**Artificial Intelligence Record Linkage Documentation**

**Project Overview**

Linkage of record in HDSS communities within the inspire network which are in two different datasets.

To allocate and link common data between the hdss dataset and the facility dataset.

**Introduction**

This project was implemented through a Django application designed to perform record linkage between two datasets: SyntheticFacilityV3 and SyntheticHdssV3. It allows users to search for records in both datasets based on various criteria and view the search results.  
  
The record linkage is done using the fellegi-sunter probabibilistic approach to data linkgae

**Project Team:**

Namuli Sylvia: As the team leader, Ms. Namuli played a pivotal role in orchestrating the project's activities. She contributed extensively to the development of the Jupyter notebook, overseeing its implementation and ensuring comprehensive documentation throughout the project's lifecycle.

Ainomugisha Priscilla: Ms. Ainomugisha actively participated as a core team member, making valuable contributions to both the Jupyter notebook and the front-end development. Her versatile skills allowed her to seamlessly integrate her efforts into multiple aspects of the project.

Aheebwomugisha Sasha Ana: Ms. Aheebwomugisha was an integral part of the team, primarily focusing her efforts on the front-end development. Her dedication and expertise significantly enhanced the user interface, contributing to the project's overall success

The repository encompasses the essential datasets provided for the hackathon, including hdss.csv and facility.csv. Additionally, it includes a comprehensive Python notebook and components developed using Django framework.

PROJECT PLAN

* Data Cleaning and Preprocessing:
* Record Linkage Logic:
* Front-end Development
* Integration and Testing:
* Documentation and Deployment:

**Requirements**

- Python (>=3.6)

- Django (>=3.2)

- pandas (>=1.3)

- numpy (>=1.21)

- asgiref (>=3.4)

- background-task (>=1.2)

**Installation**

1. Clone the repository from GitHub:

2. Create a virtual environment (optional but recommended):

```bash

python3 -m venv env

source env/bin/activate # Linux/macOS

env\Scripts\activate # Windows

3. Install the required Python packages

```bash

pip install -r requirements.txt

```

4. Apply migrations to set up the database:

```bash

python manage.py migrate

```

**Running the Application**

To run the application, execute the following command in the terminal:

```bash

uvicorn record\_project.asgi:application```

This will start the ASGI server, and your Django application will be accessible at `http://127.0.0.1:8000/`.

**IMPLEMENTATION OF THE PROJECT**

* Data Cleaning:

- Cleaned datasets by formatting dates and removing leading spaces in names.

- Handled missing values and standardized data formats for consistency

* 2. Record Linkage:

- Utilized Fellegi-Sunter, a probabilistic method for record linkage.

- Defined identifier fields and their match probabilities for comparing records.

- Developed logic to compare records across datasets and calculate match scores.

* 3. Integration with Django:

- Integrated record linkage logic into Django framework for seamless operation.

- Utilized Django ORM to interact with the database and retrieve datasets.

* 4. User Interface

- Designed a user-friendly interface for interacting with the record linkage system.

- Implemented features for searching records and displaying linked results.

* 6. Documentation and Deployment:

- Created comprehensive documentation covering installation, usage, and system architecture.

**Additional Information**

- The application uses background tasks for some operations, such as cleaning and linking records, to improve performance and responsiveness.

- Make sure to have appropriate permissions and access rights to interact with the application's features.

**Interface**  




