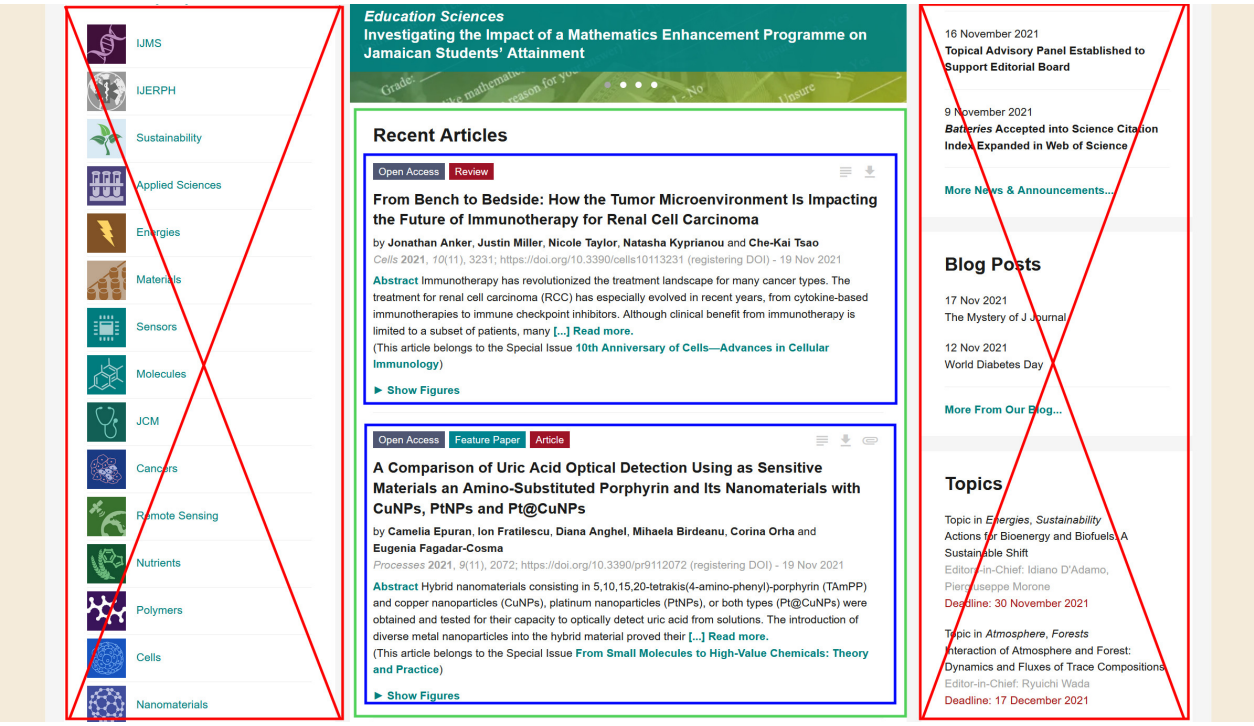
Project 1: Database Creation

Mirudula E., MS18194
Akshay Shankar, MS18117
(Equal contribution)

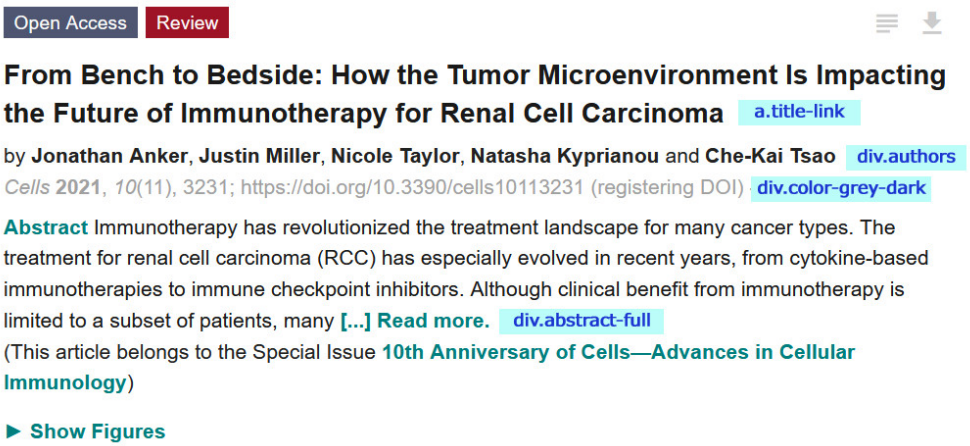
1 Introduction

In this project, we will scrape and parse several entries of specific data features from an online web-server, to construct a MySQL database in our local system using python. We have chosen the website `https://www.mdpi.com/` which maintains a large collection of research articles from various open source journals. The structure of a page is like so:



