Веб скрапинг буюу веб хуудаснаас хайчилбар аван өгөгдөл цуглуулах

Г.Махгал

2025 оны 11-р сарын 25

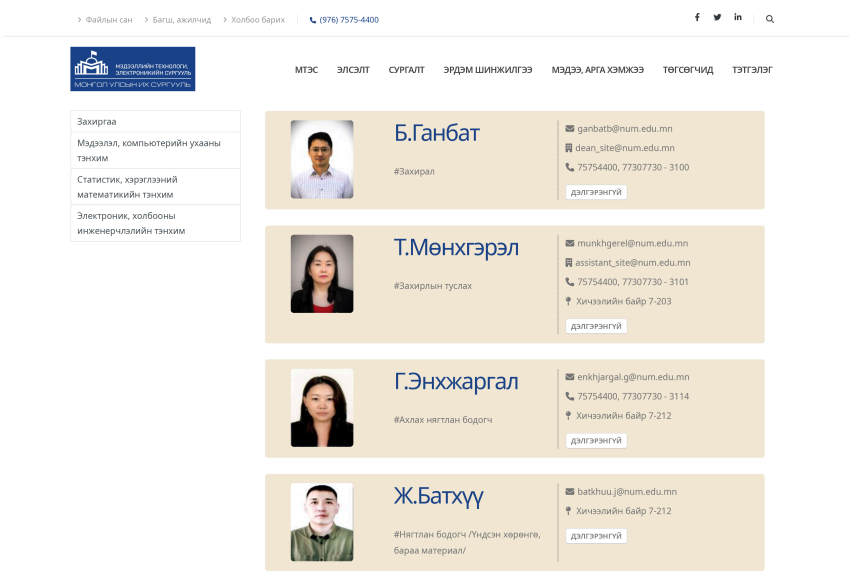
Хураангуй Веб хуудас буюу HTML форматтай документаас хэрэгтэй өгөгдлөө гарган авахыг веб скрапинг гэдэг. Веб хуудсыг сонингийн цаас мэтээр төсөөлж болох бөгөөд веб хуудас дээрээс хэрэгтэй мэдээллээ компьютерийн програм ашиглан автоматаар гаргаж авахыг веб скрапинг гэсэн нь сонин уншигчид түүн дээрх өөрт хэрэгтэй нийтлэлийг хайчлан авдагтай зүйрлэсэн нэр томъёо юм. Энэхүү төслөөр өгөгдөл цуглуулах үндсэн аргуудын нэг болоод буй веб скрапингийн талаарх ерөнхий ойлголт, үүнийг Python хэл ашиглан хэрхэн хэрэгжүүлэх тухай товч заавар ба практикийн бодит кейс зэргийг авч үзнэ. Мөн HTML-тэй төстэй XML форматтай документаас өгөгдөл ялгаж авах асуудал хөндөгдсөн. Түүнчлэн веб скрапинг ажлын үеэр баримтлах зарчимтай танилцуулаад улмаар тус зарчмыг хэрхэн сахин мөрдөж буйг үлгэрчлэн харуулна.

Агуулга

1 Бодит кейс буюу вебээс ялгаж авах өгөгдөл

Удиртгалд дурдсанчлан тодорхой жишээ буюу бодит веб сайтаас скапинг хийнэ. Ингээд МУИС-ийн Мэдээллийн технологи, электроникийн сургуулийн багш нар болон ажилчдын мэдээллийг [1] авч үзье. Тус веб хуудас https://site.num.edu.mn/staffs хаягтай. Веб хуудасны харагдацыг Зураг 1 дээрээс харж болно.

Зураг 1: Скрапинг хийх веб хуудасны харагдац



Тус веб хуудаснаас багш, ажилчдын нэр, албан тушаал, имэйл, утас, ажлын өрөө зэрэг мэдээллийг “хайчлан” авч болох нь харагдана.

2 Веб хуудасны DOM загвар

Веб хуудсыг HTML [2] хэлээр бичдэг. Энэ нь `<html>` гэж эхлээд `</html>` гэж дуусна. Үүн шиг элементийг тиг гэнэ. Хуудасны гол агуулга `<body></body>` тиг дотор байрлана. Мэдээж веб хуудас дотор ямар мэдээлэл орох бас мэдээллийг хэрхэн зохион байгуулахаас шалтгаалж үндсэн `<body></body>` тиг дотор `<div></div>`, `<p></p>`, `<img>` гэх мэтчилэн тигүүдийг тухайн зохион байгуулалтын дагуу ашиглана. Энд `<p></p>` параграф, `<img>` зураг оруулах зориулалттай.

Эдгээр элементүүд нь хоорондоо багталцсан буюу салбарласан мод мэт бүтэц үүсгэнэ. Энэ бүтэц нь веб хөтчөөр уншигдах үед Document Object Model (DOM) [3] гэдэг загвар болж хувирдаг. DOM нь HTML документыг мод хэлбэрээр төлөөлж, програмчлалын аргаар (жишээлбэл веб хөтөч дээр JavaScript хэлээр эсвэл веб скрапингийн үед Python хэлний BeautifulSoup, lxml гэх мэт сангаар) документын агуулга болон бүтэцтэй нь харьцах боломж олгодог.

Ийм DOM бүтэцтэй байдал нь веб хуудасны элементүүд рүү (selector ашиглан) хандаж улмаар мэдээллийг нь ялган авах буюу веб скрапинг хийх боломжийг бүрдүүлдэг.

