Աշխատաշուկայում թափուր տեղերի հայտնաբերման և տեղեկատվության մշակման ավտոմատացված համակարգի նախագծում Scraping տեխնոլոգիայով

Ներածություն

Համակարգը թույլ կտա աշխատանք փնտրող մարկանց ավելի արագ գտնել թափուր աշխատատեղեր , իսկ գործատուներին մի քանի անգամ ավելի շուտ լրացնել բաց հաստիքներ:

Համակարգը ստեղծված է բոլոր նրանց համար, ովքեր ցանկանում են արագ և հեշտությամբ աշխատանք գտնել: Կայքը միավորում է թափուր աշխատատեղերի վերաբերյալ հայտարարությունները՝

• Կազմակերպությունների

• Աշխատանքի տեղավորման գործակալությունների

• մարդկային ռեսուրսների հավաքագրման գծով մասնագիտացված,

• Կադրերի հավքագրման բլոկ ունեցող և

• պետական գերատեսչությունների կայքերից:

Այլևս կարիք չկա աշխատանքային հայտարարություններ փնտրելու համար ժամանակ ծախսել համացանցում գործող տասնյակ կայքերում։ Այս համակարգը Ձեր փոխարեն կանի այդ աշխատանքը:

Այս համակարգի շնորհիվ Դուք կկարողանաք մեկ տեղում տեսնել Ձեզ հետաքրքրող աշխատանքի ամենաթարմ հայտարարությունները։

Խնդրի դրվածք

Ստեծել համակարգ ՝ վեբ կայք, որը կունենա փնտրվող աշխանատքի բանալի բառի մուտքագրման համար գոտի, մուտքագրումից և հաստատումից հետո կփնտրի տեղական աշխատաշուկայում թափուր աշխատատեղերը ըստ մուտքագրված բանալի բառի , կստուգի գտնված թափուր աշխատատեղերի արդիական լինելը,

ապա կցուցադրի դրանք ըստ արդիականուտյամբ և բանալի բառին ավելի ճիշտ համընկնելու սկզբունքով:

Համակարգը պետք է այցելի բոլոր հեղինակություն վայելող աշխատանքային հայտարարություններ փնտրելու կայքերը, ճիշտ տեղում մուտքագրի բանալի բառը

Հաստատի և սպասի պատասխանին, լավ արդյունքի դեպքում ստուգի պատասխանում գտնվող բոլոր հայտարարությունրը վերցնի նրանք որոնք համապատասխանում են բանալի բառին և դեռ արդիական են և վերադարձնի սեռվեր:

Այնուհետև սեռվերը համախմբի արդյուքները և ցույց տա վեբ կայքում :

Համակարգը պեքէ կարողանա գործել մուլտի հոսքային սկզբունքով :

Համեմատական վերլուծություն

Տեղական աշխատաշուկայում , գործում են բազում աշխատանքի փնտրման կայքեր

Որոնցից առավել հեղինակություն են վայելում հետևյալները՝

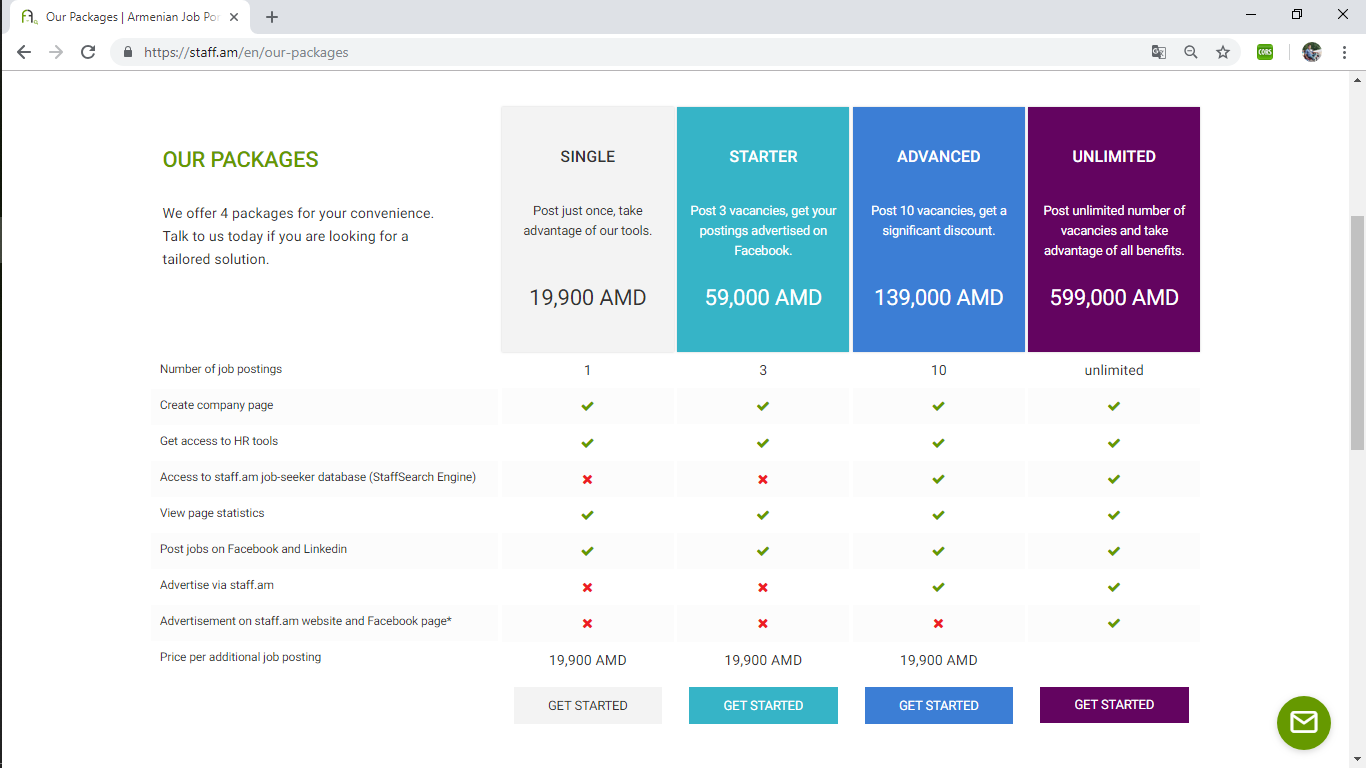
* <http://www.hr.am/>
* <https://staff.am/>
* <https://www.worknet.am/>
* <https://job.am/>
* <https://www.myjob.am/>
* <http://ijob.am/>
* <https://www.careerhouse.com/>
* <http://jobfinder.am/>
* <http://itjob.am/>
* <https://www.list.am/ru/category/29?q=ashxatanq>
* <http://jobspider.am/>

Նշվածներից միան <http://jobspider.am/> կայքն է որ մասնակի գործում է

ներկայացված համակարգի սկզբունքով ,մնացաց բոլոր կայքերը օգտվում են

միայն իրենց ներքին տվյալների բազայից և գործում են հետեվյալ սկզբունքով՝

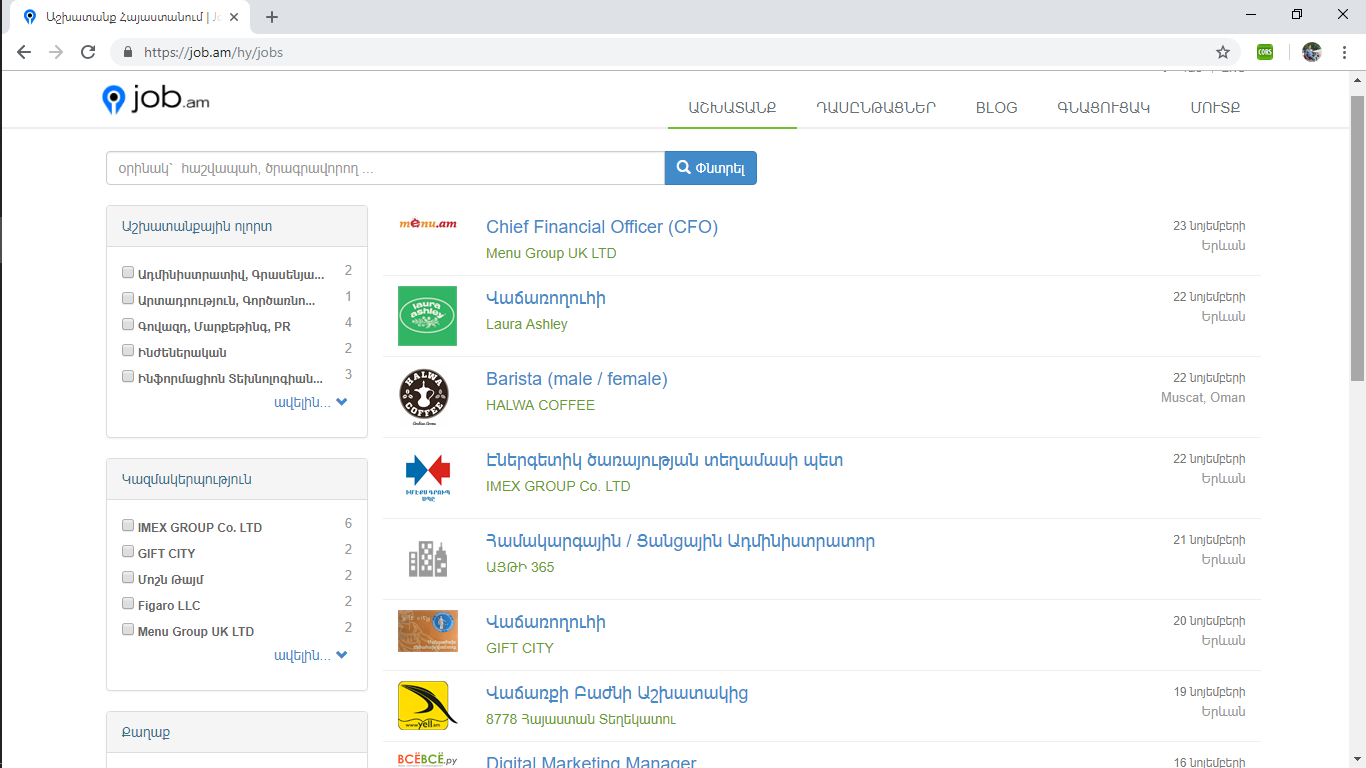
գործատու կազմակերպությունը կամ անձը մուտք է գործում կայք ,գրանցվում է, կայքում ավելացնում է ազատ աշխատատեղի մասին հայտարարություն որոշ կայքերում հայտարարությունները վճարովի են օրինակ ՝ <https://staff.am/> կայքում:



