Step 1 - Load Libraries

In [1]:
```python
import pdfplumber
import itertools
import json
import re
import spacy
from os import path
import csv
```

Step 2 - Load PDF/ Text Data

In [28]:
```python
data = ""
# filename = "Data//"
# with pdfplumber.open(file_name) as pdf:
#     for index, page in enumerate(pdf.pages):
#         data = data + str(page.extract_text())

# print("Data present in PDF document.")
# print(data)

data = """SECTION 00 3100
AVAILABLE PROJECT INFORMATION
PART 1 - GENERAL
1.1
SUBMITTALS
A.
This Section references other information relevant to the construction of this Project that is
available project information.
B.
At the request of the Owner, the information identified below represents services that have
been provided by others, not as an Architect’s Consultant, regarding conditions that affect this
Project that are beyond the responsibilities of the Architect and Architect’s Consultants.
Reference to such information herein is solely for the convenience of the Owner. Architect
makes no representation, express or implied, as to the accuracy or validity of the information.
C.
Bidders are expected to examine the site and the information available from the Owner to
determine for themselves the conditions to be encountered.
D.
If conditions other than those indicated in the information available from the Owner are
encountered before or during construction, notify the Owner before work continues.
1.2
GEOTECHNICAL REPORT
A.
The Owner’s Geotechnical Consultant has made subsurface borings at the Project site, has
performed an investigation of the geotechnical conditions, and has prepared a report of the
investigation that contains specific requirements of the Contractor.
B.
A copy is being provided as an attachment at the end of this section.
C.
The information was obtained for use in preparing the foundation design, but is indicative only
of the soil conditions where the borings are taken.
D.
The Owner retained the following company: Nova
E.
Date of Report: August 12, 2020
PART 2 - PRODUCTS
2.1
ACTION SUBMITTALS
A.
This Section references other information relevant to the construction of this Project that is
available project information.
B.
At the request of the Owner, the information identified below represents services that have
been provided by others, not as an Architect’s Consultant, regarding conditions that affect this
Project that are beyond the responsibilities of the Architect and Architect’s Consultants.
Reference to such information herein is solely for the convenience of the Owner. Architect
makes no representation, express or implied, as to the accuracy or validity of the information.
C.
Bidders are expected to examine the site and the information available from the Owner to
determine for themselves the conditions to be encountered.
D.
If conditions other than those indicated in the information available from the Owner are
encountered before or during construction, notify the Owner before work continues.
PART 3 - EXECUTION (NOT USED)
END OF SECTION
"""
```

Step 3 - Preprocessing Data

A. Use Custom NER To Extract Section Number, Section Name

In [31]:
```python
start_index = re.search(r'SECTION|DOCUMENT', data).start()
end_index= data.rindex("PART 1")

section_details = data[start_index:end_index]
```

In [32]:
```python
nlp2 = spacy.load("Spacy Custom NER Dump/")

spec_number = ""
spec_name = ""
flag1, flag2 = False, False
section_data = nlp2(section_details)
for sent in section_data.ents:
    if(sent.label_ == 'section_number'):
        spec_number = str(sent)
        flag1 = True
    elif(sent.label_ == 'section_name'):
        spec_name = str(sent)
        flag2 = True
    elif(flag1 and flag2):
        break

if(not flag1):
    spec_number = "NA"

if(not flag2):
    spec_name = "NA"

if(not flag1 and not flag2):
    sect

print("Section Number - {}".format(spec_number))
print("Section Name - {}".format(spec_name))
```

```
Section Number - 00 3100
Section Name - AVAILABLE PROJECT INFORMATION
```

B. Removed Empty Lines and End Of Section/ Document

In [33]:
```python
final_data = ""
head_flag = True
for index, line in enumerate(data.splitlines()):
    if("END OF SECTION" in line or "END OF DOCUMENT" in line):
        continue
    elif(len(line.strip()) == 0):
        continue
    else:
        final_data = final_data + line + "\n"

# print(final_data)
```

C. Find Index of PART

In [34]:
```python
start = 0
for i, l in enumerate(final_data.splitlines()):
    if(l.upper().startswith("PART")):
        start = i
        break

print("PART Starts At - {}".format(start))
```

```
PART Starts At - 2
```

