Updated the code in **test.py** file

python test.py --i ca\_clayton\_2022.xlsx --o ca\_clayton\_2022.html

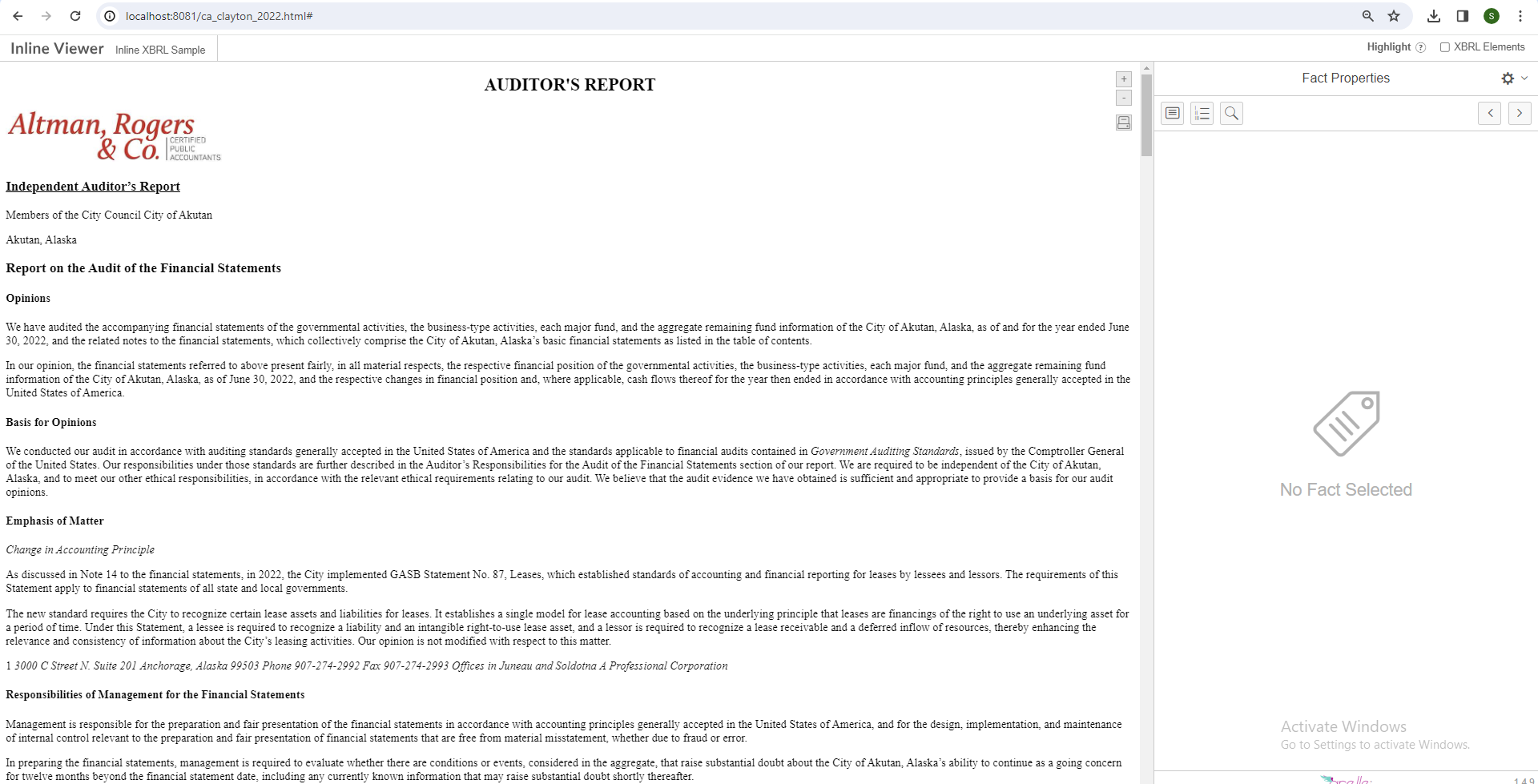
This above line is used to run the script .