Түүнчлэн HTML элемент буюу тиг нь `id`, `class` зэрэг аттрибуттай байж болдог. Тиг тодорхой аттрибуттай бол скрапинг хялбарчлагддаг. Тухайлбал `id` таг HTML хэлний дүрмээр цор ганц

байх тул веб хуудасны тодорхой хэсгийг барьж авах найдвартай барьц болдог. Жишээлбэл Зураг 1 дээр харагдаж буй ажилчдын нэрс бүхий багана орчмын DOM загварын бүтцийг ажиглавал `<div class="col-md-12" id="emps">` буюу `emps` гэсэн `id` аттрибуттай тиг олдоно. Тэгэхээр ажилчдын мэдээллийг тус тигээс цааш салбарласан элементүүдээс хайж олно. Веб хуудас дээрх тодорхой хэсгийн элементүүдийн аттрибут болон DOM бүтцийг харахдаа компьютер дээр ажиллах веб хөтөч ашиглана. Веб хөтчид тухайн веб хуудсыг ачаалж байгаад хүссэн хэсэгтээ хулганын курсорыг аваачиж байгаад хулганын баруун товчлуур дээр товшино. Товчлуур дарахад нээгдэх цэсээс “Inspect” гэснийг сонгож товшино. Нээгдэх цонхоос HTML документын аль элемент хаана, ямар харагдацтай байгааг харж нэгжсээр хүссэн элемент дээрээ очно. Хэрэв тус элемент `id` зэрэг аттрибуттай бол тэр нь харагдаж байх болно.

3 Веб скрапинг хийхэд баримтлах зарчим

Веб скрапинг хийхдээ хууль, ёс зүйн болон техникийн тодорхой хэм хэмжээнд ажиллавал зүйтэй [4].

- Мэдээлэл тархаах зориулалттай API байвал түүнийг нь ашиглах
- Сайтын robots.txt файлыг шалгаж, заасан хоригийг үл зөрчих
- Хувийн мэдээлэл, хамгаалагдсан хэсэг рүү хандахгүй байх
- Нэг дор хэт олон хүсэлт илгээхгүй байх

3.1 API

Манай кейсийн хувьд API алга байна.

3.2 robots.txt

Веб сайтын robots.txt файлыг https://site.num.edu.mn/robots.txt гэж нээн үзвэл дараах агуулга гарч ирэв.

```
User-agent: *
Disallow:
```

Энэ нь robots.txt [5] дэх дүрэм `User-agent: *` буюу бүх роботуудад хамаатай бөгөөд `Disallow:` буюу тус талбар хоосон байгаа нь ямар ч зам руу хандан орж болохыг зөвшөөрсөнийг илтгэнэ. Өөрөөр хэлбэл скрапинг хийхэд хязгаарлалт байхгүй гэжээ.

3.3 Хувийн мэдээлэл, хамгаалагдсан хэсэг

Веб скрапингийг хийхдээ admin зэрэг хаалттай, хувийн мэдээллийн хэсэг рүү нэвтрэхгүй.

Түүнчлэн энэ веб скрапингийн тайлан болон үүсэх файлуудад зөвхөн веб сайтын скрийншот бас веб скрапинг хийж буй үйлдлийн төсөөллийг тодорхой болгоход шаардлагатай HTML/XML DOM загварын бүтцийн зарим хэсгийг л харуулна. Өөрөөр хэлбэл веб скрапинг хийн гаргаж авсан бүрэн хэмжээний өгөгдлийг файлд хадгалан, ашиглахад шууд бэлэн болгож тархаахгүй.

3.4 Нэг дор илгээх хүсэлтийн тоо

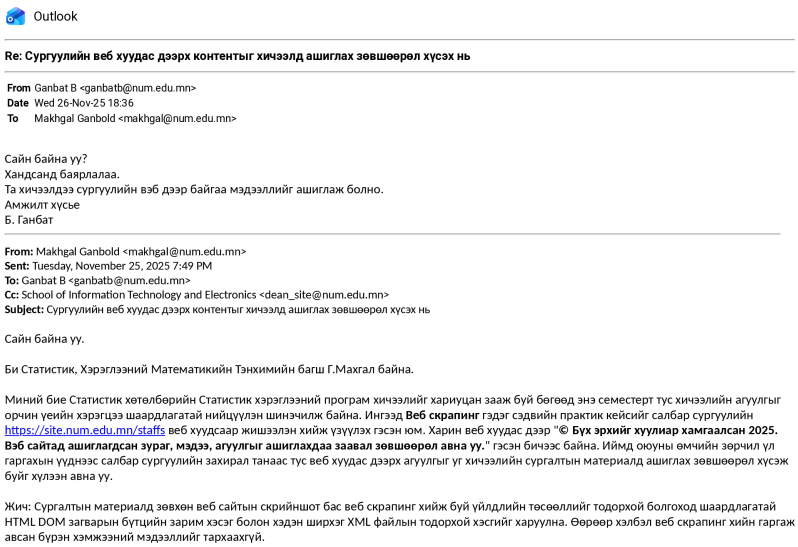
DOS халдлага [6] мэт веб серверийг ачаалахгүйн нэг дор илгээх хүсэлтийн тоог анхаарна. Иймээс кэш хувьсагч ашиглах зэргээр өмнө дуудаж авсан мэдээллийг ахин дуудахаас

зайлсхийнэ. Мөн шаардлагагүй хүсэлт илгээхгүй. Веб скрапинг ажлын програмын кодын гол хэсгийг бичсэний дараа серверт илгээх хүсэлтүүдийг бүртгэхэд 5 удаагийн дэс дараалсан хүсэлт гарсан. Харин төслийн ажлын бүх кодыг бүрэн ажиллуулбал 7 хүсэлт илгээж байна.

3.5 Оюуны өмчийн зөрчлөөс зайлсхийх

Веб скрапинг хийхээр сонгосон веб хуудас дээр оюуны өмчийн “© Бүх эрхийг хуулиар хамгаалсан 2025. Вэб сайтад ашиглагдсан зураг, мэдээ, агуулгыг ашиглахдаа заавал зөвшөөрөл авна уу” гэсэн бичээс байсан тул веб скрапинг хийхээс өмнө зохих албан тушаалтнаас зөвшөөрөл авсан. Зөвшөөрсөн хариу бүхий имэйлийг Зураг 2 дээрээс харж болно.

Зураг 2: Веб скрапинг хийх зөвшөөрөл



Outlook

Re: Сургуулийн веб хуудас дээрх контентыг хичээлд ашиглах зөвшөөрөл хүсэх нь

From Ganbat B <ganbatb@num.edu.mn>
Date Wed 26-Nov-25 18:36
To Makhgal Ganbold <makhgal@num.edu.mn>

Сайн байна уу?
Хандсанд баярлалаа.
Та хичээлдээ сургуулийн вэб дээр байгаа мэдээллийг ашиглаж болно.
Амжилт хүсье
Б. Ганбат

From: Makhgal Ganbold <makhgal@num.edu.mn>
Sent: Tuesday, November 25, 2025 7:49 PM
To: Ganbat B <ganbatb@num.edu.mn>
Cc: School of Information Technology and Electronics <dean_site@num.edu.mn>
Subject: Сургуулийн веб хуудас дээрх контентыг хичээлд ашиглах зөвшөөрөл хүсэх нь

Сайн байна уу.

Би Статистик, Хэрэглээний Математикийн Тэнхимийн багш Г.Махгал байна.

Миний бие Статистик хөтөлбөрийн Статистик хэрэглээний програм хичээлийг хариуцан зааж буй бөгөөд энэ семестерт тус хичээлийн агуулгыг орчин үеийн хэрэгцээ шаардлагатай нийцүүлэн шинэчилж байна. Ингээд **Веб скрапинг** гэдэг сэдвийн практик кейсийг салбар сургуулийн https://site.num.edu.mn/staffs веб хуудсаар жишээлэн хийж үзүүлэх гэсэн юм. Харин веб хуудас дээр "**© Бүх эрхийг хуулиар хамгаалсан 2025. Вэб сайтад ашиглагдсан зураг, мэдээ, агуулгыг ашиглахдаа заавал зөвшөөрөл авна уу.**" гэсэн бичээс байна. Иймд оюуны өмчийн зөрчил үл гаргахын үүднээс салбар сургуулийн захирал танаас тус веб хуудас дээрх агуулгыг үг хичээлийн сургалтын материалд ашиглах зөвшөөрөл хүсэж буйг хүлээн авна уу.

Жич: Сургалтын материалд зөвхөн веб сайтын скрийншот бас веб скрапинг хийж буй үйлдлийн төсөөллийг тодорхой болгоход шаардлагатай HTML DOM загварын бүтцийн зарим хэсэг болон хэдэн ширхэг XML файлын тодорхой хэсгийг харуулна. Өөрөөр хэлбэл веб скрапинг хийн гаргаж авсан бүрэн хэмжээний мэдээллийг тархаахгүй.

4 Веб скрапинг хэрэгжүүлэлт

Веб скрапинг ажлын эхэнд зайлшгүй хийх үйлдэл бол HTML документыг татаж авах явдал юм. Үүнд *requests* модул ашиглана.

```
import requests

url = "https://site.num.edu.mn/staffs"

response = requests.get(url)
html = response.text
```

4.1 JavaScript үл шаардах веб скрапинг

requests модул ашиглан HTML хуудас татаж авна. Харин HTML документыг DOM загвараар хөрвүүлэхэд *BeautifulSoup* сан ашиглана. Мөн HTML хэлний дүрмээр задлан ялгал хийх хэрэгсэл шаардлагатай. Ийм *html.parser* гэдэг Python стандарт сан байдаг ч *lxml* зэрэг илүү

сайн задлан ялгагч ашиглахыг зөвлөдөг. Эдгээрийг `pip install requests beautifulsoup4 lxml` тушаалаар суулгана.

```
from bs4 import BeautifulSoup

soup = BeautifulSoup(html, "lxml")
```

Ийнхүү HTML документыг DOM загвараар хөрвүүлэн боловсруулсаны дараа веб скрапингийн гол цөм болох өгөгдөл "хайчилж" авах ажил эхэлнэ. Жишээлэн элементийг `id` аттрибутаар нь хайн олох кодыг дор бичив.

```
# units гэсэн id аттрибуттай элементийг хайж олох
units = soup.find(id="units")
print(units)
```

Үүнээс гадна элементийн нэр, классын нэр зэргээр хайж олох арга бий. Эдгээр үйлдлийг `soup.find("p")`, `soup.find_all("div")`, `soup.find_all("div", class_="members")` гэх мэтчилэн гүйцэтгэнэ. Цаашилбал сонгосон элементийн тексtийг `item.text.strip()`, аттрибутыг `img["src"]` гэж авна.

