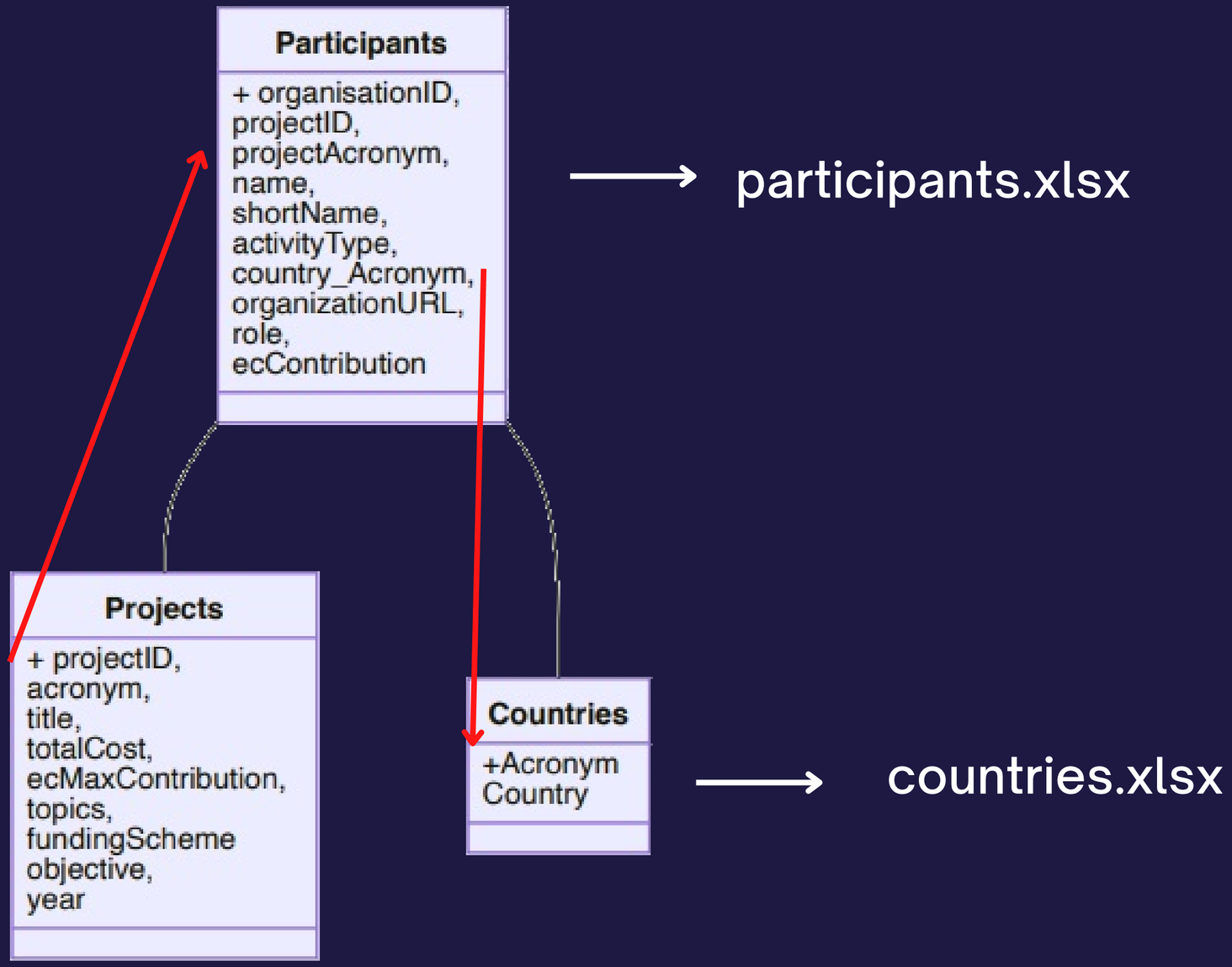
Partner Search Tool

Programming in Python

Coding Process Explanation

Given Data:
3 Excel Files



Purpose

Excel Files

SQL Database

Google Colaboratory

GitHub

Web App on Streamlit Cloud Server

Filtered Information Displayed by Country

Colab Notebook: CreateDB.ipynb

```
import sqlite3
import pandas as pd
from google.colab import drive
drive.mount('/content/drive', force_remount=True)

con= sqlite3.connect('ecsel_database.db')
df_participants=pd.read_excel('/content/drive/MyDrive/participants.xlsx')
df_countries=pd.read_excel('/content/drive/MyDrive/countries.xlsx')
df_projects=pd.read_excel('/content/drive/MyDrive/projects.xlsx')

df_projects.to_sql('projects', con, if_exists='replace', index= False)
df_countries.to_sql('countries', con, if_exists='replace', index= False)