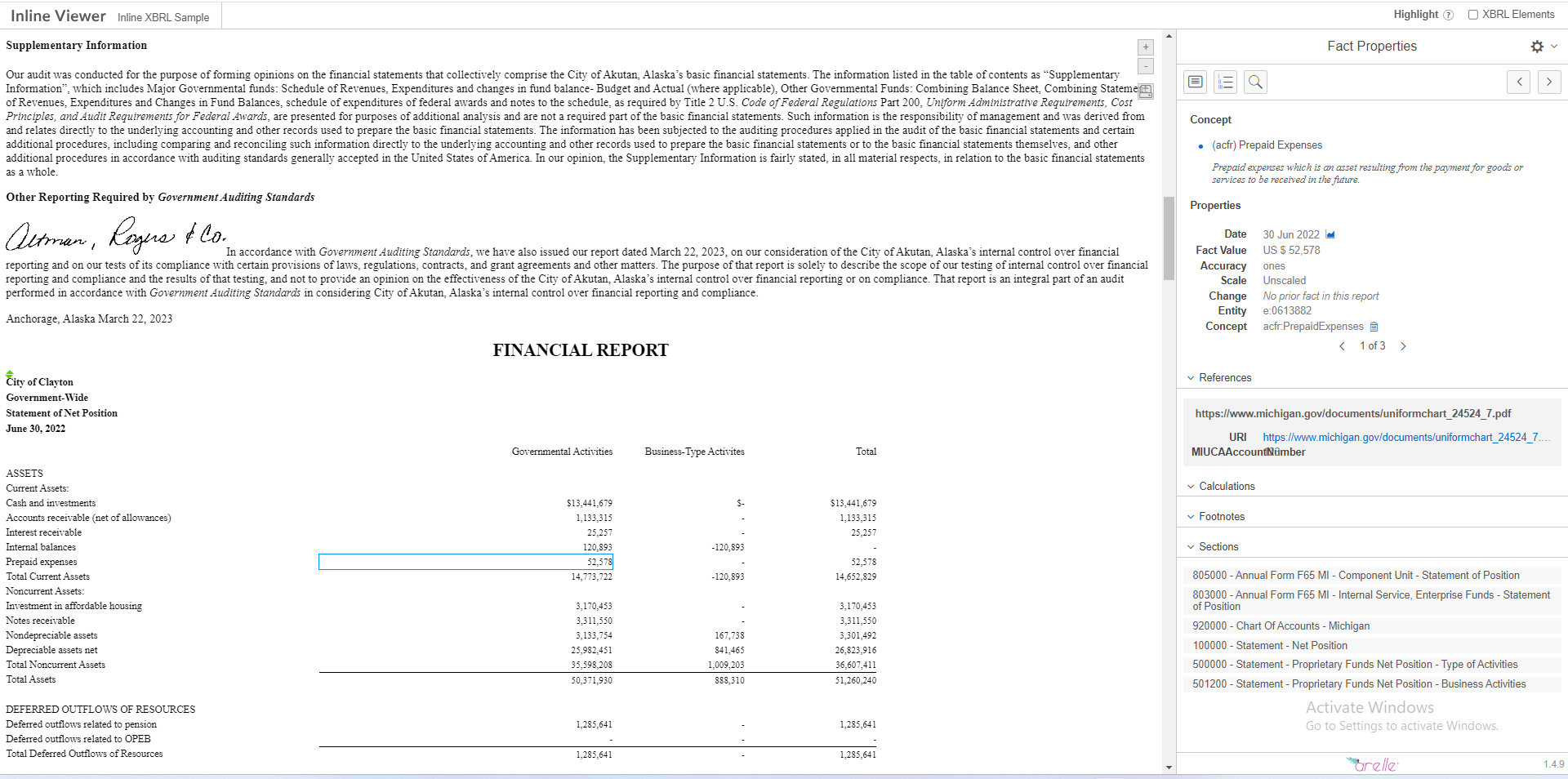
**Input files:**

1. **ca\_clayton\_2022.xlsx** - Excel file which has the financial reports
2. **Xbrl Independent Auditor.docx** - Sample word Document
3. **contexts.xlsxf**

