Meeting contact extraction

# Task overview

In the sales process any meeting is a valuable source of information about the customer. Sales executive usually must fill this information into CRM manually. In this assignment you need to use a meeting transcript to extract valuable information about a customer and create/update records in CRM.

Develop a solution that leverages Large Language Models to process Zoom meeting data (meeting information and WEBVTT files) and subsequently create or update Salesforce Contact records. The solution must consider the Salesforce schema for the Contact object, filling in all possible fields, including custom fields.

# Objectives

* **Extract relevant information** from Zoom meeting data (meeting info and WEBVTT files example provided).
* **Create or update Salesforce Contact records** using the extracted information.
* Ensure the solution adheres to the Salesforce schema for the Contact object, including custom fields.

# Deliverables

* **Source code** for the solution.
* **Documentation** explaining the implementation, setup, and usage.
* **Test cases** to validate the solution.
* **Deployment requirements:** models**,** platform, etc.

# Requirements

## Functional Requirements

1. **Data Extraction**
   * Extract relevant information from Zoom meeting info and WEBVTT files. This includes participant names, email addresses, company names, job titles, and any other relevant details. (**can be mocked, no need to integrate with Zoom API directly**)
2. **Data Processing**
   * Utilize LLMs to process the extracted data, identifying and organizing key information.
   * Ensure the data is formatted correctly for Salesforce.
3. **Salesforce Integration**
   * Create or update Salesforce Contact records based on the processed data.
   * Populate all possible fields in the Contact object, including custom fields defined in the Salesforce schema.
4. **Error Handling**
   * Implement robust error handling to manage data inconsistencies, missing information, and API errors.

## Non-Functional Requirements

1. **Performance**
   * The solution should process data efficiently, minimizing latency.
2. **Scalability**
   * Design the solution to handle a growing number of Zoom meetings and corresponding data.
3. **Security**
   * Ensure all data processing and storage adhere to security best practices.
   * Safeguard sensitive information such as email addresses and personal details.

## Technical Requirements

1. **Programming Languages**
   * Preferred: Python.
2. **Frameworks & Libraries**
   * LLM libraries: Langchain, Hugging Face Transformers, etc.
   * Salesforce SDKs: Simple Salesforce, Salesforce REST API, etc.
3. **APIs & Tools**
   * Zoom API for fetching meeting info and WEBVTT files.
   * Salesforce API for creating/updating Contact records.
4. **Data Formats**
   * Input: JSON for meeting info, WEBVTT for transcription files, Salesforce Contact schema.
   * Output: JSON for Salesforce API requests.

# Evaluation Criteria

* **Code Quality**: Clean, maintainable, and well-documented code.
* **Functionality**: Meets all functional requirements accurately.
* **Performance**: Efficient data processing and minimal latency.
* **Scalability**: Ability to handle increasing data volume.
* **Security**: Adheres to security best practices.

# Submission Instructions

* Submit the source code, documentation, and test cases via a Git repository or a ZIP file.
* Provide a deployment guide as a separate document (README.md).
* Include any additional setup instructions or dependencies.

# Useful links

* Create Salesforce developer account: <https://developer.salesforce.com/signup>
* Salesforce documentation: <https://developer.salesforce.com/docs/atlas.en-us.api_rest.meta/api_rest/intro_rest.htm>
* Salesforce objects metadata (schema): <https://developer.salesforce.com/docs/atlas.en-us.api_rest.meta/api_rest/using_resources_working_with_object_metadata.htm?q=Metadata>
* Zoom meeting API: <https://developers.zoom.us/docs/api/rest/reference/zoom-api/methods/#operation/meeting>
* WEBVTT format: https://developer.mozilla.org/en-US/docs/Web/API/WebVTT\_API