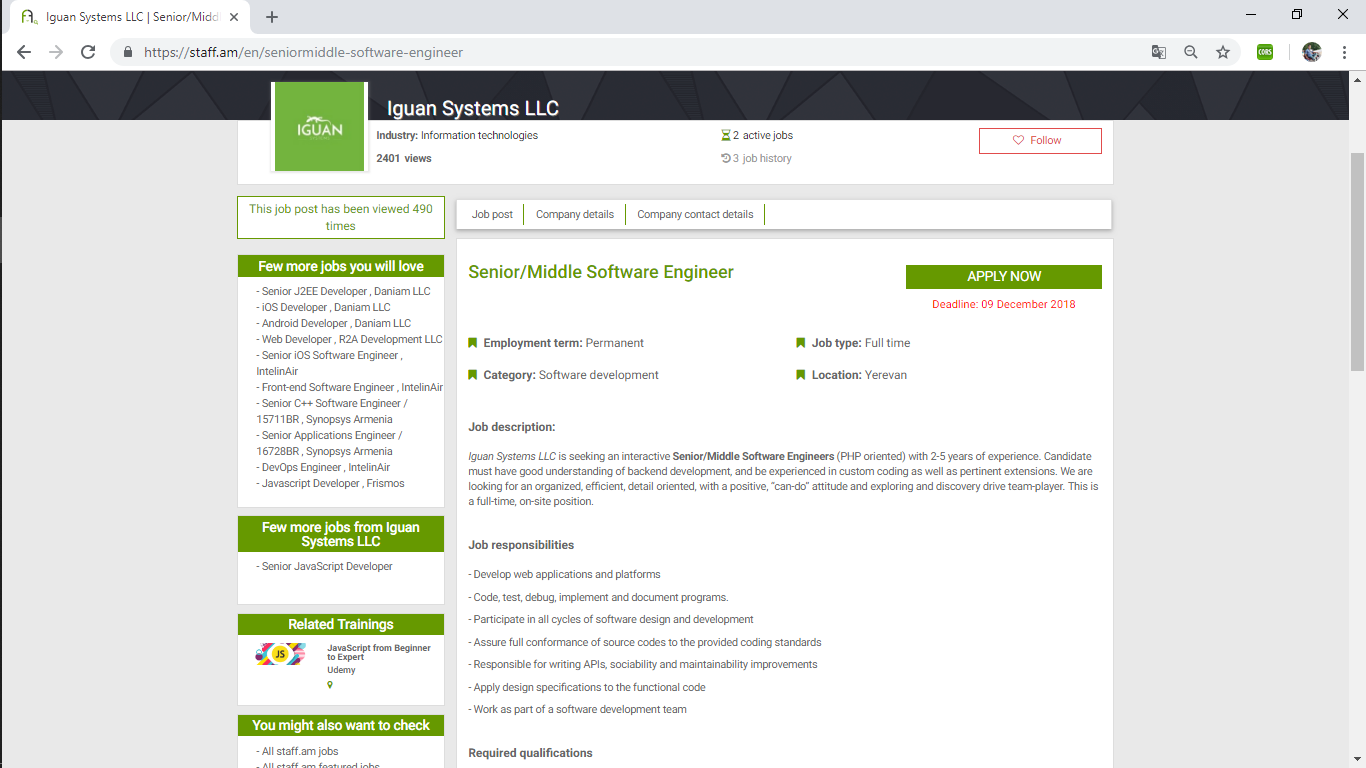
Նկար 1.

Ապա աշխատանք փնտրող մարդիկ մուտքեն գործում կայք , փնտրում հայտարարություներ բանալի բառով կամ տրվող ցուցակից , հավաստիանում են

հայտարարությունը ուժի մեջ է թէ ոչ այնուհետև դիմում գործատուին:



Նկար 2.



Նկար 3.

Ինչ թերություներ ունի այս սկզբունքը ՝ նախ բոլոր գործատու կազմակերպությունները

կամ անհատները իրենց հայտարարությունները պետքէ տեղադրեն բոլոր հասանելի կայքերում ցանկանլի արդյունքին հասնելու համար , սակայն դա ժամանակատար է և կապված հավելյալ ծախսերի հետ :

Երկրոդ թերությունը այն է որ աշխատանք փնտրող անձը թափուր հաստիք որոնելու

համար պետք է այցելի վերոհիշյալ կայքերը , փնտրի որոնման արդյունքների մեջ

իրեն հետաքրքրող հայտարարությունը ապա դիմի գործատուին և այս գործողություները կրկնի 10 –ից ավելի կայքերի համար որը իր հերթին ժամանակատար է և ոչ այդքան արդյունավետ :

Ինչ վերաբերվում է <http://jobspider.am/> կայքին, որը աշխատում է միայն մի քանի կայքերի հայտարարությունների հիման վրա , չի արտահայտում ոլորտի իրական պատկերը և չի բավարարում հաճախորդների պահանջները :

Գլուխ 1․ Ընդհանուր փաստեր վեբ scraping-ի մասին

Գրական հետազոտությունների ժամանակ հայտնաբերվել է վեբ scraping-ի մի քանի սահմանումներ: Ստորև ներկայացված բոլոր երեք հասկացությունները հիշատակվում են տվյալների ստացման տարբեր աղբյուրներում: Նրանք տարբերվում են ստացված տեղեկատվության համար նախնական աղբյուրների տեսքով:

Երբեմն անհրաժեշտ է տեղեկատվություն հավաքել վեբ կայքերից, որոնք նախատեսված են ոչ թե ծրագրային գործակալների, այլ մարդկային ընթերցողների համար:

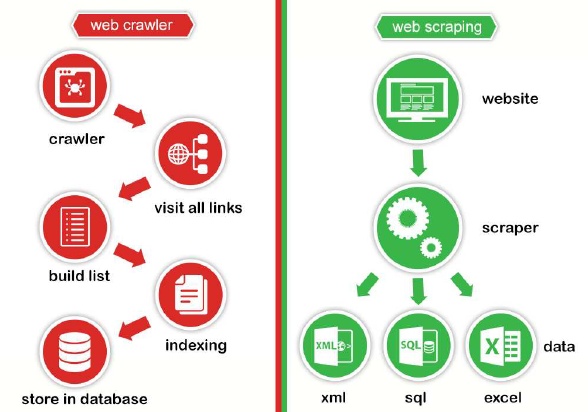
Այս գործընթացը հայտնի է որպես «վեբ scraping»:

Առաջին սահմանումը նշում է տվյալների աղբյուրները, որոնք սկզբնապես նախատեսված են ընթերցողների համար: Նման որոշումը արդեն երկար ժամանակ է համարում են արդեն հնացած: Ավտոմատացված տեխնիկայի զարգացումը հնարավորություն է ընձեռում նաև ծրագրային ապահովման ընթեռնելի աղբյուրներից արդյունահանման միջոցով: Այնուամենայնիվ, պետք է հաշվի առնել, որ հրապարակման ամսաթիվը 2009 թվականն էր: Դիմումների ծրագրավորման ինտերֆեյսը (API) Աղբյուրները այդ ժամանակ սահմանափակ էին: Հանրային API- ի դիրեկտորը, որը հասանելի է Programmable Web-ում (Berlind, 2015) ունի մոտավորապես 750 առկա աղբյուրները , համեմատած 2017 թվականի 17175-ի:

Վեբ scraping-ը, որը նաև հայտնի է որպես վեբ extraction կամ harvesting, այնպիսի մեթոդ է, որ համացանցից (WWW) տվյալները հանում և պահպանում է այլ ֆայլային համակարգում կամ տվյալների բազայում, հետագայում որոնման կամ վերլուծության համար: Ընդհանրապես, վեբ-ի տվյալները ստացվում են (HTTP) պրոտոկոլի կամ վեբ բրաուզերի միջոցով: Դա կատարվում է ձեռքով՝ օգտագործողի կողմից կամ ավտոմատ կերպով բոտի կամ վեբ զննարկչի կողմից: Շնորհիվ այն բանի, որ WWW- ում անընդհատ գեներացվում է հսկայական քանակի տվյալներ, վեբ scraping-ը լայնորեն ճանաչվում է որպես մեծ տվյալների հավաքման արդյունավետ և հզոր տեխնիկա:

Ներկայիս իրավիճակը ավելի կոնկրետ կերպով նկարագրված է երկրորդ սահմանման մեջ, երբ վեբ scraping-ը հիշատակվում է որպես մեծ տվյալների հավաքման աղբյուրներից մեկը: Սահմանում է նաև մեկ այլ տերմին - վեբ Crawler:

 Վեբ Crawler: Վեբ Crawling- ը կատարվում է ուրիշ տարբերակով և տալիս է ուրիշ արդյունք: Նկար 1-ում նկարագրվում են երկու գործողությունները: Ձախ կողմում տեսանելի գործընթացը ցույց է տալիս, որ Վեբ Crawling- ը չունի յուրահատուկ թիրախ և ընդհանրացված տեղեկատվություն է պարունակում `առանց կոնկրետ տեղեկատվության նպատակաուղղվածության: Համեմատելով աջ կողմի նկարի հետ, որտեղ Վեբ Scraper- ը ստանում է տվյաներ, գործընթացներ է իրականացնում այդ նույն տվյալների հետ և վերլուծում տվյալ աղբյուրներից ստացված տվյալները: Web Crawling-ը չի դիտարկվում այս թեմայի մեջ:



Նկար 4.