D. Correct Wrong Lines and Mapp Into Final Lines

In [35]:
```python
final_lines = []
index = -1
for line in final_data.splitlines()[start:]:
    if(line.strip().startswith("PART")):
        final_lines.append(line)
        index = index + 1
    elif(re.search(r"^[0-9]\.[0-9]", line)):
        final_lines.append(line)
        index = index + 1
    elif(re.search(r"^[A-Za-z]\.", line)):
        final_lines.append(line)
        index = index + 1
    elif(re.search(r"^[0-9]+\.", line)):
        final_lines.append(line)
        index = index + 1
    elif(re.search(r"^[0-9]+\)", line)):
        final_lines.append(line)
        index = index + 1
    elif(re.search(r"^[a-z]+\)", line)):
        final_lines.append(line)
        index = index + 1
    else:
        final_lines[index] = final_lines[index] + " " + line
```

In [28]:
```python
# print(final_lines)
```

E. Capture All Heading Present In Data

In [36]:
```python
heading = []
flag = True
for line in final_lines:
    if(re.search(r"^[0-9]+\.[0-9]+\s", line) or line.strip().startswith("PART")):
        heading.append(line)
        flag = False
    elif(re.search(r"^[A-Z]\.", line) and flag):
        heading.append(line)

print(heading)
```

```
['PART 1 - GENERAL', '1.1 SUBMITTALS', '1.2 GEOTECHNICAL REPORT', 'PART 2 - PRODUCTS', '2.1 ACTION SUBMITTALS',
'PART 3 - EXECUTION (NOT USED)']
```

F. Capture and Arange Those Heading Which Has SUBMITTAL In It and Create A Pair

In [37]:
```python
res = list(map(list, zip(heading, heading[1:])))
index_data = []
heading_list = []
for i, data in enumerate(res):
    if(i == 0):
        heading_list.append("PART 1 - GENERAL")
    if("SUBMITTAL" in data[0]):
        heading_list.append(data)
    if("PART" in data[1]):
        heading_list.append(data[1])

heading_list
```

Out[37]:
```
['PART 1 - GENERAL',
 ['1.1 SUBMITTALS', '1.2 GEOTECHNICAL REPORT'],
 'PART 2 - PRODUCTS',
 ['2.1 ACTION SUBMITTALS', 'PART 3 - EXECUTION (NOT USED)'],
 'PART 3 - EXECUTION (NOT USED)']
```

G. Create Index List Of Start Index and End Index Of Submittal Headinga Including PART

In [38]:
```python
data_lines = []
if(len(heading_list) == 0):
    final_lines = []
else:
    for item in heading_list:
        if("SUMMITAL" in item or "SUBMITTAL" in item[0] or "SUBMITTALS" in item[0]):
            x, y = final_lines.index(item[0]), final_lines.index(item[1])
            data_lines.append((x, y))
        elif("PART " in item):
            data_lines.append(item)

print(data_lines)
```

```
['PART 1 - GENERAL', (1, 6), 'PART 2 - PRODUCTS', (13, 18), 'PART 3 - EXECUTION (NOT USED)']
```

H. Generate Final Lines For Mapping Into Dictionary

In [11]:
```python
dataset = []
for pos in data_lines:
    if("PART" in pos):
        dataset.append(pos)
    else:
        for ll in range(pos[0], pos[1]):
            dataset.append(final_lines[ll])

# print(dataset)
```

```
['PART 1 - GENERAL', '1.1 SUBMITTALS', 'A. This Section references other information relevant to the constructi
on of this Project that is available project information.', 'B. At the request of the Owner, the information id
entified below represents services that have been provided by others, not as an Architect’s Consultant, regardi
ng conditions that affect this Project that are beyond the responsibilities of the Architect and Architect’s Co
nsultants. Reference to such information herein is solely for the convenience of the Owner. Architect makes no
representation, express or implied, as to the accuracy or validity of the information.', 'C. Bidders are expect
ed to examine the site and the information available from the Owner to determine for themselves the conditions
to be encountered.', 'D. If conditions other than those indicated in the information available from the Owner a
re encountered before or during construction, notify the Owner before work continues.', 'PART 2 - PRODUCTS',
'2.1 ACTION SUBMITTALS', 'A. This Section references other information relevant to the construction of this Pro
ject that is available project information.', 'B. At the request of the Owner, the information identified below
represents services that have been provided by others, not as an Architect’s Consultant, regarding conditions t
hat affect this Project that are beyond the responsibilities of the Architect and Architect’s Consultants. Refe
rence to such information herein is solely for the convenience of the Owner. Architect makes no representation,
express or implied, as to the accuracy or validity of the information.', 'C. Bidders are expected to examine th
e site and the information available from the Owner to determine for themselves the conditions to be encountere
d.', 'D. If conditions other than those indicated in the information available from the Owner are encountered b
efore or during construction, notify the Owner before work continues.', 'PART 3 - EXECUTION (NOT USED)']
```

Step 4 - Map Into Dictionary

In [39]:
```python
## Map Text Data Into Dictionary
#
dictionary = {}
part_name = ""
cnt = 0
cnt2 = 0

for index, line in enumerate(itertools.chain(final_data.splitlines()[start : start + 1], dataset)):
    if(index == 0):
        dictionary["SECTION"] = spec_number
        head = "SECTION_NAME"
        dictionary[head] = spec_name
        head = "Submittals"
        dictionary[head] = []
        if(line.startswith("PART ")):
            part_name = line
    elif(line.startswith("PART ")):
        part_name = line
    elif(re.search(r"^[0-9]+\.[0-9]+", line)):
        subsection_name = line.split()[0]
        subsection_heading = " ".join(line.split()[1:])
    elif(re.search(r"^[A-Z]\.", line.strip())):
        dictionary[head].append({"Part" : part_name})
        dictionary[head][cnt]["Sub Section"] = subsection_name + " " + line.split(".")[0]
        dictionary[head][cnt]["Sub Section Heading "] = subsection_heading
        try:
            dictionary[head][cnt]["Submittal Type "] = line.split(":")[0].split(".")[1]
            if(len(line.split(":")[1].strip()) == 0):
                dictionary[head][cnt]["Submittal Description"] = []
            else:
                dictionary[head][cnt]["Submittal Description"] = [" ".join(line.split(":")[1:])]
        except Exception as e:
            dictionary[head][cnt]["Submittal Type "] = line.split(".")[0]
            dictionary[head][cnt]["Submittal Description"] = [line.strip()]
        cnt2 = cnt
        cnt = cnt + 1
    elif(re.search(r"^[0-9]+\.", line.strip())):
        try:
            dictionary[head][cnt2]["Submittal Description"].append(line.strip())
        except Exception as e:
            cnt2 = cnt
            cnt = cnt + 1
            dictionary[head].append({"Part" : part_name})
            dictionary[head][cnt2]["Sub Section"] = subsection_name
            dictionary[head][cnt2]["Sub Section Heading "] = subsection_heading
            dictionary[head][cnt2]["Submittal Type "] = "NA"
            dictionary[head][cnt2]["Submittal Description"] = [line.strip()]
    elif(re.search(r"^[a-z]\.", line.strip())):
        dictionary[head][cnt2]["Submittal Description"].append(line.strip())
    elif(re.search(r"^[0-9]+\)", line.strip())):
        dictionary[head][cnt2]["Submittal Description"].append(line.strip())
    elif(re.search(r"^[a-z]+\)", line.strip())):
        dictionary[head][cnt2]["Submittal Description"].append(line.strip())
    elif(len(line.strip()) > 0):
        dictionary[head][cnt2]["Submittal Description"].append(line.strip())
    else:
        pass


print(dictionary)
```

```
{'SECTION': '00 3100', 'SECTION_NAME': 'AVAILABLE PROJECT INFORMATION', 'Submittals': [{'Part': 'PART 1 - GENER
AL', 'Sub Section': '1.1 A', 'Sub Section Heading ': 'SUBMITTALS', 'Submittal Type ': 'A', 'Submittal Descripti
on': ['A. This Section references other information relevant to the construction of this Project that is availa
ble project information.']}, {'Part': 'PART 1 - GENERAL', 'Sub Section': '1.1 B', 'Sub Section Heading ': 'SUBM
ITTALS', 'Submittal Type ': 'B', 'Submittal Description': ['B. At the request of the Owner, the information ide
ntified below represents services that have been provided by others, not as an Architect’s Consultant, regardin
g conditions that affect this Project that are beyond the responsibilities of the Architect and Architect’s Con
sultants. Reference to such information herein is solely for the convenience of the Owner. Architect makes no r
epresentation, express or implied, as to the accuracy or validity of the information.']}, {'Part': 'PART 1 - GE
NERAL', 'Sub Section': '1.1 C', 'Sub Section Heading ': 'SUBMITTALS', 'Submittal Type ': 'C', 'Submittal Descri
ption': ['C. Bidders are expected to examine the site and the information available from the Owner to determine
for themselves the conditions to be encountered.']}, {'Part': 'PART 1 - GENERAL', 'Sub Section': '1.1 D', 'Sub
Section Heading ': 'SUBMITTALS', 'Submittal Type ': 'D', 'Submittal Description': ['D. If conditions other than
those indicated in the information available from the Owner are encountered before or during construction, noti
fy the Owner before work continues.']}, {'Part': 'PART 2 - PRODUCTS', 'Sub Section': '2.1 A', 'Sub Section Head
ing ': 'ACTION SUBMITTALS', 'Submittal Type ': 'A', 'Submittal Description': ['A. This Section references other
information relevant to the construction of this Project that is available project information.']}, {'Part': 'P
ART 2 - PRODUCTS', 'Sub Section': '2.1 B', 'Sub Section Heading ': 'ACTION SUBMITTALS', 'Submittal Type ': 'B',
'Submittal Description': ['B. At the request of the Owner, the information identified below represents services
that have been provided by others, not as an Architect’s Consultant, regarding conditions that affect this Proj
ect that are beyond the responsibilities of the Architect and Architect’s Consultants. Reference to such inform
ation herein is solely for the convenience of the Owner. Architect makes no representation, express or implied,
as to the accuracy or validity of the information.']}, {'Part': 'PART 2 - PRODUCTS', 'Sub Section': '2.1 C', 'S
ub Section Heading ': 'ACTION SUBMITTALS', 'Submittal Type ': 'C', 'Submittal Description': ['C. Bidders are ex
pected to examine the site and the information available from the Owner to determine for themselves the conditi
ons to be encountered.']}, {'Part': 'PART 2 - PRODUCTS', 'Sub Section': '2.1 D', 'Sub Section Heading ': 'ACTIO
N SUBMITTALS', 'Submittal Type ': 'D', 'Submittal Description': ['D. If conditions other than those indicated i
n the information available from the Owner are encountered before or during construction, notify the Owner befo
re work continues.']}]}
```

Step 5 - Generate JSON/CSV File

A. Create CSV File

In [40]:
```python
## Create CSV output from dictionary
#
def create_csv_output(dictionary1, big_spec_name, spec_number, spec_name):
    headlines = ['SECTION', 'SECTION_NAME', 'PART', 'SUB SECTION', 'SUB SECTION HEADING', 'TYPE','DECRIPTION']

    big_spec_name = big_spec_name + ".csv"
    file_status = path.exists(big_spec_name)

    with open(big_spec_name, 'a', encoding = 'UTF8', newline = '') as file:
        writer = csv.writer(file)
        if(not file_status):
            writer.writerow(headlines)
        for key, item in dictionary1.items():
            if(isinstance(item, list)):
                for dicti in item:
                    writer.writerow([spec_number, spec_name, dicti['Part'], dicti['Sub Section'], dicti['Sub Se
```

In [41]:
```python
## Function Call - Generate JSON Output With Data
#
# create_json_output(dictionary, 1, spec_number, spec_name, file_name)

dictionary1 = dictionary

big_spec_name = "YYY"
spec_number = spec_number
spec_name = spec_name
## Function Call - Generate CSV Output With Data
#
create_csv_output(dictionary1, big_spec_name, spec_number, spec_name)
```

B. Create JSON File

In [42]:
```python
output_file = str(big_spec_name) + ".json"

with open(output_file, "w", encoding = 'utf-8') as outfile:
    json.dump(dictionary, outfile, indent = 4, ensure_ascii = False)
```

END - Convert Notebook to Python, PDF

In [2]:
```python
# !jupyter nbconvert --to script "submittal_extraction_v9.ipynb"
```

```
[NbConvertApp] Converting notebook submittal_extraction_v8.ipynb to script
[NbConvertApp] Writing 10120 bytes to submittal_extraction_v8.py
```

In [27]:
```python
!jupyter nbconvert --to PDFviaHTML "submittal_extraction_v9.ipynb"
```

```
[NbConvertApp] Converting notebook submittal_extraction_v9.ipynb to PDFviaHTML
[NbConvertApp] Writing 189688 bytes to submittal_extraction_v9.pdf
```

Banned Zone - Testing

In [ ]: