```
import requests
from bs4 import BeautifulSoup, SoupStrainer
import pandas as pd

cwf = "/capital-works-fund"
root = "https://www.schoolbuildings.vic.gov.au"

# METHOD 1
# select list items underneath div of given class
# list items don't need to be directly underneath
# https://beautiful-soup-4.readthedocs.io/en/latest/#css-selectors

# for a in soup.select("div .tide-grant__overview-item li"):
#    print(a)

# METHOD 2
# progressively drill down, using 'find'
# https://www.slingacademy.com/article/handling-nested-tags-and-complex-html-structures-with-beautiful-soup/#using-css-selectors

# div1 = soup.find("div", class_="tide-grant__overview-item")
# div2 = div1.find("div", class_="rpl-content")
# ul = div2.find("ul")
# li = ul.find_all("li a")

# METHOD 3
# traverse down nested tags, some with classes, in one hit
# div = soup.select("div.tide-grant__overview-item div.rpl-content ul li a")
# for a in div:
#    print(a.get("href"))

r = requests.get(root + cwf).text

soup = BeautifulSoup(r, "lxml")

anchors = soup.select("div.tide-grant__overview-item div.rpl-content ul li a")

names = [a.string for a in anchors]
hrefs = [a.get("href") for a in anchors]
projects = []
costs = []

only_cwf = SoupStrainer("div", id="capital-works-fund")

for href in hrefs:
    r = requests.get(root + href).text
    soup = BeautifulSoup(r, "lxml", parse_only=only_cwf)

    proj_para = soup.select_one("div.rpl-content > p")
    projects.append(proj_para.get_text().split(".")[0].replace("We are ", ""))

    cost_para = soup.select_one("div.rpl-callout > div.rpl-content > p")
    costs.append(cost_para.get_text().split("$")[-1].strip("."))

df = pd.DataFrame({
        "school": names,
        "href": hrefs,
        "project": projects,
        "cost": costs
        })

count = len(df[df.project.str.contains("toilets")])

df.to_csv("../vic-capital-works/parsed_projects.tsv", sep="\t")
```