Ստորև բերված սահմանման մեջ շատ մանրամասներ չեն նշվում:

Այնուամենայնիվ, դա հստակորեն գրավում է վեբ scraping -ի գործունեությունը:

Վեբ Scraping-ը ներառում է աղբյուրի հարցման, արդյունքների էջը վերցնելու և արդյունքների ձեռք բերման համար էջը վերլուծելու գործընթացը:

1.2 Վեբ Scraping -ի նպատակը և օգտագործման ոլորտները

Համաշխարհային ցանցում առկա տեղեկատվության հսկայական մասը դեռեւս հիպերտեքստի նշարկման լեզու (HTML) էջի ձեւաչափով են: Ավտոմատացված արդյունահանումը դժվար է, քանի որ նախատեսված էր մարդկանց ընթերցման համար: Այս գլխում ներկայացվում է տեղեկատվության արդյունահանման դրդապատճառները և Web Scraping- ի նպատակը :

Համաշխարհային ցանցի արագ զարգացումը զգալիորեն փոխվել է, թե ինչպես ենք մենք կիսվում, հավաքում և հրապարակում մեր տվյալները: Տեղեկատվության մեծ մասը պահվում է առցանց, ինչպես կառուցվածքային այնպես էլ ոչ կառուցվածքային ձևերով: Ինչ վերաբերում է որոշակի հարցերի կամ հետազոտությունների թեմաներին, ապա դա նոր խնդիր է առաջացրել – արդեն իսկ անհանգստացնում է տվյալների անհամապատասխանության և անհասանելիության խնդիրը, այլ ոչ թե ինտերնետի խոշոր զանգվածների հաղթահարումը:

Համաշխարհային ցանցից օգտվելը հաճախ հնարավոր է միայն ավտոմատացված վեբ Scraping-ի գոյության պատճառով: Առանց այդ տեխնիկայի, անհնար է հավաքել տվյալների մեծ քանակությունը հաճախակի և ողջամիտ ժամկետներում:

1.2.1 Վեբ Scraping -ը Շուկայական վերլուծության և հետազոտության մեջ

Ինտերնետային աղբյուրներից տվյալների հավաքումը դարձել է շուկայական հետազոտության մեթոդներից մեկը: Այն առաջարկում է ավելի արագ արձագանք, համեմատած դասական հետազոտության հետ:

Վայլ Նովիթը լավագույնն է համարում ավանդական հետազոտությունների օգտագործումը, իսկ վեբ Scraping -ը համարվում է որպես այդպիսի գործիքների համար ծախսարդյունավետ օժանդակություն: Համապարփակ պատկերացում ստանալու և շուկաներում գործիքներ ձեռք բերելու համար պետք է օգտագործել բազմաթիվ աղբյուրներ:

Սպառողները ակտիվ են առցանց աշխարհում եւ կիսվում են իրենց փորձով, վրդովմունքներով ու մոտիվացիաներով: Ընկերությունները, որոնք ցանկանում են ավելին իմանալ սպառողների մասին կարող են ավելացնել առցանց տեղեկատվության աղբյուրները:

Վեբ scraping-ը նման տվյալների հավաքման եղանակներից մեկն է:

1.2.2 Վեբ Scraping –ը կորպորատիվ տեխնոլոգիաներում

Անհամատեղելի կորպորատիվ տեխնոլոգիաները բնորոշ են խոշոր նախագծերի համար։ Դեռ պետք է միասնական տվյալների ներկայացում մի քանի համակարգերից: Որոշ հատուկ դեպքերում, լուծումը հիմնված է վեբ scraping –ի վրա: Ստորեւ բերված օրինակը ցույց է տալիս մի քանի անկախ աղբյուրներից չափումների հավաքման եւ գնահատման հնարավոր մոտեցումը: Սուրաբայի ջրամատակարարման ընկերությունը, որը կոչվում է PDAM Surabaya, ունի մի քանի ջրամբարներ, որոնք վերահսկվում են WTW IQ SensorNet 2020 XT-ի կողմից: Սակայն այս սենսորային սարքը կարող է ապահովել ջրի որակի պարամետրերի որոշ արժեքնեի տեղեկատվությունը, բայց սենսոր պասիվ է, եւ ներքին տվյալները դեռ պահվում է սենսորում: Խնդիրը լուծելու համար առաջարկվում է կիրառել տվյալների գրանցիչ՝ վեբ scraping -ի միջոցով ջրի որակի մոնիտորինգի առցանց համակարգի տվյալների հավաքագրման կառավարման համար:

1.2.3 Վեբ Scraping -ը Հասարակական Կարծիքի Հետազոտություն մեջ

Ֆիլմերի պրոդյուսերները տվյալներ են հավաքում իրենց ընթացիկ բլոկբաստերների մասին։ Նման տվյալները ներառում օգտատերերի մեկնաբանությունները, եթե դրանք առկա են: Այդ առկա տվյալների հավաքագրման համար շատ հաևմար գործիքէ է վեբ Scraping –ը :

1.2.4 Վեբ Scraping -ը Մարդկային ռեսուրսների գործակալությունների համար

Մարդկային ռեսուրսների բաժինները (HR) խոշոր ընկերություններում մշակում են իրենց ընկերությունների համար աշխատանքի բազմաթիվ դիմումներ եւ փորձում են գտնել համապատասխան թեքնածուներ:

Միայն ընդունելության դիմումները բավարար չեն թափուր թեկնածուների հաստիքները լրացնելու համար: Մարդկային ռեսուրսների բաժինները նույնպես համագործակցում են 3-րդ կողմի կազմակերպությունների հետ, որոնք կարող են առաջարկել նրանց սեփական մասնագետներին ։ Վեբ Scraping-ը կարեւոր գործիք է նման գործակալությունների համար:

1.2.5 Վեբ Scraping -ը Սոցիալական ցանցերում

Վերջին տասնամյակում սոցիալական լրատվամիջոցները (բլոգերը, սոցիալական ցանցերը, միկրո բլոգները) դարձել են խոշոր տվյալների աղբյուր հետազոտությունների համար: Կիրառելով պարզ ծրագրավորման գործիքներ, հետազոտողները կարող են դուրս հանել համապատասխան հաղորդագրությունները սոցիալական լրատվամիջոցների հարթակներից տարբեր հետազոտական նպատակներով:

Սոցիալական ցանցերի տվյալները կարող են ստացվել հանրային աղբյուրներից տարբեր միջոցներով, ինչպիսիք են Scraping -ը, կայքի կողմից տրամադրվող ծրագրերի օգտագործումը եւ շրջանցումը:

1.2.6 Վեբ Scraping -ը Կառավարության ծառայություններում

Հանցագործությունների մոնիտորինգը սոցիալական կայքերում, կարևորագույն աղբյուր է պետական մարմինների եւ իրավապահ մարմինների համար : Այդպիսի մոնիտթրինգ իրականացնելու արդյունավետ միջոց է վեբ Scraping –ը:

1.2.7 Վեբ Scraping -ը Կորպորատիվ լրտեսություն մեջ

Կորպորատիվ կոնտեքստում, վեբ գրառումները թույլ են տալիս ընկերությանը վերանայել ինչպես սեփական, այնպես էլ մրցակիցների հայտնվելը լրատվամիջոցներում: Ընկերությունը կարող է նաեւ հավաքել մանրամասներ մրցակիցների եւ նույնիսկ սեփական աշխատողների մասին:

Ձեռնարկությունների մակարդակում Web Data Extraction- ի մեթոդները առաջանում են որպես բիզնեսի եւ մրցակցային հետախուզության համակարգերում տվյալների վերլուծության, ինչպես նաեւ բիզնեսի վերամշակման համար կարեւոր գործիք:

* 1. Վեբ scraping –ի իրականացնելու մեթոդներ

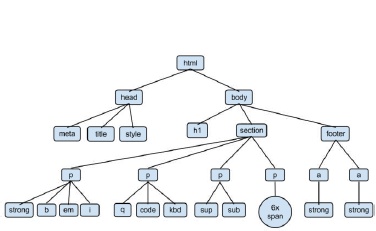
1.3.1 Ջեռքով տվըալների ստացում

Ջեռքով տվյալների ստացումը հնարավոր է, երբ՝

* տվյալների ծավալը շատ փոքր է,
* տվյալները հավաքվում են մեկ անգամ,
* ավտոմատացված համակագրի կարգավորումը ավելի երկար է տևում քան տվյալների հավաքագրումը,
* կայքը պաշտպանված է ավտոմատ համակարգերից կամ հատուկ ձև ունի:

1.3.2 HTML վերլուծություն իրականացնելով

Կայքերը միշտ չէ, որ տալիս են տվյալները հարմարավետ ձեւաչափերով, ինչպիսիք են .csv կամ .json ֆայլերը: HTML- ի էջերը ստեղծվում են սերվերի կողմից, որպես պատասխան օգտվողի հարցմանը: Այս պահին սերվերի ծրագրային ապահովումը կարևոր չէ, այլ կարեւոր է բրաուզերում արտադրանքը: HTML- ի կառուցվածքի վերլուծությունը (նկար 5-ում ներկայացված պարզ էջի նմուշը) տրամադրված ինտերնետային էջում ցույց կտա կրկնվող տարրերը: Ծրագրային լեզվի սցենարով կամ վեբ գրառման գործիքով, յուրաքանչյուր էջը նմանատիպ օրինակով կարող է օգտագործվել որպես տվյալների աղբյուր:

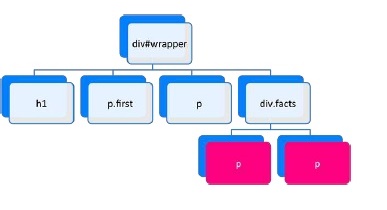


Նկար 5.

1.3.2 DOM -ի վերլուծություն իրականացնելով

Փաստաթղթի օբյեկտ մոդելի (DOM) վերլուծությունը HTML- ի վերլուծության էվոլյուցիա է, որը հիմնված է լեզվի եւ բրաուզերների զարգացումների վրա, որոնք հանգեցնում են փաստաթղթի օբյեկտի մոդելի ներդրմանը: DOM- ն մեծապես օգտագործվում է Cascading Stylesheets (CSS) եւ JavaScript- ի համար:

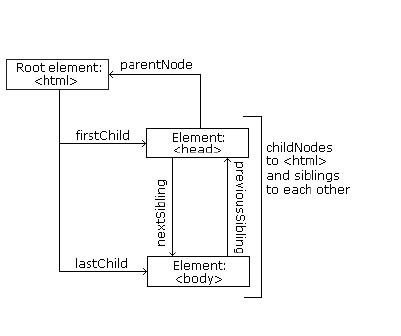
Նկար 6-ը ցույց է տալիս կոնտեյներները սեփական DOM հասցեներով: Դրանք օգտագործվում են վեբ Scraping- ում, վեբ էջի բովանդակության միջոցով ավելի հեշտ տեղաշարժի համար:



Նկար 6.

1.3.3 XPath -ի վերլուծություն իրականացնելով

Հասցեավորման հնարավորությունը նման է այն բանին, որ DOM-ն ապահովում է XPath (XML Path Language): Վերնագիրը առաջարկում XML փաստաթղթերի օգտագործումը։ Սա կիրառելի է նաեւ HTML ձեւաչափով։ XPath պահանջում է ավելի հստակ համակարգված վեբ էջ, քան DOM, եւ ունի նույն հնարավորությունը հասցեագրելով հատվածները վեբ էջի ներսում։ Նկար 7-ը ցույց է տալիս XPath-ի մեկնաբանության փաստաթղթի կառուցվածքը:



Նկար 7.

1.3.4 API –ների միջոցով

Մինչ նախորդ մեթոդները աշխատում են հավաքել հարմարավետ արդյունքներ, application Programm-ING Interface (API) ակնկալում է դիմումը որպես կապի գործընկեր։ Այսպիսով, API-ները հաճախ կոչվում են մեքենայաընթեռնելի ինտերֆեյսներ (համեմատած մարդակենտրոն): API-ի վերջնական կետ ուղարկված ստանդարտ HTTP հարցումը վերադարձնում է պատասխան սերվերից: Յուրաքանչյուր API ունի իր հստակեցումը եւ պարամետրերը։ Պատասխանի ձեւաչափը կարող է սահմանվել որպես հարցման տարբերակ: API-ի հաղորդակցության ամենատարածված ձեւաչափը JSON-ն է:

* 1. Վեբ scraping –ի օրինականությունը

Վեբ scraping -ը անօրինական չէ: Բայց կան նաեւ որոշ իրավական, բարոյական եւ էթիկական սահմանափակումներ, որոնք թույլ են չեն տալիս ընկերություններին օգտվել վեբ scraping -ից : Որոշ պատճառներ ներկայացված են հետեւյալ աղյուսակում.

1. Կայքերի օգտագործման պայմանները: Կայքերը բացահայտորեն արգելում են վեբ scraping -ը:

2. Հեղինակային իրավունքը: Քանի որ վեբ scraping -ը ներառում է պատճենումը, դա կարող է հանգեցնել հեղինակային իրավունքի խախտման:

3. Տվյալների բազայի իրավունքներ: Այս իրավունքները խախտվում են, երբ տվյալների բազայի ամբողջ կամ մի մասը վերցվում է առանց սեփականատիրոջ համաձայնության:

4. Ապրանքային նշանները, կայքի սեփականատիրոջ ապրանքային նշանների վերարտադրումը, առանց դրանց համաձայնության, կարող է հանգեցնել ապրանքային նշանի կամ դրա փոխանցման իրավունքի խախտման պահանջի:

5. Տվյալների պաշտպանություն: Առանձին անհատների (որոշ դեպքերում «անձնական տվյալներ») հավաքագրումը, առանց նրանց թույլտվության, կարող է խախտել տվյալների պաշտպանության մասին օրենքները:

Օրինականությունը լիովին կախված է օրհենսդությունից այսինքն, օրենքները տարբեր են ըստ երկրի եւ տարածաշրջանի: Հանրությանը մատչելի տեղեկատվության հավաքումը անօրինական չէ, եթե դա անօրինական է, Google- ը չի կարող գոյություն ունենալ որպես ընկերություն, քանի որ նրանք հավաքում են ինֆորմացիա աշխարհի բոլոր կայքերից:

1.5 Վեբ scraping –ի համար անհրաժեշտ ծրագրային ապահովման ուսումնասիրությունը

1.5.1 Java ծրագրավորման լեզու

Java պլատֆորմից անկախ, ընդհանուր նշանակության ծրագրավորման լեզու է, որը զուգահեռ, կլասսների վրա հիմնված, օբյեկտ – կողմնորոշված և հատուկ մշակված է, որպեսզի ունենա ինչքան հնարավոր է շատ իրագործումներ։ Այն նախատեսված է, որպեսզի «Գրվի մեկ անգամ, աշխատի ամենուր» ("Write once, run anywhere", WORA), որը նշանակում է, կոմպիլացված Java կոդը կարող է կատարվել բոլոր Java աջակցող պլատֆորմների վրա առանց նորից կոմպիլացվելու։ Java ծրագրերը կոմպիլացվում են բայթկոդ-ի, որոնք կարող են իրագործվել ցանկացած Java վիրտուալ մեքենայի (JVM) վրա, անկախ համակարգչային ճարտարապետությունից։ Մինչ 2015 թ. Java-ն հանդիսանում է ամենահայտնի օգտագործվող ծրագրավորման լեզուներից մեկը, մասնավորապես կլիենտ – սերվեր ծրագրերի համար, մոտ 9 միլլիոն գրանցված ծրագրավորողներով։ Java-ն ստեղծվել է Ջեյմս Գոսլինգի կողմից Sun Microsystems – ում (որը հետագայում Oracle կազմակերպությունը գնեց) և ներկայացվեց 1995 թ. որպես Java պլատֆորմի հիմնական մաս։ Այն ծրագրավորվել է C++-ի միջոցով։ Լեզվի սինտաքսիսը մեծ մասամբ համընկնում է C և C++ լեզուների հետ, բայց այն ունի ավելի քիչ ցածր – մակարդակի միջոցներ, քան նրանցից յուրաքանչյուրը։ Ի տարբերություն C++-ի Java-ն ունի նաև ինտեգրված լայն կլասսների գրադարան (օրինակ բազմապրոցեսային, համացանց, ֆայլային և այլն), որոնք ի շնորհիվ պլատֆորմային անկախության աշխատում են տարատեսակ օպերացիոն համակարգերում։ Որպեսզի Java լեզվով ստեղծված ծրագիրը կարողանա աշխատել ձեր համակարգչում կամ շարժական էլեկտրոնային սարքում, դուք պետք է ունենաք համապատասխան միջավայր՝ JRE (Java Runtime Environment)։ Օրիգինալ և ներդրված Java կոմպիլյատորների, վիրտուալ մեքենաների և կլասսների գրադարանի իրագործումը ի սկզբանե թողարկվել է Sun – ի պատենտավորված լիցենզիայի տակ։ 2007 թ. մայիսի դրությամբ, Sun – ը Java տեխնոլոգիայի մեծ մասի լիցենզիան փոխել է GNU General Public License:

# 1.5.2 jsoup գրադարան

jsoup -ը Java գրադարան է իրական HTML-ի հետ աշխատելու համար: Այն ապահովում է շատ հարմար API արդյունահանման եւ տվյալների մանիպուլյացիայի, օգտագործելով լավագույն DOM, CSS եւ jquery-նմանատիպ մեթոդներ:

jsoup -ը իրականացնում է WHATWG HTML5-ի հստակեցումը եւ վերլուծում է HTML-ը նույն DOM-ում, ինչպես ժամանակակից բրաուզերները:

jsoup -ը տույլ է տալիս ՝

* մաքրել եւ վերլուծել HTML-ը URL-ից, ֆայլից կամ տողից,
* գտնել եւ հանել տվյալները, օգտագործելով DOM կամ CSS selectors,
* փոփոխել HTML տարրերը, հատկանիշները եւ տեքստը,
* մաքրել օգտագործողի բովանդակությունը անվտանգ Սպիտակ ցանկից, կանխելու համար XSS հարձակումները,
* ցուցադրել Մաքուր HTML:

1.6 Կայքի ստեղծման համար անհրաժեշտ ծրագրային ապահովման ուսումնասիրությունը

1.6.1 Spring Framework

Spring Framework (կամ կարճ Spring) Java պլատֆորմի համար նախատեսված ունիվերսալ բաց կոդով Framework է:

Չնայած այն հանգամանքին, որ Spring -ը չի ապահովել որեւէ կոնկրետ ծրագրավորման մոդել, Այն դարձել է Java համայնքում տարածված հիմնականում որպես այլընտրանք ՝ Enterprise JavaBeans մոդելի փոխարեն: Spring ապահովում է ավելի մեծ ազատություն Java ծրագրավորողներին նախագծման մեջ, բացի այդ այն լավ փաստագրված է եւ հեշտ օգտագործման մեջ խնդիրները լուծելիս:

Spring-ը ապահովում է բազմաթիվ խնդիրների լուծումներ, որոնց բախվում են Java-ի ծրագրավորողները եւ կազմակերպությունները, որոնք ցանկանում են ստեղծել Java պլատֆորմի վրա հիմնված տեղեկատվական համակարգեր: Լայն ֆունկցիոնալության պատճառով դժվար է որոշել առավել նշանակալի կառուցվածքային տարրերը, որոնցից այն բաղկացած է: Spring-ը լիովին կապված չէ Java Enterprise պլատֆորմի հետ, չնայած նրա լայնածավալ ինտեգրմանը:

Մոդուլներ ՝

Spring կարելի է դիտարկել որպես Framework -ների հավաքածու: Այդ Framework -երի մեծ մասը կարող է աշխատել միմյանցից անկախ, սակայն նրանք միասին ապահովում են ավելի մեծ ֆունկցիոնալություն:

* Inversion of Control - ծրագրի բաղադրիչների կոնֆիգուրացիան եւ Java օբյեկտների կյանքի ցիկլի կառավարումը:
* Առարկայական կողմնորոշված ծրագրավորման Framework - աշխատում է գործառույթով, որը չի կարող իրականացվել առանց կորուստի օբյեկտի կողմնորոշված Java ծրագրավորման լեզվում:
* Տվյալների հասանելիության Framework  - աշխատում է Java պլատֆորմում տվյալների բազայի կառավարման համակարգերի հետ, օգտագործելով JDBC-ն եւ ORM-ը, եւ ապահովելով խնդիրների լուծումներ, որոնք կրկնվում են Java-based environments-ներում:
* Framework MVC - HTTP-ի եւ Servlet-ի վրա հիմնված շրջանակ, որը ապահովում է ընդլայնման եւ հարմարեցման բազմաթիվ հնարավորություններ:
* Java օբյեկտների կոնֆիգուրացվող փոխանցումը RPC ոճով ցանցի միջոցով, որն աջակցում է RMI-ին, CORBA-ին, HTTP-based արձանագրություններին, ներառյալ վեբ ծառայությունները (SOAP):
* Վավերացում եւ Թույլտվություն Framework - աուտենտիֆիկացիայի և թույլտվության գործընթացների կոնֆիգուրացվող գործիքակազմ:
* Հեռավոր վերահսկման framework - Java օբյեկտների կոնֆիգուրացվող ներկայացում եւ կառավարում լոկալ կամ հեռավոր կոնֆիգուրացիայի համար JMX-ի միջոցով:
* Հաղորդագրությունների փոխանակման Framework  - Հաղորդագրության ունկնդիրների կոնֆիգուրացվող գրանցում, հաղորդագրությունների թափանցիկ մշակում JMS- ի միջոցով, JMS API- ի ստանդարտի օգտագործմամբ ուղարկված հաղորդագրությունների բարելավում:
* Թեստավորում - framework որն աջակցում է կլասներ գրելու մոդուլային եւ ինտեգրացիոն թեստերի համար:

1.6.2 Spring Boot

Spring Boot-ը օգտակար պրոկտ է, որ նպատակն է պարզեցնել Spring-ի վրա հիմնված հավելվածների ստեղծումը: Այն թույլ է տալիս ստեղծել վեբ-հավելված, որը ծրագրավորողներից պահանջում է քիչ ջանքեր այն կարգավորելու եւ կոդը գրելու համար:

Spring Boot-ն ունի մեծ ֆունկցիոնալություն, սակայն դրա առավել նշանակալի առանձնահատկություններն են ' կախվածության կառավարումը, ավտոմատ կոնֆիգուրացիան եւ ներկառուցված Servlets կոնտեյնեռները:

Կախվածության կառավարման գործընթացը արագացնելու համար Spring Boot-ը անվերապահորեն փաթեթավորում է Spring-ի վրա հիմնված հավելվածի յուրաքանչյուր տեսակի համար անհրաժեշտ երրորդ կողմի կախվածությունները եւ ապահովում է մշակողին այսպես կոչված starter փաթեթների միջոցով (spring-boot-starter-web, spring-boot-starter-data-jpa եւ այլն):

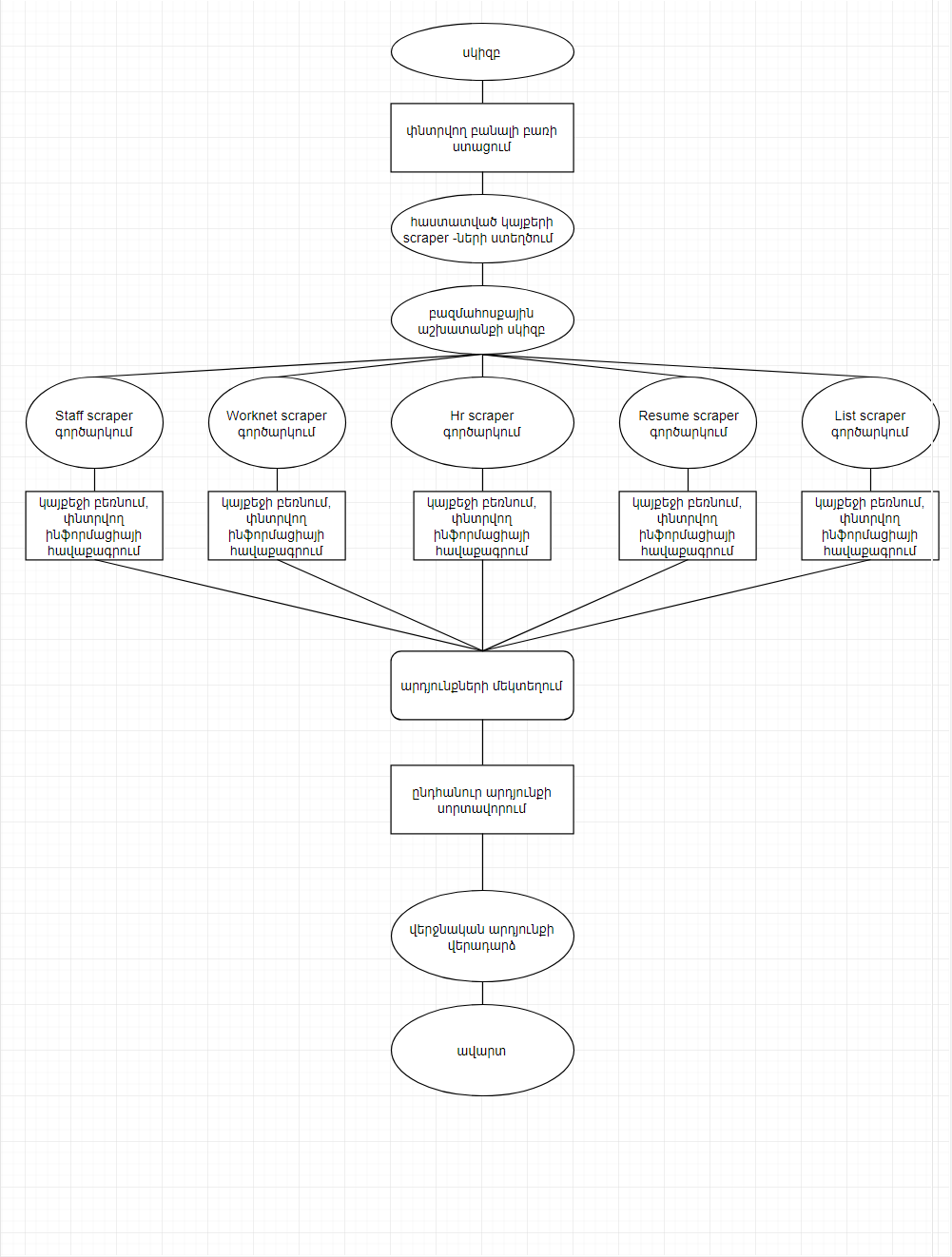
Գլուխ 2. Ծրագրային փաթեթի և ալգորիթմի մշակում

2.1 Ծրագրային պահաջներ

Ծրագրային փաթեթի մշակումը կատարվել է ՝ Java ծրագրաոման լեզվի 8 –րդ տարբերակով, spring boot framework –ի 2.0.5 տարբերակով և jsoup գրադարանի 1.13.1 տարբերակով: Ծրագրային պահանջներն են Java Runtime Environment 8 –ի առկայությունը և և համացանց մուտք գործելու հնարավորություն:

2.1 Խնդրի ալգորիթմի մշակում

Ծրագրային փաթեթի ալգորիթմը UML դիագրամի տեսքով ներկաըացված է նակար 8 ում:



Նկար 8.

2.3 Ծրագրային փաթեթի մշակում

Ծրագրային փաթեթը կազմված է երկու մասից ` դիմային մասից և սերվերային մասից:

Դիամյին մասն իր մեջ ներառում է

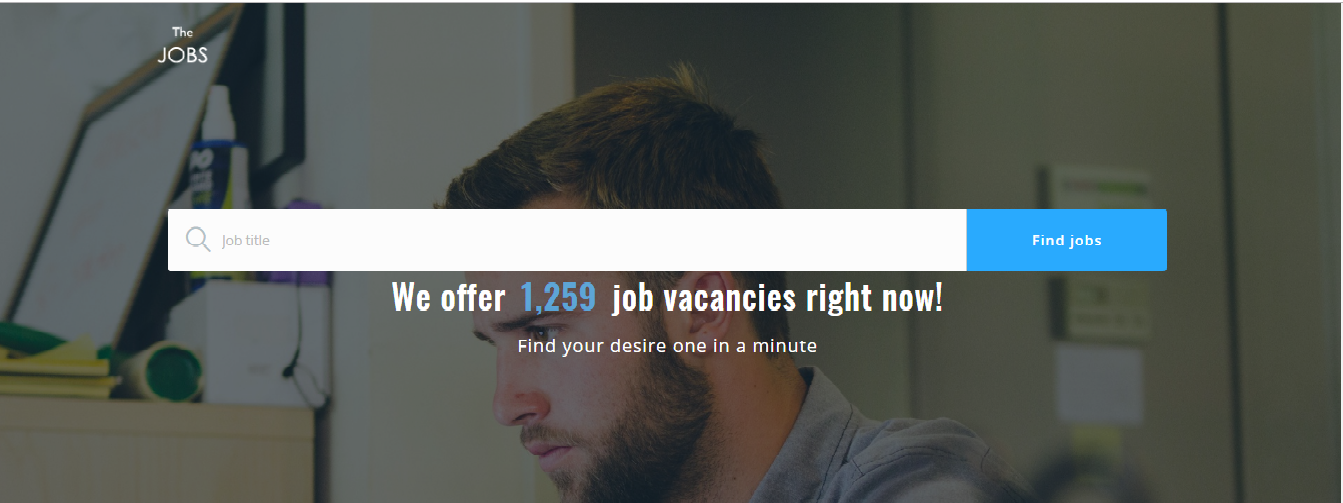
* բանալի բառի մուտքագրման հատվածից
* արդյունքների արտածման հատվածից

Սերվերային մասն իր մեջ ներառում է

* բազմահոսքային աշխատանքի ապահովման մաս
* կայքէջերի բեռնման մաս
* տվյալների հավաքման մաս
* արդյունքների սորտավորման մաս

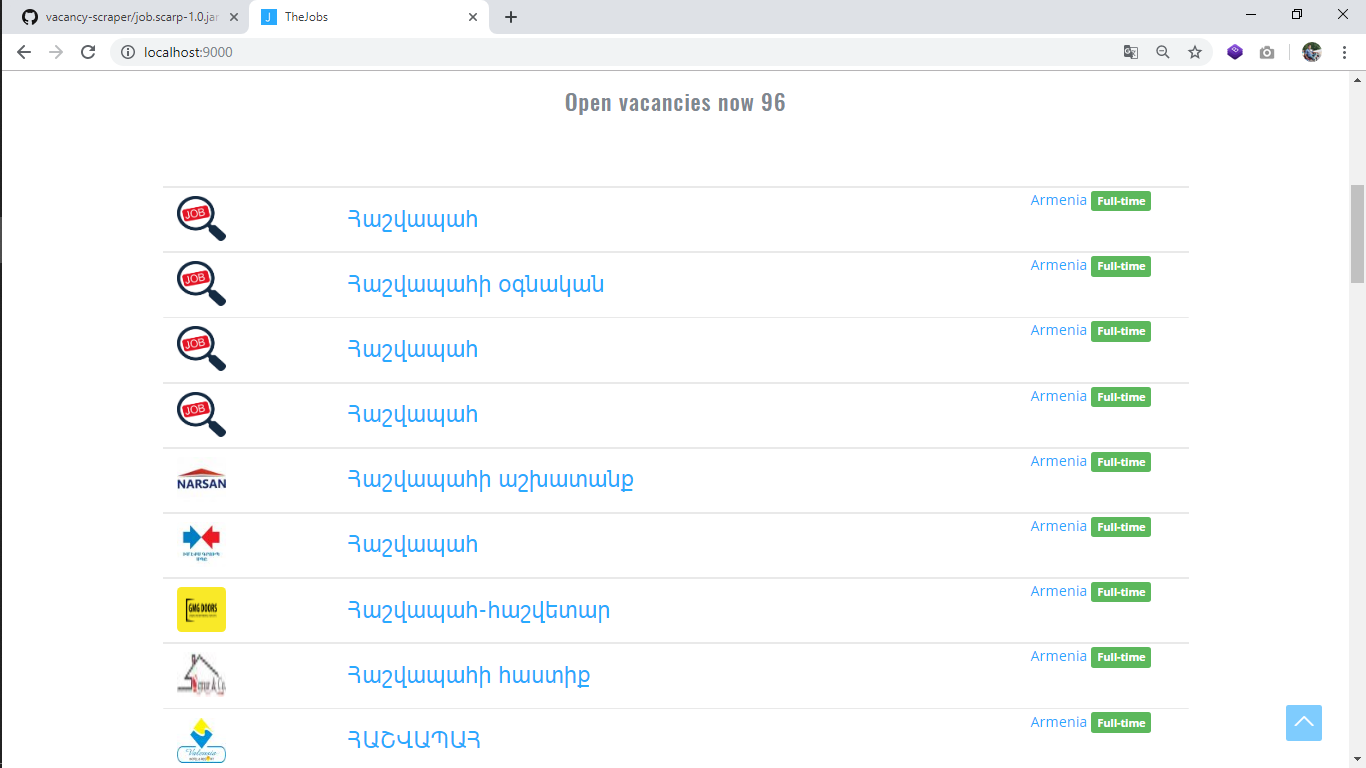
2.3.1 Դիմային մաս

Նկար 9 ում պատկերված է բանալի բառի մուտքագրման հատվածը:



Նկար 9.

Նկար 10 ում պատկերված է արդյունքների արտածման հատվածը:



Նկար 10.

Դիմային մասի կոդը ունի հետեվյալ տեսքը ՝

<!DOCTYPE html>

<html lang="en" xmlns:v-on="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<meta name="description" content="Post a job position or create your online resume by TheJobs!">

<meta name="keywords" content="">

<title>TheJobs</title>

<!-- Styles -->

<link href="assets/css/app.min.css" rel="stylesheet">

<link href="assets/css/custom.css" rel="stylesheet">

<!--<link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">-->

<!-- Fonts -->

<link href='http://fonts.googleapis.com/css?family=Oswald:100,300,400,500,600,800%7COpen+Sans:300,400,500,600,700,800%7CMontserrat:400,700'

rel='stylesheet' type='text/css'>

<!-- Favicons -->

<link rel="apple-touch-icon" href="../apple-touch-icon.html">

<link rel="icon" href="assets/img/favicon.ico">

<!-- <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>

<script src="https://cdn.jsdelivr.net/npm/vue-resource@1.5.1"></script>-->

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

<!--<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>-->

</head>

<body class="nav-on-header smart-nav">

<div id="app">

<!-- Navigation bar -->

<nav class="navbar">

<div class="container">

<!-- Logo -->

<div class="pull-left">

<a class="navbar-toggle" href="#" data-toggle="offcanvas"><i class="ti-menu"></i></a>

<div class="logo-wrapper">

<a class="logo" href=""><img src="assets/img/logo.png" alt="logo"></a>

<a class="logo-alt" href=""><img src="assets/img/logo-alt.png" alt="logo-alt"></a>

</div>

</diվ>

</div>

</nav>

<!-- END Navigation bar -->

<!-- Site header -->

<header class="site-header size-lg text-center" style="background-image: url(assets/img/bg-banner1.jpg)">

<div class="container">

<div class="col-xs-12">

<div class="header-job-search">

<div class="input-keyword">

<input id="search\_keyword" type="text" class="form-control" placeholder="Job title" value="">

</div>

<div class="btn-search">

<button type="button" onclick="find()" class="btn btn-primary">Find jobs</button>

</div>

</div>

<br><br>

<h2>We offer

<mark>1,259</mark>

job vacancies right now!

</h2>

<h5 class="font-alt">Find your desire one in a minute</h5>

<br><br><br>

</div>

</div>

</header>

<!-- END Site header -->

<!-- Main container -->

<main>

<!-- Recent jobs -->

<section>

<div class="container" id="result-list">

<header class="section-header" id="open-count">

<h2>Recent jobs</h2>

</header>

<!--<header v-if="resultSize === 0">

<h2>This Keyword not match vacancies :( </h2>

</header>

-->

<div class="row item-blocks-connected">

<!-- Job item -->

<div id="res"></div>

<!-- END Job item -->

</div>

<br><br>

<p class="text-center" id="browse\_btn">

</p>

</div>

</section>

<!-- END Recent jobs -->

<!-- Facts -->

<section class="bg-img bg-repeat no-overlay section-sm"

style="background-image: url(assets/img/bg-pattern.png)">

<div class="container">

<div class="row">

<div class="counter col-md-3 col-sm-6">

<p><span data-from="0" data-to="6890"></span>+</p>

<h6>Jobs</h6>

</div>

<div class="counter col-md-3 col-sm-6">

<p><span data-from="0" data-to="1200"></span>+</p>

<h6>Members</h6>

</div>

<div class="counter col-md-3 col-sm-6">

<p><span data-from="0" data-to="36800"></span>+</p>

<h6>Resume</h6>

</div>

<div class="counter col-md-3 col-sm-6">

<p><span data-from="0" data-to="15400"></span>+</p>

<h6>Company</h6>

</div>

</div>

</div>

</section>

<!-- END Facts -->

<!-- END How it works -->

</main>

<!-- END Main container -->

<!-- Bottom section -->

<div class="container">

<div class="row">

<div class="col-md-8 col-sm-6 col-xs-12">

<p class="copyright-text">Copyrights &copy; 2017 All Rights Reserved by <a

href="http://themeforest.net/user/thethemeio">TheThemeio</a>.</p>

</div>

</div>

</div>

<!-- END Bottom section -->

</footer>

</div>

<script src="assets/js/app.min.js"></script>

<script src="assets/js/custom.js"></script>

<!--<script src="js/main.js"></script>-->

<!--<script src="http://localhost:9000/main.js"></script>-->

<script src="/js/search.js"></script></body></html>

2.3.2 Սերվերային մաս

Սերվերային մասի աշխատանքը սկսում է բանալի բառի մուտքագրումից և հաստատումից հետո: Բնալաի բառի ստացումից հետո ֆիկսվում է աշխատանքային հոսքերի քանակը այնուհետեվ գործի են անցնում scraper -ները :

Բանալի բառի ստացման և բազմահոսքային ծրագրի սկիզբը կոդով ներկայացվում է այսպես ՝

@RestController

@RequestMapping("/scrap")

public class MainController {

@GetMapping("/job")

public List<Job> getJobsByKeyword(@RequestParam("keyword") String keyword) {

List<Job> dest = new CopyOnWriteArrayList<>();

long start = System.nanoTime();

ExecutorService executorService = Executors.newFixedThreadPool(5);

executorService.execute(new StaffScraper(keyword, dest));

executorService.execute(new WorknetScraper(keyword, dest));

executorService.execute(new HrScraper(keyword, dest));

executorService.execute(new ResumeScraper(keyword,dest));

executorService.execute(new ListScraper(keyword,dest));

executorService.shutdown();

while (!executorService.isTerminated()) {

}

System.out.println("\nFinished all threads");

System.out.println("Scraped job size - " + dest.size());

dest.sort(new WordMatchComparator(keyword));

long end = System.nanoTime();

double seconds = (double) (end - start) / 1\_000\_000\_000.0;

System.out.println("Work end with " + seconds + " seconds");

return dest;

}

}

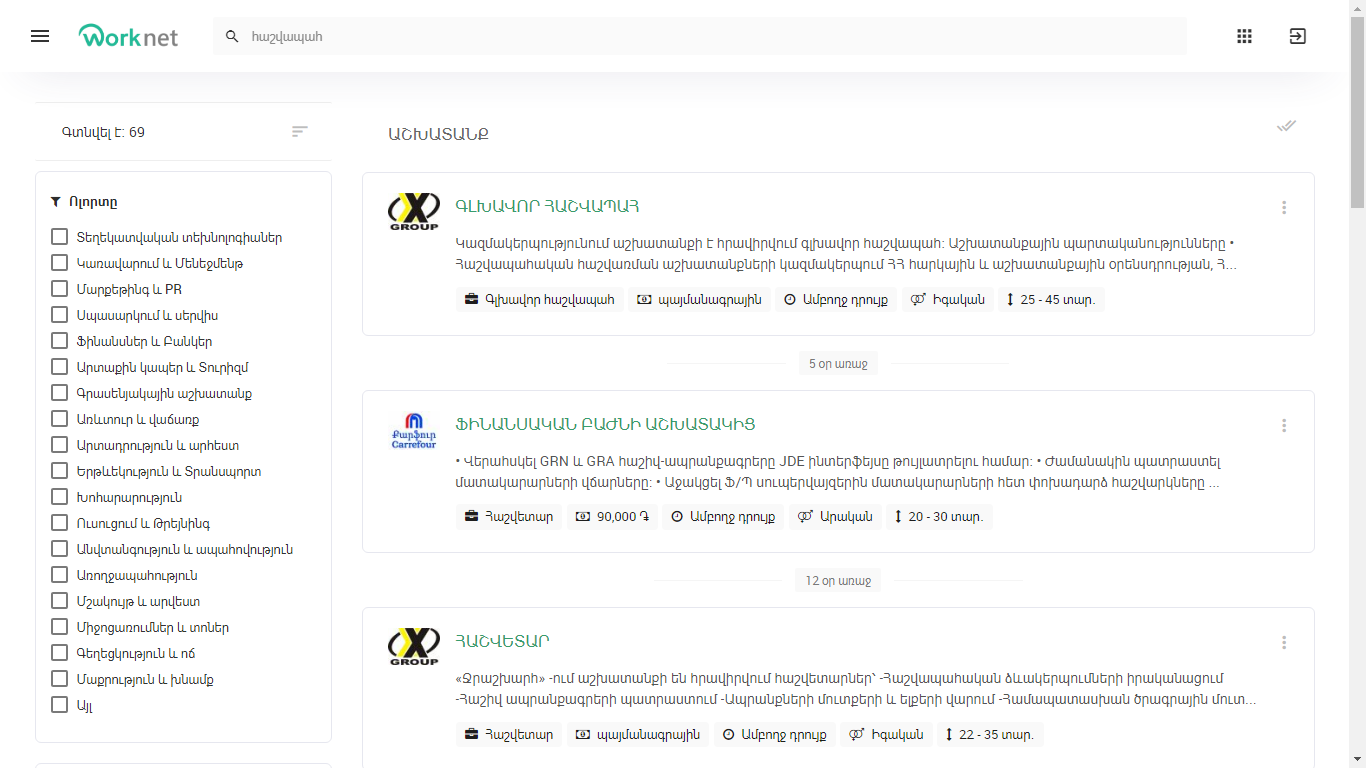
Այս աշխաըանքի ընթացքում տեղական աշխատանքի շուկայից կդիտարկենք առավել հաճախակի այցելություն ունեցող 5 կայքեր ` staff.am, worknet.am, hr.am, resume.am, list.am :

Բոլոր 5 կայքերն ունեն տարբեր կառուցվածք և ամեն կայքի համար գործում է առանձին scraper :

Դիտարկենք worknet.am կայքի scraper –ը, կայքեջի մեզ հետաքրքրող հատվածի հասցեն հետեվյալն է՝ <https://www.worknet.am/hy/jobs?query=հաշվապահ> ,

Տվյալ հասցեն փոխանցում ենք jsoup գրադարանին և բեռնում էջը, այնուհետեվ սկսում ենք հավաքել մեզ հետաքրքրող ինֆորմացիան:

Որոնման արդյունքն ունի հետեվյալ տեսքը ՝ նկար 11



Նկար 11.

Մեզ հետաքրքող հատվածի HTML կոդը այսպիսնն է ՝

<div class="listview\_\_item">

<img src="/file/33267/resize/60x60" class="listview\_\_img logo" alt="">

<div class="listview\_\_content">

<div class="listview\_\_heading"><a href="/job/գլխավոր-հաշվապահ-3732" data-aside-content="">Գլխավոր հաշվապահ</a></div>

<p>Կազմակերպությունում աշխատանքի է հրավիրվում գլխավոր հաշվապահ:

Աշխատանքային պարտականությունները

• Հաշվապահական հաշվառման աշխատանքների կազմակերպում ՀՀ հարկային և աշխատանքային օրենսդրության, Հ...</p>

<div class="listview\_\_attrs">

<span><i class="fa fa-briefcase"></i> Գլխավոր հաշվապահ</span>

<span><i class="fa fa-money"></i> պայմանագրային</span>

<span><i class="fa fa-clock-o"></i> Ամբողջ դրույք</span><span>

<i class="fa fa-venus-mars"></i>Իգական </span>

<span><i class="fa fa-arrows-v"></i> 25 - 45 տար․</span>

</div></div>

<div class="actions listview\_\_actions">

<div class="dropdown actions\_\_item">

<i class="zmdi zmdi-more-vert" data-toggle="dropdown"></i>

<div class="dropdown-menu dropdown-menu-right">

<a class="dropdown-item" href="/company/x-group-866" data-aside-content="">X-GROUP</a>

<a class="dropdown-item" href="#sendMessage?userID=27839"><i class="fa fa-envelope-o"></i> Ուղարկել նամակ</a>

<a class="dropdown-item" href="#"><i class="fa fa-map-marker"></i> Նոր Նորք </a>

</div></div></div></div>

Օգտվելով DOM էլեմնտներից վերցնւմ ենք անհրաժեշտ ինֆորմացիան

կոդից հատված ՝

public void getJobs(){

try {

Document worknetPage = Jsoup.connect("https://www.worknet.am/hy/jobs?query="+keyword).get();

Elements jobList = worknetPage.select("div[class='listview\_\_item']");

int count = 0;

for (Element sjob : jobList) {

if(count == 20){

break;

}

Element titleElement =sjob.getElementsByTag("a").first();

String title = titleElement.text();

String imgSrc ="https://www.worknet.am"+ sjob.getElementsByTag("img").first().attr("src");

String link = "https://www.worknet.am"+ titleElement.attr("href");

String description = sjob.getElementsByTag("p").first().text();

Job job = new Job(title,description,link,imgSrc);

if(job.getCompanyLogo() == null){

job.setCompanyLogo("assets/img/job.png");

}

this.result.add(job);

}

} catch (IOException e) {

e.printStackTrace();

}

}

Կատարելով նույն գործողությունը մնացաց բոլոր կայքերի համար միաժամանակ, արագ և էֆեկտիվ հավաքագրում ենք տափուր աշխատատեղերը մեկ վայրում այնուհետև սորտավորում արդյունքները ըստ բանալի բառի և վերադարձնում արդյունքը:

Սերվերային մասի աշխատանքը միջինում տեվում է 3 ից 6 վայրկյան:

Գլուխ 3. Տնտեսագիտական հիմնաորում և բնապահպանական խնդիրներ

3.1 Էներգետիկական ռեսուրսները

Էներգետիկական ռեսուրսները նույնպես լինում են 2 տեսակի՝ վերականգնվող և չվերականգնվող: Չվերականգնվող ռեսուրսներն են ածուխը, նավթը, գազը, սլանեցները, տորֆը, որոնք համարվում են թեթև տարրեր (ջրածին, հելիում, լիթիում), ատոմային վառելիքը: Վերականգնվող ռեսուրսներն են համարվում ֆուրոսինթեզի գործընթացի էներգիայի, արևային ճառագայթների օգտագործման էներգիան, հիդրոէներգիան, ալիքների և մակընթացության էներգիան, գոլորշիացման էներգիան և նստվածքների գոյացումը, քամու էներգիան, հիդրաջերմային (ընդերքի ջերմային), որը հիմնված է մթնոլորտային և ցամաքային, ջրային մակերեսի ջերմաստիճանի տարբերության վրա:

Չնայած որ էներգիայի տեսակները զանազան են՝ մարդն օգտագործում է դրանցից մի մասը: 120-150 տարի առաջ էներգիայի գլխավոր աղբյուրներն էին մարդկային և կենդանական ուժը, փայտը, քամին, իսկ այժմ այդ աղբյուրներն են նավթը, գազը, ածուխը, հիդրոէներգիան և ատոմային էներգիան: Էներգիայի մյուս տեսակների օգտագործումը հաճախ չի հանդիպում: Ընդերքից կարելի է հանել 66,16 մլն տոննա ուրանի պաշար: Ծովերն ու օվկիանոսները պարունակում են 4 մլրդ տոննա ուրան, սակայն ջրից այն հանելը 5 անգամ ավելի թանկ է, քան հանքից: Ատոմային վառելիքի պոտենցիալ ռեսուրսներն ավելի հաճախ են օգտագործվում, քան օրգանական վառելիքի բոլոր տեսակները միասին: Օրինակ՝ լիթիում, որն օգտագործում են ծովի ջրում սինթեզի համար (պայմանական վառելիքի միավոր ) 8.1017 տ է, իսկ դեհտերիումը 6.1020 տ: Այնուամենայնիվ, այս ռեսուրսները պիտանի չեն օգտագործման համար: Ներկայումս հիմնական էներգետիկական ռեսուրս է համարվում ածուխը, որը 20-30 անգամ գերազանցում է նավթին և 30-50 անգամ բնական գազին:

Եթե տարեկան 7 մլրդ տոննա պայմանական վառելիք է լինում, և վառելիք ստանալու ջանքերը յուրաքանչյուր 2 տարին մեկ անգամ կրկնապատկվում են, ապա հաշվարկները ցույց են տալիս, որ 2050թ-ին վառելիքի հնարավոր բոլոր պաշարները սպառված կլինեն: Այս փաստը վերաբերվում է հատկապես գազին և նավթին: Որոշ երկրներ արդեն ունեն նավթի սուր պահանջ:

Բոլոր տեսակի վառելիքների այրումը ախտոտում է շրջակա միջավայրը, մթնոլորտում գոյանում է մեծ քանակությամբ CO2 և ահա թե ինչու անհրաժեշտ է որոնել էներգիայի նոր աղբյուրներ և նոր միջոցներ նրա մրրկման համար:

Դեռ հին ժամանակներում մարդիկ օգտագործել ել էներգիայի այնպիսի տեսակներ, ինչպիսիք են քամու էներգիան, գետերի ջրի ծանրության ուժը, հէկ-ները: Էներգիայի այս աղբյուրները գրեթե չեն սպառվում: Դրանց օգտագործման սղությունը պայմանավորված է երկրի վրա ոչ հավասարաչափ բախշմամբ և դրանց ժամանակին ոչ համընթաց օգտագործմամբ: Ջրից ստացվող էներգիան մարդիկ հին ժամանակներից սկսած օգտագործել են ջրամատակարարման համար, իսկ ավելի ուշ հիդրոէլեկտրակայանների: Հիդրոէներգետիկան հանդիսանում է էներգետիկայի բաղադրիչ մասը: Կառուցվել են խոշոր հէկ-ներ Դնեպր, Վոլգա, Ենիսեյ, Լենա և շատ ուրիշ գետերի վրա: Ներկայումս հիդրոէներգետիկական կառույցները նախատեսված են լուծել մի շարք այնպիսի հարցեր, ինչպիսիք են՝ էլեկտրոէներգիան, հողատարածքների ջրամատակարարումը, ձկնորսությունը, նավարկությունը:

Հիդրոռեսուրսների օգտագործումը 1966թ. Ապահովեց աշխարհի էներգիայի պահանջարկի 6%, 1980 5%, իսկ 2000թ 2-2,5% կապահովի: Հիդրոկայանների կառուցումը ոչ բարենպաստ ազդեցություն ունի բնության վրա: Ջրամբարների կառուցման անհրաժեշտությունը առաջացնում է ջրասուզում, որն ազդում է շրջակա տարածքի ռելեֆի և կլիմայի վրա: Այդ իսկ պատճառով էներգիայի նոր աղբյուրների և նոր միջոցների որոնումներ են կատարվում: Օրինակ՝ նախատեսվում են արևային ռադիացիայի, հիդրոէներգիայի, քամու, ծովի էներգիայի աղբյուրների ստեղծում:

Արևը կյանքի սկիզբն է երկրի վրա, որը լույս և ջերմություն է տալիս, ապահովում է ածուխի, սլանեցի, նավթի, գազի, հիդրոռեսուրսների առաջացումը: Արևի յուրաքանչյուր վայրկյան էներգիա է հաղորդում երկրին, որը հավասար ե 16,8.1013 կգՋ: Մթնոլորտի արտաքին սահմանին հասնող արևային ռադիացիային ընդհանուր ծավալի միայն կեսն է հասնում երկրի մակերեսին: Երկիր մոլորակի կողմից սպառվողէներգիան 35000 անգամ ավելի շատ է, քան բնակչության կողմից օգտագործվող էներգիայի տարեկան չափը: Արևը 5000 անգամ ավելի շատ է էներգիա հաղորդում երկրին քան մյուս բոլոր էներգիայի աղբյուրները միասին: Արևային Էներգիայի օգտագործումը կապված էէլեկտրաքիմիական ռեակցիաների և լուսային ճառագայթների օպտիկական կոնցեպտրացիայի միջոցով ստեղծվող ջերմային կամ էլեկտրոէներգիայի հետ:Արևային Էներգիայի օգտագործումը սահմանափակված է ինժեներա-տեխնիկական միջոցների հավաքագրման, ստեղծման և արևային ճառագայթների օգտագործման մակարդակով:

Քամին նույնպես հանդիսանում էներգիայի հինավուրց աղբյուրներից մեկը, որն օգտագործվում է նավարկության մեջ: Այն իրենից ներկայացնում է օդի հոսք, որը գոյանում է տարբեր տեղանքների ջերմաստիճանի տարբերության դեպքում: Քամու ուժգնությունը մեծանում է բարձրության հետ: Քամու էներգիան օգտագործող էներգիայի կայանի աշխատանքի բարենպաստ պայմաններն ապահովված են, եթե քամու ուժգնությունը 10 և 14 մ/վ է: Աշխարհում այդպիսի կայան կառուցվել է Կուրսկ քաղաքում ԽՍՀՄ-ի ժամանակաշրջանում 1929-1930թթ: ԽՍՀՄ-ում, Անգլիայում, Ֆրանսիայում, Կանադայում կատարվում են աշխատանքներ ժամանակակից քամու էլեկտրոկայան կառուցելու համար: Հեռանկարային է ծովից ստացվող էներգիայի օգտագործումը, որը կապվաղ է ջրային շերտէրի ջերմաստիճանի փոփոխության հետ: Ստորին շերտերը սառն են մնում, իսկ վերին շերտերա տաքանում են արևից ստացված էներգիայով: Հաշվարկված է, որ օվկիանոսից ստացվող էներգիայի քանակը 300 անգամ ավելի, քան բոլոր տեսակի էներգիայի ժամանակակից օգտագործման ծավալը և համարժեք է տարեկան 18 մլրդ նավթի օգտագործմանը:

Ծովերից և Օվկիանոսներից ստացվող էներգիայի օգտագործման նոր ուղղությունը մակընթացության էլեկտրոնիկայի մշակումն է: Առաջընթաց և հետընթացի մեկի ցիկլը 8ղ10 12 ԿՎՏ է: Սա 100 անգամ գերազանցում է աշխարհի հէկ-ներից ստացվող էներգիայի թվին: Առաջին փորձառու 400 կվտ հզորությամբ քամու էլեկտրոկայանը փորձարկվել է ԽՍՀՄ-ի ժամանակաշրջանում Բարենցի ծովում: Նախատեսվում է կառուցել 14 մլն ԿՎՏ հզորությամբ մակընթացության Էլեկտրոկայան Սպիտակ ծովում: