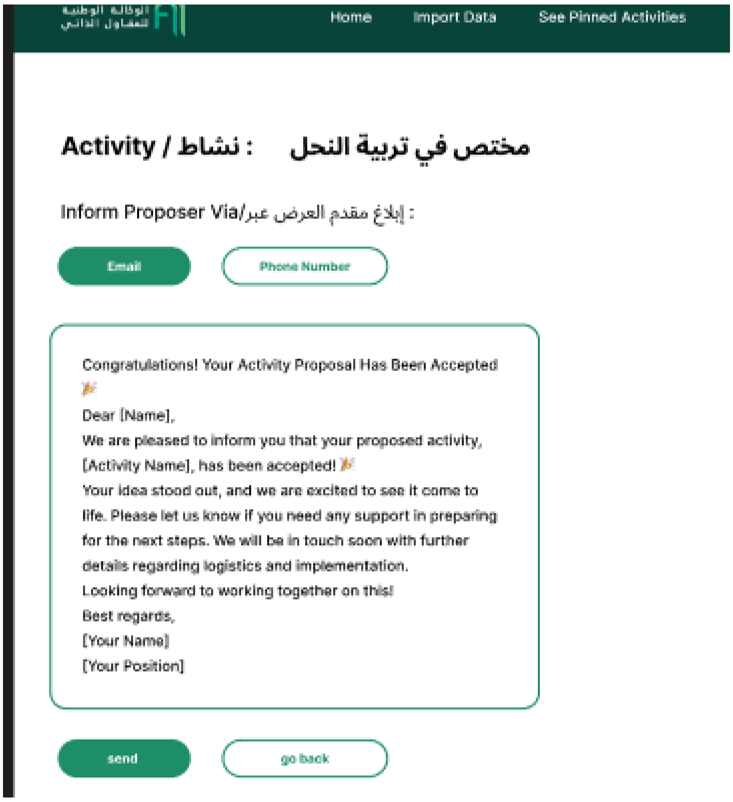
Proposed Activity Details | تفاصيل النشاط المقترح

Field/مجال : Services de loisirs et de récréation

Activity/نشاط : L الخياطة و تقليف مقاعد السيارات

Description/وصف : n أعمال تركيب وصيانة وفحص وتشغيل وإصلاح أنظمة السباكة

- Acceptance/Rejection for Activity Page (optional) : a page that allows the administrator to either accept/reject an activity and inform the proposer via email ro phone number



Activity / نشاط :   مختص في تربية النحل

Inform Proposer Via/إبلاغ مقدم العرض عبر :

Congratulations! Your Activity Proposal Has Been Accepted 🎉
Dear [Name],
We are pleased to inform you that your proposed activity, [Activity Name], has been accepted! 🎉
Your idea stood out, and we are excited to see it come to life. Please let us know if you need any support in preparing for the next steps. We will be in touch soon with further details regarding logistics and implementation.
Looking forward to working together on this!
Best regards,
[Your Name]
[Your Position]

Backend (API & Logic Layer)

- **Authentication Module** (JWT-based or session-based).
- **Activity Management API** (CRUD operations for activities).
- **User Management API** (for authentication and profile settings).
- **Analytics Module** (optional, to track engagement).

Database (Storage Layer)

- **Proposal Activity Table:** Stores activities Proposed by users (proposerName, proposerPhoneNumber, proposerEmail, proposerWilaya, activityName, activityField, activityDescription).

Evaluation & Results

1. Results

after model, we end up with 3 different labeled activities:

- **Redundant or Rejected Data Removal:** Entries identified as redundant, invalid, or irrelevant were automatically removed from the dataset to enhance data quality.

- **Uncertain Entries Left for Review:** For cases where model could not definitively determine whether a text should be included, the data was left for manual review. This allows the decision-makers to evaluate and decide if the text should be added.
- **Improved Data Consistency:** The cleaning and normalization processes significantly improved the structure of the dataset, making it more suitable for further analysis or classification tasks.

This approach balances automation with human oversight, ensuring that only meaningful and high-quality text is retained for NLP applications.