Дээрх кодыг ажиллуулбал дараах үр дүн гарна.

```
<div class="col-md-3" id="units">
</div>
```

Өөрөөр хэлбэл элемент хоосон байна. Гэвч инспект хийхэд тус `<div></div>` элемент хоосон байгаагүй. Энэ нь HTML документын агуулгын зарим хэсгийг DOM хөрвүүлж дууссаны дараа JavaScript ажиллуулан ачаалсаны шинж байж болно. Ингээд илүү няхуур инспект хийтэл ажилчдын нэрс ч JavaScript код ажилласнаар ачаалагдаж байгаа нь мэдэгдлээ. Тэгэхээр JavaScript дэмждэг веб скрапинг хэрэглүүр ашиглах хэрэгтэй.

4.2 JavaScript шаардлагатай веб скрапинг

Веб скрапингаар авах мэдээлэл нь HTML документыг DOM загварт хөрвүүлсэний дараа JavaScript код ажиллахад веб серверээс дуудагдан ачаалагддаг бол JavaScript дэмждэг веб скрапинг хийнэ.

JavaScript ажиллуулан веб скрапинг хийхэд ашиглаж болох модул гэвэл *playwright* байна. Үүнийг `pip install playwright` тушаалаар нэмж суулгана. Мөн энэ нь веб хөтөч дээр суурилж ажиллах тул түүнийг нь `playwright install` тушаалаар суулгана. Тус тушаалаар Chromium, Firefox, WebKit зэрэг веб хөтчийн бинар суулгац татагдан автоматаар сууна.

```
import asyncio
from playwright.async_api import async_playwright

async def scrap(url):
  async with async_playwright() as p:
    browser = await p.chromium.launch(headless=True)
    page = await browser.new_page()
```

```
    await page.goto(url)
    html = await page.content()
    await browser.close()
    return html

html = await scrap("https://site.num.edu.mn/staffs")
```

4.3 HTML документыг үзэмжтэй болгох

HTML документын бичиглэл хүн уншихад төвөгтэй байлаа. Өөрөөр хэлбэл `<div>` гэж эхэлсэн элемент чухам хаана `</div>` гэж хаагдаж буйг харахад бэрхшээлтэй байна. Хэрэв элементийн эхлэл төгсгөл бүрийг нэг мөрд бичиж бас эдгээр нь цогц нэг элемент тул мөрийн эхнээс ижил зай бүхий догол мөрд жигдлэн бичвэл HTML документыг уншихад хялбар болдог. Ийнхүү үзэмжтэй болгоход *BeautifulSoup* классын `prettify()` функц ашиглана.

```
from bs4 import BeautifulSoup

soup = BeautifulSoup(html, "html.parser")
html = soup.prettify()
```

Эцэст нь гарсан үр дүнг “htmls/staffs.html” файлд хадгалав.

```
with open("htmls/staffs.html", "w", encoding="utf-8") as f:
  f.write(html)
```

4.4 Өгөгдөл ялгаж авах

Одоо HTML документын DOM загварын дагуух бүтцэд тулгуурлан хэрэгтэй өгөгдлөө ялгаж авна. Чухамдаа энэ нь л веб скрапингийн гол ажил болох “хайчилбар” авах үйлдэл юм.

```
from bs4 import BeautifulSoup

soup = BeautifulSoup(html, "lxml")

emps = soup.find(id="emps")

members = emps.find_all("div", class_="members")

results = []

for m in members:
  # (1) Ажилчдын нэр (<h1> -> <a> -> text)
  h1 = m.find("h1")
  name = h1.get_text(strip=True) if h1 else None

  # (2) Албан тушаал (<h1> элементийн яг дараагийн <p>)
  # <h1> -> parent element -> first <p> element
  first_p = h1.find_parent().find_all("p") if h1 else []
```