(a) Outline of the webpage

We require only the content inside the blue boxes, each of which contains the metadata of an article (the content in the red boxes are not relevant to us). Our goal is to scrape the following data from each of these article entries: *Title, Authors, DOI, Journal, Publication date* and *Abstract*. Below we have indicated the relevant CSS selectors to extract this data from the website (identified using the browser inspector).



(a) Relevant CSS selectors to scrape data

---

2 Implementation

2.1 Scraping the data

We begin by requesting the HTML page and parsing it using BeautifulSoup:

```
import requests, re, time # for HTML requests, and regex
from bs4 import BeautifulSoup # for scraping
from datetime import datetime # for date-time manipulation

#pg = page number; kw = keywords; auth = authors
url = "https://www.mdpi.com/search?sort=pubdate&page_no="\
      + str(pg) +"&page_count=15&year_from=1996&year_to=2021&q="\
      + kw + "&authors=" + auth + "&view=default"

p = requests.get(url) #get the html content of the page
s = BeautifulSoup(p.content, "html.parser") #parse the page
```

The raw data is then extracted using the CSS selectors.

```
titles = [title.text for title in s.select("a.title-link")]
raw_authors = [author.text for author in s.find_all("div", {"class": "authors
    "})]
raw_dois = [doi.text for doi in s.find_all("div", {"class": "color-grey-dark"
    })]
raw_abstracts = [abstract.text for abstract in s.find_all("div", {"class": "
    abstract-full"})]
```

2.2 Cleaning the raw data

Now we utilize some regex, and date-time manipulation to clean the data from the HTML extraction. We note that the title, authors and abstract can be easily extracted, with minimal parsing; but the other fields are combined in a single CSS selector which we will have to split up manually.

2.2.1 Title

```
titles[0]
```

```
'The Application of Lean Methods in Corporate Sustainability—A Systematic Literature Review'
```

2.2.2 Authors

```
raw_authors[0]
```

```
'\nby\nFrank Bertagnolli, Kerstin Herrmann, Isabel Rittmann and Tobias Viere '
```

```
def cleanAuthor(auth):
    auth = re.sub("\\nby\\n", "", auth)
    auth = re.sub(" and ", ", ", auth)
    return auth
```

2.2.3 DOI, Date, Journal

```
raw_dois[0]
```

```
'\nSustainability 2021, 13(22), 12786; https://doi.org/10.3390/su132212786 (registering\xa0DOI) - 19 Nov 2021\n'
```

```
url = re.compile(" (https://doi.org/.*?) ")
date = re.compile(" (\d{2} \w{3} \d{4})")

def cleanDOI(doi):
    return url.search(doi).group(1)

def cleanDate(doi):
    date_obj = datetime.strptime(date.search(doi).group(1), '%d %b %Y')
    return date_obj.strftime('%Y-%m-%d')

def cleanJournal(doi):
    journ = re.compile("(.*);")
    return journ.search(doi).group(1)
```

2.2.4 Abstract

```
raw_abstracts[0]
```

```
'\nThis paper reviews the application of lean methods for corporate sustainability and highlights demands for future
research. With the help of a systematic literature review, papers at the interface of lean and sustainability were id
entified and matched to a standardized list of lean methods to assess their frequency in the context of sustainabilit
y. In a further step, papers containing actual case studies were analyzed in more detail regarding specific applicati
on settings, sustainability dimensions, measurability of sustainability impact, and other criteria. The quantitative
analysis of 363 publications shows frequent use of lean methods such as just in time and value stream mapping in the
context of sustainability, and a surprisingly low use of other approaches such as karakuri, milk run, or chaku chaku.
The in-depth analysis of 81 case studies reveals the primacy of intra-company and ecological assessments in the lean
context, while social and inter-company aspects remain rather underexposed. This study complements existing research
on lean and sustainability by systematically analyzing specific lean methods in the context of sustainability and by
further exploring the sustainability characteristics of such lean applications.\nFull article\n'
```

```
def cleanAbstract(abst):
    abst = re.sub("(Full article|\\n)", "", abst)
    return abst
```

2.3 Creating the data entries

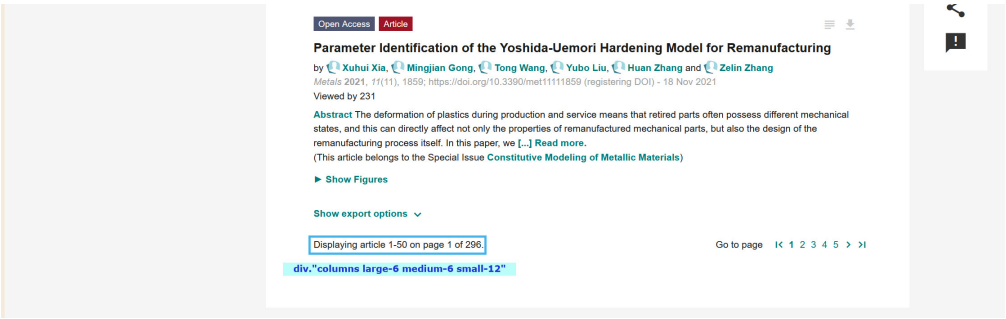
We now clean the data using the subroutines shown above, and construct a list of tuples where each tuple contains the metadata of a single article.

```
authors = [cleanAuthor(author) for author in raw_authors]
dois = [cleanDOI(doi) for doi in raw_dois]
journals = [cleanJournal(doi) for doi in raw_dois]
dates = [cleanDate(doi) for doi in raw_dois]
abstracts = [cleanAbstract(abstract) for abstract in raw_abstracts]

#Create tuples of records (title, authors, journal, date, doi, abstract)
records = []
for i in range(len(titles)):
    records.append((titles[i], authors[i], journals[i], dates[i], dois[i],
    abstracts[i]))
```

2.4 Finding the max. pages of search results (for the final script)

We get the page number from the following CSS selector, and use the script below to clean and extract it from the first page of the results. We then compare it with a user input, max_pg so there is a fixed limit for the pages we scrape.




```
def getMaxPages(max_pg, kw = "", auth = ""):

    #replace space with plus in the queries
    kw = re.sub(" ", "+", kw)
    auth = re.sub("[ ,]", "+", auth)

    #Webscrape the results page for list of articles
    url = "https://www.mdpi.com/search?sort=pubdate&page_no=1"\
          +"&page_count=15&year_from=1996&year_to=2021&q="\
          + kw + "&authors=" + auth + "&view=default"

    print(url)
    p=requests.get(url) #get the html content of the page
    s=BeautifulSoup(p.content, "html.parser") #parse the page

    #get the max page number
    pg = re.compile("of (\d+).")
    pgtext = s.find("div", {"class":"columns large-6 medium-6 small-12"}).
    text

    return min(max_pg, int(pg.search(pgtext).group(1)))
```

2.5 Creating the MySQL database

We will use the mysql-connector package to manipulate the SQL database from python. We have already set up a user "idc409" with password "pwd", and a database "test" in MySQL from the terminal application. A connection is established with the server using these credentials:

```
from mysql.connector import connect

connection = connect(
        host="localhost",
        user="idc409",
        password="pwd", database="test")
```

For ease of use, we have defined the two SQL queries which we will be using:

```
create_table_query = """
CREATE TABLE articles(
    id INT AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(1000),
    authors VARCHAR(1000),
    journal VARCHAR(1000),
    published_date DATE,
    doi VARCHAR(500),
    abstract VARCHAR(5000)
)"""

insert_query = """
INSERT INTO articles
(title, authors, journal, published_date, doi, abstract)
VALUES (%s, %s, %s, %s, %s, %s)
"""
```

The table "articles" is then created, and the data entries from the list of tuples "records" is fed into the table:

```
#failsafe; drop table incase of any mistake
#with connection.cursor() as cursor:
#    cursor.execute("DROP TABLE articles")
#    connection.commit()

#construct table
with connection.cursor() as cursor:
    cursor.execute(create_table_query)
    connection.commit()

#insert data
with connection.cursor() as cursor:
    cursor.executemany(insert_query, records)
    connection.commit()
```