2. Evaluation

Our model achieved an accuracy between 60% and 65%, which indicates moderate performance but leaves room for improvement. Enhancing the dataset quality, and utilizing a more powerful GPU for better training efficiency could help improve performance.

Strengths and weaknesses of the solution

1. Strengths

1. Robust Handling of Large and Unstructured Data

The algorithm is engineered to process large volumes of data efficiently, even when the input data is unstructured or uncleaned. It incorporates advanced text preprocessing techniques, including normalization, special character removal, and handling of missing or inconsistent entries, ensuring reliable performance across diverse datasets.

---

2. Multilingual Support for Arabic, French, and English

Leveraging a multilingual BERT model (`paraphrase-multilingual-mpnet-base-v2`), the algorithm seamlessly processes and analyzes data in Arabic, French, and English. This capability ensures consistent and accurate results across multilingual datasets, making it suitable for diverse user bases and regions.

---

3. Ensemble of Multiple Models for Enhanced Accuracy

The algorithm employs a **model ensemble approach**, combining **TF-IDF**, **fuzzy matching**, and **semantic embeddings** to calculate similarity scores. This multi-faceted methodology ensures robust and precise matching by:

- Capturing term frequency and importance through TF-IDF.
- Accounting for minor variations (e.g., typos, abbreviations) using fuzzy matching.
- Understanding contextual meaning and semantic relationships via BERT embeddings.

  The final similarity score is a weighted combination of these metrics, ensuring a balanced and accurate evaluation.

4. Domain-Specific Batch Processing for Robustness

To enhance efficiency and accuracy, the algorithm compares proposed activities with existing activities **only within the same domain**. This domain-specific batch processing ensures that comparisons are contextually relevant and reduces computational overhead. Additionally, it minimizes false positives by focusing on activities that share the same field or category.

5. Double-Checking Mechanism for Declined Activities

The algorithm incorporates a **double-checking mechanism** to validate declined activities. This step ensures that activities are not incorrectly rejected due to potential errors, such as misleading or incorrectly entered domains by users. By re-evaluating declined activities, the system maintains fairness and reduces the risk of false rejections.

2. Weaknesses

One of the main weaknesses of our project is the limitation of our GPU infrastructure. The current setup is not powerful enough to efficiently process large amounts of text, which results in slower execution times and higher latency. This affects the overall performance of our model, especially when handling complex NLP tasks. To improve this, we need a better GPU that can support deep learning models more effectively, allowing for faster processing and smoother execution. Upgrading to a more powerful GPU would help enhance scalability and make our system more efficient.

Potential improvements

- since we had limited time, especially since running and testing took a lot of times and the reason is the use of our local gpus which are not powerful time, we could not integrate the fully model with our interface (website), so for the near future we would love to link both model performance and website for it to be real time filtering
- also our web app can be intergated easily to the admin dashboard, so in the future, we can make the filtering easy for the responsibles of filtering systems.
- in the form for the proposer, it should make the description as a required field, since we have noticed that most of proposers can use random domains for their activities while it their activity does not even match the domain selected, therefore a description being required would help for better activity filtering.
- instead of manually inserting activities that could be most of the time redundant, there should be a platform when it comes to proposing a commerce to directly let know the proposer that no need to send the request since the activity already exist
- add fields to add details about the existence of “registre de commerce”, since most activities proposed by the proposers they do not explain well how they practice their fields : meaning if they either work independetly (freelancing) or they do have “un registre de commerce” which is not eligble.

Suggestions for improvements and better data collections

- in the form for the proposer, it should make the description as a required field,
  since we have noticed that most of proposers can use random domains for

their activities while it their activity does not even match the domain selected,
therefore a description being required would help for better activity filtering.

- instead of manually inserting activities that could be most of the time redundant, there should be a platform when it comes to proposing a commerce to directly let know the proposer that no need to send the request
since the activity already exist.
- add fields to add details about the existence of “registre de commerce”, since
most activities proposed by the proposers they do not explain well how they practice their fields : meaning if they either work independetly (freelancing) or
they do have “un registre de commerce” which is not eligble