df_participants.to_sql('participants', con, if_exists='replace', index= False)

con.close()
```

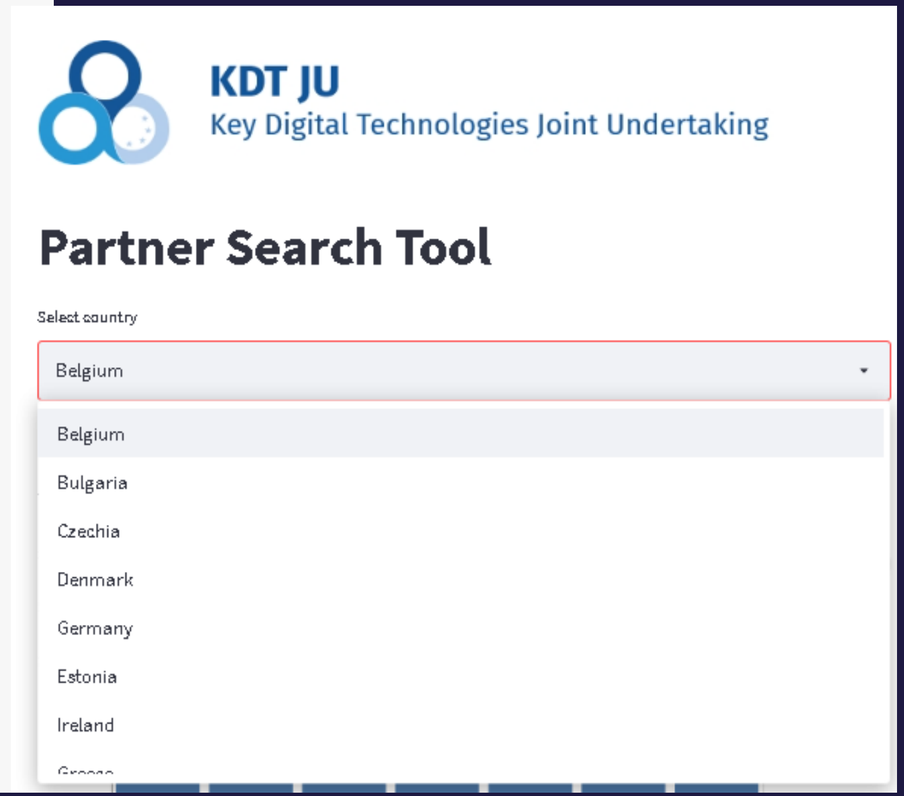
ecsel_database.db

Final.py 1/3

```
import sqlite3
import pandas as pd
import streamlit as st
from PIL import Image

database = 'ecsel_database.db'
selects= {
'grants':
'''SELECT j.year, SUM(p.ecContribution) AS grants
  FROM participants p JOIN projects j ON p.projectID==j.projectID
  WHERE p.country = '{}'
  GROUP BY j.year''',

'participants':
'''SELECT shortName, name, activityType, organizationURL, COUNT(ecContribution) n_projects, SUM(ecContribution)
  FROM participants p
  WHERE p.country = '{}'
  GROUP BY name ORDER BY SUM(ecContribution) DESC''',

'coordinators':
'''SELECT p.shortName, p.name, j.acronym
  FROM participants p JOIN projects j ON p.projectID = j.projectID
  WHERE p.country='{}' AND p.role = 'coordinator' ''
}
```

Final.py 2/3

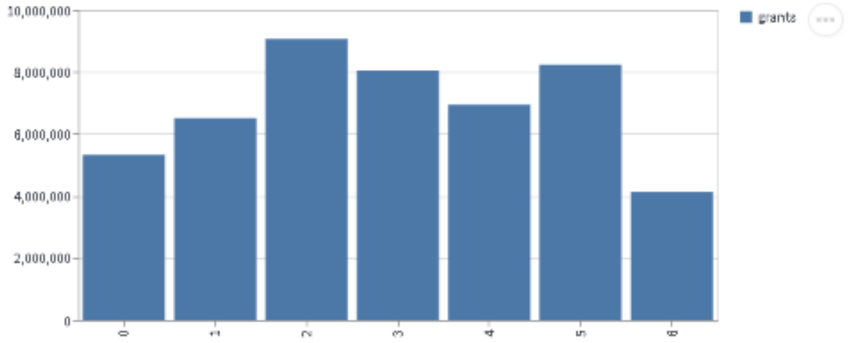
```
image=Image.open('descarga.png')
st.image(image)
st.title('Partner Search Tool')

conn=sqlite3.connect(database)
df_countries= pd.read_sql('SELECT * FROM countries', conn)
countries=list(df_countries.Country)

ct= st.selectbox('Select country', countries)
country = df_countries[df_countries.Country== ct].Acronym.item()

st.write(f'You selected: {country}-{ct}')

dfs={}
for key, sel in selects.items():
  dfs[key]=pd.read_sql(sel.format(country), conn)
```



```
st.subheader(f'Yearly EC contribution in {ct} (€)')
st.bar_chart(dfs['grants'])

st.subheader(f'Participants in {ct}')
st.dataframe(dfs['participants'])
csv_p=dfs['participants'].to_csv().encode('utf-8')

st.download_button(
    label= 'Download participants data as CSV',
    data=csv_p,
    file_name=f{country}_participants.csv',
    mime='text/csv',
)
```

```
st.subheader(f'Project coordinators in {ct}')
st.dataframe(dfs['coordinators'])
csv_c=dfs['coordinators'].to_csv().encode('utf-8')

st.download_button(
    label= 'Download coordinators data as CSV',
    data=csv_c,
    file_name=f{country}_coordinators.csv',
    mime='text/csv',
)

conn.close()
```

Yearly EC contribution in Spain (€)



Participants in Spain

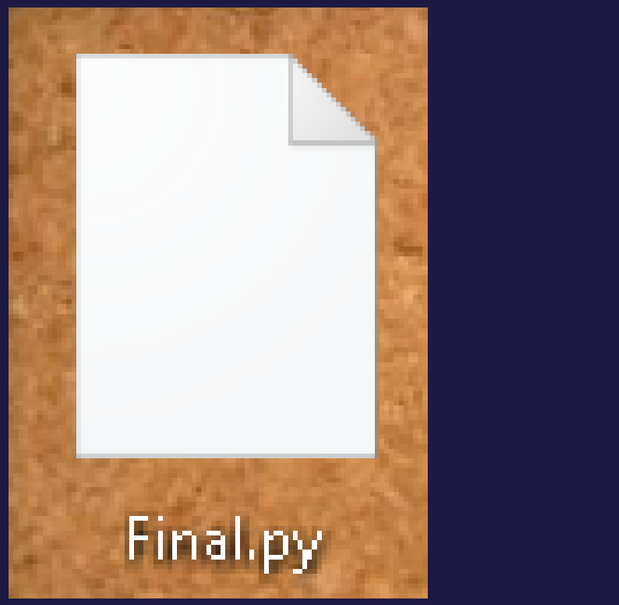
	shortName	name	activityType	organizationURL
0	TECNALIA	FUNDACION TECNALIA R...	REC	http://www.tecna
1	IKERLAN	IKERLAN S. COOP	REC	http://www.ikerla
2	INDRA	INDRA SISTEMAS SA	PRC	http://www.indra.
3	CSIC	AGENCIA ESTATAL CONS...	REC	http://www.csic.e
4	ITI	INSTITUTO TECNOLOGIC...	REC	www.iti.es
5	MGEP	MONDRAGON GOI ESKOL...	HES	www.eps.mondra
6	UPM	UNIVERSIDAD POLITECN...	HES	http://www.upm.e
7	HI-IBERIA	HI IBERIA INGENIERIA Y P...	PRC	www.hi-iberia.es
8	SEVEN SOLUTIONS SL	SEVEN SOLUTIONS SL	PRC	www.sevensols.co
9	THE REUSE COMPANY	KNOWLEDGE CENTRIC S...	PRC	<NA>

Download participants data as CSV

Project coordinators in Spain

	shortName	name	acronym
0	UPM	UNIVERSIDAD POLITECN...	AFarCloud
1	MGEP	MONDRAGON GOI ESKOL...	MANTIS
2	IKERLAN	IKERLAN S. COOP	FRACTAL
3	INDRA	INDRA SISTEMAS SA	COMP4DRO...
4	UPM	UNIVERSIDAD POLITECN...	SWARMs
5	TECNALIA	FUNDACION TECNALIA R...	AMASS
6	TAS-E	THALES ALENIA SPACE E...	AQUAS

Download coordinators data as CSV



Final.py 3/3

keywords.py

```python
def extract_nonstops(text):
    nlp = spacy.load("en_core_web_sm")
    punctuation = '!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
    result = []
    doc = nlp(text.lower())
    for token in doc:
        if not (token.text in nlp.Defaults.stop_words or token.text in punctuation):
            result.append(token.lemma_)
    return result


def extract_all_words(text):
    nlp = spacy.load("en_core_web_sm")
    punctuation = '!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
    result = []
    pos_tag = [ 'ADJ', 'NOUN']
    doc = nlp(text.lower())
    for token in doc:
        if (token.pos_ in pos_tag) and not (token.text in nlp.Defaults.stop_words or token.text in punctuation):
            result.append(token.lemma_)
    return result

def words_count(words):
    kwds = {}
    for w in words:
        if w not in kwds:
            kwds[w]=1
        else:
            kwds[w]+=1
    return kwds

def common_words(words_dict, n=3):
    vals = list(words_dict.values())
    vals.sort(reverse=True)
    words = [k for k,v in words_dict.items() if v >= vals[n-1]]
    return words

def extract_words(text, n=3):
    hotwords = extract_all_words(text)
    words_dict = words_count(hotwords)
    vals = list(words_dict.values())
    vals.sort(reverse=True)
    words = [k for k,v in words_dict.items() if v >= vals[n-1]]
    words = ', '.join(words)
    return words
```

```python
# Extract default values for 'unitest'

if __name__ == '__main__':

    # Test NLP functions
    text = '''Python is one of the most popular programming languages today
and is easy for beginners to learn because of its readability.
It is a free, open-source programming language with extensive support modules
and community development, easy integration with web services, user-friendly
data structures, and GUI-based desktop applications. '''

    nostops = extract_nonstops(text)
    hotwords = extract_words(text)
    words = extract_all_words(text)
    words_dict = words_count(words)
    keywords = common_words(words_dict)
```

test.py

```python
import spacy
import unittest
import keywords as kw
```

```python
text = '''Python is one of the most popular programming languages today
and is easy for beginners to learn because of its readability.
It is a free, open-source programming language with extensive support modules
and community development, easy integration with web services, user-friendly
data structures, and GUI-based desktop applications. '''


# String to perform the assessment in the 'extract_words' test
s = 'programming, language, easy'

# Dictionary to perform the assessment in the 'test_count' test
d = {'popular': 1, 'programming': 2, 'language': 2, 'today': 1, 'easy': 2, 'beginner': 1, 'readability': 1, 'free': 1, 'open
```

```python
class TestStringMethods(unittest.TestCase):


    def test_hotwords(self):
        # This function tests the 'kw.extract_words(text)' function
        # Do not modify the function name
        # Insert your code here:
        extractedwords=kw.extract_words(text)
        self.assertEqual(extractedwords,s)

    def test_count(self):
        # This function tests the 'kw.words_count' function
        # Requires to use 'kw.extract_all_words(text)' as an intermediate step to get
        # Do not modify the function name
        # Insert your code here:
        allwords=kw.extract_all_words(text)
        nofwords=kw.words_count(allwords)
        self.assertEqual(nofwords,d)


if __name__ == '__main__':
 unittest.main()
```

modules imported
unitest to permit test execution
keywords to use created functions

Data given to realize the test

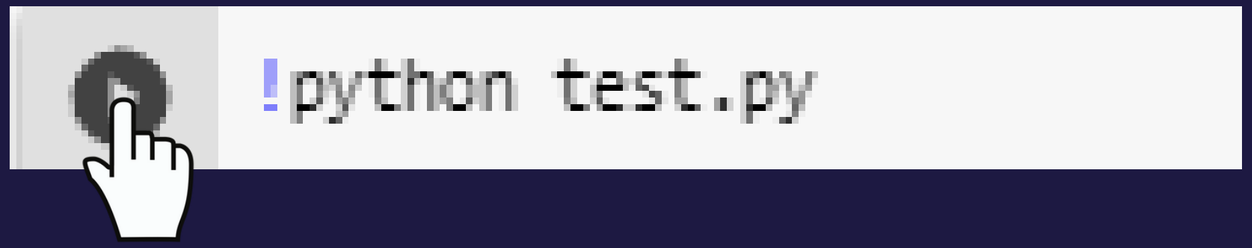
Signify the begining of functions to be tested

test whether or not s is equal to the value of hotswords as it is defined in keywords.py

```
hotwords = extract_words(text)
```

test whether or not d is equal to words_dict as it is defined in keywords.py

```
words_dict = words_count(words)
```




```
F.
======================================================================
FAIL: test_count (__main__.TestStringMethods)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "test.py", line 36, in test_count
    self.assertEqual(wordsnum,d)
AssertionError: {'pop[160 chars] 1, 'community': 1, 'development': 1, 'integra[123 chars]': 1} != {'pop[160 chars] 1, 'module': 1, 'community': 1, 'development'[137 chars]': 1}
  {'application': 1,
   'beginner': 1,
   'community': 1,
-  'data': 1,
?      ^

+  'datum': 1,
?      ^^

   'desktop': 1,
   'development': 1,
   'easy': 2,
   'extensive': 1,
   'free': 1,
   'friendly': 1,
   'gui': 1,
   'integration': 1,
   'language': 2,
+  'module': 1,
   'open': 1,
   'popular': 1,
   'programming': 2,
   'readability': 1,
   'service': 1,
   'source': 1,
   'structure': 1,
   'support': 1,
   'today': 1,
   'user': 1,
   'web': 1}

----------------------------------------------------------------------
Ran 2 tests in 1.318s

FAILED (failures=1)
```

The test failed !

We have the detail of where the test failed.

2nd function : nofwords =! d

The detail of differences are in nofwords we have "data" in d we have datum

"module" is present in d but not in nofwords

Do you have **any questions?**

Thank you so much!

Aya and Ana - Group 6