**Mini Search Engine Documentation**

## Introduction

The Mini Search Engine is a basic search application implemented in Python. It uses a dictionary for in-memory storage and performs a linear search on data loaded from a CSV file or a MySQL database.

## Features

1. **In-memory Dictionary:**
   * Fast and efficient searching using a Python dictionary.
2. **CSV File Storage:**
   * Persistent storage using CSV files to store data between sessions.
3. **MySQL Database Integration:**
   * Ability to store and retrieve data from a MySQL database for scalability.

## Installation

### Prerequisites

* Python 3.x
* Pandas library (**pip install pandas**)
* mysql-connector-python library (**pip install mysql-connector-python**)

## Usage

1. **In-memory List:**
   * Initialize the search engine with an empty List.
   * search\_engine = []

* Search for documents based on a query.

for i in low\_prior:  
 if i.lower() in combination or i in combination:  
 combination.remove(i)  
 elif i.upper() in combination:  
 combination.remove(i)  
  
print(combination)  
  
for y in combination:  
 for x in person:  
 for i in x:  
 if y.lower()==i.lower() or y==i:  
 for ai in x:  
 print(ai,end=' ')  
  
  
 elif y.upper()==i.upper():  
 for bi in x:  
 print(bi,end=' ')

1. **CSV File Storage:**

* Save the in-memory dictionary to a CSV file.

def extract\_tables\_as\_csv():  
 counter=1  
 global amount\_of\_tables  
 for i in range(amount\_of\_db):  
 connection.database=databases\_as\_str[i]  
  
 cursor.execute("show tables;")  
  
 result=cursor.fetchall()  
 for table in result:  
 a=''.join(map(str,table))  
 query=f"select \* from {a}"  
 output=pd.read\_sql(query,connection)  
 output.to\_csv(f"table{counter}.csv",index=False)  
 counter+=1  
 amount\_of\_tables+=1

1. **MySQL Database Integration:**

* Connect to a MySQL database.

connection=db.connect(  
 host="localhost",  
 user="root",  
 password="password"  
)

* Save the databases available in MySQL Server as a list

cursor=connection.cursor()  
cursor.execute("Show databases;")  
  
all\_database=cursor.fetchall()  
for database in all\_database:  
 databases.append(database)

* Load data from the MySQL database into the in-memory list.

for i in range(1,amount\_of\_tables+1):  
 with open(f'table{i}.csv','r') as file:  
 records = file.readlines()  
 key\_list = records[0].strip().split(',')  
 for record in records:  
 person.append(record.split())

## Conclusion

The Mini Search Engine offers a basic solution for search functionalities using a dictionary. Users can choose between in-memory storage, CSV file storage, or MySQL database storage based on their needs. The provided Python script encapsulates the implementation details, including the linear search approach when querying data loaded from CSV or MySQL. Feel free to extend the functionality or adapt the code to suit your specific requirements.