---

3 Results

We have combined all the above snippets into a single script looping over every page (till the max_page) in a jupyter notebook (which is attached at the end). The script is executed for the first few pages of the website results, with no specific authors or keywords:

```
driver(10, kw = "", auth = "")
```

```
https://www.mdpi.com/search?sort=pubdate&page_no=1&page_count=15&year_from=1996&year_to=2021&q=&authors=&view=default
Page 1 completed...
Page 2 completed...
Page 3 completed...
Page 4 completed...
Page 5 completed...
Page 6 completed...
Page 7 completed...
Page 8 completed...
Page 9 completed...
Page 10 completed...
```

Below we show the data that was stored in the database (the abstracts are not shown since they are too long):

```
mysql> select id, title, authors, journal, published_date, doi from articles \G;
*************************** 1. row ***************************
           id: 1
        title: Channel Migration of the Meandering River Fan: A Case Study of the Okavango Delta
      authors: Xue Yan, Jinliang Zhang, Yang Li, Long Sun
      journal: Water 2021, 13(23), 3319
published_date: 2021-11-23
          doi: https://doi.org/10.3390/w13233319
*************************** 2. row ***************************
           id: 2
        title: HPLC-DAD Analysis of Hemp Oil Supplements for Determination of Four Cannabinoids: Cannabidiol, Cannabidiolic Acid, Cannabinol and Delta 9-Tetrahydrocannabinol
      authors: Katarzyna Madej, Aleksandra Chmiołek, Kamila Szlachta, Wojciech Piekoszewski
      journal: Separations 2021, 8(12), 227
published_date: 2021-11-23
          doi: https://doi.org/10.3390/separations8120227
*************************** 3. row ***************************
           id: 3
        title: Tunable Spacing Dual-Wavelength Q-Switched Fiber Laser Based on Tunable FBG Device
      authors: Nurnazifah M. Radzi, Amirah A. Latif, Mohammad F. Ismail, Josephine Y. C. Liew, Noor A. Awang, Han K. Lee, Fauzan Ahmad, Siti F. Norizan, Harith Ahmad
      journal: Photonics 2021, 8(12), 524
published_date: 2021-11-23
          doi: https://doi.org/10.3390/photonics8120524
*************************** 4. row ***************************
           id: 4
        title: Characterization of a Chickpea Mutant Resistant to Phelipanche aegyptiaca Pers. and Orobanche crenata Forsk
      authors: Shmuel Galili, Joseph Hershenhorn, Evgeny Smirnov, Koichi Yoneyama, Xiaonan Xie, Orit Amir-Segev, Aharon Bellalou, Evgenia Dor
      journal: Plants 2021, 10(12), 2552
published_date: 2021-11-23
          doi: https://doi.org/10.3390/plants10122552
*************************** 5. row ***************************
           id: 5
        title: Preclinical Efficacy of Pro- and Anti-Angiogenic Peptide Hydrogels to Treat Age-Related Macular Degeneration
      authors: Amanda Acevedo-Jake, Siyu Shi, Zain Siddiqui, Sreya Sanyal, Rebecca Schur, Simon Kaja, Alex Yuan, Vivek A. Kumar
      journal: Bioengineering 2021, 8(12), 190
published_date: 2021-11-23
          doi: https://doi.org/10.3390/bioengineering8120190
*************************** 6. row ***************************
           id: 6
        title: Perspectives on Molecular Materials—A Tribute to Professor Peter Day
      authors: Lee Martin, Scott S. Turner, John D. Wallis, Hiroki Akutsu, Carlos J. Gómez-García
      journal: Magnetochemistry 2021, 7(12), 152
published_date: 2021-11-23
          doi: https://doi.org/10.3390/magnetochemistry7120152
*************************** 7. row ***************************
           id: 7
        title: Boosting the H2 Production Efficiency via Photocatalytic Organic Reforming: The Role of Additional Hole Scavenging System
      authors: Yamen AlSalka, Osama Al-Madanat, Amer Hakki, Detlef W. Bahnemann
      journal: Catalysts 2021, 11(12), 1423
published_date: 2021-11-23
          doi: https://doi.org/10.3390/catal11121423
*************************** 8. row ***************************
           id: 8
```

We have successfully scraped data from the MDPI web server and created our own local database containing the metadata of articles from various open-source journals!