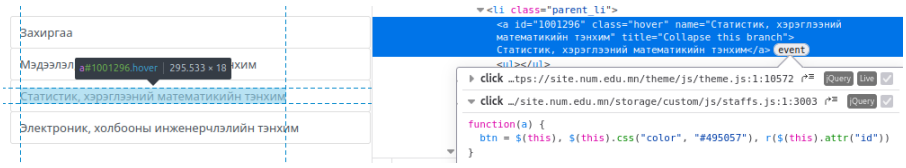
```
  position = first_p[0].get_text(strip=True)[1:] if len(first_p) > 0 else None

  # (3) имэйл, утас, ажлын өрөө агуулсан <p> элементүүд
  right_col = m.find_all("div", class_="col-md-4")
  p_items = []

  if len(right_col) > 1:
    for p in right_col[1].find_all("p"):
      text = p.get_text(strip=True)
      p_items.append(text)

  # (4) Үр дүнг хадгалах
  results.append({
    "name": name,
    "position": position,
    "details": p_items    # [имэйл, утас, ажлын өрөө]
  })

print(results)
```

```
[{'name': 'Б.Ганбат', 'position': 'Захирал', 'details': ['ganbatb@num.edu.mn',
'dean_site@num.edu.mn', '75754400, 77307730 -  3100']}, {'name': 'Т.Мөнхгэрэл',
'position': 'Захирлын туслах', 'details': ['munkhgerel@num.edu.mn',
'assistant_site@num.edu.mn', '75754400, 77307730 -  3101', 'Хичээлийн байр
7-203']}, {'name': 'Г.Энхжаргал', 'position': 'Ахлах нягтлан бодогч', 'details':
['enkhjargal.g@num.edu.mn', '75754400, 77307730 -  3114', 'Хичээлийн байр
7-212']}, {'name': 'Ж.Батхүү', 'position': 'Нягтлан бодогч /Үндсэн хөрөнгө,
бараа материал/', 'details': ['batkhuu.j@num.edu.mn', 'Хичээлийн байр 7-212']},
{'name': 'Г.Буянхишиг', 'position': 'Талбайн үйлчлэгч', 'details': []}, {'name':
'Г.Уранцэцэг', 'position': 'Талбайн үйлчлэгч', 'details': []}, {'name':
'Д.Золжаргал', 'position': 'Талбайн үйлчлэгч', 'details': []}, {'name':
'Н.Батнасан', 'position': 'Талбайн үйлчлэгч', 'details': []}, {'name':
'Н.Дагиймаа', 'position': 'Талбайн үйлчлэгч', 'details': []}]
```

Дээрх кодын `first_p[0].get_text(strip=True)[1:]` мөрийн төгсгөлд `[1:]` гэсэн нь ажилчдын албан тушаалын өмнөх # тэмдгийг гээж орхих зорилготой.

Одоо нэг асуудал гарсан нь тэнхимүүдийн багш, ажилчдын мэдээллийг авах явдал юм. Асуудал гэсний шалтгаан уг мэдээлэл веб хуудсын зүүн гар талын баганад буй цэс дэх линк дээр товшиход бас л JavaScript кодын тусламжтай ачаалагдаж байгаад оршино. Линк дээр товшиход ямар JavaScript ажиллахыг Зураг 3 зураг дээр үзүүлсэн шиг инспект хийн харж болно.

Зураг 3: Линк дээр товшиход ажиллах JavaScript код



Улмаар чухам ямар код ажиллан юу болж буйг мэдэхийн тулд “staff.js” файлыг нээгээд `r($(this).attr("id"))` кодоос эхлэн уншина. Ийнхүү нээтэл minify хийсэн файл байв. Кодыг beautify хийн уншихад хялбар болгов.

```
var array_unit = [],
    array_emp = [],
    array_all = [],
    array_emp_filtered = [],
    k = 0,
    un = "",
    btn = "",
    lng = 1;
$(function() {
    function r(a) {
        array_emp.length = 0, array_all.length = 0, $.ajax({
            type: "GET",
            url: "https://portal.num.edu.mn/handler/myhandler.ashx?nr=9&lng=1&
par1=" + a,
            dataType: "xml",
            cache: !1,
            async: !1,
            success: function(a) {
                $(a).find("row").each(function(a) {
                    var r = {
                        id: $(this).attr("id"),
                        nm: $(this).attr("nm"),
                        mail: $(this).attr("mail"),
                        workmail: $(this).attr("workmail"),
                        bu: $(this).attr("bu"),
                        po: $(this).attr("po"),
                        pos: $(this).attr("pos"),
                        un: $(this).attr("un"),
                        uh: $(this).attr("uh"),
                        uid: $(this).attr("uid"),
                        pt: $(this).attr("PosType")
                    };
                    array_emp.push(r)
                }), 0 < array_emp.length && function() {
                    var a = "";
                    a += '<div class="row">';
                    for (var r = 0; r < array_emp.length; r++) {
                        var t = "",
                            i = "";
                        if ("" != array_emp[r].mail)
                            for (var s = array_emp[r].mail.split("#"), n = 0; n
< s.length; n++) t += '<p><i class="fas fa-envelope"></i>' + s[n] + "</p>";
                        if ("" != array_emp[r].workmail)
                            for (var e = array_emp[r].workmail.split("#"), n =
0; n < e.length; n++) i += '<p><i class="fas fa-building"></i>' + e[n] + "</p>";
                        var l = "";
                        "" != array_emp[r].po && (l = '<p><i class="fas fa-
```

```
phone"></i>' + array_emp[r].po + "</p>");
                        var p = "";
                        "" != array_emp[r].bu && (p = '<p><i class="fas fa-map-
pin"></i>' + array_emp[r].bu + "</p>"), a += '<div class="col-md-12"><div
class="members"><div class="row">', a += '<div class="col-md-3"><img
src="https://sisi.num.edu.mn/sisi_v4/Modules/getpic.ashx?id=' + array_emp[r].id
+ '"style="height:100px;width:100px; object-fit:cover;display: block; margin: 0
auto;" class="img-circle"  alt="staff"/></div>', a += '<div class="col-md-4"><h1
class="font-weight-normal line-height-1"><a href="/staffs/details?id=' +
array_emp[r].id + "&uid=" + array_emp[r].uid + '">' + array_emp[r].nm + "</a></
h1><p>" + array_emp[r].pos + '</p></div><div class="col-md-4">' + t + i + l + p
+ "<a href='/staffs/details?id=" + array_emp[r].id + "&uid=" + array_emp[r].uid
+ "' class='button btn btn-xs btn-light text-1 text-uppercase staff-
more'>Дэлгэрэнгүй</a></div></div></div><br>"
                    }
                    $("#emps").html(a)
                }()
            },
            error: function(a, r, t) {
                $("#emps").html("")
            }
        })
    }
    $.ajax({
        type: "GET",
        url: "https://portal.num.edu.mn/handler/myhandler.ashx?nr=14&lng=1&par1=
1002076",
        dataType: "xml",
        cache: !1,
        async: !1,
        success: function(a) {
            $(a).find("row").each(function() {
                var a = {
                    id: $(this).attr("ID"),
                    un: $(this).attr("Un"),
                    pu: $(this).attr("Pu"),
                    ot: $(this).attr("Ot")
                };
                array_unit.push(a)
            }), a = "", a += '<div class="tree">', a += function a(r) {
                var t = "<ul>";
                for (var i = 0; i < array_unit.length; i++) array_unit[i].pu ==
r && (0 == k && (un = array_unit[i].id), 1002076 == r || array_unit[i].pu ==
un ? t += "<li>" : t += '<li class="">', k++, t += '<a id="' + array_unit[i].id
+ '" name="' + array_unit[i].un + '" class="hover">' + array_unit[i].un + "</
a>", t += a(array_unit[i].id), t += "</li>");
                t += "</ul>";
                return t
            }(1002076), a += "</div>", $("#units").html(a)
        },
        error: function(a, r, t) {
```

```
            $("#myaccordion").html("")
        }
    }), $(".tree li:has(ul)").addClass("parent_li").find(" > a").attr("title",
"Collapse this branch"), $(".all_units").hide("fast"),
$("#uname").text($("#1001028").attr("name")), r(1002081), btn = $("#1002081"),
$(".tree li.parent_li > a").on("click", function(a) {
        btn = $(this), $(this).css("color", "#495057"), r($(this).attr("id"))
    })
});
```

`r($(this).attr("id"))` код файлын төгсгөлд байна. Энд дуудан ажиллуулах `r()` функцийг файлын бараг эхэнд зарлажээ. Функцийг бие дэх кодыг уншвал тус функц `"https://portal.num.edu.mn/handler/myhandler.ashx?nr=9&lng=1&par1="+a` линк рүү GET аргаар AJAX хүсэлт илгээгээд XML [7] форматтай хариу хүлээн авах нь мэдэгдэнэ. Түүнчлэн линк нь `r(a)` функцийн a аргументаар ямар утга дамжуулсанаас хамаарч эцэслэн тодорхой болно.

`r(a)` функцийг "staff.js" файл дотор `$(function() {` гэж эхэлсэн, DOM бэлэн болмогц ажиллах функц дотор зарлажээ. Кодыг цааш уншвал `r(a)` функцийг файлын төгсгөл хавьд `r(1002081)` гэж дуудсан ба энэ нь DOM бэлэн болмогц ажиллахаар бичигджээ. Ингээд `"https://portal.num.edu.mn/handler/myhandler.ashx?nr=9&lng=1&par1=1002081"` линкээр GET [8] хүсэлт илгээв. GET аргаар хүсэлт илгээхийн тулд ердөө веб хөтчийн хаягийн мөрд тус линкийг оруулаад Enter товчлуур дарахад хангалттай. Серверээс ирсэн хариуг дор хуулж орууллаа.

```
<ROOT>
<row id="86B5490F-7E03-41A9-A52B-D9398B7B7399" nm="Б.Ганбат" bu="" po="75754400,
77307730 - 3100" pos="#Захирал" PosOrder="120" PosType="1" uid="1002081"
un="Захиргаа" uh="МУИС, МТЭС, 3" mail="ganbatb@num.edu.mn"
workmail="dean_site@num.edu.mn" saunit="0"/>
<row id="E12165F7-39AB-4028-9568-CB9194ADD1FF" nm="Т.Мөнхгэрэл" bu=" Хичээлийн
байр 7-203" po="75754400, 77307730 - 3101" pos="#Захирлын туслах" PosOrder="228"
PosType="2" uid="1002081" un="Захиргаа" uh="МУИС, МТЭС, 3"
mail="munkhgerel@num.edu.mn" workmail="assistant_site@num.edu.mn" saunit="0"/>
<row id="31BA5606-8B18-4CB4-A943-930A49937B8A" nm="Г.Энхжаргал" bu=" Хичээлийн
байр 7-212" po="75754400, 77307730 - 3114" pos="#Ахлах нягтлан бодогч"
PosOrder="312" PosType="2" uid="1002081" un="Захиргаа" uh="МУИС, МТЭС, 3"
mail="enkhjargal.g@num.edu.mn" workmail="" saunit="0"/>
<row id="AE54C397-B996-4062-AB74-EF21CC83A3F5" nm="Ж.Батхүү" bu=" Хичээлийн байр
7-212" po="" pos="#Нягтлан бодогч /Үндсэн хөрөнгө, бараа материал/"
PosOrder="313" PosType="2" uid="1002081" un="Захиргаа" uh="МУИС, МТЭС, 3"
mail="batkhuu.j@num.edu.mn" workmail="" saunit="0"/>
<row id="5FA26970-43FF-4340-99A2-EDF01B6D8398" nm="Г.Буянхишиг" bu="" po=""
pos="#Талбайн үйлчлэгч" PosOrder="443" PosType="3" uid="1002081" un="Захиргаа"
uh="МУИС, МТЭС, 3" mail="" workmail="" saunit="0"/>
<row id="8D5352EB-AF3A-414D-9EB1-1B8BB3B172E8" nm="Г.Уранцэцэг" bu="" po=""
pos="#Талбайн үйлчлэгч" PosOrder="443" PosType="3" uid="1002081" un="Захиргаа"
uh="МУИС, МТЭС, 3" mail="" workmail="" saunit="0"/>
<row id="4FD62D80-FC43-4CE1-858D-45601E0CC358" nm="Д.Золжаргал " bu="" po=""
pos="#Талбайн үйлчлэгч" PosOrder="443" PosType="3" uid="1002081" un="Захиргаа"
```

```
uh="МУИС, МТЭС, 3" mail="" workmail="" saunit="0"/>
<row id="FE815EFF-8B27-4621-90F9-D3AB47277BF3" nm="Н.Батнасан" bu="" po=""
pos="#Талбайн үйлчлэгч" PosOrder="443" PosType="3" uid="1002081" un="Захиргаа"
uh="МУИС, МТЭС, 3" mail="" workmail="" saunit="0"/>
<row id="82B4534C-5012-4F23-BA51-AFA89FDE8CD3" nm="Н.Дагиймаа" bu="" po=""
pos="#Талбайн үйлчлэгч" PosOrder="443" PosType="3" uid="1002081" un="Захиргаа"
uh="МУИС, МТЭС, 3" mail="" workmail="" saunit="0"/>
</ROOT>
```

Ийнхүү веб хуудсыг дуудаж ачаалахад захиргааны харьяа ажилчдын мэдээлэл гарч ирсэний учир тодорхой болов. Нөгөө талаас энэ мэдээллийг аль хэдийн “хайчлаад” авчихсан тул ингэсхийн орхиж харин бусад линк дээр товшиход гарч ирэх тэнхимүүдийн багш, ажилчдын мэдээллийг гаргаж авахад л анхаарлаа хандуулна.

Салбар сургуулийн захиргаа болон тэнхимүүдийн мэдээлэл “staff.js” файлд буй хоёр дахь AJAX хүсэлтийн хариуд ирэх ажээ. Эхний AJAX үйлдэл шиг GET аргаар `"https://portal.num.edu.mn/handler/myhandler.ashx?nr=14&lng=1&par1=1002076"` линкээр хүсэлт илгээн XML хариу хүлээн авч байна. XML документаас зөвхөн салбар сургуультай хамаатай хэсгийг дор хуулж оруулав.

```
<root>
<row ID="1002081" Un="Захиргаа" Ab="МУИС, МТЭС, 3" Hp="https://site.num.edu.mn"
Pu="1002076" Ot="10"/>
<row ID="1001298" Un="Мэдээлэл, компьютерийн ухааны тэнхим" Ab="МУИС, Мтэс,
Мкут" Hp="http://seas.num.edu.mn/dep/ics" Pu="1002076" Ot="9"/>
<row ID="1001296" Un="Статистик, хэрэглээний математикийн тэнхим" Ab="МУИС,
Мтэс, Схмт" Hp="http://seas.num.edu.mn/dep/am" Pu="1002076" Ot="9"/>
<row ID="1001297" Un="Электроник, холбооны инженерчлэлийн тэнхим" Ab="МУИС,
Мтэс, Эхит" Hp="http://seas.num.edu.mn/dep/ece" Pu="1002076" Ot="9"/>
</root>
```

“staff.js” файл дахь JavaScript кодыг цааш уншвал XML файлаар ирсэн мэдээллээс `pu` аттрибутын утга `1002076` утгатай тэнцүү `row` элементүүдийг ялган авч байна. Мөн эдгээр элемент дэх `id` аттрибутын утга нь дээр дурдсан `r($(this).attr("id"))` код руу дамжих “id” утга аттрибутын утга болно. Ийнхүү сүүлд татаж авсан XML документын `<row>` элементүүдийн `id` аттрибутын 1002081, 1001298, 1001296, 1001297 утга нэг бүрчлэн `"https://portal.num.edu.mn/handler/myhandler.ashx?nr=9&lng=1&par1=1002081"`, `"https://portal.num.edu.mn/handler/myhandler.ashx?nr=9&lng=1&par1=1001298"` гэх мэтчилэн GET хүсэлтийг сервер рүү илгээвэл багш, ажилчдын мэдээлэл XML форматаар ирнэ.

1002081, 1001298, 1001296, 1001297 дөрвөн утгыг гаргаж авах үйлдлийг ч програмчилж болно.

```
import requests
from bs4 import BeautifulSoup

url = "https://portal.num.edu.mn/handler/myhandler.ashx?nr=14&lng=1&par1=
1002076"
```

```
response = requests.get(url)

soup = BeautifulSoup(response.text, "lxml-xml")

num_units = soup.find_all("row")
site_units = []
for unit in num_units:
  pu = unit.get("Pu")
  if pu == '1002076':
    id = unit.get("ID")
    site_units.append(id)

print(site_units)
```

Дээрх кодыг ажиллуулбал дараах үр дүн гарна.

```
['1002081', '1001298', '1001296', '1001297']
```

```
import requests
from bs4 import BeautifulSoup

site_emps = []
for unit in site_units:
  url = "https://portal.num.edu.mn/handler/myhandler.ashx?nr=9&lng=1&par1=" +
unit
  response = requests.get(url)
  soup = BeautifulSoup(response.text, "lxml-xml")
  emps = soup.find_all("row")
  for emp in emps:
    site_emps.append({
      "name": emp.get("nm"),
      "unit": emp.get("un"),
      "position": emp.get("pos")[1:].split("#"),
      "email": emp.get("mail"),
      "phone": emp.get("po"),
      "office": emp.get("bu").strip()
    })

print(site_emps[59:64])
```

```
[{'name': 'Д.Баянжаргал', 'unit': 'Статистик, хэрэглээний математикийн тэнхим',
'position': ['Тэнхимийн эрхлэгч', 'Профессор'], 'email':
'bayanjargal@num.edu.mn', 'phone': '75754400, 77307730 -  3600', 'office':
'Хичээлийн байр 7-401'}, {'name': 'А.Галтбаяр', 'unit': 'Статистик, хэрэглээний
математикийн тэнхим', 'position': ['Профессор'], 'email':
'galtbayar@num.edu.mn', 'phone': '75754400, 77307730 -  3603', 'office':
'Хичээлийн байр 8-308'}, {'name': 'А.Энхбаяр ', 'unit': 'Статистик, хэрэглээний
математикийн тэнхим', 'position': ['Профессор'], 'email':
'enkhbayar.a@num.edu.mn', 'phone': '75754400, 77307730 -  3603', 'office':
```

```
'Хичээлийн байр 8-308'}, {'name': 'Г.Баттөр', 'unit': 'Статистик, хэрэглээний
математикийн тэнхим', 'position': ['Профессор'], 'email': 'battur@num.edu.mn',
'phone': '75754400, 77307730 -  3604', 'office': 'Хичээлийн байр 8-100e'},
{'name': 'Ж.Сонинбаяр', 'unit': 'Статистик, хэрэглээний математикийн тэнхим',
'position': ['Зөвлөх дэд профессор'], 'email': 'jsoninbayar@num.edu.mn',
'phone': '75754400, 77307730 -  3606', 'office': 'Хичээлийн байр 8-100e'}]
```

Эцэст нь нь үүнийг *pandas.dataFrame* болгоод зарим мөрийг хэвлэж харуулъя. Үүний тулд эхлээд `pip install pandas` тушаалаар тус багцыг суулгана. Веб скрапинг ажлын үр дүн болох өгөгдлийн зарим хэсгийг Хүснэгт 1 дээрээс харж болно.

```
import pandas as pd

df = pd.DataFrame(site_emps)
df[["name","unit"]].iloc[59:64]
```

Хүснэгт 1: Веб скрапинг аргаар цуглуулсан өгөгдлийн эхний хоёр багана дахь зарим мөр

	name	unit
0	Д.Баянжаргал	Статистик, хэрэглээний математикийн тэнхим
1	А.Галтбаяр	Статистик, хэрэглээний математикийн тэнхим
2	А.Энхбаяр	Статистик, хэрэглээний математикийн тэнхим
3	Г.Баттөр	Статистик, хэрэглээний математикийн тэнхим
4	Ж.Сонинбаяр	Статистик, хэрэглээний математикийн тэнхим

Дүгнэлт

1. Зориулалтын API байгаагүй тул HTML DOM бүтцийг задалж унших аргаар веб скрапинг хийлээ.
2. Албан ёсны зөвшөөрөлтэй веб скрапинг хийв.
3. Веб хуудасны үндсэн контент JavaScript кодоор ачаалагдаж байсан тул тус кодыг уншиж, зохих код бичин скрапинг хийв.
4. Веб скрапинг хэрэгжүүлэх явцад буюу энэхүү тайлан дахь Python кодыг эхнээс дуустал ажиллуулахад веб сервер рүү нийтдээ 7 удаа хүсэлт илгээсэн байна.

Ашигласан материал

[1] М. У. И. Сургууль, “Багш, ажилтан — МУИС.” https://site.num.edu.mn/staffs, 2025.

[2] W3Schools, “HTML tutorial.” Accessed: Nov. 26, 2025. [Online]. Available: https://www.w3schools.com/html/

[3] Wikipedia contributors, “Document object model — Wikipedia, the free encyclopedia.” https://en.wikipedia.org/w/index.php?title=Document_Object_Model&oldid=1316812336, 2025.

[4] DataCamp, “Ethical web scraping: Principles and practices.” Accessed: Nov. 25, 2025. [Online]. Available: https://www.datacamp.com/blog/ethical-web-scraping

[5] Wix, “Robots.txt file: The complete guide.” Accessed: Nov. 25, 2025. [Online]. Available: https://www.wix.com/seo/learn/resource/robots-txt-file

[6] Cloudflare, “What is a denial-of-service (DoS) attack?” Accessed: Nov. 26, 2025. [Online]. Available: https://www.cloudflare.com/learning/ddos/glossary/denial-of-service/

[7] W3Schools, “XML tutorial.” Accessed: Nov. 26, 2025. [Online]. Available: https://www.w3schools.com/xml/

[8] W3Schools, “HTTP request methods.” Accessed: Nov. 26, 2025. [Online]. Available: https://www.w3schools.com/tags/ref_httpmethods.asp