Data Engineer Exercise

The primary objective of this test is to evaluate the candidate's proficiency in data scraping, data cleansing, storage, and transformation. It's designed to assess both the technical acumen and the strategic thinking required to handle real-world data engineering challenges.

- Your primary task is to parse the content from the following file: **English Test Paper**. Extract relevant data, structure it appropriately, and store it in a format that facilitates easy retrieval and analysis.
- **Version Control with GitHub (Public or Private)**: Ensure your code is version-controlled using GitHub. It's imperative that your repository is systematically organized, and each commit clearly conveys its purpose.
- You are not restricted from using ChatGPT to expedite your tasks.
- Please ensure that your code is readable and flexible.
- The results don't necessarily have to be perfect (Paragraphs, Questions, or Options). Spending one to two days to complete the testing is enough. If you have encountered any issues, please note them in Task 4. The test goal is to observe the design architecture, software engineering concepts, and the trial process.

Task 1: Architectural Design for JSON Data Storage

Objective:

Construct a robust JSON schema to encapsulate the content of the provided English test paper, ensuring scalability and ease of data retrieval.

Example Schema Structure:

- `source`: The URL from where the test paper is sourced.
- `date`: The date on which the data was retrieved.
- `passages`: An array of passages from the test paper.
  - `id`: A unique identifier for each passage.
  - `content`: The text content of the passage.
  - `questions`: An array of questions related to the passage.
    - `id`: A unique identifier for each question.
    - `text`: The text content of the question.
    - `options`: An array of multiple choice options for the question.

```
{
    "source": "https://www.ceec.edu.tw/files/file_pool/1/0n045359274947649605/02-112%E5%AD%B8%E6%B8%AC%E8%8B%B1%E6%96%87%E8%A9%A6%E5%8D%B7.
    "date": "2023-09-04",
    "passages": [
        {
            ...
        },
    ]
}
```

Please note that you do not need to follow this format, you can propose the format that you think is the best and most convenient to handle.

Task 2: Constructing the Data Pipeline using Python

Objective:

Engineer a Python-based pipeline to parse the English test from the provided source and transform the content to align with the designed JSON schema. Feel free to use any method or library to extract text from the English test PDF document.

Requirements:

1. Implement a robust PDF text extraction mechanism.
2. Accurately identify and extract passages, associated questions, and multiple-choice options.
3. Populate the JSON schema with the parsed data, ensuring data integrity.
4. Implement error-handling mechanisms to gracefully manage potential parsing discrepancies or anomalies in the PDF content.

Deliverables:

- A Python script capable of ingesting the PDF URL and outputting the structured JSON data.
- Supplementary scripts or modules, if any, aiding in the parsing or transformation process.

Task 3: Comprehensive README Documentation

Objective:

Craft a detailed README file elucidating the setup, execution, and underlying logic of the Python parsing script, ensuring clarity for end-users and potential contributors

Task 4: Describe the Challenges That You Meet When Taking the Test

Objective:

Reflect on the process of completing the tasks and identify any challenges or obstacles faced during the test.

Example:

1. **Parsing Complexities**: ...
2. **Schema Design Decisions**: ...
3. **Coding Hurdles**: ...
4. **Documentation Intricacies**: ...
5. **Time Management**: ...