**Expected Output:**

This will append the word content in the xbrl format

****

****

[give the sample input file]

word\_file\_path = 'Xbrl Independent Auditor.docx'

from docx2python import docx2python

def extract\_text\_from\_docx(file\_path):

[Extract the content using docx2python library]

doc\_html\_result = docx2python(file\_path, html = True)

text\_content = ""

for page in doc\_html\_result.body[0][0][0]:

for paragraph in page:

for element in paragraph:

if not isinstance(element, dict) or ('type' in element and element['type'] != 'footer'):

if isinstance(element, str):

text\_content += element

text\_content += '\n'

doc\_html\_result.close()

return text\_content

word\_text = extract\_text\_from\_docx(word\_file\_path)

html\_in += soup.prettify("utf-8").decode("utf-8")

[Removed the unusual font size in the word content]

word\_text = re.sub(r'style="[^"]\*"', '', word\_text)

[Added the function for insert the line spacing for the paragraph content]

def add\_paragraph\_tags(html\_input):

soup = BeautifulSoup(html\_input, 'html.parser')

for line in soup.find\_all(string=True):

if line.strip() and line.parent.name not in ['h3','h4','span','b','u','i']:

paragraphs = line.strip().split('\n') # Split text into paragraphs based on double newlines

for paragraph in paragraphs:

p\_tag = soup.new\_tag("p")

p\_tag.string = paragraph.strip()

line.insert\_before(p\_tag) # Insert the <p> tag before the current line

line.insert\_before("\n") # Add a newline after the <p> tag for formatting

line.extract()

return str(soup).replace('</p><i>','<i>').replace('</p>\n<i>','\n<i>').replace('</i><p>','</i>')

word\_text = add\_paragraph\_tags(word\_text)

from docx import Document

[Used this Document library to get the image from the word document]

def extract\_images(docx\_file):

doc = Document(docx\_file)

images = []

for rel in doc.part.rels.values():

if "image" in rel.reltype:

image\_data = rel.target\_part.blob

image\_name = rel.target\_ref[:] # Get the image name

images.append((image\_name, image\_data))

return images

images = extract\_images(word\_file\_path)

for idx, (image\_name, image\_data) in enumerate(images):

with open(f"{image\_name}", "wb") as f:

f.write(image\_data)

[ Used this function for replace the image with image name ]

def replace\_image\_placeholders(content):

pattern = r'----(.\*?)----'

replaced\_content = re.sub(pattern, r'<img src="\1" style="width:300px;" />', content)

return replaced\_content

word\_text = replace\_image\_placeholders(word\_text)

[ added a header “**AUDITOR'S REPORT”** and the word content]

ix\_header = conf.ix\_header\_start + ix\_header\_content + conf.ix\_header\_end + '<ix:nonNumeric> \n <h2 align="center"> AUDITOR\'S REPORT </h2> '+ word\_text+'</ix:nonNumeric>'

[ added a header **“FINANCIAL REPORT”** before the excel content]

html\_out = conf.new\_header + '\n' + ix\_header.replace("$place\_id$", conf.place\_id) + '\n<ix:nonNumeric> \n <h2 align="center"> FINANCIAL REPORT </h2> </ix:nonNumeric>'

for line in html\_in.splitlines():

html\_out = html\_out + line + '\n'

html\_out = html\_out +'</body></html>'

html\_out = html\_out.replace("&lt;", "<")

html\_out = html\_out.replace("&gt;", ">")

html\_out = html\_out.replace("xbrli:startdate", "xbrli:startDate")

html\_out = html\_out.replace("xbrli:enddate", "xbrli:endDate")

os.remove("temp0")

os.remove("temp")

encoded\_html\_out = html\_out.encode('utf-8')

decoded\_html\_out = encoded\_html\_out.decode('utf-8') # Decode the encoded string using UTF-8 encoding

[ The html file (Word content + Excel content) will be saved as output\_file (--o file\_name which we gave while run the script) ]

with open(output\_file, 'w', encoding='utf-8') as f:

f.write(decoded\_html\_out)

print(f"Successfully converted to {output\_file}")